Norges miljø- og biovitenskapelige universitet

Masteroppgave 2023 30 stp
Fakultet for realfag og teknologi

Bruk av kunstig intelligens til medisinsk beslutningstøtte: Sammenligning av Dynamic Ensemble Selection med klassiske ML-algoritmer i kreftprediksjon

Use of Artificial Intelligence for Medical Decision Making: A Comparative Study of Dynamic Ensemble Selection and Classical ML-Algorithms for Cancer Prediction

Majorann Thevarajah og Saranjan Anpalagan
Industriell økonomi

Forord

Denne masteroppgaven er skrevet ved Fakultet for realfag og teknologi (REALTEK) ved Norges Miljø- og Biovitenskapelige Universitet (NMBU), våren 2023. Oppgaven markerer avslutningen på et fem-årig masterprogram som sivilingeniørstudenter i industriell økonomi, med en fordypning innen datavitenskap. Oppgaven har et omfang på 30 studiepoeng.

Vi vil benytte anledningen til å rette en stor takk til vår hovedveileder ved NMBU, førsteamanuensis Oliver Tomic, for rettledning, gode tilbakemeldinger, samt vist stort engasjement og interesse gjennom hele arbeidsprosessen med masteroppgaven. Vi vil også rette en stor takk til biveilederne, professor Cecilia Marie Futsæther ved NMBU og universitetslektor Jesper Frausig ved NMBU, som har kommet med gode innspill og interessante diskusjoner på våre ukentlige møter. Takk til seniorrådgiver Natalia Kunst ved Helsedirektoratet som alltid har vært tilgjengelig for spørsmål, gitt grundige tilbakemeldinger og ikke minst har hjulpet oss med å komme i kontakt med en klinisk ekspert, spesialist i ortopedisk kirurgi og overlege Hanne Osnes-Ringen. Vi setter stor pris på all den tiden dere har investert i oss og oppgaven.

Masteroppgaven symboliserer avslutningen på vår akademiske reise som har vært et innholdsrikt semester i løpet av mastergraden. Prosessen har vært både lærerikt og spennende, hvor vi har fått tilliten og friheten til å utarbeide vår oppgave, samtidig gjennomføre det i samsvar med våre og veiledernes preferanser. Vår tid ved NMBU har vært preget av både nedturer og oppturer, men vi verdsetter all den kunnskapen vi har tilegnet og opparbeidet oss i løpet av denne tiden.

Avslutningsvis vil vi uttrykke vår dype takknemlighet til både familie og venner for deres støtte, forståelse og omsorg som har blitt vist gjennom hele vår ferd mot å fullføre denne oppgaven. Gjennom våre fem uforglemmelige år har vi også hatt gleden av å dele minner med noen fantastiske medstudenter, og vi vil gjerne takke dem for å gjøre vår studietid så spesiell.

Majorann Thevarajah

Majorann Thevarajah

Saranjan Anpalagan

Ås, 15.mai 2023

Sammendrag

I en verden med teknologiske fremskritt som stadig påvirker ulike næringer, har maskinlæring (ML) vist seg å være en banebrytende teknologi som kan revolusjonere ulike sektorer. Helsesektoren, som i stor grad står overfor kritiske og komplekse utfordringer, er en sektor som kan dra stor nytte av ML. Maskinlæring er en gren av datavitenskapen som bruker algoritmer og statistiske modeller til å forbedre datamaskiners ytelse, og er en fundamental byggestein i utviklingen av kunstig intelligens.

Denne oppgaven tar for seg problemstillingen om å analysere prediksjon av kreftpasienter ved å anvende ulike ML-algoritmer. I denne sammenhengen ble Dynamic Ensemble Selection (DES) undersøkt for å evaluere om det kan gi bedre resultater for prediksjon av kreftpasienter enn kjente klassiske ML-algoritmer. Flere ML-teknikker ble brukt til å utføre prediksjonstester og øke forståelsen av algoritmene. Videre ble en MCDA-analyse benyttet for å sammenligne resultatene med den nåværende beslutningsprosessen, som tar hensyn til kliniske og etiske retningslinjer samt pasientens behov og interesse. Studien vil gi innsikt i hvilken grad DES og de klassiske ML-algoritmene kan bidra til å forbedre dagens situasjon om å støtte medisinsk beslutningstaking i kreftbehandling.

Datasettene som ble brukt til å trene de prediktive modellene inneholdt omfattende klinisk informasjon om pasienter behandlet ved Oslo universitetssykehus (OUS). Datasettene inkluderte en gruppe på 192 pasienter som gjennomgikk behandling for kolorektal kreft i tidsrommet 2013-2017, samt en annen gruppe på 197 pasienter som ble behandlet for hode- og halskreft i perioden 2007-2013. Åtte klassifiseringsalgoritmer ble trent på disse datasettene med kliniske egenskaper for generell overlevelse (OS), progresjonsfri overlevelse (PFS) og sykdomsfri overlevelse (DFS). Resultatene ble validert ved å måle nøyaktighet, F1-score for positiv og negativ, Matthews korrelasjonskoeffisient (MCC) og ROC AUC. Videre ble modellen for hode- og halskreft testet på et eksternt datasett bestående av 99 behandlede pasienter ved MAASTRO-klinikken i Nederland.

Funnene fra oppgaven tyder på at det er flere muligheter å dra nytte av i forhold til å anvende ulike ML-algoritmer. De klassiske algoritmene presterer generelt bedre enn DES med hensyn til nøyaktighet, prediksjonsytelse, og antall feilaktig klassifisering. I følge MCDA-analysen blir også de klassiske algoritmene sett på som den beste løsningen i kombinasjon av den

eksisterende beslutningsprosessen. Den nye løsningen skal ikke være en erstatning, men bli sett på som et mulig beslutningsstøtteverktøy. Det er viktig å merke seg at ulike algoritmer og teknikker vil respondere forskjellig og gi ulike svar på forskjellige typer data og problemer. Dermed er denne anbefalingen gitt for de datasettene og algoritmene som denne oppgaven har basert seg på.

For videre forskning anbefales det å samle et større og mer dagsaktuelt datasettet, som kan bidra til å optimalisere prognosen og overlevelsesraten for kreftpasienter. Dette kan gi mer presise og pålitelige prediksjoner om hvilken behandling som vil gi best resultat for den enkelte pasient. Resultatene fra denne oppgaven kan danne grunnlag for utvikling av modeller som kan identifisere optimal kreftbehandling for en pasient og brukes som beslutningstøtteverktøy av helsepersonell ved behandling av nye kreftpasienter.

Abstract

In a world of technological advancements that continue to impact various industries, machine learning (ML) has proven to be a ground-breaking technology that can revolutionise various sectors. The health sector, which largely faces critical and complex challenges, is a sector that can greatly benefits from ML. Machine learning is a part of computer science that deals with using algorithms and statistical models to learn and improve computer performance based on feedbacks and experiences from previous data and is a fundamental in the development of artificial intelligence.

The master thesis deals with the problem of analyzing the prediction of cancer patients using different ML algorithms. In this context, several ML techniques are used to perform prediction tests and increase the understanding of the algorithms. Furthermore, an MCDA-analysis is used to compare the results with the current solution, which is based on clinical and ethical guidelines as well as the patients' needs and interests. The aim is to investigate whether Dynamic Ensemble Selection (DES) gives better results for predicting cancer patients than existing models, like random forest and logistic regression. The study will provide insight into the extent to which the DES algorithms and the classical algorithms can contribute to improving the current situation of supporting medical decision-making in cancer treatment.

The datasets used for training the predictive models consisted of clinical information from 192 patients who were treated for colorectal cancer in the period 2013 to 2017, and 197 patients who were treated for head and neck cancer in period 2007 to 2013 at Oslo University Hospital, OUS. Eight classification algorithms were trained on these datasets with clinical characteristics of overall survival (OS), progression-free survival (PFS), and disease-free survival (DFS). The results were validated by measuring accuracy, F1-score for positive and negative, Matthew's correlation coefficient (MCC) and ROC AUC. Furthermore, an external data set consisting of 99 patients who received treatment at the MAASTRO clinic in the Netherlands was used to test head and neck cancer models.

The findings from the thesis indicate that several opportunities can benefit from the use of different ML algorithms. The classical algorithms generally outperform DES when it comes to accuracy, prediction performance, and number of misclassifications. According to MCDA-analysis, the classic algorithms are also seen as the best solution in combination with the

current situation. The new solution should not be a replacement but be seen as a possible decision-support tool. It is also important to note that different algorithms and techniques will respond differently and give another output to different type of data and problems. This recommendation is therefore given for the datasets and algorithms on which this task is based on.

A challenge with the datasets that are used in this thesis is that they were limited and contained little information. For further research, a larger and more up-to-date data set should be collected, which can help to optimize cancer patients' prognosis and survival rate This can provide more precise and reliable predications about which treatment will give the best result for the individual patient. The results from this thesis can form the basis for the development of models that can identify optimal cancer treatment for a patient and be used as a decision-support tool by healthcare professionals when treating new cancer patients.

Innholdsfortegnelse

Figurer [1]

[1] Figurer i oppgaven ble genert ved bruk av programmeringsspråket Python, regnearket Excel eller ble designet selv ved hjelp av Diagrams.net (drawio, u.å). Ingen av figurene i oppgaven er hentet fra litteratur eller andre nettbaserte kilder. I de tilfeller der inspirasjon fra eksisterende figurer har blitt brukt, er det inkludert en referanse i figurteksten.

Tabeller

Begrepsliste

Begrep	Betydning
Arrays	I datavitenskap betyr å samle data av samme datatype i en minnebasert struktur for å behandle og lagre store mengder med data. Dette gir fordeler som rask tilgang og effektiv håndtering av data.
Bias	I en analytisk sammenheng refererer begrepet "bias" til en systematisk tendens til feil eller skjevhet i data/analyser som kan føre til unøyaktige resultater og konklusjoner. Bias kan oppstå når dataene som brukes i analysen ikke er representative for den populasjonen som studeres.
Biopsi-fri markør	Brukes i en medisinsk sammenheng, som referer til en diagnostisk metode som ikke krever en smertefull prosedyre for å samle informasjon om en tilstand eller sykdom.
DataFrame	Datastruktur som representerer data i tabellform, med kolonner og rader.
Datakvalitet	I denne oppgaven blir «datakvalitet» brukt til å referere og forklare hvor relevant, komplett og konsist dataene er i datamengden.
Konsensusprediksjonsresultat	En metode til å danne en samlet prediksjon, hvor flere individuelle metoder eller prediksjoner er kombinert og sammensatt.
Korrelerte egenskaper	Referer til to egenskaper som er sterkt relatert til hverandre. Et eksempel er om en endring av en verdi for en variabel kan påvirke verdien for en variabel.
Maskering	Å skjule en informasjon
Metrikk	Måleenheter som brukes for å evaluere resultater fra eksempelvis modeller. I maskinlæring brukes metrikk til å evaluere

	hvor godt en modell yter. Eksempler på slike måleenheter er F1 score, ROC-AUC, MCC, og mer.
One-vs.-All (OvA)	En teknikk som brukes for å trene modeller til å skille en klasse fra en annen klasse om gangen i et multiklassifiseringsproblem. Om et datasett inneholder klassene 1, 2, 3, vil én modell trenes for skille 1 fra 2. En annen modell vil trenes for å skille 1 fra 3, og én tredje modell som trenes for å skille 2 fra 3. Ved prediksjon av en ny prøve vil prøven bli klassifisert på alle de tre modellene, og prøven bli klassifisert som den klassen som får flest stemmer.
One-vs.-Rest (OvR)	En teknikk som brukes for å trene modeller til å skille en klasse fra andre klasser i et multiklassifiseringsproblem. Om et datasett inneholder klassene 1, 2, 3, vil én modell trenes for skille 1 fra 2 og 3. En annen modell vil trenes for å skille 2 fra 1 og 3, og én tredje modell som trenes for å skille 3 fra 1 og 2. Ved prediksjon av en ny prøve vil prøven bli klassifisert på alle de tre modellene, og prøven blir klassifisert som den klassen som for høyest predikasjonsverdi.
Overtilpasning *(eng. overfitting)*	Når en modell har blitt trent for mye på treningsdataen, og dermed yter dårligere på nye, ukjente data. Modellen mangler generaliseringsevnen som er nødvendig for å håndtere nye datasett. Dette kan skyldes av at modellen blir for kompleks og spesifikk for treningsdataen, eller at den blir påvirket av støy eller små variasjoner.
Sentrert	Et datasett som er sentrert rundt null, ved å trekke gjennomsnittet av hver variabel fra alle observasjoner i datasettet. Dette resulterer i en justering som sikrer at hver variabel har et gjennomsnitt på null.

Series (Pandas)	En «en-dimensjonal» datastruktur som ligner på en liste og som kan inneholde variabler av forskjellige datatyper. En kolonne i en tabell.
Standardisert/skalert	I maskinlæring blir «standardisering» brukt i den forstand at numeriske verdier i et datasett blir transformert. Vanligvis innebærer dette å trekke fra gjennomsnittet og dele på standardavviket for hver numerisk variabel i datasettet. Dette gjøres for å unngå at variabler med høye verdier har større innvirkning på modellens prediksjon enn variabler med lave verdier.
Undertilpasning *(eng. underfitting)*	Modellen er underlæret på treningsdataen og klarer ikke har presterer godt på treningsdataen. Det vil heller ikke prestere respektabel for ukjent data, da den vil ha problemer med å generalisere nye dataprøver. Oppstår som oftest når modellene er for enkel og er lite komplekst til å finne underliggende mønster i datasettet.

Forkortelser

Forkortelse	Betydning
AUC	Area Under Curve
CRISP-DM	Cross-Industry Standard Process for Data Mining
CT	Computertomografi
DES	Dynamic Ensemble Selection
DES-P	Dynamic Ensemble Selection Performance
DFS	Sykdomsfri overlevelse (*eng. Desease-Free Survival*)
ECDF	Empirisk Kumulativ Fordelingsfunksjon (*eng. Empirical Cumulative Distribution Function)*
ECOD	Efficient Cumulative - based Outlier Detection
FN	Falsk Negativ (*eng. False Negative)*
FP	Falsk Positiv (*eng. False Positive*)
FPR	Falsk Positiv Rate (*eng. False Positive Rate*)
HPV	Humant Papiliomavirus
KDP	Kernel Density Plot
KNN	K-Nærmeste Naboer (*eng. K-Nearst Neighbors)*
KNORA-E	KNearest Oracle - Eliminate
KNORA-U	KNearest Oracle - Union
LRC	Lokalt eller regionalt tilbakefall (*eng. Local-Regional Control)*
MCC	Matthews korrelasjonskoeffisient (*eng. Matthews Correlation Coefficent*)
MCDA	Flermålsanalyse (*eng. Multi-Criteria Decision Analysis)*
ML	Maskinlæring (*eng. Machine Learning)*

NaN-values	Manglende verdier (*eng. Not a Number-values*)
OS	Generell overlevelse (*eng. Overall Survival)*
OUS	Oslo universitetssykehus
OvA	En-mot-alle (*eng. One-vs.-All*)
OvR	En-mot-resten (*eng. One-vs.-Rest*)
PCA	Prinsipalkomponentanalyse *eller* hovedkomponentanalyse (*eng. Principal Component Analysis)*
PET	Positronemisjonstomografi (*eng. Positron Emission Tomography)*
PFS	Progresjonsfri overlevelse (*eng. Progression-Free Survival)*
PRE	Presisjon (*eng. precision*)
QDA	Kvadratisk diskriminantanalyse (*eng. Quadratic Discriminant Analysis)*
REC	Gjenkalling (*eng. recall*)
RENT	Repetert elastisk nett teknikk (*eng. Repeated Elastic Net Technique*)
ROC	Receiver Operator Characteristics
Status quo	Nåværende eller eksisterende tilstand
TN	Sann Negativ (*eng. True Negative*)
TP	Sann Positiv (*eng. True Positive)*
TPR	Sann Positiv Rate (*eng. True Positive Rate*)
WBS	Arbeidsnedbrytningsstruktur (*eng. Work Breakdown Structure*)

Symboler

Forkortelse	Betydning
e	Euler
p	Sannsynligheten
log	Den naturlige logaritmen
z	Den standardiserte/transformerte verdien
$\boldsymbol{\phi}(\boldsymbol{z})$	Sigmoid funksjon, er en verdi mellom 0 og 1
y	(2.3) Klassetilhørigheten
	(2.4 & 2.5) En vektor som representerer centroidene for en av de klassene i datasettet
x	(2.3) Verdiene til de forklarende variablene
	(2.4 & 2.5) Vektor som representer den nye prøven
	(3.1) verdien i datasettet
k	En av K mulige klassene
L	Antall klasser
$\boldsymbol{\mu}$	Gjennomsnittet av x
$\boldsymbol{\sigma}$	Standardavviket av x

Kapittel 1

Introduksjon

1.1 Bakgrunn og motivasjon

Kreft har blitt en stadig større trussel for folkehelsen, med økende antall tilfeller sammenlignet med forrige århundre (Grimsrud et al., 2021). Ifølge helsestatistikken fra Kreftregisteret, er kreft den vanligste dødsårsaken i Norge, og omtrent 30 000 mennesker dør av kreft hvert år (Larsen et al., 2022). Det kan observeres en økning i antall krefttilfeller over tid, og prognoser indikerer at denne trenden vil fortsette å stige fram mot 2040. Noen av hovedårsakene til denne økningen er større befolkning, populasjon blir eldre, og økt levealder i befolkning (Larsen et al., 2014). I 2021 ble det rettet betydelig oppmerksomhet mot den potensielle påvirkningen Covid-19-pandemien kunne ha for kreftdiagnostikken i Norge (Larsen et al., 2022). På bakgrunn av den økende trenden av krefttilfeller og ventetiden, ser helsesektoren på digital teknologi og digitalisering som et potensielt alternativ for å forbedre og effektivisere ventetiden, behandlingen og kreftdiagnostikken (NOU 2023: 4).

I dagens samfunn blir en rekke næringer stadig mer påvirket av digitalisering, som er definert som bruken av teknologi for å forbedre, forenkle og fornye ulike tjenester (Laudon & Traver, 2018). Rapporten *NOU 2023:4, Digitalisering og teknologisk utvikling i helse- og omsorgstjenestene*, utgitt av den norske regjeringen hevder at digital teknologi i helsesektoren kan forbedre kvaliteten på tjenestene som tilbys, øke effektiviteten og gi bedre pasientopplevelser (NOU 2023: 4). Digitalisering gir mulighet for innovasjon og forbedret effektivitet i tjenesteleveranse til brukere. En spesifikk form for digitalisering som har blitt mer relevant er maskinlæring (ML), en gren innenfor kunstig intelligens. ML muliggjør at maskiner kan lære å gjøre prediksjoner og ta beslutninger ved å analysere data ved hjelp av algoritmer (Alpaydin, 2020). Derfor kan det være fordelaktig å øke forståelsen for hvordan maskinlæring kan påvirke og endre beslutningsprosesser, samt forstå mulighetene og utfordringene som følger med denne teknologien.

1.2 Formål

Formålet med oppgaven er å evaluere prediksjonsevnen til Dynamic Ensemble Selection (DES) og eksisterende klassiske algoritmer på helsedata for pasienter som lider av kolorektal kreft eller hode- og halskreft. En grundig vurdering av DES og andre anerkjente klassiske modeller vil bli utført, og sett i lys av potensialet for å utvikle medisinske applikasjoner som kan brukes for å støtte beslutningstaking innen kreftmedisinsk sammenheng.

I studien har det blitt fokusert på Dynamic Ensemble Selection algoritmer fra DESlib-biblioteket, et relativt nytt og moderne bibliotek som har fått lite forskningsoppmerksomhet så langt (Cruz et al., 2020). Gitt at biblioteket er nytt og algoritmene i pakken er avanserte, er det en mulighet for at modeller som benyttes av DES-algoritmer kan oppnå bedre resultater på helsedata enn klassiske maskinlæringsalgoritmer. DES er en ensemble metode som handler om å forme en optimal klassifiseringsmodell basert på flere individuelle modeller. DES skiller seg fra andre klassiske ensemble algoritmer ved å benytte en dynamisk tilnærming til å velge individuelle modeller som inkluderes i ensemblet for hver prediksjon (Cruz et al., 2020). Dette er i kontrast til de klassiske ensemble-modellene som har et felles ensemblesett for alle prediksjoner. DES-algoritmene har altså potensialet til å tilpasse og endre seg i henhold til dataene, noe som kan øke mulighetene for forbedret prediksjonsytelse. En annen styrke med DES er evnen til å inkludere og kombinere flere læringsalgoritmer i ensemblet. Det er av vesentlig betydning å påpeke at under utførelsen av oppgaven var DESlib-pakken fortsatt under utvikling.

På lengre sikt kan denne studien bidra til å gi legene bedre verktøy for å ta informerte beslutninger om hvilken behandling som burde utføres basert på pasientdata. Det kan igjen bidra til å forbedre prognosene og overlevelsesraten for kreftpasienter ved å gi mer presise og pålitelige prediksjoner om hvilken behandling som vil gi best resultat for en bestemt pasient

1.3 Problemstilling og forskningsspørsmål

I lys av disse dagsaktuelle trendene, retter denne masteroppgaven søkelyset mot å undersøke DES-algoritmers evne til å predikere behandlingsutfallet for kreftpasienter sammenlignet med andre klassiske maskinlæringsalgoritmer. I særdeleshet vil oppgaven undersøke om maskinlæringsalgoritmer kan fungere som et beslutningsstøtteverktøy i en kombinasjon med dagens situasjon for å styrke beslutningene som tas av helsepersonell. Med utgangspunkt i beskrevet formål vil følgende problemstilling bli belyst:

Hvordan predikerer Dynamic Ensemble Selection på helsedata om kreftpasienter i forhold til andre klassiske maskinlæringsalgoritmer, og hvilke av de vurderte algoritmene kan brukes for å støtte beslutningstaking innen kreftmedisinsk sammenheng?

En systematisk tilnærming til forskningsspørsmål er avgjørende for å bryte ned en problemstilling i mindre områder, og for å sikre at forskningen utføres på en strukturert og målrettet måte. For å oppnå dette, har det blitt formulert følgende forskningsspørsmål som kan gi en dypere forståelse av problemstillingen og hjelpe med å avdekke relevante data og innsikt:

Forskningsspørsmål 1: I hvilken grad skiller nøyaktigheten for modellene som bruker DES eller klassiske algoritmer seg fra hverandre?

Forskningsspørsmål 2: Hvilke muligheter og utfordringer er til stede ved benyttelse av DES-algoritmer?

Forskningsspørsmål 3: Kan bruken av MCDA-analyse tilrettelegge implementering av maskinlæringsmetoder for å støtte medisinsk beslutningstaking?

Første forskningsspørsmål fokuserer på nøyaktighetsmålinger av algoritmene og sammenligner dem med hverandre. Formålet med dette spørsmålet er å identifisere hvilke algoritmer som gir de beste prediksjonene basert på en gitt datamengde. Det andre forskningsspørsmålet fokuserer på svakheter og styrker knyttet til bruk av DES-algoritmer. Målet er å undersøke om det eksisterer tilstrekkelig potensial i å benytte DES-algoritmer istedenfor vanlige klassiske ML-algoritmer. Det tredje og siste forskningsspørsmålet fokuserer på bruken av MCDA-analyse for å evaluere algoritmene og vurdere hvordan de utfører i forhold til den nåværende beslutningsprosessen

1.4 Avgrensning

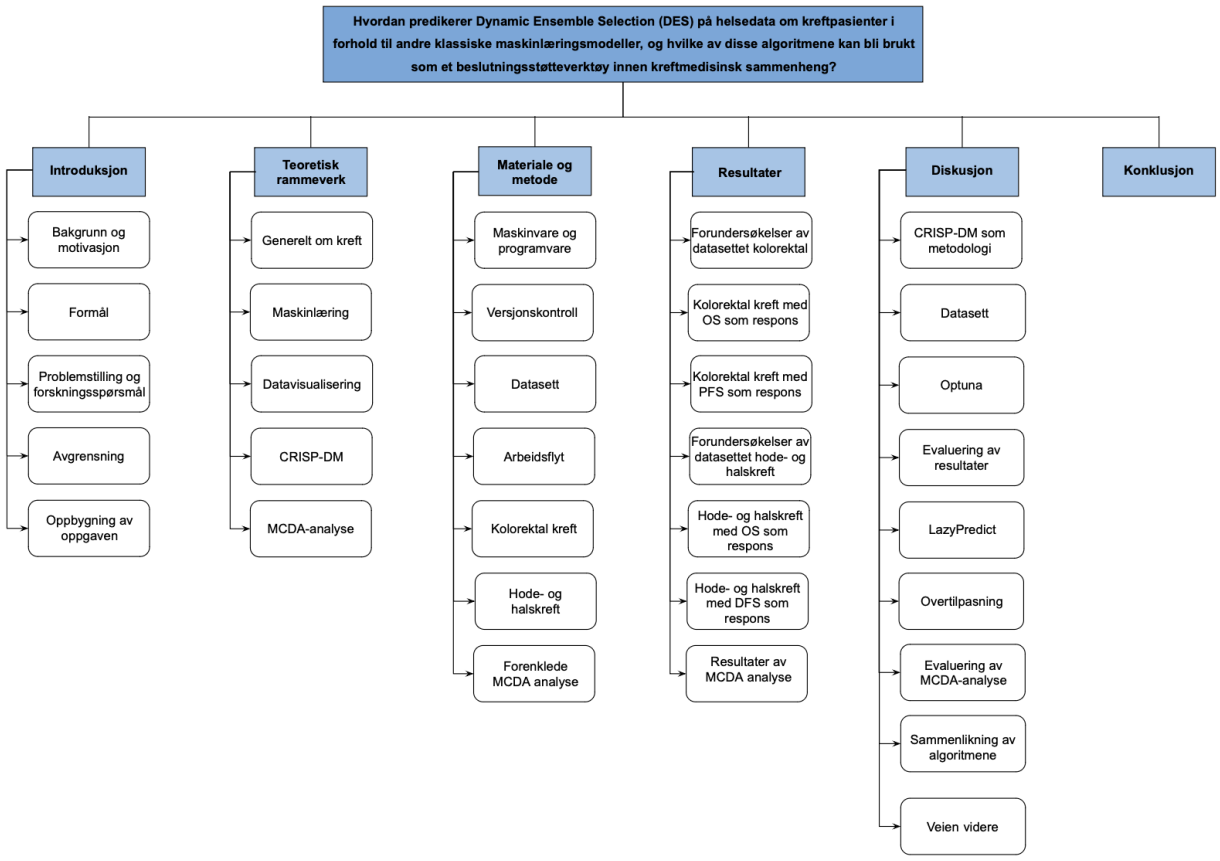
I samarbeid med oppgavens fire veiledere har det blitt forankret enkelte avgrensninger for å sikre en tilfredsstillende besvarelse av problemstilling og forskningsspørsmålene i oppgaven. Dette har blitt gjort for å ivareta oppgavens formål. Noen av disse avgrensningene kan også fungere som inspirasjon for fremtidig forskning, og anbefalinger for slik forskning kan bli funnet i kapittel 5.9.

Oppgaven har et primært fokus på bruken av maskinlæring innen helsesektoren. Det er begrenset antall lignende tilfeller med samme grad av kompleksitet som er undersøkt, og det er ikke kjent om de anbefalingene som utarbeides i denne oppgaven er generaliserbare. Tidligere masteroppgaver har undersøkt og analysert de datasettene som benyttes i denne oppgaven. Imidlertid har ingen av de tidligere gjennomførte masteroppgavene tatt for seg en lignende problemstilling, som spesifikt fokuserer på Dynamic Ensemble Selection (DES).

Det er viktig å påpeke at både elimineringsprosessene som er utført og de satte evalueringskriteriene for MCDA-analysen, er gjort av mennesker. Det kan derfor være en viss grad av subjektivitet i vurderingene. Det er også verdt å merke seg at elimineringen av variabler kan føre til tap av informasjon som kan være nyttig for modelleringen. Derfor er det viktig å utføre grundige vurderinger og balansere mellom å eliminere irrelevante variabler og å beholde informasjonsrike variabler. Det er dessuten av betydning å påpeke at mangelen på dedikert kontaktperson ved et sykehus har gitt opphav til utfordringer som kan påvirke kvaliteten på resultatene fra de ulike testene og analysene.

1.5 Oppbygning av oppgaven

Med hensikten om å systematisere oppgaven og besvare problemstillingen har det blitt utarbeidet en arbeidsnedbrytningsstruktur, også kjent som Work Breakdown Structure (WBS). En WBS-struktur er et verktøy for å dele i en hierarkisk nedbrytning av arbeidsoppgaver som prosjektet skal utføre, og kan bidra til å sikre en god oversikt over prosjektets omfang (Rolstadås et al., 2020). Figur 1 illustrerer denne arbeidsnedbrytningsstrukturen. I denne oppgaven er utgangspunktet i problemstillingen som deretter brytes ned i seks kapitler med tilhørende underkapitler.



Figur 1: WBS-struktur, oppbygning av oppgaven

Kapittel 1 omhandler introduksjon og bakgrunn for masteroppgaven. Deretter presenteres Kapittel 2, teorikapittel, som skal gi det solide og grundige rammeverket for teorien som legger grunnlaget av oppgaven. I Kapittel 3 gir en beskrivelse av materialet og metodene som er benyttet for å modellere datasettene med flere maskinlæringsalgoritmer. Deretter presenteres forskningsresultatene i Kapittel 4, etterfulgt av en diskusjon av resultatene og forslag til videre arbeid i Kapittel 5. Til slutt, i Kapittel 6, presenteres oppgavens konklusjon.

Kapittel 2

Teoretisk rammeverk

2.1 Generelt om kreft

Kreft er en sykdom som skyldes ukontrollert vekst av anormale celler i kroppen. Disse cellene kan invadere nærliggende vev og kan også spre seg til andre deler av kroppen gjennom blant annet blodbanen (Grimsrud et al., 2021). Hvis det bare er noen få syke celler, kan kroppen kanskje klare å tilintetgjøre dem på egenhånd ved hjelp av signalstoffer eller immunforsvaret. De vanligste kreftformene blant den norske befolkningen er tykktarms-, lunge-, bryst- og/eller hudkreft (Larsen et al., 2014).

Årsakene til kreft kan klassifiseres i tre hovedgrupper: 1) genetiske og fysiologiske faktorer, 2) livsstil og adferd, og 3) ytre faktorer som miljø og sosioøkonomiske forhold (NOU 1997: 20). Arvelige faktorer spiller en dominerende rolle i de fleste tilfeller av kreft. I sum viser forskning at årsakene til kreft er komplekse og sammensatte, og krever en helhetlig tilnærming for å redusere risikoen og forebygge sykdommen (NOU 1997: 20). I denne avhandlingen vil to forskjellige krefttyper bli undersøkt i detalj: kolorektal kreft og hode- og halskreft.

2.1.1 Kolorektal kreft

Tykktarms- og endetarmskreft, også kalt kolorektal kreft, er en type kreftsvulst som oppstår i tykktarmen *(eng. kolocancer*) eller endetarmen *(eng. rektalcancer)* (Folkehelseinstituttet, 2020). Det er en av de mest vanlige former for kreft, og risikoen øker med alderen. Symptomer inkluderer blod i avføringen, endringer i tarmvanene, vekttap og magesmerter. Behandlingen avhenger av alvorlighetsgraden av sykdommen, men kan inkludere kirurgi, strålebehandling eller kjemoterapi (OUS, 2023). Det er viktig å få regelmessige screeningtester for å oppdage tarmkreft i sine tidlige stadier, når det er enklere å behandle.

Ifølge en rapport utgitt av Kreftregistret er tykk- og endetarmskreft et av de hyppigste kreftformene her til lands for begge kjønn (Larsen et al., 2023). Selv om forekomsten av denne krefttypen øker, viser statistikken til at mange flere av dem som for denne diagnosen lever lengere. Samtidig som forekomsten øker, lever mange flere av de som får denne

tarmkreft lenger. I 2021 fikk omtrent 1697 norske kvinner og 1517 norske menn tykktarmskreft. Norge diagnostiseres det omtrent 5 000 nye tilfeller av tykktarmskreft hvert år, og sykdommen er en av de vanligste kreftformene i landet (Kreftregisteret, 2022).

2.1.2 Hode- og halskreft

Hode- og halskreft er en samlebetegnelse på ulike type kreft som oppstår i hode og halsregionen, inkludert munnhulen (cavum oris), svelget (pharynx), nesen, bihulene og strupehodet (Kreftforeningen, 2023). Røyking og alkoholforbruk er de viktigste risikofaktorene for hode- og halskreft, og infeksjoner med visse typer virus, som HPV (Humant Papiliomavirus), kan også øke risikoen (Nygård, u.å).

Ifølge statistikk fra Kreftforeningen ble 737 tilfeller av kreft i leppe, munnhule, og svelg registrert i Norge i løpet av et år (Kreftregisteret, 2021). Blant disse tilfellene var 477 menn og 260 kvinner. I samme periode ble 47 tilfeller av kreft i nese/bihule diagnostisert, hvorav 25 var menn og 22 var kvinner. 123 personer er diagnostisert med strupekreft, hvorav 108 er menn og 15 er kvinner. Det har vært en svak økning i antall nye tilfeller av hode- og halskreft i Norge de siste årene (Kreftregisteret, 2021).

2.2 Maskinlæring

Maskinlæring (ML) er en del av datavitenskapen som omhandler å bruke algoritmer og statistiske modeller for å gi datamaskiner muligheten til å lære og forbedre ytelsen på oppgaver, basert på tilbakemeldinger og erfaringer fra tidligere data (Raschka & Mirjalili, 2019). I maskinlæringsfaglig sjargong kalles det «å lære» for «å trene opp» en modell. For å utføre denne treningen brukes en eller flere maskinlæringsalgoritme(r). Disse algoritmene kan ha ulike formål og blir brukt til å trene en prediksjonsmodell på data. Det kreves altså datasett som kan brukes til å lage en modell og dette datasettet deles oftest opp i to: et treningssett og et testsett (Raschka & Mirjalili, 2019). Treningssett for å trene modellen og testsett for å evaluere modellen.

2.2.1 Læringsteknikker innen maskinlæring

For at maskinen skal være i stand til å utføre komplekse oppgaver bør den ved hjelp av maskinlæring læres opp (Raschka & Mirjalili, 2019). Det finnes flere typer læringsteknikker, og de mest vanligste er overvåket læring (*eng. supervised learning),* ikke-overvåket læring

(eng. unsupervised learning) og forsterkningslæring *(eng. reinforcement learning).* På norsk er begrepene overvåket og ikke-overvåket læring også kjent som veiledet og ikke-veiledet læring.

Overvåket læring

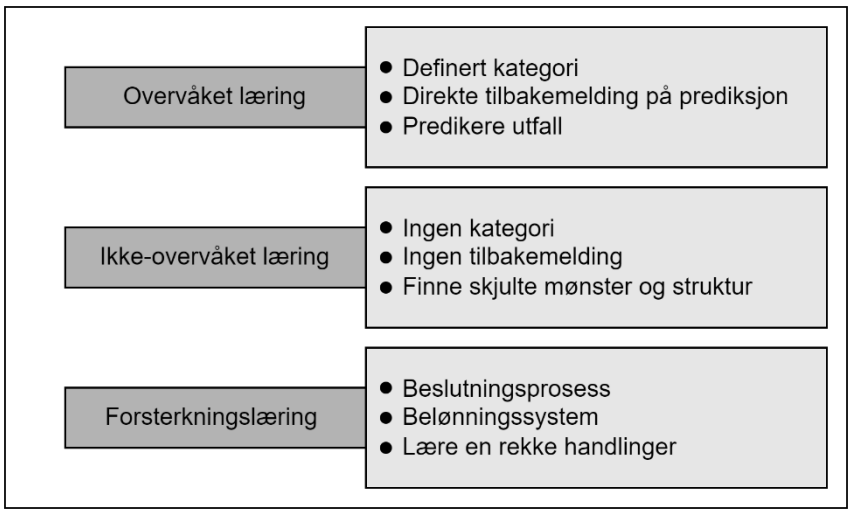
Overvåket læring en læringsteknikk som innebærer å trene en modell ved å gi den data med riktig svar, slik at modellen kan generalisere og forutsi nye data (Raschka & Mirjalili, 2019). Det er altså en respons som har hjulpet modellen til å trene for å predikere responsen for nye data. Denne læringsteknikken består av tre faser. I den første fasen sørger man for å benytte et kategorisert datasett. I den andre fasen velger man egenskapene ved dataene, basert på datatype og relevans, som kan brukes til læring. I den siste fasen bygger man en modell basert på disse egenskapene som er definert, til å gi merkelapp på hva som er korrekt svar.

Ikke-overvåket læring

I ikke-overvåket læring gir man data til modellen uten å gi de riktige svarene, slik at den kan finne mønstre og strukturer i dataen på egenhånd (Raschka & Mirjalili, 2019). Ikke-overvåket læring har dermed ingen respons som den kan lene seg opp mot. Denne læringsteknikken består også av tre faser. I den første fasen sørger man for å innhente data, og i dette tilfelle vil det være fordelaktig med en viss grad av likheter og mønstre i datagrunnlaget. I den andre fasen vil systemet avdekke likhetene og mønstrene, før den i siste fase vil lage en modell som er i stand til å skille og gjenkjenne mønstrene.

Forsterkningslæring

Forsterkningslæring er en teknikk som brukes for å trene modeller til å ta beslutninger basert på en serie av interaksjoner, hvor modellen blir belønnet eller straffet basert på utfallet av beslutningen (Raschka & Mirjalili, 2019). Dette av betydningen om at algoritmene finner den beste strategien ved å feile, prøve og bli korrigert av sine egne erfaringer underveis. Forsterket læring handler derfor om å utføre handlinger som kan bidra til å maksimere belønning i en gitt situasjon. Figur 2 presenterer et kort sammendrag av læringsteknikkene.



Figur 2: Et kort sammendrag av de presenterte læringsteknikkene. Inspirert av (Raschka & Mirjalili, 2019)

I denne oppgaven blir overvåket læring brukt som læringsteknikk.

2.2.2 Overvåket læring i praksis

I overvåket læring er det typisk å utføre en datasettdekomponering i to matriser, presentert som *X* og y (Raschka & Mirjalili, 2019). Som vist i figur 3, inneholder matrisen *X* alle variablene og observasjonene i datasettet som skal brukes for å trene en modell. I kontrast til *X,* består matrisen y av en liste over de respektive responsverdiene til hver observasjon.

X

Observasjon	Variabel 1	Variabel 2	·····	Variabel m
1	X_{11}	X_{12}	·····	X_{1m}
2	X_{21}	X_{22}	·····	X_{2m}
⋮	⋮	⋮	⋮	⋮
n	X_{n1}	X_{n2}	·····	X_{nm}

y

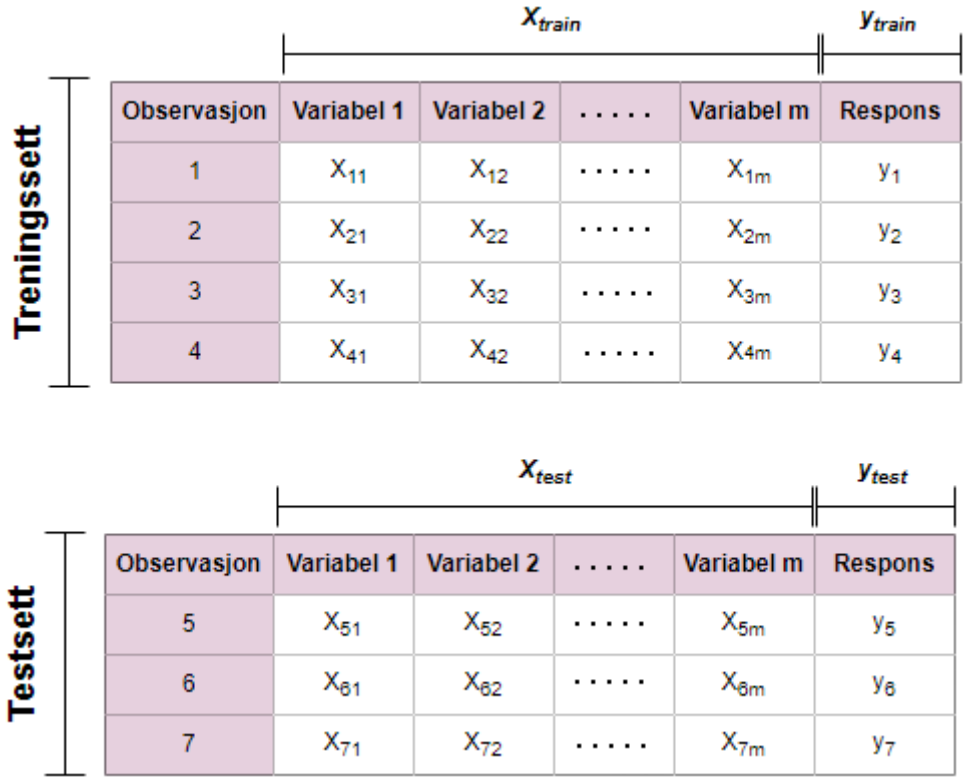
Observasjon	Respons
1	y_1
2	y_2
⋮	⋮
n	y_n

Figur 3: Presenterer hvordan et datasett kan splittes i en X-matrise med en korresponderende liste y med responsverdi for n prøver

Ved modellering er det vanlig å dele datasettene inn i et treningssett og et testsett/valideringstest. Av disse brukes treningssettet til å trene klassifiseringsmodellene i kontrast til at testsettet brukes for predikere ytelsen på modellen (Fosse et al., 2020).

Testsettet brukes altså for å sjekke om modellen klarerer å predikere riktig og tildele riktig klasse til ny data basert på hva modellen har lært fra treningssettet (Boyd & Crawford, 2012).

Treningssettet består av X_{train} og y_{train} , der X_{train} inneholder deler av X-matrisen og y_{train} inneholder responsene som hører til observasjonene i X_{train} (Géron, 2022). På lik linje som treningssettet, består testsettet av en matrise, vanligvis kalt X_{test} og en liste kalt y_{test}, med responsene til observasjonene i X_{test}. Matrisen med X_{test} inneholder de gjenværende observasjonene fra matrise X som ikke er inkludert i X_{train}. Figur 4 illustrerer et eksempel på et datasett som består av syv observasjoner, hvor de første fire observasjoner har blitt trukket ut som treningssett og de tre gjenværende observasjoner har blitt trukket ut som testsett.



Figur 4: Et datasett med syv prøver er splittet i trening og testsett med m variabler. Disse kan brukes for å trene en modell med treningssettet og evaluere modellen med testsettet.

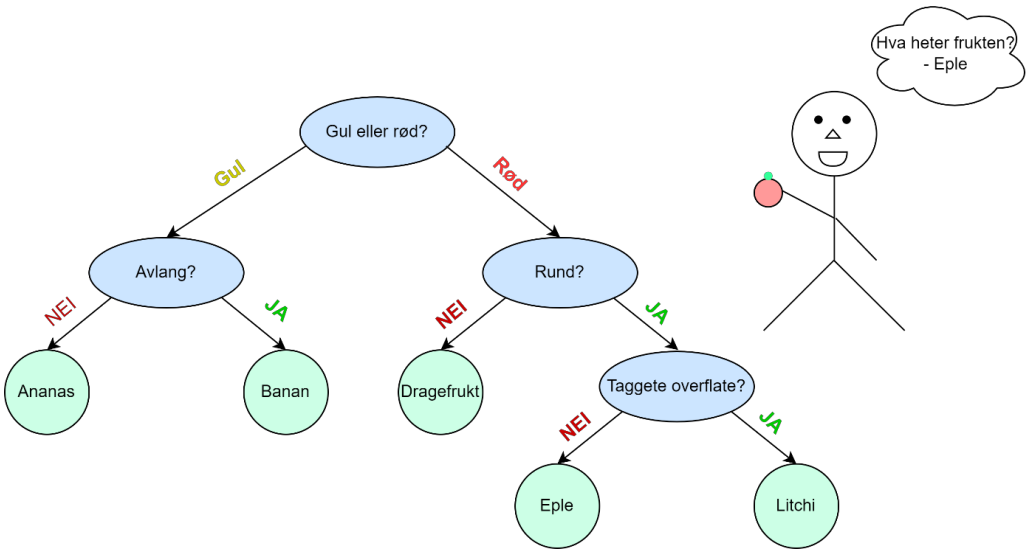
2.2.3 Klassifiseringsalgoritmer

Det eksisterer en mangfoldighet av klassifiseringsalgoritmer som kan benyttes for å utvikle modeller innenfor maskinlæring. Det er klassifiseringsalgoritmer som hjelper modeller til å predikere hvilken kategori en ny observasjon tilhører basert på datasettene modellene er trent med tidligere. Hovedformålet med algoritmene er å skille mellom to eller flere klasser. Nedenfor blir en detaljert beskrivelse av hvilke klassifiseringsalgoritmer som er brukt i denne oppgaven, samt en tabelloversikt over fordelene og ulempene for den beskrivende algoritmen gitt. Det er imidlertid viktig å huske på at disse kan variere betydelig avhengig av egenskapene til datasettet og det spesifikke problemet som skal løses. Algoritmene som er benyttet i oppgaven, er som følger:

Random forest

Random forest algoritmen er en ensemble-teknikk for å utføre regresjon og klassifisering ved hjelp av en samling av beslutningstrær (Raschka & Mirjalili, 2019). I dette tilfelle, blir ordlyden «ensemble» brukt i betydningen av at flere individuelle modeller med forskjellige styrker og svakheter blir kombinert for å bedre blant annet nøyaktighet (Raschka & Mirjalili, 2019). Et beslutningstre tar utgangspunkt i et treningssett som består av en serie med spørsmål, hvor disse informasjonene bistår med nedbrytningen av dataen. Videre bruker algoritmen en teknikk som heter «feature bagging», der det velges en tilfeldig undermengde av egenskaper fra et utvalg for å trene hvert tre. Dette bidrar også til å øke variasjonen mellom trærne og redusere overtilpasning (Raschka & Mirjalili, 2019). En detaljert forklaring om overtilpasning er gitt i kapittel 2.2.4 og i begrepslisten.

Random forest er en av de mest foretrekkende algoritmene i maskinlæring, da den har evne til å brukes for klassifiseringsproblemer, samt bildegjenkjenningsproblemer (Géron, 2022). Algoritmen har også evnen til å håndtere store mengder data, i tillegg til å gi høye nøyaktigheter. Modeller med random forest kan enkelte ganger være tilbøyelig for overtilpasning når treningsdataene blir splittet helt til alle bladene i trærne. For å unngå dette er det viktig å sette en grense på hvor dypt splittelsen kan gå.



Figur 5: Illustrerer på et eksempel random forest, beslutningstre som skal hjelpe gutten med å finne hvilken frukt han holder.

Figur 5 illustrerer et eksempel med en gutt som ikke kan navnet på en frukt. Han følger et bestemmelsestre for å finne ut av hvilken frukt han har tatt fra en fruktkurv som inneholder flere forskjellige fruktsorter. Dette multiklasseproblemet kan løses ved å stille en serie med spørsmål om frukten. I dette eksempelet holder gutten et eple, og ved å følge bestemmelsestreet kan han identifisere navnet på frukten.

Tabell 1: Fordeler og ulemper med random forest

Fordeler	Ulemper
• Kan anvendes både for klassifiserings- og regresjonsoppgaver • Kan behandle både numeriske og kategoriske egenskaper • Kan håndtere store og komplekse datamengder med mange egenskaper, spesielt støy. • Relativt enkel å sette opp modellen og kjent for å gi gode nøyaktighetsresultater ved klassifisering	• Kompleksitet, krever mye datamaskinkraft og kan være tidkrevende å kjøre for store datasett • Til tider være vanskelig å tolke resultatene, spesielt når det gjelder å forstå de interaksjonene mellom egenskapene

I denne studien benyttes *RandomForestClassifier* fra maskinlæringsbiblioteket *scikit-Learn* (Pedregosa et al., 2011).

Logistisk regresjon (*eng. Logistic regression)*

Logistisk regresjon er en lineær maskinlæringsalgoritme som blir anvendt til å løse binære klassifiseringsproblemer. Modeller med logistisk regresjon er enkelt å implementere samt en av de mest brukte i maskinlæring (Raschka & Mirjalili, 2019). Selv om logistisk regresjon er en binær klassifiseringsalgoritme, kan den bli utvidet til å løse multiklassifiseringsproblemer ved å blant annet bruke teknikker som en-mot-resten (*eng. OvR, One-vs.-Rest*) eller en-mot-alle (*eng. OvA, One-vs.-All*).

Algoritmen beregner sannsynligheten ved hjelp av en logistisk funksjon for ulike utfall i en binær problemstilling. Funksjonen kalles for *logit function* (Raschka & Mirjalili, 2019) og er gitt ved

$$logit = \log\left(\frac{p}{1-p}\right) \tag{2.1}$$

hvor *p* står for sannsynligheten for positivt utfall, og *log* definerer den naturlige logaritmen. Funksjonen er bygget opp av odds ratio som tar for seg sammenhengen mellom to hendelser, og hvis begge hendelsene er like store, vil odds ratio være lik 1. Logaritmefunksjonen tar inn verdier mellom 0 og 1 (Raschka & Mirjalili, 2019). Disse verdiene blir videre transformert som gjør at hele nummerspekteret blir dekket. Den inverse formen av likningen kjent som sigmoid funksjonen, benyttes til å predikere sannsynligheten for at en observasjonsprøve tilhører en spesifikk klasse. Sigmoid funksjonen er definert som,

$$\phi(z) = \frac{1}{1+e^{-z}} \tag{2.2}$$

hvor *z* representerer ulike prøvene som er matet inn i modellen. Resultatene fra sigmoid funksjonen er en verdi mellom 0 og 1, og gir sannsynligheten for at observasjonen tilhører klasse 1. Dersom $\phi(z)$ er større enn terskelverdien, vil en terskelfunksjon tildele observasjonen klasse 1, og hvis ikke klasse 0.

Tabell 2: Fordeler og ulemper med logistisk regresjon

Fordeler	Ulemper
• Godt egnet for binære klassifiseringsproblemer • Kan være enklere å forstå, sammenlignet med mer komplekse maskinlæringsalgoritmer • Kan håndtere både kontinuerlige og kategoriske prediksjoner	• Kan være følsom for ekstremverdier, som kan ha en stor innvirkning på modellens prediksjonsytelse • Fungerer best når dataene kan separeres lineært. Dersom dataene ikke kan separeres, kan det påvirke ytelsen negativt

I denne studien benyttes *LogisticRegression* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Kvadratisk diskriminantanalyse, QDA (*eng. Quadratic Discriminant Analysis)*

Kvadratisk diskriminantanalyse (QDA) er en statistisk teknikk som brukes for klassifiseringsproblemer. QDA er en av flere varianter av Discriminant Analysis (DA), som brukes til å analysere og identifisere forskjeller mellom to eller flere grupper (Bruce et al., 2020). QDA er en ikke-lineær metode som er spesielt nyttig når det er komplekse sammenhenger mellom inputvariablene.

Målet med QDA er å forutsi klassemedlemskapet til en ny observasjon basert på en rekke forklarende variabler eller prediksjoner. QDA antar at sannsynlighetsfordelingen til de forklarende variablene for hver klasse er normalfordelt, med gjennomsnitt og kovariansmatriser som kan variere mellom klassene. Med en kjent klassefordeling for en gruppe observasjoner, estimerer QDA parameterne for normalfordelinger for hver klasse ved hjelp av maksimal estimering. Sannsynligheten og prediksjonen blir utført av Bayes teoremet,

$$P(y = k \mid x) = \frac{P(x \mid y = k)P(y = k)}{P(x)} \tag{2.3}$$

Her er y klassetilhørigheten, x er verdiene til de forklarende variablene, k er en av K mulige klassene, P (y = k | x) er sannsynligheten for at x hører til klassen k gitt verdiene til de forklarende variablene (Pedregosa et al., u.å).

Tabell 3: Fordeler og ulemper med QDA

Fordeler	Ulemper
• Kan håndtere ikke-lineære sammenhenger • Kan håndtere variansforskjeller mellom klasser • Effektiv for høydimensjonale data	• Kan være mer utsatt for overtilpasning • Vil prestere bedre med et stort antall observasjoner

I denne studien benyttes *QuadraticDiscriminantAnalysis* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Gaussian Naive Bayes, GaussianNB

Gaussian Naive Bayes (GaussianNB) er en av de mest brukte klassifiseringsalgoritmene innenfor maskinlæring. Algoritmen tilhører den såkalte Naive Bayes-familien av klassifiseringsalgoritmer, som alle bruker Bayes teorem som grunnlag for klassifiseringen (VanderPlas, 2016). GaussianNB er spesielt effektiv når det gjelder å håndtere kontinuerlige verdier som er normalfordelt.

Metoden bruker en modell som antar at alle forklarende variabler i datasettet er uavhengige av hverandre gitt klassetilhørigheten. Dette er en naiv antakelse, derav navnet Naive Bayes, men ofte viser det seg at denne antakelsen ikke er så naiv som den høres ut (VanderPlas, 2016). GaussianNB antar at forklarende variabler for hver klasse er normalfordelt, og formelt kan denne sannsynligheten uttrykkes som presentert i formel 2.3.

For å beregne sannsynligheten P(x | y = k) i formel 2.3 antar GaussianNB at forklarende variabler for hver klasse er normalfordelt, og bruker gjennomsnittsverdien og standardavviket til hver variabel i klassen til å estimere den betingede sannsynligheten. Deretter beregnes den a priori sannsynligheten P(y = k) ved å telle antall observasjoner i hver klasse og dele på totalt antall observasjoner i datasettet (VanderPlas, 2016).

Tabell 4: Fordeler og ulemper med GaussianNB

Fordeler	Ulemper
• Kan være enkel å implementere og forstå • Effektiv på store datasett og høydimensjonale data • Robust med overfitting og kan håndtere ubalanserte datasett • Krever ikke store datamengder for å gi resultater sammenlignet med andre klassifiseringsalgoritmer	• Kan være følsom for ekstremverdier • Resultatene kan være misvisende om dataene ikke er normalfordelt • Kan ha problemer med å håndtere skjeve datasett, der en klasse dominerer

I denne studien benyttes *GaussianNB* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Nearest centroid

Nearest centroid er en enkel klassifiseringsalgoritme som har sin metode i først definere centroid (Raschka & Mirjalili, 2019). Centroid er et sentral punkt for hver klasse i datasettet som beregnes og settes ved trening. Deretter måles avstanden mellom datapunktet og det sentrale punktet, centroid. Algoritmen klassifiserer datapunktet for den centroid-klassen som har minst avstand. Avstandene kan regnes ved hjelp av forskjellige sammenhenger, hvor de mest vanligste er euklidsk avstand *(eng. euclidean distance)* og manhattan avstand *(eng. manhattan distance)*. Euklidsk avstand kan utrykkes som

$$d(x,y) = \sqrt{\sum_{j=1}^{m} \left(x_j - y_j\right)^2} \tag{2.4}$$

, og manhattan avstand er definert som

$$d(x,y) = \sum_{j=1}^{m} |x_j - y_j| \tag{2.5}$$

, hvor *x* står for vektor som representer den nye prøven, og *y* står for en vektor som representerer centroidene for en av de klassene i datasettet (Raschka & Mirjalili, 2019).

Tabell 5: Fordeler og ulemper med nearest centroid

Fordeler	Ulemper
• Enkel og rask implementering • Effektiv på store datasett og høydimensjonale data • Kjent for å ha rask utførelsestid • Egnet for både numeriske og kategoriske data	• Følsom for ekstremverdier og støy • Overlappende klynger (*eng. cluster*) kan gi redusert ytelse for modellene • Kan ha problemer med ubalansert data. Fungerer best når klyngen har omtrent samme størrelse. • Kan ha problemer med å yte godt om datasettet har komplekse klyngemønstre.

I denne studien benyttes *nearest centroid* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Dynamic Ensemble Selection, DES

Dynamic Ensemble Selection (DES) er en maskinlæringsmetode som primært tar sikte på å forbedre prediksjonsytelsen til ensemble-modeller ved å velge en optimal kombinasjon av undermodeller fra en større samling av modeller. DES-teknikker er ment til å være svært effektive i å forbedre ytelsen til ensemble-læringsmodeller, spesielt når det gjelder å håndtere komplekse og ustrukturerte datasett. DES-teknikker kan også være spesielt nyttige i situasjoner der dataene endrer seg over tid eller når man har begrensede ressurser til rådighet (Cruz et al., 2020).

Tabell 6: Fordeler og ulemper med Dynamic Ensemble Selection (DES)

Fordeler	Ulemper
• Kan forbedre ytelsen til individuelle basismodeller • Kan håndtere endringer i datasettet • Kan redusere feil og usikkerhet i klassifiseringsbeslutninger	• Kan føre til økt tids- og ressursbruk • Har tendens til å overtilpasse raskt • Krever ofte mye tid og arbeid for å optimalisere • Kan føre til økt kompleksitet i modellen

I denne studien benyttes biblioteket *DESlib*, et ensemblelæringsbibliotek som fokuserer på implementeringen av de nyeste teknikkene for dynamisk klassifisering og ensemblevalg. Det er verdt å merke at denne metoden er *«work in progress»* som vil si at den er ikke ferdigutviklet (Cruz et al., 2020). På bakgrunn av at DESlib-biblioteket er ett nytt bibliotek og fortsatt i utvikling, er det også mangel på studier og litteratur som oppgaven kan støtte seg opp mot. Nedenfor presenteres algoritmene fra DESlib-pakken som har blitt avendt i denne oppgaven.

KNearest Oracle – Eliminate, KNORA-E

KNearest Oracle - Eliminate (KNORA-E) er en metode innenfor ensemble læring som brukes til å eliminere dårlige individuelle modeller for å forbedre prediksjonsytelsen (Cruz et al., 2020). Algoritmen består av to hovedkomponenter: K-nearest neighbors (KNN) og elimineringsteknikker. KNORA-E-algoritmen anvender ensemble-teknikker for å integrere flere KNN-modeller med varierte avstandsfunksjoner og antall naboer med det formål å bedre resultater. For å benytte algoritmen må man først bygge flere modeller av k-nearest neighbors (KNN) med forskjellige hyperparametere og/eller avstandsfunksjoner. Disse modellene brukes deretter til å klassifisere et nytt datapunkt ved å ta en avstemning over alle modellene og deres respektive klassifiseringer (Souza et al., 2018).

For eksempel, hvis man har tre KNN-modeller som klassifiserer et datapunkt som klasse A, klasse B, og klasse A, så vil KNORA -E klassifisere datapunkt som klasse A, da den velger den klassen som er mest populær blant alle modellene.

Formelt kan KNORA-E-algoritmen beskrives ved å følge disse trinnene:

1. Velg en mengde KNN-modeller med forskjellige hyperparametere og/eller avstandsfunksjoner
2. For hvert datapunkt i testsettet:
   a. Bruk hver KNN-modell til å finne de k nærmeste naboene til datapunktet
   b. Ta en avstemning over alle KNN-modellene for å bestemme klassen til datapunktet
3. Evaluer ytelsen til algoritmen ved å beregne metrikker som nøyaktighet, presisjon og gjenkalling

KNORA-E-algoritmen kan brukes til både binær og multiklassifisering, og den har vist seg å være effektiv i mange anvendelser (Souza et al., 2018). Algoritmen kan tilpasses datatyper og problemområder, og det er viktig å utføre en grundig evaluering av algoritmens ytelse for å sikre at den passer til det spesifikke problemet som skal løses.

K-Nearest Oracle – Union, KNORA-U

K-Nearest Oracle - Union (KNORA-U) er en tilsvarende algoritme som KNORA-E, der forskjellen i metoden er at den er basert på en kombinasjon av k-nearest neighbor (KNN) algoritmen og en metode som kalles Union (Souza et al., 2018). Union er en teknikk som brukes til å håndtere situasjoner der det er flere mulige klassifikasjoner for en observasjon. Dette kan være nyttig når modellene har forskjellige styrker og svakheter, og man ønsker å dra nytte av den totale informasjonen som er tilgjengelig (Souza et al., 2018).

KNORA-U kombinerer disse to teknikkene ved å først bruke KNN for å finne de K nærmeste naboene til en ukjent observasjon. Deretter brukes Union til å velge den mest sannsynlige klassifikasjonen av disse naboene. Dette gjøres ved å kreve enighet mellom forskjellige KNN-modeller som bruker ulike avstandsmålinger eller ulike verdier for K. Den endelige klassifikasjonen er deretter basert på flertallet av de enige klassifikasjonene fra modellene (Souza et al., 2018).

Et eksempel på KNORA-U kan være klassifisering av epler og appelsiner. Anta at modellen med KNORA-U i dette eksempelet har trent med tre separate modeller basert på et datasett som består av informasjon om epler og appelsiner. Hver av disse modellene vil gi en individuell prediksjon for er nytt frukt som skal predikeres. Konsensusprediksjonsresultat genereres ved å benytte k-nærmeste naboer til å identifisere de nærmeste nabofruktene til den aktuelle frukten som skal predikeres. Dersom flertallet av de nærmeste naboene predikerer at den nye frukten er et eple, vil modellen konkludere med at den nye frukten er et eple som sin endelige prediksjon.

Tabell 7: Fordeler og ulemper med KNORA-E og KNORA-U

Fordeler	Ulemper
• Bedre klassifikasjonsresultater enn vanlig KNN fordi den kombinerer flere modeller og kan ignorere avvikende naboer som kan gi klassifisering • Bedre kontroll over valget av K-parameteren for KNN ved å bruke flere modeller med forskjellige verdier for K	• Kan være enkel å implementere, men tids- og ressursbruken er høy • Kan være følsom og sensitiv for «missing values» • Kan være sensitiv om dataene inneholder støy eller avvik • Kan være sårbar for ubalanserte datasett, som kan føre til overtilpasning

I denne studien benyttes KNORA-E og KNORA-U fra DESlib-biblioteket (Cruz et al., 2020).

Dynamic Ensemble Selection Performance, DES-P

Dynamic Ensemble Selection Performance (DES-P) er en metode som velger alle basis klassifisere som oppnår en klassifiseringsytelse som er høyere enn *random classifier (RC)* (Cruz et al., 2020). Ytelsen til den tilfeldige RC er definert som

$$RC = \frac{1}{L} \tag{2.6}$$

, der L er antall klasser.

En av de største utfordringene med DES-P, er at det ikke ligger nok studie eller litteratur om algoritmen til å innhente nødvendig informasjon om hvordan den fungerer. Dette kan være av at DESlib pakken fortsatt er under utviklingsfasen.

Tabell 8: Fordeler og ulemper med DES-P

Fordeler	Ulemper
• Kan forbedre nøyaktigheten til klassifiseringsmodellen ved å velge de beste modellene i ensemblet og kombinere dem • Kan hjelpe modellen med å generalisere bedre ved å ta hensyn til mangfoldet i ensemblet og unngå overtilpasning	• Kan være ressurskrevende for å trene og validere, siden den krever å kjøre flere modeller • Avhengig av kvaliteten på datasettet og kan ikke garantere nøyaktighet dersom datasettet er liten • For mange modeller i ensemble kan være til potensiell overtilpasning

I denne studien benyttes DES-P fra DESlib-biblioteket (Cruz et al., 2020).

2.2.4 Modelltilpasning

I de fleste tilfellene kan en modell gi gode resultater når den blir trent på treningsdata. Imidlertid er det viktig å være oppmerksom på at en modell også kan bli overtilpasset (*eng. overfitting)* eller undertilpasset (*eng. underfitting),* og dermed resultere i dårligere prediksjonsytelse (Raschka & Mirjalili, 2019). En overtilpasset modell vil ha vanskeligheter med å generalisere til nye og ukjente data på grunn av en altfor sterk tilpasning til treningsdataene. På den annen side vil en undertilpasset modell ha begrensninger i sin evne til å fange opp viktige trekk med både trenings- og testdata. Dette vil dermed resultere i dårlig ytelse på alle typer data, samt ha høy bias i sin modell (Owen, 2022).

En av de viktigste faktorene som påvirker ytelsen til en klassifiseringsalgoritme er dens hyperparametere (Owen, 2022). For å oppnå best mulig ytelse, og for å unngå over- eller undertilpasning, er det essensielt å utføre hyperparameteroptimalisering. Ved å optimere parameterne vil modellen oppnå en optimal konfigurasjon som vil føre til forbedret ytelse og økt generaliserbarhet for fremtidige, ukjente datasett. Noen eksempler på slike hyperparametere kan være antall trær i et beslutningstre (Random Forest), antall naboer i k-nearest neighbor (KNN) og parameteren C i logistisk regresjon som tar for seg reguleringstyrken for en modell (Pedregosa et al., 2011).

2.2.5 Algoritmer for inspeksjon av ekstremverdier

Ekstremverdier er verdier som signifikant avviker fra de øvrige verdiene i et datasett (Jafari, 2022). Avvik vil være i form av at det er svært høye eller lave verdier i forhold til de øvrige verdiene. Viktighet av inspeksjon og behandling av ekstremverdier for denne oppgaven er dekket detaljert under steg 5 i kapittel 3.4.2.

Det finnes mange ulike teknikker som kan brukes for å undersøke ekstremverdier og i denne oppgaven har PyOD biblioteket blitt benyttet (Zhao et al., 2019) .PyOD er spesielt utviklet for å oppdage ekstremverdier i både enkle og komplekse datasett. Pakken inneholder en rekke algoritmer som den kan bruke for å finne ekstremverdier. Nedenfor følger en detaljert beskrivelse av algoritmene som har blitt brukt i PyOD for å finne mulige ekstremverdier i datasettene for oppgaven.

K-nærmeste naboer, KNN (*eng. K-Nearest Neighbor)*

K-nærmeste naboer (KNN) er en enkel algoritme innenfor maskinlæring som brukes for både klassifisering og regresjon. KNN kan enten bli betraktet som en klassifiseringsalgoritme eller for å finne ekstremverdier (Atewan, 2022). For klassifisering vil KNN se på k-nærmeste naboene til et ukjent datapunkt, og merke det ukjente punkt som tilhørende den klassen som er mest vanlig blant disse k-naboene. Når det kommer til undersøkelse av ekstremverdier, måler algoritmen avstanden fra hvert datapunkt i datasettet til sine k-nærmeste naboer. Om et datapunkt har stor avstand til sine nærmeste naboer, vil datapunktet bli betraktet som en ekstremverdi.

I denne oppgaven vil KNN bli brukt som en metode for å undersøke ekstremverdier. Undersøkelsene har blitt kjørt med normalverdier (*eng. default parameters)*. Dette betyr at algoritmen har brukt fem naboer og minkowski som metrisk for å regne avstanden mellom k-nærmeste naboer. Minkowski er en av mange metoder for å regne avstand mellom punkter.

For å illustrere dette, kan man tenke seg at en har et datasett som inneholder informasjon om bolig og boligpriser. Formålet i dette eksempelet er å finne om det er noen enkelte datapunkter som skiller seg fra resten i datasettet. KNN-algoritmen vil da finne de k-nærmeste naboene for hvert datapunkt, for eksempel 5 naboer. For hvert datapunkt vil

naboenes gjennomsnittspris bli beregnet. Dersom boligprisen avviker betydelig fra gjennomsnittsprisen, vil datapunktet bli definert som en ekstremverdi.

Kort fortalt, KNN for undersøkelse av ekstremverdier består av følgende trinn (Auffarth, 2020).:

1. Velge en verdi for parameteren *k* som gir antall nærmeste naboer som skal brukes i klassifiseringen. I tillegg må avstanden mellom datapunktene defineres ved en metrisk funksjon, for eksempel euklidsk, Manhattan eller Minkowski avstand.
2. Lokalisere de *k* nærmeste naboene til datapunktet som skal sammenlignes opp mot. Dette gjøres ved å beregne avstanden fra det aktuelle datapunket til alle datapunkter i datasettet, og deretter velge de k-datapunktene med kortest avstand. Dette gjentas for alle datapunkter i datasettet.
3. Sammenligne gjennomsnittsverdien av k-nærmeste naboer og det aktuelle datapunktet. Dersom datapunktet avviker betydelig fra gjennomsnittet til naboene, vil datapunktet bli definert som en ekstremverdi.

Empirical Cumulative - based Outlier Detection, ECOD

ECOD (Efficient Cumulative - based Outlier Detection) er en metode for å identifisere avvik i datasett som ikke har kategorier. ECOD er altså en ikke-overvåket læring og bruker empirisk kumulativ fordelingsfunksjon (ECDF) for å finne mulige ekstremverdier (Zhao et al., 2019). Fordelingsfunksjon ECDF gir datasettet en sannsynlighetsfordeling basert på empiriske observasjoner. ECOD er algoritmene som er godt egnet for undersøkelser av ekstremverdier i datasett som har store mengder med data og høy dimensjon.

ECOD bruker fordelingsfunksjonen til finne fordelingen av normal data og for å finne datapunkter som ligger langt fra denne fordelingen. Algoritmen konstruerer to fordelingsfunksjoner hvor den første tar for seg normal data og den andre tar for seg unormal data. Dersom den unormale fordelingsfunksjonen blir klassifisert som høy verdi for et datapunkt av ECOD, blir datapunktet betraktet som ekstremverdi (Zhao et al., 2019).

For å illustrere dette, kan man tenke seg at en har et datasett som inneholder informasjon om alder og lønn i en arbeidsplass. ECOD vil først finne den empiriske kumulativ fordelingsfunksjon. Det kan blant annet utføres ved å sortere variabelen lønn i stigende

rekkefølge og deretter finne kumulativ sannsynlighet. Når dette er klart kan ECOD identifisere mulige ekstremverdier som har verdier som ligger langt fra hovedklyngen.

2.2.6 Validering av modellytelse

Validering av modellytelse er en kritisk del av maskinlæringsprosessen som hjelper til med å sikre at modellen kan gi pålitelige og nøyaktige resultater når den brukes til å løse en bestemt oppgave (Raschka & Mirjalili, 2019). Prosessen med validering av modellytelse innebærer vanligvis å bruke en kombinasjon av test- og treningssett for å evaluere modellens ytelse. Det finnes flere måter å evaluere modellenes ytelse på, avhengig av type problem og datasett som brukes. Noen vanlige målinger inkluderer nøyaktighet, presisjon, og følsomhet. Disse målingene kan hjelpe til med å bestemme hvor godt modellen kan klassifisere dataene i de ulike kategoriene, og hvor nøyaktige resultatene er.

Når det gjelder å evaluere modellytelse, kan det være hensiktsmessig å dele de predikerte resultatene inn i fire grupper (Raschka & Mirjalili, 2019).:

1. **Sann positiv prediksjon** (**TP,** *eng. true positive*): Antallet positive tilfeller som er korrekt klassifisert av modellen
2. **Sann negativ prediksjon** (**TN,** *eng. true negative*)**:** Antallet negative tilfeller som er korrekt klassifisert av modellen
3. **Falsk positiv prediksjon** (**FP,** *eng. false positive*): Antallet negative tilfeller som er feilaktig klassifisert som positive av modellen
4. **Falsk negativ prediksjon** (**FN,** *eng. false negative*): Antallet positive tilfeller som er feilaktig klassifisert som negative av modellen

Tabell 9 er en *confusion matrix* som gir en oversikt over de presenterte gruppene, og brukes for å evaluere resultatene av en klassifiseringsalgoritme (Raschka & Mirjalili, 2019). Matrisen gir en enkel oversikt over modellens ytelse ved å vise hvordan modellen har klassifisert prøvene i forhold til de faktiske klassene i datasettet.

Tabell 9: Confusion matrix presenterer de predikerte sanne og falske resultatene. Inspirert av (Raschka & Mirjalili, 2019)

	Predikert negativ (*eng. predicted negative*)	Predikert positiv (*eng. predicted positive*)
Sann negativ *(eng. actual negative)*	TN	FP
Sann positiv (*eng. actual positive*)	FN	TP

I denne oppgaven vil fire ulike valideringstyper av modellytelse bli vurdert, og det er:

Accuracy

I maskinlæring referer accuracy (nøyaktighet) til andelen av korrekte prediksjoner som en modell har gjort totalt sett. Det kan beregnes ved å dividere antall korrekte prediksjoner med det totale antallet prediksjoner som modellen har gjort (Raschka & Mirjalili, 2019).

Accuracy kan uttrykkes ved:

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \tag{2.7}$$

, der TP (*eng. True Positive)* er antall sanne positive prediksjoner, TN (*eng. True Negative)* er antall sanne negative prediksjoner, FP (*eng. False Positive)* er antall falske positive prediksjoner, og FN (*eng. False Negative)* er antall falske negative prediksjoner.

Accuracy kan være en nyttig metrikk for å evaluere en modell når datasettet er balansert. Hvis datasettet er ubalansert, der det ene resultatet er mer vanlig enn det andre, kan accuracy være en misvisende metrikk og andre metrikker som MCC, F1-score på begge klasser eller ROC-AUC være mer passende (Saleh, 2018).

F1-Score

F1-score er en vanlig metode å evaluere ytelsen til en klassifiseringsmodell. Det er en gjennomsnittlig metrikk som tar hensyn til både presisjon (*eng. precision,* PRE) og gjenkalling (*eng. recall,* REC) av en modell (Raschka & Mirjalili, 2019). F1-score er nyttig når modellen har et ujevnt distribuert datasett med en liten andel av positive tilfeller. Den gir en mer balansert og nøyaktig måling av modellens ytelse ved å ta hensyn til både presisjon og gjenkalling. Disse

metrikkene kan sammenlignes opp mot TPR og FPR, og kan beregnes ved å bruke følgende formel:

$$F1 = 2\frac{\mathrm{PRE} \cdot \mathrm{REC}}{\mathrm{PRE} + \mathrm{REC}} \tag{2.8}$$

, der

$$PRE = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.9}$$

og

$$REC = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}} \tag{2.10}$$

F1-score varierer mellom 0 og 1, hvor en verdi på 1 indikerer en perfekt modell og en verdi på 0 indikerer en svært dårlig modell (Raschka & Mirjalili, 2019). En høy F1-score indikerer at modellen har en høy presisjon og gjenkalling, noe som betyr at den er i stand til å identifisere positive tilfeller med høy nøyaktighet og lav feilrate. F1-scoren kan deles i to kategorier:

- **F1-positiv** er et mål på hvor godt systemet identifiserer positive objekter i en samling av objekter. For eksempel kan det være relevant i en situasjon der systemet må identifisere tilstedeværelsen av sykdom i en prøve. I dette tilfelle vil presisjonen måle andelen positive prøver som systemet har identifisert korrekt. Gjenkalling vil måle andelen av positive prøver som systemet har identifisert korrekt av alle positive prøver i samlingen (Raschka & Mirjalili, 2019).

  Denne metoden beregnes som en kombinasjon av presisjon og gjenkalling, og tar hensyn til både falske positive og falske negative resultater. Dette gjør at F1-positiv kan gi et mer balansert bilde av systemets evne til å identifisere positive objekter.

- **F1-negativ** er et mål på systemets evne til å identifisere negative objekter i en samling av objekter. Denne metoden beregnes på samme måte som F1-positiv, men fokuserer på negative objekter enn positive. F1-negativ regner altså nøyaktigheten på lik linje som F1-positivt, men med motsatt respons.

I denne studien benyttes *metrics.f1_score* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Matthews korrelasjonskoeffisient, MCC (*eng. Matthews Correlation Coefficent*)

Matthews korrelasjonskoeffisient (MCC) er en måte å evaluere kvaliteten på resultatene til en binær klassifiseringsmetrikk, og godt egnet måling for evaluering av modeller med ubalansert klassefordeling. MCC tar hensyn til alle fire verdiene i en 2x2 krysstabell som viser antall sann positive, sann negative, falsk positive og falsk negative resultater (Chicco & Jurman, 2020).

MCC kan da beregnes ved å bruke følgende formel:

$$MCC = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN})}} \tag{2.11}$$

MCC er en verdi mellom -1 og 1, der 1 indikerer en perfekt prediksjon, 0 indikerer tilfeldig prediksjon, og -1 indikerer en totalt motsatt prediksjon (Chicco & Jurman, 2020).

I denne studien benyttes *metrics.matthews_corrcoef* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011).

Receiver Operating Characteristic - Area Under the Curve, ROC-AUC

Receiver Operating Characteristic (ROC) kurve er vanligvis bygget ved å plotte andelen sann positive rate (TPR) mot andelen falsk positive rate (FPR) for ulike terskelverdier. En TPR og FPR er gitt ved:

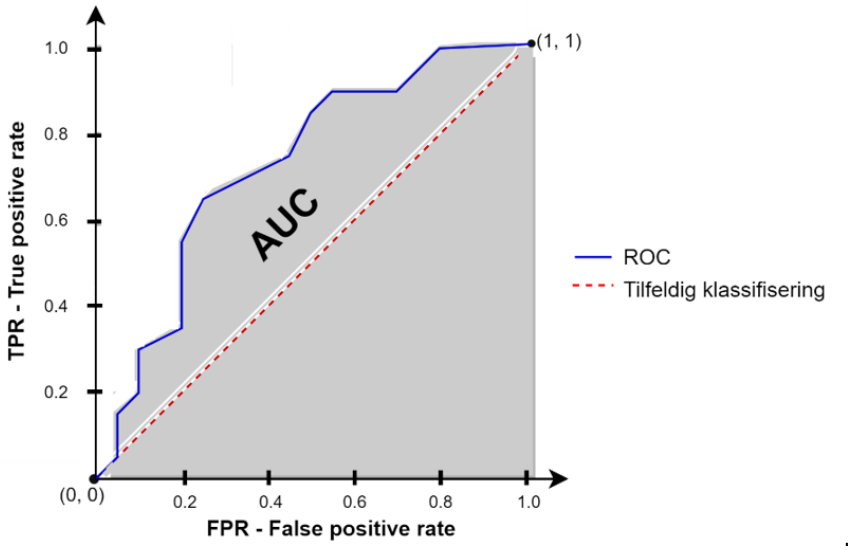
$$TPR = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}} \tag{2.12}$$

og

$$FPR = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2.13}$$

Sann positive rate (*eng. True Positive Rate,* TPR) og falsk positive rate (*eng. False Positive Rate,* FPR) er begge viktige mål for å evaluere ytelsen til en prediktiv modell eller klassifiseringsalgoritme (Raschka & Mirjalili, 2019).

For å forstå hvordan ROC-kurven fungerer, blir et eksempel på dette illustrert i figur 6 med FPR langs x-aksen og TPR langs y-aksen i et koordinatsystem. Som vist i figuren strekker ROC-kurven fra origo, altså punkt (0, 0) til punktet (1, 1). En rød stripet linje strekker seg gjennom de to punktene og refererer til modellytelsen som kan tilsvare tilfeldig gjetting, der TPR og FPR er like. Dette betyr at det vil være like mange korrekt klassifiserte prøver som har tilhørende klasse 1, som det er feilklassifiserte prøver som tilhører klasse 0. En kurv som ligger over den rette linjen, indikerer bedre modellytelse enn tilfeldig gjetting. Om kurven skulle havne under den rette linje, betyr det at modellen presterer dårligere enn tilfeldig gjetting.



Figur 6: En vilkårlig ROC-kurve er vist i blått, og arealet mellom kurven og x-aksen representerer ROC - AUC. En rød, stiplet linje viser ytelser til en modell som tilfeldig gjetter klasser. Inspirert av (Raschka & Mirjalili, 2019).

En perfekt klassifiseringsalgoritme vil ha en ROC-kurve som ligger helt i det øverste, venstre hjørnet av koordinatsystemet. Dette betyr at TPR vil være lik 1, som indikerer at alle faktisk positive tilfeller blir identifisert riktig. I kontrast til at FPR vil være lik 0, som betyr at ingen faktisk negative tilfeller blir feilaktig identifisert som positive. Når ROC-kurven er tegnet, kan man beregne *area under the curve* (AUC), som gir en numerisk verdi for ytelsen til klassifiseringsmodellen. AUC for ROC-kurven varierer mellom 0 og 1, hvor en verdi på 1 indikerer en perfekt modellytelse, og en verdi på 0 indikerer en modellytelse tilsvarende tilfeldig gjetting.

I denne studien benyttes pakken *metrics.roc_auc_score* fra maskinlæringsbiblioteket *scikit-learn* (Pedregosa et al., 2011)

2.3 Datavisualisering

Datavisualisering er prosessen med å presentere dataene grafisk for å gjøre det lettere for mennesker å forstå og analysere dem (Wilke, 2019). Det er et viktig verktøy for å gjøre informasjonen fra dataene mer tilgjengelig og forståelig for folk flest. Datavisualisering kan bidra til å identifisere mønstre, trender og relasjoner i dataene som kan være svært vanskelige å oppdage ved å analysere tallene alene. Det kan også hjelpe til å kommunisere dataene på en klar og engasjerende måte til et bredere publikum (Dougherty & Ilyankou, 2021).

Det finnes mange verktøy og teknologier som kan brukes for å lage og presentere datavisualiseringer, som for eksempel Python og R-studio (Wilke, 2019). Det er viktig å velge riktig verktøy og teknikk for å formidle dataene på en klar og effektiv måte. I denne avhandlingen benyttes teknikkene Principal Component Analysis (PCA), Violinplot og Clustermap i Python for å utføre en visuell analyse av datasettene.

2.3.1 Prinsipalkomponentanalyse (PCA)

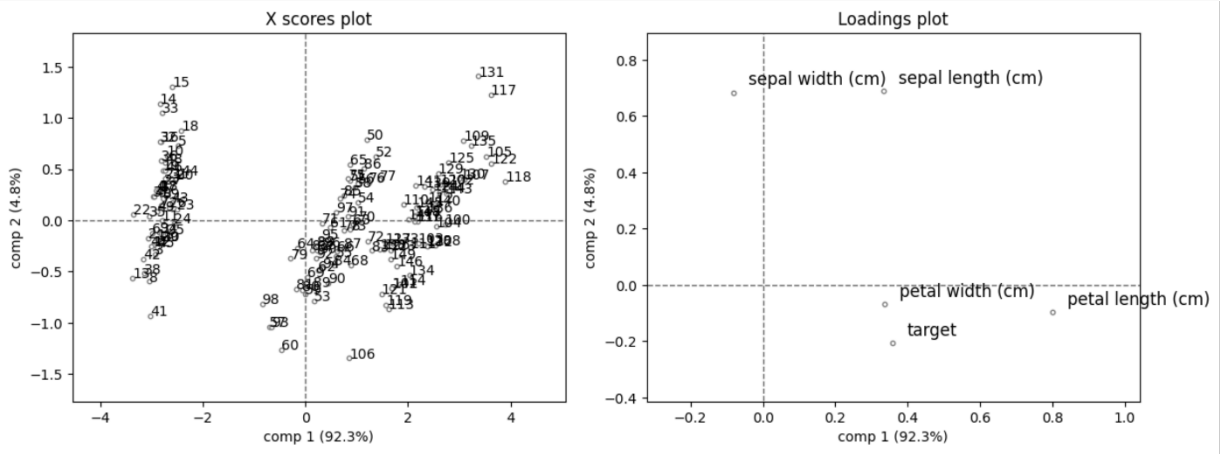
PCA står for «Principal Component Analysis» på engelsk, og er en statistisk metode som brukes for å redusere dimensjonen i et datasett (Abdi & Williams, 2010). Dette gjøres ved å finne de mest informative komponentene i datamaterialet, og deretter transformere dataene slik at disse komponentene blir de nye koordinatene. På norsk kan PCA også kalles for prinsipalkomponentanalyse eller hovedkomponentanalyse. Disse hovedkomponentene er lineære kombinasjoner av de opprinnelige variablene, og kan brukes til å representere dataene med færre dimensjoner (Abdi & Williams, 2010).

PCA er en vanlig metode, innen Python, for å analysere og visualisere høy dimensjonale data. Det er en datareduksjonsmetode som brukes for å identifisere de viktigste underliggende faktorene eller dimensjonene i datamateriale (Abdi & Williams, 2010). Dette gjøres ved å finne de lineære kombinasjonene av de opprinnelige variablene som gir maksimal varians. Disse lineære kombinasjonene kalles for hovedkomponentene, og kan visualiseres som en rotasjon av de opprinnelige koordinatene (Cohen, 2022). På denne måten kan PCA redusere dimensjonen i dataene mens man likevel beholder mest mulig av variansen.

PCA-plott kan være av ulike dimensjoner, som blant annet i 2D eller 3D. Den vanligste dimensjonen for en PCA-plott er 2D, som består av en x-akse og en y-akse. Den horisontale

aksen, også kjent som x-aksen, presenterer den første hovedkomponenten (PC-1) som inneholder mest variasjon i datasettet. Tilsvarende presenterer y-aksen, den andre hovedkomponenten (PC-2) som inneholder mest variasjon blant de gjenværende dataene etter at PC1 har blitt tatt i betraktning. PC1 og PC2 gir dermed en ide om de største variasjonene som finnes i datasettet. Dette kan være med på å skille mellom ulike grupper eller kategorier av data, eller til å identifisere mønstre i dataene som ikke er synlige i det opprinnelige datarommet (Abdi & Williams, 2010).

I denne oppgaven har det blitt benyttet en programpakke ved navn «Hoggorm» for å utføre PCA-analyse på datasettene og presentere disse som scoreplot og loadingplot (Tomic et al., 2019a). I et scoreplot er hver prøve presentert som et punkt i enten to-dimensjonal eller tre-dimensjonalt koordinatsystem som tar for seg prinsipalkomponentene. Dette plottet kan brukes til å identifisere verdier som skiller seg ut fra grupper av prøver som deler lignende egenskaper. En loadingplot, viser korrelasjon mellom variablene og komponentene i en modell. Figur 7 illustrerer et eksempel på et scoreplot og loadingplot av et velkjent datasett: Iris data (Fisher, 1936).



Figur 7: Presenter score- (V.S) og loadingplot (H.S) på Iris data. Det er tydelig at datasettet har tre ulike grupper som kan representere klassene Setosa, Versicolour, and Virginica. Komponent 1 forklarer 92.3% varians og komponent 2 forklarer 4.8% i varians. Figuren er generert med Python-script hvor datasettet er hentet fra scikit-learn (Pedregosa et al., 2011) og PCA-analysen uført på sentrert data med hoggorm biblioteket (Tomic et al., 2019a).

Metoden kan brukes i en rekke forskjellige områder, som for eksempel statistisk modellering og bildebehandling. Det kan også brukes for å oppdage skjulte mønstre eller relasjoner i dataene, som for eksempel å skille mellom forskjellige typer av en gitt datamengde. Videre kan bruken av PCA også være nyttig for datakomprimering, hvor man ønsker å redusere datamengden ved å kun beholde de viktigste komponentene (Cohen, 2022). Dette kan gjøre

det mulig å lagre eller overføre dataene mer effektivt, og dermed spare både tid og ressurser. Det er imidlertid viktig å merke seg at PCA er en lineær metode, og kan derfor ikke alltid passe for alle typer av data. I så fall finnes det andre metoder som kan være mer passende. Tabell 10 presenterer en oversikt over fordeler og ulemper med PCA og PCA-plott.

Tabell 10: Fordeler og ulemper med PCA og PCA-plott

Fordeler	Ulemper
• Kan brukes til å oppdage og fjerne ekstremverdier og unøyaktige data fra datasett • Kan brukes til å identifisere de mest informative variablene i datasettet	• Kan være utfordrende å tolke og forstå PCA-resultatene hvis det er mange variabler og hovedkomponenter • Kan føre til informasjonstap hvis en stor del av variasjonen blir redusert • Kan være ressurskrevende avhengig av størrelsen på datasettet og antall variabler

I denne studien benyttes pakken *hoggorm* (Tomic et al., 2019a) og *hoggorm-plot* (Tomic et al., 2019b).

2.3.2 Violinplot

Violinplot er en avansert visualiseringsmetode som viser typisk fordelingen og spredningen av dataene langs en vertikal akse, hvor bredden på hver del av plottet presenterer tettheten av dataene (Atewan, 2022). Dermed kan man enkelt identifisere om dataene er samlet i bestemte områder eller er spredt jevnt over hele området. Metoden kombinerer funksjoner fra både boxplot og kernel density estimate plot (KDE) (Atewan, 2022). Boxplot viser typisk de fem sentrale statistikkene for datamengden; minimum, maksimum, median og første og tredje kvartil. KDE gir en estimering av sannsynlighetsfordelingen til dataene, som normalt er en kurve som presenterer tettheten.

Denne visualiseringsmetoden er spesielt nyttig for å identifisere variabler som skiller seg ut fra normalfordelingen, og gir en indikasjon på mulige ekstremverdier i datasettet. Ved å

kombinere egenskapene til boxplot og KDE, gir violinplot en bedre forståelse av formen og spredningen av dataene i en enkel og oversiktlig format (Chen, 2022). Ved å bruke formen på violin, som er basert på kernel density estimatene, kan man identifisere om dataene er skjeve eller symmetriske, og om det er flere moduser eller toppunkt i fordelingen. Dette kan være spesielt nyttig når man ønsker å undersøke om det er noen avvik fra normalfordelingen i datasettet, eller om det er noen subgrupper som skiller seg ut.

En annen fordel med violinplot er at den tillater sammenlikning av flere grupper eller kategorier samtidig. Dette kan gjøres ved å plassere flere violinplot ved siden av hverandre og fargekode dem etter ulike kategorier eller grupper i datasettet (Atewan, 2022). Ved å gjøre dette kan man se hvordan fordelingen av dataene varierer mellom ulike grupper, og om det er noen signifikante forskjeller mellom dem.

Det er verdt å merke seg at selv om violinplot kan være nyttig visualiseringsmetode, kan den også ha noen begrensninger. For eksempel kan det være vanskelig å sammenligne størrelsen på dem direkte, spesielt hvis de er overlappende eller har ulik bredde. Det kan også være vanskelig å lese av eksakte tallverdier fra violinplot, og det kan derfor være nødvendig å supplere med andre typer visualiseringer eller statiske analyser. Tabell 11 viser en enkel oversikt over fordeler og ulemper med violinplot.

Tabell 11: Fordeler og ulemper med violinplot

Fordeler	Ulemper
• Kan tydeliggjøre eventuelle variasjoner i et datasett som ikke ville vært like synlige ved bruk av andre plott • Kan gi mer nøyaktig informasjon om fordelingene • Gjør det mulig å visualisere både median, kvartiler og tetthetsfordeling i ett plott	• Kan gi et misvisende plott hvis det er for lite antall datapunkter • Kan kreve større beregningskraft for større datasett • Kan være vanskelig å sammenligne med andre plott

I denne studien benyttes *violinplot* fra biblioteket *Seaborn* (Waskom et al., 2017).

2.3.4 Clustermap

Clustermap (også kjent som klyngekart eller hierarkisk klyngeanalyse på norsk) er en teknikk innen datavitenskap og statistikk som brukes til å utforske sammenhenger mellom variabler i et datasett (Raschka & Mirjalili, 2019). Det brukes vanligvis til å visualisere og gruppere data basert på likheter og forskjeller mellom dem. En clustermap viser hvordan forskjellige datapunkter er organisert og gruppert i klynger basert på hvor like de er. Dette gjøres ved å beregne avstanden mellom datapunktene og deretter gruppere dem basert på denne avstanden (Raschka & Mirjalili, 2019). Avstanden kan beregnes ved hjelp av forskjellige metoder, for eksempel Euclidean som er presentert i formel 2.4.

En vanlig metode for å presentere en clustermap er gjennom et varmekart, der klyngene er symbolisert ved hjelp av fargekodede firkanter (Belorkar et al., 2020). Klyngene kan være organisert hierarkisk, med flere nivåer av underklynger. Clustermap-teknikken er nyttig for å oppdage mønstre og strukturer i komplekse datasett, og kan gi innsikt i sammenhenger mellom variabler som ikke er åpenbare ved en enkel gjennomgang av datasettet. Tabell 12 viser fordeler og ulemper med clustermap.

Tabell 12: Fordeler og ulemper med clustermap

Fordeler	Ulemper
• Kan gi en visuell representasjon av gruppestruktur i dataene, noe som kan hjelpe med å identifisere mønstre og trender i dataene • Kan gi en indikasjon om det er noen duplikater av prøver i datasettet. • Kan identifisere korrelasjoner mellom variabler og dermed hjelpe med å identifisere viktige funksjoner	• Kan ikke brukes for å visualisere fordeling av dataene • Kan være vanskelig å tolke hvis det er store datasett med mange variabler

I denne studien benyttes *clustermap* fra biblioteket *Seaborn* (Waskom et al., 2017)

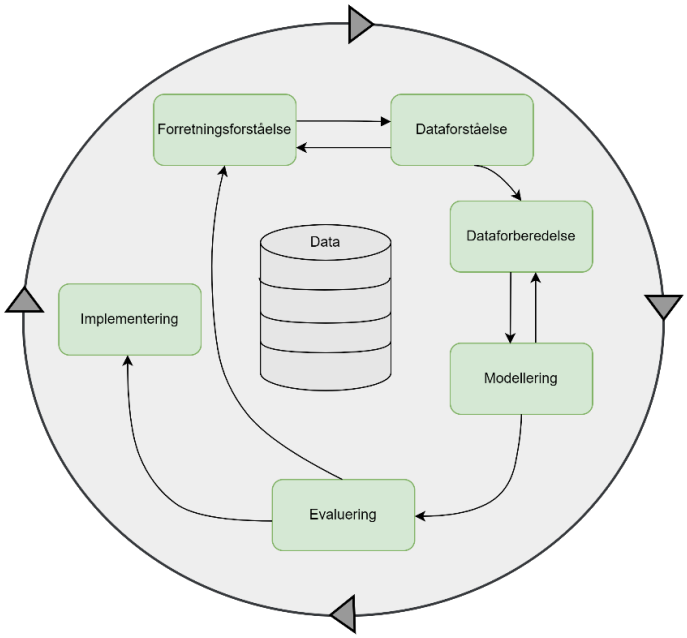
2.4 CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) er en prosessmetodologi som har blitt utviklet for å tilby et omfattende rammeverk for prosjektledelse og dataanalyse (Chapman et al., 2000). Formålet med CRISP-DM-metoden er å gi en standardisert og systematisk tilnærming til prosjekter, som hjelper organisasjoner med å forstå og analysere dataene sine på en effektiv og konsistens måte. Metodologien består av seks faser som hjelper til med å definere, planlegge, implementere og evaluere et prosjekt. Disse fasene er:

1. **Forretningsforståelse**: Dette trinnet omhandler å få en grundig forståelse av problemet som skal løses. Dette inkluderer å definere problemet og målsetningene klart og tydelig, og å identifisere hvilke data som trengs for å løse problemet. Det kan også innebære å identifisere eventuelle begrensninger eller utfordringer som kan påvirke prosjektet.
2. **Dataforståelse**: Dette trinnet involverer datautforskning, inkludert innsamling og undersøkelse av dataene for å forstå strukturen og identifisere eventuelle problemer eller mangler. Et eksempel på data som kan være relevant innen helsesektoren kan være pasientjournaler. Disse inneholder informasjon om pasientenes medisinske historie, diagnoser, behandlinger og medisiner som kan brukes til å utvikle prediktive modeller for sykdommer.
3. **Dataforberedelse**: Dette trinnet omhandler å forberede dataene man skal bruke til å løse problemet. Dette inkluderer å rense og formatere dataene, å integrere data fra forskjellige kilder hvis nødvendig, og å sørge for at dataene er i en form som kan brukes i analysen.
4. **Modellering**: Dette trinnet omhandler å utvikle en eller flere modell(er) som kan brukes til å løse problemet. Dette betyr å velge relevante dataanalyseverktøy og -teknikker, og å utvikle og teste modellene.
5. **Evaluering**: Dette trinnet omhandler å evaluere modellene man har utviklet, for å avgjøre hvor godt de løser problemet. Dette inkluderer å evaluere modellenes nøyaktighet, pålitelighet og effektivitet, og å justere modellene hvis nødvendig basert på evalueringen.
6. **Implementering**: Dette trinnet omhandler å implementere løsningen man har utviklet. Mer spesifikt, å integrere løsningen i virksomheten eller organisasjonen, å

sørge for at de nødvendige ressursene er på plass for å støtte løsningen, og å overvåke og vedlikeholde løsningen over tid for å sikre at den fortsetter å løse problemet.

Prosessmodellen, som illustrert i figur 8, viser til en oversikt over syklusen for et digitaliseringsprosjekt, som viser til de respektive fasene og relasjonen mellom dem. Da prosessen er en syklus, vil det være en prosess som er iterativ og det legges opp til å benytte seg av den tilegnede kunnskapen underveis til å forbedre resultat. Den sykliske metoden legger dermed opp til strukturering av problemet i utgangspunktet, i tillegg til repeterbar og konsistens fremgang mot det ønskede mål og resultat (Chapman et al., 2000).



Figur 8: Prosessmodellen for CRISP-DM-metoden. Inspirert av (Chapman et al., 2000)

CRISP-DM er en metodologi som blir mye brukt innen helsesektoren i dag for å analysere store datasett og hente ut verdifull innsikt fra dataene. En artikkel publisert i *National Library of Medicine* i 2020 med tittelen *Adaptations of data mining methodologies: a systematic literature review* undersøkte bruken av CRISP-DM-metoden i et helsedata-prosjekt. Studien diskuterte tilpasninger av data mining-prosjekter som kunne føre til bedre resultater og mer nøyaktige analyser, men også oppmerksomheten på potensielle begrensninger og utfordringer ved å tilpasse slike metoder (Plotnikova et al., 2020). Eksempelvis kan forskere tilpasse og justere CRISP-DM-prosessen for å imøtekomme spesifikke utfordringer og krav i forskjellige anvendelser. I denne oppgaven blir CRISP-DM-metoden brukt til å etablere en

standardisert arbeidsflyt som skal følges for hvert datasett. Dette kan gi kommende prosjektgrupper en enkel og overførbar metode som kan brukes som en mal for fremtidige prosjekter.

2.5 MCDA-Analyse

MCDA-analyse står for Multi-Criteria Decision Analysis på engelsk eller flermålsanalyse på norsk, og er en beslutningsstøttemetode som brukes når det er flere mål eller kriterier som skal vurderes opp mot hverandre (Rolstadås et al., 2020). Denne metoden bruker en strukturert tilnærming for å hjelpe beslutningstakere med å evaluere komplekse situasjoner. Det finnes to hovedmetoder i MCDA: kvantitative og kvalitative metode. Kvantitativ metode innebærer å trekke sammenhenger fra empiriske data og uttrykke disse i form av tallverdier (Blumberg et al., 2014). Kvalitativ metode bygger i stor grad på empiriske observasjoner og fanger opp det som ikke kan måles numerisk (Dalland, 2000). Den primære forskjellen mellom disse to metodene avhenger om informasjonen kan bli uttrykt i numerisk eller tekstlig form.

MCDA-analysen kan hjelpe beslutningstakere med å håndtere risiko og usikkerhet i beslutningsprosessen. Metoden kan også identifisere de mest relevante og realistiske alternativene for beslutningstakere basert på deres preferanser og vurderinger (Deshpande et al., 2020). Det finnes ulike fremgangsmåter som kan benyttes for å utføre en MCDA-analyse, et eksempel på en slik metode kan være følgende:

1. **Identifisere problemet og målene:** Det første trinnet i MCDA-analysen er å definere problemet som skal løses og identifisere de ulike målene som skal oppnås. Dette kan kreve en grundig analyse av situasjon og en innsikt i beslutningstakerens preferanser og krav.
2. **Valg av alternativer:** Neste trinn er å identifisere alle de mulige alternativene som skal løse problemet og oppnå målene. Det kan være fordelaktig å involvere eksperter og interessenter i denne prosessen for å sikre at alle relevante alternativer blir vurdert.
3. **Identifisere kriteriene:** Deretter identifisere de ulike kriteriene eller faktorene som skal brukes for å evaluere og sammenligne de ulike alternativene. Disse kriteriene kan ha ulike vekting, avhengig av deres relative betydning.

4. **Datainnsamling:** Innsamling av data som kan brukes til å vurdere hvert alternativ i forhold til hvert kriterium. Disse dataene kan være kvantitative eller kvalitative, avhengig av kriteriene og tilgjengelige datakilder.
5. **Analysere dataene:** Dataene analyseres for å evaluere hvert av alternativ i forhold til hver av kriteriene. Eksempelvis ved å bruke nyttekostnadsanalyse eller ulike algoritmer for å predikere et utfall.
6. **Presentere resultatene:** Resultatene fra analysen presenteres og sammenligner de ulike alternativene basert på deres totale score.
7. **Velg det beste alternativet:** Til slutt blir resultatene fra analysen benyttet til å velge det beste alternativet som oppfyller målene og kravene som er definert i trinn 1. Samtidig kan mulige konsekvenser av valget vurderes.

MCDA-analysen blir stadig mer brukt i helsesektoren for å ta beslutninger om prioritering av ressurser, behandlingsvalg og risikovurdering (Baltussen et al., 2010). En vitenskapelig artikkel publisert i tidsskriftet *BMC Health Services Research* undersøkte og diskuterte utfordringene knyttet til bruken av MCDA i helsesektor og dens teknologivurdering (Oliveira et al., 2019). Studien gjorde oppmerksom på tolv ulike utfordringer som kan oppstå i forbindelse med bruken av MCDA-metoden, og foreslo et rammeverk som kan bidra til å sikre en korrekt bruk av metode. Blant de utfordringer som identifiseres var blant annet problemet knyttet til modellering av data, inkludert usikkerhet og mangelfull datakvalitet.

Formålet med MCDA-analysen i denne oppgaven er å evaluere om de introduserte algoritmene kan bli brukt som et beslutningsstøtteverktøy for helsevesenet. Analysen skal undersøke om algoritmene kan brukes som en supplerende beslutningsstøtte, og ikke som en erstatning, i tråd med kliniske og etiske retningslinjer og med pasientenes interesser og behov i fokus.

Kapittel 3

Materiale og metode

Det følgende kapittel tar utgangspunktet i materialet og metodene som er benyttet for å besvare oppgavens problemsstilling og forskningsspørsmål best mulig. Avhandlingen baseres på to ulike datasett som representerer krefttypene kolorektal, og hode- og halskreft. Kapittelet er strukturert ved å opplyse hvilke maskinvarer og programvarer som har blitt brukt, samt hvor arbeidet er lagret. Deretter introduseres en detaljert arbeidsflyt som baseres på CRISP-DM-metodikken, samt presenteres hvilke andre metoder som ble brukt for oppnåelsen av resultatene.

3.1 Maskinvare og programvare

Passende maskinvare er essensielt for å kunne håndtere programvarene, grunnet de omfattende og ressurskrevende arbeidsoppgavene som skal utføres. For at utførelsen skal være effektivt og pålitelig, bør maskinvarene oppfylle de forskjellige kravene som en programvare stiller. Maskinkravene kan være eksempelvis prosessorkraft, minne, lagringsplass og grafikkort. I denne masteroppgaven blir følgende maskinvarer brukt:

- Huawei MateBook X Pro Signature Edition med en RAM på 16 GB hvor 15,8 GB kan brukes. Maskinen kjører på 1,80 GHz med Intel Core i7-8550 CPU. Operativt system: Windows 10, 64-bit.
- Apple MacBook Pro 2020 med M1-chip, 8-kjerners prosesser med 4 ytelseskjerner og 4 effektivitetskjerner. Maskinen har en RAM på 8 GB og lagringsplass på 512 GB. Operativt system: macOS Venture 13.3.1.

De overnevnte maskinvarene har sine primære oppgaver i å skrive og lese et programmeringsspråk i en rekke programmeringsverktøy. Python med versjon 3.7.9 (Van Rossum & Drake, 2009) er programmeringsspråket, som ble benyttet i programmerings-verktøyene Google Colab (Google Colaboratory, u.å) og Jupyter Notebook i regi Anaconda (Anaconda., 2020).

For å utføre databehandlingen i denne studien blir det benyttet Python-bibliotekene NumPy (Harris et al., 2020) og Pandas (McKinney, 2010). I tillegg ble de nødvendige

klassifiseringsalgoritmene for oppgaven hentet fra maskinlæringsbiblioteket scikit-learn (Plotnikova et al., 2020) og DESlib (Cruz et al., 2020). For å visualisere dataene ble bibliotekene Matplotlib (Hunter, 2007), Hoggorm (Tomic et al., 2019a), Hoggorm-plot (Tomic et al., 2019b) og Seaborn (Waskom et al., 2017) tatt i bruk. En oversikt over de forskjellige pakkene som ble brukt i studien er som følger:

NumPy (versjon 1.22.4)

NumPy (Numerical Python) er et Python-bibliotek som tilbyr funksjoner og metoder for å arbeide med store mengder numeriske data, spesielt matriser og arrays. Biblioteket gir et enkelt og effektivt grensesnitt for å utføre matematiske beregninger, statistisk analyser og manipulasjon av store datasett (Harris et al., 2020).

Pandas (versjon 1.5.3)

Pandas er et velkjent og populært bibliotek som gir en rekke nyttige verktøy for å håndtere store mengder av strukturerte data i Python (McKinney, 2010). Biblioteket gir funksjonaliteter for å importere data fra ulike datakilder, inkludert CSV, Excel, SQL og mer. Pandas er bygget på to viktige teknikker av hovedstrukturer, Series og DataFrame, som gir en fleksibel tilnærming til å manipulere og organisere data. En Series er en endimensjonal datastruktur som ligner på en liste og som kan inneholde variabler av forskjellige datatyper. En kolonne i en tabell. En DataFrame, derimot, er en todimensjonal struktur som består av rader og kolonner som tilsvarer de ulike datapunktene og variablene i datasettet.

scikit-learn (versjon 1.2.2)

scikit-learn er et åpent kildekodebibliotek for maskinlæring i Python (Pedregosa et al., 2011). Det gir en rekke verktøy for datavisualisering, dataforberedelse, modellering, evaluering og optimalisering av ulike maskinlæringsmodeller, som eksempelvis klassifisering, regresjon, klustering, og dimensjonsreduksjon gjennom et enkelt og konsistent brukergrensesnitt.

DESlib (versjon 0.4.dev)

DESlib er et bibliotek for ensemblelæring. Pakken har satt sitt fokus på implementering av dynamisk klassifisering og ensemblevalg (Cruz et al., 2020). Biblioteket er under utvikling ved oppgavens utforming. Det er nettopp denne pakken som har blitt utforsket og brukt for å hente Dynamic Ensemble Selection algoritmer.

PyOD (versjon 1.0.9)

Python Outlier Detection (PyOD) er en Python-pakke som brukes for å identifisere ekstremverdier i datasett (Zhao et al., 2019). Pakken støtter både overvåket og ikke-overvåket læringsteknikker, og gir mulighet til å kombinere flere algoritmer for å forbedre ytelsen. I kapittel 4 i boken *«Python Data Cleaning Cookbook»* diskuteres hvor godt PyOD fungerer for inspeksjon av ekstremverdier med eksempler til stede (Walker, 2020).

Optuna (versjon 3.1.1)

Optuna er en Python-basert kildekodepakke for hyperparameteroptimalisering i maskinlæring (Akiba et al., 2019). Pakken bruker en bayers tilnærming til optimalisering, for den kontinuerlig finner nye kombinasjoner av hyperparametere basert på tidligere funn. Dette gjør at Optuna kan effektivt navigere gjennom det store hyperparameterrommet og finne optimale hyperparametere på kortere tid enn tradisjonelle tilnærminger som rutenettsøk. I tillegg har pakken en rekke funksjoner for visualisering og logging av resultater, som gjør det enkelt å analysere resultatene og finne de beste hyperparameterne.

Seaborn (versjon 0.12.2)

Seaborn er et Python-bibliotek som gir et bredt spekt av plottefunksjoner som gjør det enkelt å visualisere statistiske data (Waskom et al., 2017). Biblioteket er bygget på toppen av Matplotlib, og tilbyr integrasjon med andre Python-baserte verktøy som NumPy og Pandas. Biblioteket tilbyr en rekke tema og fargepaletter som er spesielt designet for å gjøre det enkelt å visualisere informative plott.

Matplotlib (versjon 3.7.1)

Matplotlib er en Python-basert pakke for visualisering av data (Hunter, 2007). Den brukes ofte i maskinlæring for å visualisere resultater fra trening og evaluering av modeller, samt for å analysere dataene som brukes til modelltreningsprosessen. Matplotlib gir et bredt spekter av muligheter for visualisering, inkludert linjediagrammer, punktdiagrammer, søylediagrammer, varmediagrammer og mer avanserte figurer som tredimensjonale plott.

Hoggorm (versjon 0.13.3)

Hoggorm er et bibliotek som inneholder metoder som blant annet PCA (*Principal Component Analysis)* og PCR (*Principal Component Regression)*. I kontrast til scikit-learn, har hoggorm som mål om å forstå og tolke variasjon i dataene ved hjelp av de nevnte metodene (Tomic et al., 2019a).

Hoggorm-plot (versjon 0.13.2)

Hoggorm-plot er et bibliotek utviklet for å visualisere resultater som er analyser fra hoggorm-pakken (Tomic et al., 2019b).

LazyPredict (versjon 0.2.12)

LazyPredict er en programvarepakke i Python som gir en tilnærming for å utføre maskinlæringseksperimenter (Pandala, 2022). Ved å bruke LazyPredict kan brukere enkelt trene flere modeller og evaluere deres ytelse på en gitt oppgave. Dette kan være særlig nyttig når det er begrenset tid eller ressurser til rådighet. LazyPredict har innebygde modeller som kan brukes til både klassifiserings- og regresjonsoppgaver. Disse inkluderer flere populære algoritmer som random forest, logistisk regression, GaussianNB og mange flere. Videre kan LazyPredict også gi en rapport om modellens ytelse og sammenligne resultatene fra forskjellige modeller for å hjelpe brukeren med å velge den som gir best resultat.

Det er viktig å legge merke til at LazyPredict forslår algoritmer ved å lage modeller med normale *(eng. default)* hyperparameter og presenterer modellen som kommer best ut. Dette kan bety at modellen den foreslår kommer best ut når modellene kun tar i bruk de normale hyperparameterne, men at andre modeller har potensial for å prestere bedre når det utføres hyperparameteroptimalisering. En detaljert forklaring om hyperparameteroptimalisering blir gitt i kapittel 3.4.3

3.2 Versjonskontroll

All kode som blir benyttet i dette prosjektet, er tilgjengelig på GitLab: https://gitlab.com/majorann_saranjan/master-v23-io. Tabell 13 viser siste versjonsnummer for ipynb-filene (Jupyter Notebook filene) som er tilgjengelige på GitLab på det tidspunktet denne oppgaven ble skrevet.

Tabell 13: Oversikt over GitLab repository`s ipynb-filer med siste oppdaterte versjon

Kolorektal kreft		**Hode- og halskreft**	
Fil	**Git hash**	**Fil**	**Git hash**
Data prosessering	981a62ed	Data prosessering	1f1d082d
Visualisering	67c26157	Visualisering	3e0b109e
Ekstremverdi_inspeksjon	58d337c0	Ekstremverdi_inspeksjon	ca52bccf
Dropper_ekstremverdier	d599379c	Dropper_ekstremverdier	be0dd011
Parameteroptimalisering_OS	cf063f79	Parameteroptimalisering_OS	808d327e
Klassiske_modeller_OS	b974fe3c	Klassiske_modeller_OS	afb4832e
DES_modeller_OS	5ba69c5c	DES_modeller_OS	a7f622a1
Vanskelige_pasienter_OS	e4c9f04b	Vanskelige_pasienter_OS	0a346f8f
Parameteroptimalisering_PFS	7ab052f7	Parameteroptimalisering_DFS	46dcfd98
Klassiske_modeller_PFS	50975bcc	Klassiske_modeller_DFS	ed8ebe85
DES_modeller_PFS	84b1a57f	DES_modeller_DFS	ed72c988
Vanskelige_pasienter_PFS	409b7120	Vanskelige_pasienter_DFS	d10941a9

3.3 Datasett

Denne oppgaven har tatt for seg to eksisterende datasett: kolorektal kreft og hode- og halskreft. Datasettene ble samlet av Oslo universitetssykehus i 2007 og 2013, og har tidligere blitt brukt i en rekke studier og undersøkelser. Noen av forskningene ble gjort av uteksaminerte masterstudenter på Norges Miljø- og Biovitenskapelige Universitet ved Ås.

Masterstudentene Lars J.S Engeseth og Alise D. Midfjord var to av flere studenter som har brukt de samme datasettene i deres arbeid. Engeseth anvendte kolorektal datasettet for å undersøke hvordan RENT (Repeated Elastic Net Technique) kan implementeres for variabel seleksjon og hvordan dette påvirker prediksjonene (Engesæth, 2022). Midfjord brukte hode- og halskreft datasettet for prediksjon av behandlingsutfall ved hjelp av radiomics fra PET- og CT-bilder (Midtfjord, 2018). Videre har Jon M. Moan skrevet en studie basert på datasettet

for hode- og halskreft (Moan et al., 2019), og datasettet for kolorektal stammer fra en egen studie med navnet OxyTarget (Røe, 2018). Tabell 14 gir en oversikt over datasettene og deres generelle innhold.

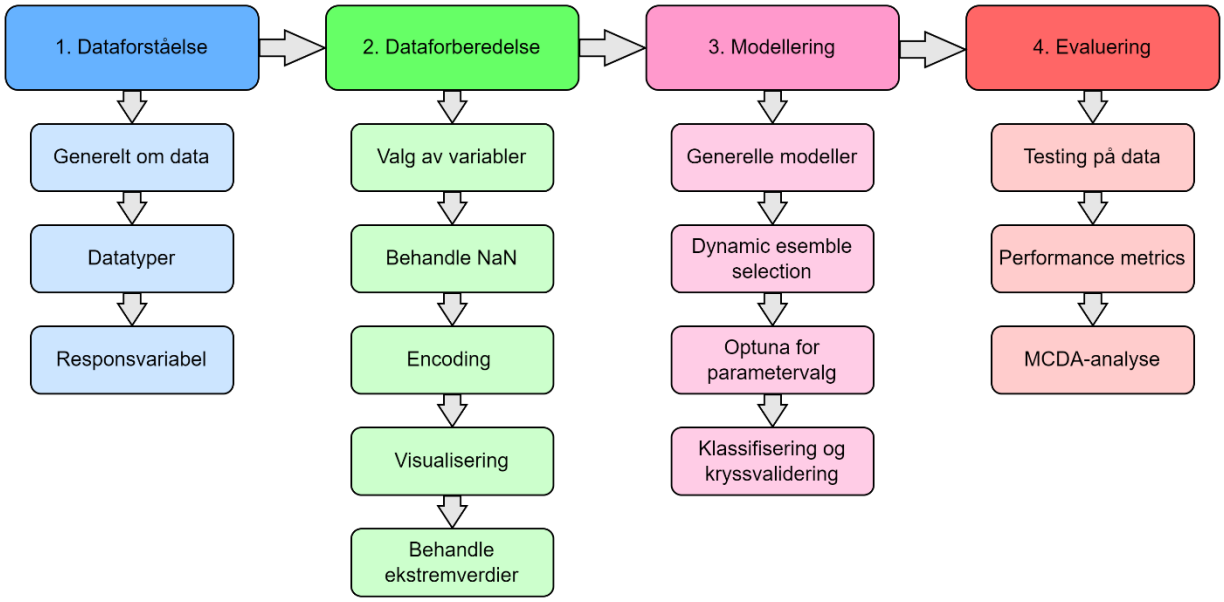
Tabell 14: En overordnet oversikt over innholdet i datasettene

	Kolorektal kreft	**Hode- og halskreft**
Kilde (hvor datasettet stammer fra)	Oslo universitetssykehus (OUS)	Oslo universitetssykehus (OUS)
Periode	2013 – 2017	2007 – 2013
Antall pasienter	192	197
Antall manglende verdier (NaN-values)	6334	104
Responsvariabler	**OS** (*eng. Overall Survival*) – generell overlevelse **PFS** (*eng. Progression-Free Survival*) - Progresjonsfri overlevelse	**OS** (*eng. Overall Survival*) – generell overlevelse **DFS** (*eng. Desease-Free Survival)* – sykdomsfri overlevelse **LRC** (*eng. Local-Regional Control)* – Lokalt eller regionalt tilbakefall

Mer detaljert og grundigere forklaring av datasettene, og behandling av disse blir presentert i kapitlene 3.5 kolorektal kreft og 3.6 hode- og halskreft.

3.4 Arbeidsflyt

For å etablere en standardisert metodisk struktur for hvert av datasettene, ble en spesifikk arbeidsflyt fulgt. Formålet med arbeidsflyten er å strukturere arbeidet med datasettene, modellere maskinlæringsmodellene og evaluere disse, samt overføre den samme databehandlingen fra ett datasett til et annet. Dette kan bidra til å effektivisere arbeidet og gjøre arbeidsprosessen smidigere. Arbeidsflyten er illustrert i figur 9.



Figur 9: Arbeidsflyten som visualiserer prosessen som følges for å bygge modeller og evaluere disse.

I planleggingen, organiseringen og implementeringen av prosjekter som involverer datanalyse, er det avgjørende å ha en effektiv arbeidsflyt som kan sikre nøyaktigheten og relevansen til resultatene. Den presenterte arbeidsflyten er inspirert av CRISP-DM (Cross-Industry Standard Process for Data Mining), en prosessmetodologi bestående av seks faser: *forretningsforståelse, dataforståelse, dataforberedelse, modellering, evaluering og implementering*.

I denne avhandlingen ble fire av de seks fasene brukt: *dataforståelse, dataforberedelse, modellering og evaluering*. Dette skyldes behovet for økt fokus på relevans og kvalitet, med tanke på omfanget og kompleksiteten i problemstillingen som ble undersøkt. Ved å følge CRISP-DM-metoden kan kommende forskere anvende en systematisk tilnærming av dataanalysen, som informerer at de viktige aspektene ved analysen ble grundig adressert.

Det er viktig å merke seg at prosedyren rundt dataforståelse og dataforberedelse varierer fra datasett til datasett. Imidlertid er fasene modellering og evaluering felles for alle datasettene. Siden denne oppgaven tar for seg to ulike datasett, blir det også presentert to separate kapitler, 3.5 og 3.6, som forklarer dataforståelse og dataforberedelse spesifikt for datasettene kolorektal og hode- og halskreft. På bakgrunn av at modelleringsfasen og evalueringsfasen er til felles for begge datasettene, presenteres metodikken for de samlet i kapitelene 3.4.3 og 3.4.4. Nedenfor følger underseksjoner som tar for seg de fire hovedfasene fra arbeidsflyten med detaljerte beskrivelser av stegene i hver fase.

3.4.1 Dataforståelse

Dataforståelse referer til innsamling av relevante data for et prosjekt, og å oppnå en forståelse av den innsamlede data (Chapman et al., 2000). Dette steget omfatter å identifisere avvik og utfordringer i datasettet, og nødvendige forberedelsene som kreves for å preprosessere dataene. Denne informasjon kan gi verdifull innsikt i behovet for forbedring av datainnsamlingsmetoden, samt kvaliteten på de innsamlede dataene. Datakvaliteten i en oppgave referer til graden av relevans, kompleksitet, og konsistens av informasjonen i datamengden som blir benyttet i oppgaven.

En fellesnevner for de to datasettene som ble analysert og undersøkt i denne studien, er at de inneholder helsedata. Dette inkluderer informasjon om pasientenes medisinske tilstand og annen relevant helseinformasjon som er samlet fra forskjellige kilder som bildedata, medisinsk behandling og histologi. Det er verdt å merke seg at helsedata er spesielt sensitive og konfidensielle, og krever spesielle hensyn og etiske retningslinjer ved behandling og deling. For å sikre en grundig forståelse av de anvendte datasettene, ble det fulgt fire trinn som er illustrert i figur 9. Resultatene av disse er presentert i kapittel 3.5 og 3.6.

3.4.2 Dataforberedelse

Preprosessering av datasettene er en avgjørende del av den initiale dataanalyseprosessen og spiller en betydelig rolle med tanke på å organisere og forbedre kvaliteten på dataene (Chapman et al., 2000). Denne fasen sikrer at modeller kan trenes på dataene og dermed bidra til å øke prediksjonsytelsen. I denne avhandlingen har det vært en prioritering å utføre grundig preprosessering av datasettene ved å følge arbeidsflytens steg for steg for å klargjøre dataene.

Arbeidsflyten for denne fasen består av flere steg, som illustrert i figur 9. Disse stegene inkluderer valg av variabler, behandling av manglende verdier, behandling av kategoriske verdier, visualisering av data og håndtering av ekstremverdier. En grundig forklaring av stegene er nærmere forklart senere i denne seksjonen.

Det bør påpekes at visse datasett kan kreve fullføring av alle stegene i dataforberedelsesfasen, og andre kan kun kreve gjennomføring av utvalgte steg. Beslutningene om hvilke datasett som hadde behov for å gjennomgå disse trinnene, ble

basert på strukturen og informasjonen som var tilgjengelig i de ulike datasettene for denne oppgaven. De spesifikke trinnene og metodene som var nødvendige for hvert av datasettene i denne oppgaven, er dekket i kapittel 3.5 og 3.6.

STEG 1 - Valg av variabler

Det første steget i dataforberedelsen er å undersøke variablene for å avdekke eventuelle faktorer som kan påvirke datakvaliteten. Dette steget involverer eliminering av mindre viktige variabler og observasjoner som mangler verdier, også kjent som «missing values». Fjerning av disse kan forbedre presisjon og hastighet på modellutvikling, men kan også hindre visse modeller i å fungere når de manglende verdiene er til stede (Jafari, 2022). Videre kan denne prosessen redusere muligheten for overtilpasning, som kan påvirke generaliserbarheten av modellene.

Valg av hvilke teknikker som skal brukes for filtrering av irrelevante variabler og observasjoner, avhenger av egenskapene til det gitte datasettet. Det er avgjørende å finne en passende balanse mellom beholdte og fjernede variabler, ettersom et stort antall variabler kan føre til overtilpasning av en modell. For få variabler kan føre til at en modell ikke klarer å fange opp viktige sammenhenger i dataene. En optimal teknikk er dermed essensielt for valg av variabler og observasjoner basert på datasettets struktur og egenskap. En detaljert beskrivelse av de ulike teknikkene som er benyttet for valg av variabler og observasjoner i de ulike datasettene er dekket i kapitlene 3.5 og 3.6.

STEG 2 - Behandle manglende verdier (NaN-values)

Manglende verdier, også kjent som «NaN-values», representerer manglende eller udefinerte data. Manglende verdier i et datasett kan skyldes av ulike årsaker, deriblant mangel på innsamling av data, innsamlingsfeil eller utilgjengelighet av verdier på grunn av spesifikke begrensinger (Emmanuel et al., 2021).

Datasett med manglende verdier kan ha negativ effekt på analyser og modellering, da mange modeller ikke vil være stand til å fungere som ønsket (Jafari, 2022). Dette kan føre til at analyseresultatene blir misvisende eller ha en form for usikkerhet. For å unngå disse negative konsekvensene, kan det være nødvendig å fjerne noen observasjoner fra datasettet eller bruke andre metoder for å håndtere dem. Mange metoder finnes for å håndtere manglende

verdier, men det er viktig å velge riktig metode for å håndtere dem på en hensiktsmessig måte slik at kvaliteten på dataen bevares (Khan & Hoque, 2020).

Måten disse manglende verdiene (NaN-values) blir behandlet på, er avhengig av hva som er formålstjenlig for den spesifikke analysen eller modellen. Noen av de vanligste teknikkene for å behandle disse verdiene på, kan være:

1. **Fjerning av rader eller kolonner**: Hvis det er relativt få rader eller kolonner som inneholder mange NaN-verdier, kan man velge å fjerne dem fra datasettet. Dette kan gjøres med funksjoner som «dropna()» i pandas biblioteket (McKinney, 2010).
2. **Fylling av verdier:** Hvis det er viktig å beholde mye av informasjon, kan man velge å fylle inn manglende verdi med andre verdier, for eksempel gjennomsnittsverdien av variabelen, medianverdien på variablene, eller en annen passende verdi. Dette kan gjøres med funksjoner som «fillna()» i pandas biblioteket (McKinney, 2010).
3. **Maskering:** Man kan velge å beholde NaN-verdiene som de er, men maskere dem slik at de ikke påvirker resultatene av beregningene eller modellen. Dette kan gjøres med funksjoner som «numpy.isnan()» i NumPy biblioteket (Harris et al., 2020).

STEG 3 - Encoding av kategoriske verdier

Encoding av kategoriske verdier handler om å konvertere kategoriske variabler til numeriske verdier (Dahouda & Joe, 2021). Dette er en viktig prosesseringsdel som gjør det mulig å bruke disse variablene til statistiske modeller som har krav om numeriske verdier som input. I maskinlæringsverden er det behov for at hele datasettet er numerisk for å kunne bruke det sammen med algoritmer (Yang, 2018). Derfor er det veldig viktig å encode alle kategoriske verdier til numeriske, slik at datasettet er klart til å bli kjørt sammen med maskinlæringsalgoritmer.

Det finnes mange forskjellige metoder og teknikker for encoding av kategoriske verdier. One-Hot Encoding, LabelEncoding og Binary Encoding er noen av disse teknikkene og i denne oppgaven brukes LabelEncoder fra biblioteket til scikit-learn (Pedregosa et al., 2011). LabelEncoder er en metode som gir hvert element i en variabel, en unik numerisk verdi (Yang, 2018). Et eksempel på encoding kan være en variabel som inneholder elementene «Banan», «Eple» og «Appelsin». Ved å anvende LabelEncoder på variabelen, vil den tilordne

tallverdiene 0 til Banan, 1 til Eple og 2 til Appelsin for hver av kategoriene. Dermed har variabelen endret seg fra å være kategoriske til numeriske verdier.

Behovet for encoding av kategoriske variabler har ikke vært like avgjørende og fundamental for begge datasettene. Det var kun datasettet med kolorektal kreft som hadde behov for behandling av kategoriske variabler og grunnen til dette blir forklart i detaljer i kapittel 3.5.2.

STEG 4 - Visualisering av datasettene

Etter å ha behandlet datasettene for manglende og kategoriske verdier, ble disse visualisert for å gjøre dataene forståelig. Med visualisering er det blant annet enklere å forstå komplekse data og informasjon. Dette hjelper analytikere til å se sammenhenger som kan være utfordrende å avdekke ved kun å betrakte numeriske verdier (Géron, 2022). Visualisering kan også avdekke avvik eller ekstremverdier i datasettet. Dette hjelper igjen med å identifisere unyttige observasjoner eller variabler som må elimineres fra datasettet for å øke verdien av datasettet ved modellering.

I denne avhandlingen ble datasettene visualisert ved hjelp av clustermap, PCA-plott og violinplot. Hvilke nytte disse visualiseringstypene gir og hvordan disse praktiseres, er nærmere forklart i det teoretiske rammeverk under kapittel 2.3. Resultatene fra plottene er presentert i kapitelene 4.1.1 og 4.4.1.

STEG 5 - Inspeksjon og behandling av ekstremverdier *(eng. outliers)*

Den siste delen av prosesseringen er å undersøke om ekstremverdier er til stede i datasettene. Selv om en har mulighetene til å finne potensielle ekstremverdier med visualiseringene, vil det gi mer sikkerhet knyttet til analysen ved å ta i bruk en test som tar for seg identifisering av ekstremverdier (*eng. outlier detection*). Ekstremverdier er verdier som signifikant avviker fra de øvrige verdiene i et datasett (Agrawal, 2021). Avvik vil være i form av at det er svært høye eller lave verdier i forhold til de øvrige verdiene. Dette kan være verdier som skyldes av menneskelige faktorer, som målefeil eller skrivefeil (Foxwell, 2020).

Ekstremverdier kan ha en negativ innvirkning på resultatene av modellene som er trent, og kan føre til økt feil i prediksjon eller klassifisering (Jafari, 2022). Videre kan de forstyrre modellens forståelse av dataen, og føre til at modellen tilpasser seg ekstremverdiene i stedet for å være av de vanlige observasjonene. Dette kan øke sannsynligheten for overtilpasning,

noe som igjen kan påvirke nytten av modellen negativt. Derfor er det av stor betydning å oppdage og håndtere ekstremverdier for å unngå slike negative konsekvenser.

Det finnes flere metoder for å detektere ekstremverdier i datasett, og som nevnt i teoretisk rammeverk benytter oppgaven teknikker fra biblioteket PyOD for å analysere mulige ekstremverdier. To algoritmer fra PyOD, KNN (K-Nearest Neighbors) og ECOD (Outlier Detection Using Empirical Cumulative Distribution Functions), ble benyttet for å inspisere ekstremverdier i de to datasettene som ble undersøkt i oppgaven.

Det er verdt å merke seg at KNN er en overvåket metode, i kontrast til at ECOD er en ikke-overvåket metode. I denne oppgaven ble begge disse metodene valgt bevisst for å sammenligne deres ytelse og for å evaluere om en overvåket metode vil identifisere de samme ekstremverdiene som en ikke-overvåket metode. Observasjoner som blir identifisert som ekstremverdier, kan gi verdifull informasjon om hvordan disse verdiene kan påvirke ytelsen til modellen og dataanalysen. Det kan dermed være nødvendig å eliminere slike observasjoner fra datasettene for å unngå støy eller bias i resultatene. Dette vil gi et solid grunnlag for videre vurdering og forbedring av modellene og dataanalysen.

Resultatene fra analysen om deteksjon av ekstremverdier, er detaljert beskrevet i henholdsvis i kapittel 4.1.2 og 4.4.2 i oppgaven. Disse kapitlene gir en grundig analyse av hvilke observasjoner som har blitt identifisert som ekstremverdier basert på KNN og ECOD, samt en gjennomgang av hvordan disse ekstremverdiene er behandlet.

3.4.3 Modellering

Modellering er en viktig del av maskinlæringsprosessen og utgjør fase tre i arbeidsflyten, som vist i figur 9. I denne fasen er fokuset på å utvikle et rammeverk og trene maskinlæringsalgoritmene for å oppnå de ønskede resultatene. Denne seksjonen om modellering vil derfor beskrive i detalj de ulike metodene og teknikkene som er brukt for å bygge modellene i oppgaven, samt valg av algoritmer, parameterinnstillinger, modellvalidering og metoder for å forbedre modellytelsen. Det er viktig å understreke betydningen av modellering, ettersom det er gjennom denne prosessen at man kan besvare forskningsspørsmålene og konkludere med problemstillingen i oppgaven.

Valg av algoritmer

For å få gode resultater, er det viktig å ha tilstrekkelig med modeller som setter grunnlag for sammenligning av modeller med hverandre. For å sikre korrekt sammenligningsgrunnlag, ble de samme algoritmene benyttet på både kolorektal samt hode- og halsdatasettet.

Som nevnt innledningsvis, benytter DES-algoritmer ensemble læring. I sammenligning med andre velkjente ensemble-algoritmer, tar DES i bruk en dynamisk tilnærming for å finne individuelle modeller som inkluderes i ensemblet for hver prediksjon. De individuelle modellene blir valgt basert på egenskapene til *hver prøve* som skal klassifiseres. DES har altså potensialet til å tilpasse og endre seg i forhold til dataen, og derved muliggjøre forbedret prediksjonsytelse. En annen styrke med DES, er dens evne til å inkludere og kombinere flere læringsalgoritmer i ensemblet, som gjør det mulig å integrere ulike typer informasjon i modellen og dermed forbedre dens generaliseringsytelse.

For å etablere et solid sammenligningsgrunnlag ble det utviklet tre modeller med forskjellige algoritmer fra DESlib-pakken. Videre ble det trent tre andre modeller med klassiske ML-algoritmer som er kjent for å gi gode resultater i klassifiseringsoppgaver. For å utforske ytterligere alternativer, ble LazyPredict-pakken brukt for å bestemme to ytterligere klassifiseringsalgoritmer som var best egnet for hvert av datasettene i denne oppgaven. For datasettet om kolorektal kreft ble GaussianNB utmerket som den beste, og nearest centroid ble utpekt som den beste klassifiseringsalgoritmen for datasettet om hode- og halskreft. Tabell 15 gir en oversikt over de algoritmene som har blitt brukt i denne oppgaven.

Tabell 15: Oversikt over klassifiseringsalgoritmer benyttet i oppgaven

Dynamic Ensemble Selection	**Klassiske ML-algoritmer**
K-Nearest Oracle-Eliminate (KNORA-E)	Random Forest
K-Nearest Oracle Union (KNORA-U)	Logistisk Regresjon
Dynamic Ensemble Selection Performance (DES-P)	Quadratic Discriminant Analysis (QDA)
-	GaussianNB (LazyPredict)
-	Nearest Centroid (LazyPredict)

Hovedfokuset for oppgaven er rettet mot de åtte algoritmene som er presentert ovenfor, men i modelleringsfasen ble også de algoritmene som er oppført i tabell 16 anvendt for å lage

noen modeller. Resultatene for disse klassifiseringsalgoritmene har ikke blitt gitt samme grad av oppmerksomhet og vekt som de åtte presenterte algoritmene. For interesserte lesere er disse resultatene vedlagt i oppgaven, som gir en ekstra innsikt i hvordan disse algoritmene har prestert. Det er viktig å legge merke til at MCB og OLA ligger i kategorien DCS (Dynamic Classifier Selection) og ikke DES (Dynamic Ensemble Selection) i DESlib-pakken. Mer om dette er beskrevet i vedlegg A.5.

Tabell 16: Oversikt over klassifiseringsalgoritmer som ikke er inkludert i oppgaven, men resultatene fra disse er presentert i vedleggene som følger med i oppgaven.

Andre algoritmer fra DESlib	**Klassiske ML-algoritmer**
Meta Learning for Dynamic Ensemble Selection (META-DES)	Support Vector Classifier (SVC)
Multiple Classifier Behaviour (MCB)	K-Nearest Neighbors Classifier (KNN)
Overall Local Accuracy (OLA)	-

Standardisering

En viktig preprosesseringsteknikk for å sikre effektiv læring av variablene under modelleringen er standardisering (Géron, 2022). Standardisering av datasett handler om å transformere skalaen til variablene, slik at disse har likt gjennomsnitt på 0 og standardavvik på 1. Det er vanlig at datasettet inneholder variabler med forskjellig skala. Eksempelvis kan noen variabler ha verdier i tusener og andre i millioner, altså rekkevidden (*eng. range*) er forskjellig mellom variablene.

For å standardisere datasettene i denne oppgaven har StandardScaler fra scikit-learn blitt benyttet (Pedregosa et al., 2011). Formel 3.1 viser likningen for standardisering, hvor **z** er den standardiserte/transformerte verdien, **x** er verdien i datasettet, $\boldsymbol{\mu}$ er gjennomsnittet av **x**, og $\boldsymbol{\sigma}$ er standardavviket av **x** (Raschka & Mirjalili, 2019).

$$\mathcal{Z} = \frac{x - \mu}{\sigma} \tag{3.1}$$

Optimalisering av parametere

I metodefasen har det vært fokus på å justere modellparametere (hyperparametertuning) for å forbedre og øke prediksjonsytelsen. For å finne de optimale parameterne til en algoritme, kan forskjellige teknikker benyttes. Noen av de mest kjente teknikkene er grid search og random search, som er tilgjengelig i scikit-learn biblioteket (Pedregosa et al., 2011). I denne oppgaven blir en pakke som heter Optuna brukt for å finne de beste kombinasjonene med modellparametere (Akiba et al., 2019).

Optuna er et åpent kodebibliotek som ble lansert i 2019, og det brukes til automatisert hyperparameteroptimalisering (Akiba et al., 2019). Optuna anvender ulike teknikker som tilfeldig søk (random search), rutenett-søk (grid search) og bayesiansk optimalisering for å finne de optimale hyperparameterne for en modell. Biblioteket er også diskutert i boken skrevet av Agrawal. T med Optuna som en metode for hyperparameteroptimalisering (Agrawal, 2021).

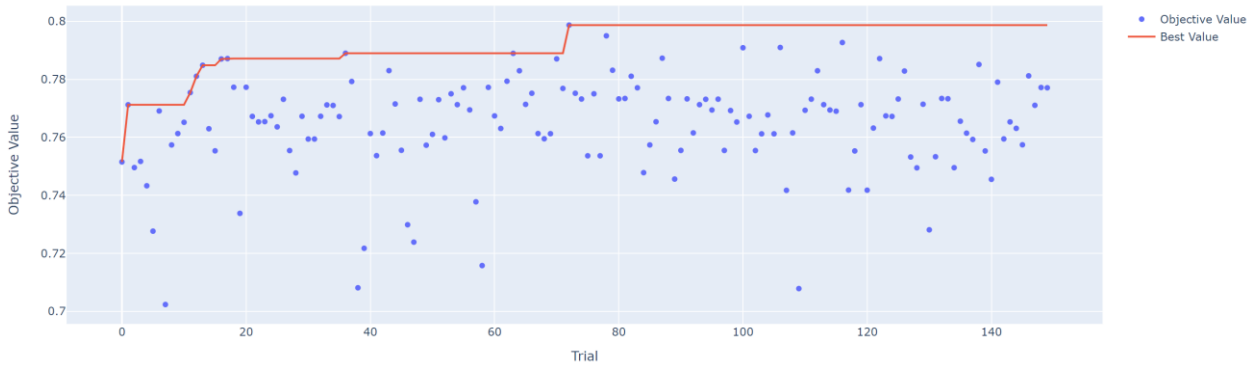
Funksjonaliteten til Optuna er å utføre en systematisk prøv-og-feil-metode for å finne de optimale parameterne. For å identifisere optimale parameterverdier for en gitt modell, er det nødvendig å definere hvilke parametere som skal optimaliseres på forhånd. Deretter må man bestemme hvilke områder av det såkalte «hyperparameterrommet» som skal søkes. Dette rommet består av en samling av mulige parameterkombinasjoner som Optuna vil bruke til å trene og evaluere modellen. Optuna bruker resultatene fra evalueringen til å justere hyperparameterområdet og fokusere søket på områder med tidligere suksess, og dermed utelukke områder med mislykkede resultater (Akiba et al., 2019). Dette øker effektiviteten ved å fokusere på de mest lovende områdene for å finne de beste parameterkombinasjonene. Til slutt vil de beste kombinasjonene av hyperparameterne bli returnert. Antall evalueringsforsøk som skal utføres, bestemmes på forhånd ved å definere parameteren «trials» i Optuna. I denne oppgaven er «trials» satt til 200, og dermed bruker Optuna 200 forskjellige kombinasjoner for å finne de beste hyperparameterkombinasjonene for hver algoritme.

I vedlegg A presenteres en oversikt over de parameterne som har blitt optimalisert, samt resultatene av disse optimaliseringene. Tabell 17 er en del av dette vedlegget, og gir en detaljert framstilling av hyperparameterresultatene for KNORA-E.

Tabell 17: Optuna's forslag til hyperparameterkombinasjoner for modellene med KNORA-E etter 200-trials

Datasett	n_estimators	Pool classifiers	K	Voting	knn_metric
Kolorektal (OS)	319	Random forest	2	'hard'	mahalanobis
Kolorektal (PFS)	361	[Random forest, Bagging Classifier]	5	'soft'	minkowski
Hode- og hals (OS)	76	[Random forest, Bagging Classifier]	7	'hard'	minkowski
Hode- og hals (DFS)	164	[Random forest, Bagging Classifier]	2	'hard'	minkowski

Beskrivelse av parameterne er nærmere forklart i vedlegg A. Hyperparameteroptimalisering ble kjørt med 200 trials med Optuna. Nedstående figur 10 illustrerer et eksempel på optimering av KNORA-E for kolorektal kreft med generell overlevelse (OS) som responsvariabel. Figuren viser tydelig at trial 72 gir den mest optimale hyperparameterkombinasjonen. Denne kombinasjonen er presentert i tabell 17.



Figur 10: Illustrerer resultatet over hyperparameteroptimalisering med Optuna for modellen med KNORA-E for kolorektal datasettet.

Kryssvalidering

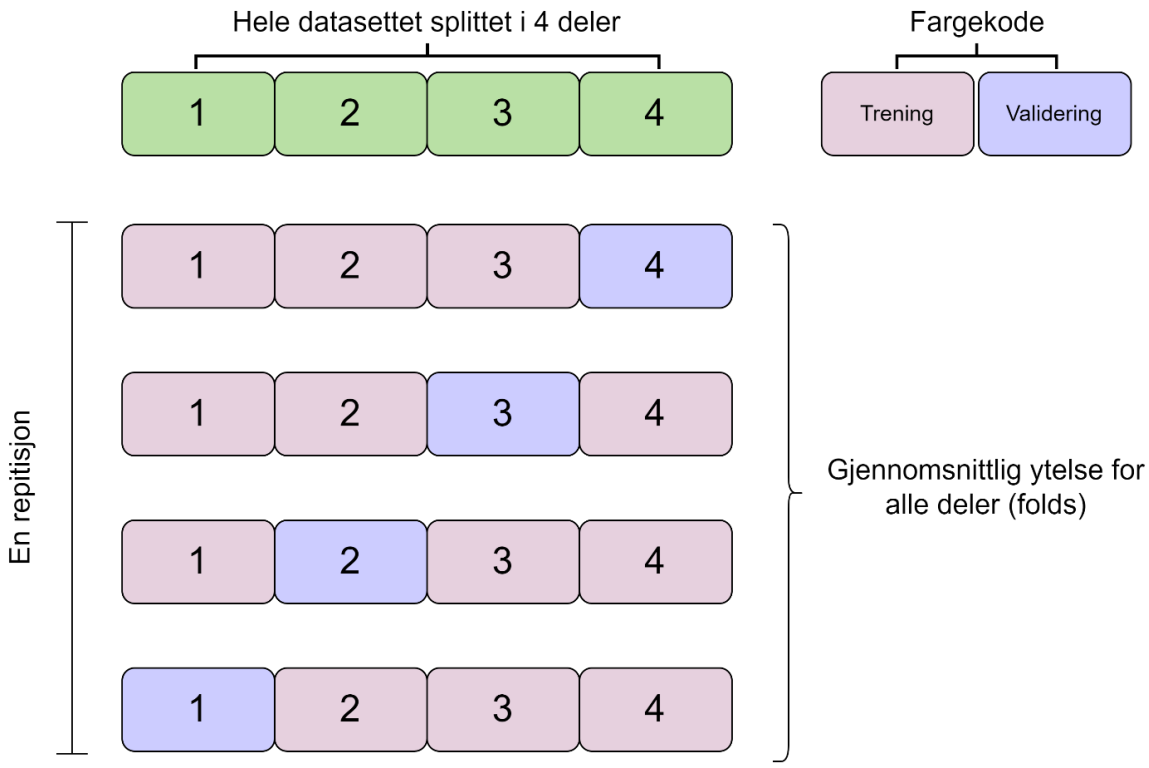
Kryssvalidering er et viktig element som brukes i maskinlæring for å evaluere modellens ytelse og estimere dens evne til å generalisere til ukjente data. Vanligvis deler man datasettet i to separate deler; treningsdata og testdata (Raschka & Mirjalili, 2019). Treningsdataen brukes til å trene modellen, mens testdataen brukes til å evaluere modellens ytelse. Kryssvalideringen erstatter den enkle tilnærmingen og deler heller datasettet i mindre grupper av prøver som kalles for "folds" (Raschka & Mirjalili, 2019). Av de fordelte gruppene, blir modellen trent på alle deler utenom en gruppe. Den siste gruppen brukes til å teste/validere modellen. Dette gjentas i kombinasjoner frem til alle grupper har vært testdata en gang. Etter at alle

kombinasjoner er gjennomført, regnes gjennomsnittsresultatet. Dette er med på å gi et nøyaktigere estimat av modellens ytelse enn om det kun hadde blitt brukt én trenings- og testdel. I kryssvalidering er det også mulig å bestemme antall repetisjoner som referer til hvor mange ganger kryssvalidering skal gjentas. Ved å gjennomføre flere repetisjoner med kryssvalideringer vil det komme frem en mer robust og pålitelig evaluering av modellens ytelse siden gjennomsnittlig ytelse baserer seg på flere modeller.

Hvilke metoder som skal brukes av enten kryssvalidering eller den enkle tilnærmingen med splitting av trening og test, er basert på selve datasettet (VanderPlas, 2016). Et datasett med veldig mange observasjonsprøver vil den enkle tilnærmingen med en trening og testdata gi nøyaktig estimat. Dersom antallet observasjonsprøver i et datasett er lavt, vil kryssvalidering være en avgjørende teknikk for å oppnå et stabilt estimat. I denne oppgaven består datasettene for kolorektal kreft og hode- og halskreft av relativt få prøver, henholdsvis 192 og 197 prøver, noe som gjør kryssvalidering nødvendig for å oppnå stabile resultater.

Ved modellering kan overtilpasning oppstå når modellen tilpasses for mye til treningsdataene, og dermed miste evnen til å generalisere til nye og ukjente data (Raschka & Mirjalili, 2019). Kryssvalidering kan bidra til å gi en mer pålitelig indiksjon på modellens evne til å generalisere til nye data. Dette kan være en stor fordel for å unngå overtilpasning og sikre modellen kan håndtere ukjent data på en effektiv og nøyaktig måte.

Figur 11 illustrerer hvordan kryssvalidering fungerer for et datasett som er splittet i 4 grupper (folds). I denne oppgaven har RepeatedStratifiedKFold fra scikit-learn blitt brukt for kryssvalidering (Pedregosa et al., 2011). Alle modellene i denne oppgaven har blitt kjørt med fire grupper (folds), som har blitt repetert tusen ganger.



Figur 11: Illustrerer et eksempel for kryssvalidering. Et datasett er delt inn i fire deler. I en repetisjon blir hver del brukt til å validere på modellen som er trent med de tre andre delene. I løpet av en repetisjon, skal hver del ha vært valideringssett én gang. Gjennomsnittet av prediksjonene fra hver iterasjon vil rapporteres som resultat. Inspirert av (VanderPlas, 2016).

3.4.3 Evaluering

Evaluering er en fase i CRISP-DM der modellene som er utformet, gjennomgår en omfattende vurdering. Kvaliteten på evalueringen avhenger av hvor godt modellene blir utformet i modelleringsfasen. Ved å gjennomføre en grundig evaluering av modellene, er det mulig å generere resultater som indikerer modellenes ytelse og sammenligne ulike modeller for å identifisere de mest effektive (VanderPlas, 2016). Dette gjør det mulig å vurdere kvaliteten på modellene og fastslå i hvilken grad de oppfyller kravene for å løse problemstillingen. Evalueringen spiller også en viktig rolle i å trekke konklusjoner og identifisere områder der modellene kan forbedres.

Performance metrics

Modellene i oppgaven er evaluert med testdata/valideringsdata fra kryssvalidering som ikke har vært benyttet i opplæringen av maskinlæringsmodellene. Testdata gir mulighet for å evaluere modellens ytelse på nytt og gir et mer realistisk bilde av hvordan en modellen vil prestere på nytt ukjent data. Resultatene til modellene er presentert med fem forskjellige scoretyper: *Accuracy, F1-positiv, F1-negativ, MCC og ROC-AUC*. Ved å presentere resultatene

med alle fem scoretypene, får man en helhetlig vurdering av modellytelsen. Hver scoretype gir informasjon om spesifikke aspekter i modellen, og ved å kombinere disse, får man et balansert bilde av modellens styrker og svakheter. Dette kan bidra til å ta viktige beslutninger i forhold til videre bruk av modellen eller forbedringer som kan gjøres. En detaljert beskrivelse med formler og sammenhenger er beskrevet i kapittel 2.2.6. I oppgaven har det også blitt vagt å presentere resultatene fra modellene som confusion matrix. Med confusion matrix vil det være mulig å vurdere hvordan modellene klarer å unngå falske negative prediksjoner (FN) og falske positive prediksjoner (FP).

MCDA-analyse (Flermålsanalyse)

Som en annen del av evalueringsfasen, blir også en MCDA-analyse (Multiple Criteria Decision Analysis), gjennomført. Ved bruk av MCDA, også kjent som flermålsanalyse på norsk, blir en metodisk tilnærming som er utviklet for å hjelpe beslutningstakere med å ta beslutninger som involverer flere kriterier og alternativer benyttet. En grundig beskrivelse av hvordan metoden fungerer i praksis er beskrevet i kapittel 3.7.

I denne oppgaven presenteres to forskjellige alternativer, og hvilke alternativer disse er og hvilke antagelser som har blitt gjort for utførelsen av analysen blir presentert i kapittel 3.7.2. Det er viktig å påpeke at ingen av de foreslåtte alternativene skal betraktes som en erstatning for den nåværende beslutningsprosessen, som bygger på kliniske retningslinjer og individuelle vurderinger fra et tverrfaglig team samt pasientenes interesser og behov. Alternativene skal dermed anses som et beslutningsstøtteverktøy, som kan brukes i kombinasjon med dagens løsning.

3.5 Kolorektal kreft

Det første datasettet som ble analysert og preprosessert, omhandlet kolorektal kreft. Dette datasettet ble opprinnelig samlet inn som en del av et forskningsprosjekt kjent som OxyTarget, hvor studiet pågikk mellom 2013 og 2017 (Røe, 2018). Studienes formål var å etablere en biopsifri markør for å identifisere aggressive krefttilstander hos pasienter som lider av tykktarmkreft. I dette kapittelet presenteres det som er unikt for dette datasettet. Som påpekt i innledningen av kapittel 3.4, vil fokuset i dette kapittelet være på kun to av de fire fasene i CRISP-DM-metoden, nemlig dataforståelse og dataforberedelse.

3.5.1 Dataforståelse

Det første steget etter at datasettet var tildelt, var å oppnå en grundig forståelse av datasettet slik det er beskrevet i *Fase 1 - Dataforståelse*, illustrert i figur 9. Dette trinnet er essensielt for å gi analytikerne en solid forståelse av datasettet, skaffe en helhetlig oversikt over dataene og identifisere hvordan de kan brukes i modelleringsprosessen. I datasettet bestående av pasienter med kolorektal kreft, har alle trinnene i arbeidsflyten blitt utført på en grundig måte. I det følgende vil en detaljert beskrivelse av de observerte fenomenene under utførelsen av trinnene i fase 1 bli presentert.

Generelt om data

OxyTarget datasettet ble levert i et Excel-format og bestod av to regneark; 1) et infoskriv om variablene og 2) selve datasettet. Datasettet bestod av totalt 192 pasienter, hvorav syv av disse frivillig trakk tilbake sitt samtykke til å bruke deres data i studiene. Disse individenes data ble identifisert i datasettet ved bruk av merkelappen «Withdrawn consent». I tillegg omfattet datasettet 101 variabler, som ga en dimensjon på (192, 101).

Totalt antall manglende verdier i datasettet, også kjent som NaN-values på engelsk, var 6334 som tilsvarer 32.99% av hele datasettet. Det ble observert at noen resultater var merket med bindestrek (-), og det ble besluttet å betrakte disse som manglende verdier i oppgaven.

Den yngste pasienten som var inkludert i datasettet var en 36 år gammel mann, mens den eldste pasienten var en 93 år gammel mann. En tabell som viser fordelingen mellom kjønn og alder i datasettet, presenteres i tabell 18.

Tabell 18: Kjønns- og aldersfordelingen i kolorektal datasettet

Variabel	Kategori	Fordeling
Kjønn	Mann	64.89 %
	Kvinne	35.11 %
Alder	36-50	12.23 %
	51-65	38.30 %
	66-80	36.17 %
	81-93	13.30 %

Datatyper

Kolorektal datasettet bestod av 101 variabler med ulike datatyper som inkluderte flyttall (*eng. float*), heltall (*eng. integer*), tekststreng (*eng. string*) og objekt (*eng. object*). En liste over de ulike datatypene som inngikk i datasettet er oppført nedenfor:

- pandas._libs.tslibs.timestamps.Timestamp
- pandas._libs.tslibs.nattype.NaTType
- datetime.datetime
- numpy.float64

Responsvariabler

I datasettet finnes to mulige endepunkter/responsvariabler som kan brukes for trening av prediksjonsmodeller; generell overlevelse (OS event, *eng. Overall Survival*) og progresjonsfri overlevelse (PFS event, *eng. Progression-Free Survival*). Begge responsene var definert som binære hendelser. I henhold til oppfølgingsprogrammet ble pasientene fulgt opp i en periode på fem år (Røe, 2018). Nedenfor følger en kort beskrivelse av de ulike responsvariablene.

- **OS:** En pasient som overlevde (OS) gjennom oppfølgingsperioden fikk tildelt kategori 0, og de som døde ble tildelt kategori 1.
- **PFS:** En pasient ble tildelt kategori 0 for PFS dersom de forble i live og ikke opplevde lokalt eller metastatisk tilbakefall. Om en av hendelsene inntraff innen oppfølgingsperioden, fikk pasienten tildelt kategori 1.

En tabell som viser responsene og fordelingen mellom klassene, presenteres i tabell 19. I denne oppgaven ble begge responsvariabler brukt for å lage modeller.

Tabell 19: Klassefordelingen for responsvariablene OS og PFS

Respons	Kategori	Fordeling
OS- event	Ikke dø: 0	69.68 %
	Dø: 1	30.32 %
PFS - event	Ikke tilbakefall: 0	59.04 %
	Tilbakefall: 1	40.95 %

3.5.2 Dataforberedelse

Datasettet for kolorektal kreft gjennomgikk de fem stegene som ble presentert i dataforberedelsesfasen i figur 9 for å sikre at datasettet var klar for modellering. Som beskrevet i dataforståelsen om kolorektal kreft, hadde datasettet flere variabler, manglende verdier og kategoriske variabler som måtte behandles før det var egnet for modellering. Nedenfor følger en detaljert beskrivelse av de konkrete tiltakene som ble gjennomført i de første tre fasene for det aktuelle datasettet. Stegene med visualisering og inspeksjon av ekstremverdier er nærmere beskrevet i kapittel 4.1

STEG 1 - Valg av variabler og prøver (*eng. feature selection*)

Det aktuelle datasettet for kolorektal kreft inneholdt en stor mengde variabler, hvorav mange av disse variablene hadde manglende verdier i sine observasjoner. Som beskrevet i kapittel 3.3, besto datasettet av totalt 101 variabler og hele 6334 manglende verdier. For å sikre at variablene var av god kvalitet og for å vurdere om det var nødvendig å eliminere spesifikke variabler, ble to undersøkelsestrinn utført:

I det første trinnet ble datasettet undersøkt manuelt for å vurdere om noen variabler og pasientprøver kan elimineres fra datasettet. Som beskrevet i kapittel 3.5.1, var det syv pasienter som ikke ønsket å delta i forskningen, og disse ble markert som «Withdrawn consent» og eliminert fra datasettet. I tillegg ble noen variabler eliminert manuelt basert på vurderinger av hvor liten verdi variabelen hadde for datasettet. Tabell B.1 i vedlegg B gir en oversikt over hvilke variabler som ble eliminert manuelt og begrunnelsen for elimineringen.

Etter den manuelle elimineringen av enkelte variabler, fortsatte prosessen til det andre trinnet i datasettprosesseringen. To forskjellige tester ble gjennomført for å evaluere kvaliteten på de gjenværende variablene. Den første testen innebar å beregne prosentandelen av manglende verdier per variabel i forhold til det totale antallet prøver som var tilgjengelig. Deretter ble det kalkulerte prosenttallet sammenlignet med en nedre grense som var satt til 25%. Hvis det kalkulerte prosenttallet var over 25%, ville testen eliminere den aktuelle kolonnen. Tabell B.2 i vedlegg B inneholder en tabell med detaljert informasjon om hvilke variabler som har blitt eliminert etter testen.

En lignende test ble gjennomført på pasientprøvene der en nedre grense på 40% ble satt. I tillegg til dette var det to observasjoner i datasettet som manglet responsvariabel. Disse ble også eliminert fra datasettet. Resultatene av dette prosesseringstrinnet førte til betydelige endringer i datasettdimensjonene, hvor antall funksjonsvariabler ble redusert fra 101 til 42 (inkludert responsvariablene) og antall pasientprøver ble redusert fra 192 til 179.

Bøkene *Hands-ON Machine Learning with scikit-learn and Tensorflow* (Géron, 2022) og *Learning predictive Analytics with Python* (Kumar, 2016) diskuterer hvordan variabler og prøver kan elimineres og hvilke konsekvenser det bør legges oppmerksomhet på. Begge bøkene diskuterer eliminering av variabler og prøver som er utsatt av mange manglende verdier. Det er ingen fasit svar på terskelen for å eliminere, da det er varierer avhengig av datasett. For å redusere risikoen for å miste mange variabler som kan ha viktig betydning for å finne mønstre i modellen, ble en terskel på 25 % satt. Terskelen for observasjonsprøver ble satt litt høyere, på 40%, for å unngå for mange observasjoner med substituerte verdier som kunne avvike mye fra de opprinnelige verdiene. Dersom det er for mange substituerte verdier for variablene for en observasjon, kan dette svekke informasjonen om pasienten samt kvaliteten på hele datasettet.

STEG 2 – Behandling av manglende verdier (NaN-values)

Etter at variablene ble valgt i *steg 1 – valg av variabler,* hadde datasettet for kolorektal 502 manglende verdier, noe som fortsatt utgjør en betydelig andel av datasettet. Det finnes en rekke tilnærminger som kan bli benyttet for å behandle manglende verdier. Gjennomsnittverdi, median og frekvens er tradisjonelle metoder som ofte blir brukt for å estimere manglende verdier. Nyere teknikker har blitt introdusert for å oppnå mer presise estimater (Khan & Hoque, 2020).

I dette prosjektet har KNN-imputer blitt brukt, som er en teknikk som bruker algoritmen k-nærmeste naboer (KNN), for å fylle de gjenværende 502 manglende verdier i datasettet (Pedregosa et al., 2011). KNN-algoritmen estimerer manglende verdier ved å bruke verdiene til nærliggende observasjoner i datasettet (Per & Claes, 2006). Algoritmen har blitt konfigurert til å bruke fem naboer, som betyr at den finner de fem nærmeste observasjonene som ligner mest på observasjonen med den manglende verdien. Deretter blir den kalkulerte

gjennomsnittsverdien av disse fem nærmeste observasjonene brukt til å fylle inn den manglende verdien i datasettet.

En betydelig fordel ved å benytte KNN-metoden er at den tar hensyn til å finne forholdet mellom observasjonene i datamengden, og gir dermed mer nøyaktige estimater enn hvis man hadde brukt gjennomsnittsverdien eller medianen fra variablene i datasettet (Per & Claes, 2006). Imidlertid kan det være en ulempe å bruke KNN-metoden hvis datasettet har et stort antall manglende verdier eller betydelige variasjoner i datamengden. Dette kan føre til avvik i de estimerte verdiene som fylles inn, og dermed svekke kvaliteten på analysen.

Noen få NaN-verdier var ikke hensiktsmessige å bruke KNN på, da disse var knyttet til kategoriske data og gjaldt følgende variabler: *Place Surgery, Type of Surgery, Histology Descroption, Mucinous og Blood Type*. Disse manglende verdiene ble behandlet ved å legge til et nytt og unikt element i variablene, nemlig å erstatte NaN-verdiene med teksten «mangler». Variabel *Place Surgery* hadde for eksempel kategoriene *['Ahus', 'DNR', 'Ullevål', NaN]* og dette ble endret til *['Ahus', 'DNR', 'Ullevål', 'mangler']*. Variablen fikk altså en ny kategori som erstattet NaN-verdiene. Ved å anvende denne løsningen, ble alle NaN-verdiene i datasettet behandlet.

STEG 3 – Encoding av kategoriske verdier

For å optimalisere datasettet for modellering og analyse, ble variablene med kategorisk data behandlet ved hjelp av LabelEncoding-teknikken, som beskrevet i steg 3 i kapittel 3.4.2. LabelEncoding er en teknikk som konverterer kategorisk data til numeriske verdier, slik at det kan behandles av maskinlæringsalgoritmer. En oversikt over hvilke variabler det gjelder og resultatet av LabelEncoding er presentert i vedlegg B.2

Etter utførelsen av LabelEncoding på datasettet ble det avdekket en mindre feil i variabelen «Histology description». LabelEncoding resulterte i 9 forskjellige elementer i variabelen, mens datasettet kun skulle ha hatt 7 forskjellige elementer. En nærmere analyse avdekket at elementet «*Adenocarcinoma»* var lagret på tre forskjellige måter, nemlig "*Adenocarcinoma*", «*Adenocarcinoma* » og «*Adenocarcinoma* ». Disse representerer samme element, men på grunn av mellomrommene ble disse tolket som forskjellige elementer og fikk utdelt hver sin klasse ved LabelEncodingen. Feilen ble oppdaget for pasientene med ID 190 og 192, og ble raskt rettet opp når feilen ble oppdaget

3.6 Hode- og halskreft

Det andre datasettet som ble analysert og preprosessert, omhandlet hode- og halskreft. Datasettet inkluderte hode- og halskreft pasienter som ble behandlet ved Oslo universitetssykehus mellom perioden 2007 og 2013 (Moan et al., 2019). Datasettet var bestående av klinisk informasjon om pasientene med tilhørende PET-parametere. I dette kapittelet presenteres særegenheter ved dette datasettet, samt en beskrivelse av hvordan datasettet har blitt preprosessert.

3.6.1 Dataforståelse

Det første trinnet i analysen av datasettet, som i likhet ble gjennomført for datasettet med kolorektal kreft, bestod i å oppnå en grundig forståelse av datasettet som beskrevet i *Fase 1 - Dataforståelse*, presentert i figur 9. Nedenfor blir en detaljert beskrivelse presentert av de observerte fenomenene ved gjennomføringen av trinnene i fase 1.

Generelt om data

Det kliniske datasettet for hode- og halskreftspasienter ble levert som tre Excel-dokumenter; 1) klinisk datasett, 2) pet parametere til klinisk datasett og 3) respons til klinisk datasett. Tre text-dokumenter levert sammen med Excel-dokumentene, hvor disse inneholdt informasjon om innholdet som ble funnet i Excel-dokumentene.

Kliniske datasettet inkluderte 197 pasienter og 12 variabler som utgjør (197, 12) i dimensjon. Det ble oppdaget totalt 104 manglende verdier i datasettet. Den yngste pasienten som var inkludert i datasettet var en 39 år gammel mann, mens den eldste pasienten var en 79 år gammel kvinne. En tabell som viser fordelingen mellom kjønn og alder i datasettene, presenteres i tabell 20.

Tabell 20: Kjønns- og aldersfordelingen i hode- og halsdatasettet

Variabel	Kategori	Fordeling
Kjønn	Mann	75.13 %
	Kvinne	24.87 %
Alder	39-50	9.14 %
	51-65	61.93 %
	66-79	28.93 %

Excel-dokumentet med PET-parametere inneholder tilhørende PET-parametere for pasienter i det kliniske datasettet. Dokumentet inkluderer tre variabler som er trukket ut av PET bilder og blitt presentert som kontinuerlige SUV-verdier. Tabell 21 presenterer variablene i dokumentet med PET-parametere og hvordan disse har blitt beskrevet i text-dokumentet som fulgte med.

Tabell 21: Beskrivelse av PET-parameterne i hode- og halsdatasettet

Variabel	**Beskrivelse**
SUVpeak	Maksimal gjennomsnittsverdi for SUV (standardisert opptaksverdi) i en kule med volum 1 cm^3 hvor sentrum av kulen (interesseregion for SUVpeak) må tilhøre tumorvolumet.
MTV	Metabolsk svulstvolum: Volum med $SUV \geq 0.5 \cdot SUVpeak$ [cm3]
TLG	Total Lesion Glycolysis: MTV · SUVmean [cm3] (SUVmean = gjennomsnittsverdi SUV i MTV)

Datatyper

Det kliniske datasettet inneholder datatypene av flyttall (float) og heltall (integer). Av 12 variabler var kun to av disse numeriske mens de resterende var binære variabeltyper. Variablene «age» og «pack years» var de med numeriske verdier i datasettet. Tabell 22 presenterer en oversikt over variablene med tilhørende beskrivelse.

Tabell 22: Beskrivelse av variablene i det kliniske datasettet

Variabel	**Beskrivelse**
Age	Alder på pasienten i år
Female	Om pasienten er kvinne eller mann. Klasse 0 = Mann. Klasse 1 = Kvinne
cavum_oris	Om primærsvulsten er i munnhulen eller ikke. Klasse 0 = Nei. Klasse 1 = Ja
oropharynx	Om primærsvulsten er i munnsvelg eller ikke. Klasse 0 = Nei. Klasse 1 = Ja
hypopharynx	Om primærsvulsten er i stupesvelg eller ikke. Klasse 0 = Nei. Klasse 1 = Ja

larynx	Om primærsvulsten er i strupehode eller ikke. Klasse 0 = Nei. Klasse 1 = Ja
histgrade_high	Historisk grad (G) av svulstvev Klasse 0 = G1-G2. Klasse 1 = G3
hpv_related	Om pasienten har positiv HPV Klasse 0 = Negativ status. Klasse 1 = Positiv status
ecog	Oversikt over ecog status Klasse 0 = Status 0. Klasse 1 = Status 1-3
charlson	Oversikt over Charlson Comorbidity Index (multisykelighets score) Klasse 0 = index 0. Klasse 1 = Index 1-6
pack_years	Oversikt over antall år med røyking av 20 sigaretter per dag.
uicc8_III-IV	Oversikt over pasientens kreftstadium Klasse 0 = Stadie 1-2. Klasse 1 = Stadie 3-4.

Responsvariabler

I datasettet er det tre mulige endepunkter/responsvariabler som kan brukes for å trene prediksjonsmodeller; generell overlevelse (OS event, *eng. Overall Survival*), sykdomsfri overlevelse (DFS event, *eng. Disease-Free Survival*) og lokalt-regionalt tilbakefall (LRC – *eng. Local-Regional Control*). Alle tre responsene var definert som binære hendelser. I henhold til oppfølgingsprogrammet ble pasientene fulgt opp i en periode på fem år (Moan et al., 2019). Nedenfor følger en kort beskrivelse av de ulike responsvariablene og deres klasse.

- **OS:** En pasient som forble i live (OS) innen oppfølgingsperioden, fikk tildelt kategori 0. Dersom pasienten døde fikk den tildelt kategori 1.
- **DFS**: En pasient som forble i live og ikke opplevde tilbakefall av kreften i form av regional, lokalt eller metastatisk innenfor gitt oppfølgingsperiode, ble tildelt kategori 0 for DFS. Dersom pasienten opplevde et av disse tilbakefallsformene, ble pasienten tildelt kategori 1 for DFS.
- **LRC:** En pasient som verken fikk lokalt eller regionalt tilbakefall, fikk utdelt kategori 0 for LRC, og kategori 1 dersom pasienten opplevde lokal eller regionalt tilbakefall.

En tabell som viser responsene og fordelingen mellom kategoriene, presenteres i tabell 23. Av disse tre responsene ble OS-event og DFS-event brukt til å lage modeller. Dette ble gjort for å sette et godt sammenligningsgrunnlag når resultatene med begge krefttypene, altså når kolorektal og hode- og halskreft sammenlignes. Det ble ikke gjort modeller med LRC-event som responsverdi da fordelingen mellom kategoriene er veldig skjevfordelt.

Tabell 23: Klassefordelingen for responsvariablene OS, DFS og LRC for hode- og halskreft datasettet

Respons	**Kategori**	**Fordeling**
OS- event	Ikke dø: 0	61.42 %
	Dø: 1	38.58 %
DFS - event	Sykdomsfri overlevelse: 0	54.31 %
	Enten død eller fått tilbakefall: 1	45.69 %
LRC – event	Ikke - lokalt-regionalt tilbakefall: 0	75.63 %
	Lokalt -regionalt tilbakefall: 1	24.37 %

MAASTRO datasett

I tillegg til det presenterte datasettet for hode- og halskreft ble et lignende datasett levert med navnet MAASTRO. Dette er et eksternt datasett fra MAASTRO-klinikken i Nederland. Ifølge tekst-dokumentet som fulgte med, bestod det opprinnelige datasettet av 198 pasienter som senere ble redusert til 99 pasienter. Datasettreduksjonen var forårsaket av manglende klinisk data og ufullstendige bildedata for enkelte pasienter. Datasettet ble samlet i perioden mellom 2008 og 2014. I denne oppgaven har MAASTRO datasettet kun blitt brukt som testdata.

MAASTRO datasettet inneholder 11 variabler som utgjør dimensjonen (99, 11). Den største forskjellen mellom MAASTRO datasettet og datasettet fra OUS ligger i antall variabler. Datasettet fra OUS har en ekstra variabel med variabelnavnet «ecog» som tar for seg ECOG performance status. Dette er en variabel som har blitt anbefalt å droppe i text-dokument «read_me_clincal.txt» med en begrunnelse om at dette ikke er tilgjengelig i MAASTRO datasettet.

I MAASTRO datasettet er den yngste pasienten 41 år gammel mann, mens den eldste pasienten er en kvinne på 83 år gammel. En tabell som viser fordelingen mellom kjønn og alder i datasettene, presenteres i tabell 24.

Tabell 24: Kjønns- og aldersfordelingen i MAASTRO datasettet

Variabel	Kategori	Fordeling
Kjønn	Mann	73.73 %
	Kvinne	26.26 %
Alder	41-50	13.13 %
	51-65	51.51 %
	66-83	35.36 %

I datasettet fra MAASTRO klinikken er det også mulige å bruke alle de tre responsene som ble presentert i datasettet fra OUS. Tabell 25 viser responsene og fordelingen mellom kategoriene.

Tabell 25: Klassefordelingen for responsvariablene OS, DFS og LRC (MAASTRO datasettet)

Respons	Kategori	Fordeling
OS- event	Ikke dø: 0	46.46 %
	Dø: 1	53.54 %
DFS - event	Sykdomsfri overlevelse: 0	40.41 %
	Enten død eller fått tilbakefall: 1	59.59 %
LRC – event	Ikke - lokalt-regionalt tilbakefall: 0	75.75 %
	Lokalt -regionalt tilbakefall: 1	24.25 %

3.6.2 Dataforberedelse

Datasettene med hode- og halskreft hadde ikke behov for å gjennomgå alle de fem stegene som ble illustrert i figur 9 for dataforberedelsesfasen. *Steg 3 - encoding* var ikke relevant for datasettene fra OUS og MAASTRO, da disse ikke inneholdt kategoriske verdier. Nedenfor følger en beskrivelse på hva som ble utført spesifikt i de to første stegene av prosesseringsfasen. Stegene med visualisering og inspeksjon av ekstremverdier er presentert i kapittel 4.4

STEG 1 - Valg av variabler og prøver *(eng. feature selection)*

Som presentert i dataforståelse hadde datasettet fra OUS en ekstra variabel enn datasettet fra MAASTRO klinikken. Variabelen tok for seg informasjon om «ecog» ytelsesstatus. På bakgrunn av at dette er en variabel som ikke er inkludert i MAASTRO datasettet, ble den eliminert fra OUS datasettet. Dette var også anbefalt av tekst-dokumentet som fulgte med.

Med hensyn til at datasettene for hode- og halskreft hadde færre variabler enn datasettet for tykktarm, ble det ikke nødvendig å gjennomføre flere tiltak i denne fasen for begge datasettene fra OUS og MAASTRO.

STEG 2 – Behandling av manglende verdier (NaN-values)

Datasettet fra OUS hadde totalt 104 manglende verdier i kontrast til MAASTRO som hadde ingen. Dermed var det kun datasettet fra OUS som trengte behandling. I datasettet fra OUS var det kun variablene «hpc_related» og «uicc8_III-IV» som hadde 52 manglende verdier hver. Variablene inneholder kategoriene 0 og 1. Dermed ble de manglende verdiene i variablene behandlet ved gi dem kategori 2. Tabell 26 presenterer fordelingen mellom kategoriene for begge variablene etter at de manglende verdiene ble behandlet med å tildele kategori 2.

Tabell 26: Fordelingen etter behandling av variabler med manglende verdier

Variabel	Fordeling	
hpv_related	Kategori 0:	31.98%
	Kategori 1:	41.62%
	Kategori 2:	26.40%
uicc8_III-IV	Kategori 0:	37.06%
	Kategori 1:	36.55%
	Kategori 2:	26.39%

3.7 Forenklede MCDA-analyse

I dette delkapittelet vil en detaljert beskrivelse av oppgavens MCDA-analyse bli gitt. Her vil antagelser og begrensninger bli introdusert sammen med alternativer og kriterier. Dette gir en helhetlig og systematisk tilnærming til analysen, som vil sikre en grundig vurdering av ulike løsninger samt gir et solid grunnlag for å velge den optimale løsningen.

3.7.1 Antagelser og begrensninger

For å sikre høy kvalitet på MCDA-analysen har det vært nødvendig å foreta flere antagelser og begrensningene, både på grunn av begrenset informasjon i hvert datasett og offentlig informasjon. Det er verdt å merke seg at det *ikke* har blitt utført en behovsanalyse, noe som kan anses som en fordelaktig faktor i utførelsen av en MCDA-analyse (Siekelova et al., 2021).

Behovsanalyse er en analyse av behovene for å identifisere og definere problemet eller muligheten som anskaffelsen er ment å løse, og for å sikre at alternativene i MCDA-analysen er hensiktsmessige og formålstjenlige (Rolstadås et al., 2020). Dårlig eller mangelfull behovsanalyse kan skyldes flere faktorer. I denne oppgaven var det mangel på kontaktperson med tilstrekkelig ekspertise innenfor sykehusdrift, og tilstedeværelsen av sensitiv informasjon som hindrer tilgang til offentlige kilder. Dette gjør at mulighetsstudie ikke ble basert på det prosjektutløsende behovet. Det er derfor viktig å forstå og klargjøre disse begrensningene, samt ta hensyn til dem når man vurderer resultatene. På samme måte kan en grundig behovsanalyse avgjørende for å sikre at analysen adresserer de viktigste faktorene og tar hensyn til relevante interessenter og deres preferanser.

En annen konsekvens i tilgangen på data er at MCDA-analysen er sårbar for manglende kvantitative data. Dette kan føre til en subjektiv vurdering av alternativene, og dermed begrense helsepersonellets evne til å ta velinformerte beslutninger. For å overvinne dette problemet bør beslutningstakere ta hensyn til både kvalitative og kvantitative egenskaper ved analysen av alternativer i en MCDA-analyse. Videre har det blitt antatt at alle alternativene vil være tilgjengelige og gjennomførbare, selv om noen av alternativene kan være teknologisk avanserte eller kreve spesiell ekspertise. Dette kan være en urealistisk antagelse, da noen av alternativene kan være utilgjengelige på grunn av økonomiske, teknologiske eller praktiske fordeler.

Sammenfattende må beslutningstakere, i dette tilfelle helsepersonell, være oppmerksomme på disse antagelsene og begrensningene. Dette kan ha innvirkning på analysens pålitelighet og validitet, og det er viktig å være bevisst på disse når man vurderer resultatene av analysen. Det er også viktig å huske på at disse antagelsene kan endres dersom det kommer til ny informasjon eller data som utfordrer dem.

3.7.2 Mulighetsstudie

Det undersøkes to ulike alternativer for å øke støtten for beslutningstaking innen kreftbehandling. Formålet med denne analysen er å undersøke hvilke av de to nye alternativene som har størst potensial for å øke beslutningsstøtten for helsepersonell som står ovenfor komplekse valg. Det er viktig å påpeke at ingen av de foreslåtte alternativene skal betraktes som en erstatning for den nåværende beslutningsprosessen.

Dagens løsning

Dagens praksis i norske sykehus innebærer at kreftpasienter gjennomgår behandling basert på kliniske og etiske retningslinjer, samt individuelle vurderinger fra et tverrfaglig team av helsepersonell (Meld.St. nr.34 (2015-2016)). Teamet kan bestå av leger, sykepleiere, fysioterapeuter, sosionomer og andre som kan bidra til å gi best mulig behandling og støtte til pasienten (Kreftforeningen, u.å). Helsepersonell bruker informasjon fra pasientens helsetilstand og diagnose for å bestemme behandlingsmetoder og medisiner som gir best mulig resultat. Beslutninger om behandling tas av helsepersonell basert på deres erfaring og fagkunnskap samt pasientenes interesser og behov. Dette innebærer at det ikke brukes noen automatisert presisjonsteknologi som maskinlæringsalgoritmer for å støtte beslutninger om behandling (Osnes-Ringen, 2023).

Ved å inkludere nullalternativet, kan man vurdere om det er bedre å ikke gjøre noen endringer og beholde status quo, eller om det er bedre å implementere et av de foreslåtte alternativene. Alternativene som presenteres anses som et beslutningsstøtteverktøy, som skal brukes i kombinasjon med dagens løsning. De foreslåtte alternativene er som følger:

Alternativ 1 – De klassiske maskinlæringsalgoritmene + dagens løsning

Dette alternativet referer til en kombinasjon av dagens løsning og bruken av klassiske maskinlæringsalgoritmer for å predikere og understøtte helsepersonellet sine beslutninger. I dette tilfelle, vil dette være algoritmer som random forest, QDA, logistisk regresjon. Ved bruk av disse algoritmene, kan man lage en modell som kan i videre forskning predikere hvilken kreftbehandling som mest sannsynlig er effektiv for en pasient basert på ulike kriterier som pasientens alder, kjønn, sykdomsstadium og så videre. Med andre ord, dette alternativet gir et bredt spekter av algoritmer å velge mellom, som kan gi god prediksjonskraft for behandling. Samtidig kan det være utfordrende å velge den mest hensiktsmessige algoritmen for en gitt oppgave, da det kan være stor variasjon i kvalitet og nøyaktighet mellom ulike algoritmer.

Alternativ 2 - Dynamic Ensemble Selection (DES) + dagens løsning

Dette alternativet referer også en kombinasjon av dagens løsning, men bruken av en spesifikk type algoritmepakke kalt Dynamic Ensemble Selection (DES). DES-algoritmene (i dette tilfelle, KNORA-E, KNORA-U og DES-P) fungerer ved å selektere de beste undermodellene fra en gruppe av modeller basert på dataene som presenteres. Dette skal i teorien gi en mer presis og pålitelig prediksjon, siden DES-algoritmene velger de beste undermodellene basert på den aktuelle datamengden som presenteres. Den burde være bedre tilpasset til ulike typer data enn de klassiske maskinlæringsalgoritmene. DES kan derfor være et viktig alternativ å vurdere i MCDA-analyse, da den i videre forskning kan gi bedre prediksjon om valg av kreftbehandling og påvirke behandlingsforløpet på en mer effektiv måte.

3.7.3 Evalueringskriterier

Innenfor rammen av dette evalueringskriteriet, tildeles hvert av alternativene en poengsum som reflekterer i hvilken grad alternativet tilfredsstiller kriteriene: prediksjonsnøyaktighet, datakvalitet, og funksjonalitet. Kriteriene er fremstilt systematisk med en individuell karakter på en skala fra 1 – 5, hvor 1 tilsvarer lavest poengsum og 5 tilsvarer høyest poengsum. Jo bedre alternativet tilfredsstiller kriteriet, desto høyere score får den. Tabell 27 viser en detaljert oversikt over evalueringskriterier for å oppnå de ulike poengscorene.

Tabell 27: Evalueringsmålene for å oppnå de ulike poengscorene for MCDA-analyse

Score	Prediksjonsnøyaktighet	Data- og modellkvalitet	Funksjonalitet
1	Lav pålitelighet eller nøyaktighet i prediksjonsmodellene	Et lite datasett med få variabler og/eller manglende verdier som påvirker klassifiseringsprosessen negativt	Lav ytelse eller hastighet i teknologiene som kan føre til forsinkelser, store utfordringer med integrasjon
2	Noe pålitelig eller nøyaktig i prediksjonsmodellene	Et lite datasett med noe få variabler og/eller manglende verdier som påvirker klassifiseringsprosessen negativt	Fortsatt noe lav ytelse eller hastighet i teknologi som kan føre til forsinkelser, noe utfordringer med integrasjon
3	Relativ nøyaktighet i prediksjonsmodellene	Et lite datasett med noe få variabler, men klarer å klassifisere korrekt i de fleste tilfellene	Relativ lav ytelse eller hastighet i teknologi som kan føre til forsinkelser, litt integrasjonsutfordringer
4	Meget nøyaktig i prediksjonsmodellene	Et relativt stort datasett som gir god innsikt, ok kvalitet og klarer å klassifisere korrekt i de fleste tilfellene	Relativ høy ytelse eller hastighet i teknologi, men fortsatt noe forsinkelser
5	Høy pålitelighet eller nøyaktighet i prediksjonsmodellene	Stort og variert utvalg av data som gir omfattende innsikt, høy kvalitet og pålitelige, tilstrekkelig datagrunnlag	Høy ytelse eller hastighet i teknologiene, ingen vanskeligheter med integrasjon av nye funksjoner

Prediksjonsnøyaktighet

Referer til hvor godt en modell kan forutsi resultatene av et bestemt datasett. I denne sammenhengen vil prediksjonsevne være et viktig kriterium for å vurdere nøyaktigheten og effektiviteten til DES sammenlignet med andre klassiske ML-modeller. Prediksjonsnøyaktighet vil bli evaluert ved å sammenligne accuracy sammen med de resterende performance metrics scorene, samt se på modellenes evne til å forutsi kreftpasientenes helsetilstand (Qayyum et al., 2020). En høy score i prediksjonsevne vil indikere at modellen gir mer nøyaktige og pålitelige prediksjoner.

- Nøyaktighet: Hvor godt predikerer de ulike alternativene kreftpasienters helsetilstand og utfall?

Data- og modellkvalitet

Referer til klassifiseringen av pasientene basert på det gitte datasettet som brukes i modellen, og deres oppførsel til ulike typer datasett med varierende datainnhold og utfordringer. I denne sammenhengen vil datagrunnlag være det viktigste kriterium for å evaluere hvor godt DES-modellen og klassiske algoritmer kan bruke helsedata om kreftpasienter til å gi pålitelige anbefalinger. Det vil vurderes om modellene bruker tilgjengelige data til å klassifisere riktig, og om den tar hensyn til forskjellige typer data som kan påvirke anbefalingene (Rasheed et al., 2022). En høy score i datagrunnlag vil indikere at modellen klassifiserer pasientene korrekt og gir riktige verdier til hver hendelse, og at den tar hensyn til forskjellige typer data som kan påvirke anbefalingene.

- Klassifisering: Hvor godt klassifiseres dataene som brukes av alternativene, og hvor godt reflekterer de faktiske forholdene til kreftpasientene?

Funksjonalitet

Referer til hvor godt ulike metoder og teknikker kan integreres i modellen for å kunne effektivisere den samt hvor lang tid det tar før modellen gir et svar. I denne sammenhengen vil funksjonalitet bli ansett på som et kriterium for å evaluere relevansen av DES-modellen eller klassiske algoritmene for å integrere alternativer i eksisterende infrastruktur på norske sykehus (Qayyum et al., 2020). Det vil vurderes om modellen støtter ulike metoder og funksjoner som kan effektivere prosessen til modellen og gi en raskere modelleringstid.

- Integrering: Hvor godt kan funksjoner og metoder integreres i alternativene med hensikten om å optimalisere og effektivisere modellene?
- Modelleringstid: Hvordan er ressurs- og tidsbruken til de ulike modellene ved å utføre en test på datasettene?

3.7.4 Vekting av evalueringskriterier

Tabell 28 presenterer en systematisk vekting i prosent av konseptene, sammen med en begrunnelse for valget av vektingen. Noen kriterier er viktigere enn andre, og avgjørelsen om hvilke kriterier som er viktigere avhenger av krav og behov gitt av prosjektet. I denne avhandlingen blir kriteriene tilskrevet ulik vektlegging på grunnlag av diskusjonene og forankringen med en klinisk ekspert, Hanne Osnes-Ringen. Denne tilnærmingen reflekterer

en forsiktig tilnærming til å håndtere situasjoner der det er usikkerhet om betydningen og relevansen av kriteriene som skal vurderes. Samtidig kan det være hensiktsmessig å videre undersøke og analysere de ulike kriteriene og deres relative betydning for å forbedre beslutningsprosessen på en mer informert og presis måte.

Tabell 28: Begrunnelse for vektleggingen av kriteriene for MCDA analysen

Kriterier	**Vekt**
Prediksjonsnøyaktighet I helsesektoren kan prediksjonsnøyaktighet være spesielt viktig i forbindelse med prediktive modeller for diagnostisering og prognostisering av sykdommer. En høy prediksjonsytelse er viktig for å sikre at beslutningene som tas er basert på pålitelige og nøyaktige prognoser. Dette kan bidra til å redusere risikoen for feildiagnostisering og feil behandling av pasienter, og kan også bidra til å optimalisere ressursbruken og redusere kostnadene knyttet til helsevesenet. Årsaken til at dette kriteriet er satt på nesten samme nivå som neste kriteriet, datakvalitet, er for å sikre at algoritmene gir gode resultater. Nøyaktighetsscore er et mål på hvor godt algoritmen fungerer i forhold til det faktiske resultatet. Å ha høy nøyaktighetsscore vil bidra til å styrke tilliten til alternativet, og dermed øke sannsynligheten for at alternativet vil bli brukt og implementert i praksis.	35%
Data- og modellkvalitet Data- og modellkvalitet er det viktigste kriterium siden det utgjør en av de mest grunnleggende byggesteinene i ethvert datadrevet prosjekt eller virksomhet. For å sikre en optimal bruk av datagrunnlaget er det viktig å ha en grundig forståelse av dataene som inngår, samt hvordan de skal brukes. Det er også viktig å ha plass tiltak for å sikre datakvalitet og dataintegritet. I tillegg er det viktig å velge riktig teknologi og verktøy for å håndtere analysere datagrunnlaget på en effektiv og pålitelig måte. For uten at dataene er sikre og pålitelige, kan dette gi feil utfall og grunnlag for helsepersonellet.	40%

Årsaken til at dette kriteriet er vektlagt høyest, er for å sikre tillit og omdømme. Klinisk ekspertise, Hanne Osnes-Ringen, har anbefalt at høy kvalitet på datagrunnlag og minimal feilklassifisering av pasienter er essensielt for å opprettholde helsepersonellenes rykte og ivareta pasientens helse. Dersom klassifiseringen av pasienter er usikre og feilaktig på grunn av dårlig kvalitet på data, kan dette resultere i alvorlige konsekvenser. Kombinasjon av feilaktig klassifisering og lav nøyaktighetsscore kan føre til feil behandling av pasienter og potensielt sette deres helse i fare. Videre kan dette påvirke helsepersonellets omdømme og tillit, som kan ha negative konsekvenser for deres evne til å tilby behandling og omsorg. Dette understreker noe av viktigheten av å vekte dette kriteriet høyt for å sikre sykehusets og helsepersonalets tillit.	
Funksjonalitet Funksjonalitet fremstår som det minst viktigste kriteriet med tanke på at modelleringstiden og at modellene kan akseptere integrasjon av ulike funksjoner og metoder, ikke er like kritisk og viktig. Til syvende sist er det viktigste at svaret som blir gitt, er korrekt og sikker. Uavhengig om modelleringstiden er lang. Den løsningen som sykehuset ender opp med å ta nå, kan mest sannsynlig anses som en dårlig løsning i fremtiden med tanke på teknologien endrer seg. Årsaken til at dette kriteriet gis lav vekt er på grunn av at helsepersonell ikke anser effektivisering som oppnås gjennom modelleringstid og metodeintegrasjon som en kritisk faktor. For helsepersonell er det viktigste at resultatene som fremkommer er troverdige og nøyaktige, derfor vil disse faktorene ha høyere prioritet enn «bonusfaktorene» som modelleringstid og integrasjon. Selv om disse bonusfaktorene kan gjøre arbeidet lettere og raskere, vil feilaktige svar eller manglende nøyaktighet være av større betydning for helsepersonell i deres arbeid.	25%

3.7.5 Beregning av totalscore

Etter å ha identifisert kriteriene, vil hvert alternativ bli vurdert og scoret i henhold til hvert enkelt kriterium. Deretter vil hver score bli multiplisert med kriteriets vekting i prosent. Dette vil gi en vektet score for hvert kriterium, som vil danne grunnlaget for den totale scoren. Den alternativløsningen med den høyeste totale scoren vil være det foretrukne alternativet i henhold til MCDA-analysen. Den totale scoren kan beregnes som følger:

$$Totalscore = \sum_{i=0}^{n} (\text{score}_\text{i} \text{ x vekting}_\text{i} \ (\%)) \tag{3.2}$$

- **Totalscore**: Totalscore for alternativløsningen
- $\mathbf{Score_i}$**:** Rangering av alternativløsningen for kriterium, en skala fra 1 til 5
- $\mathbf{Vekting_i}\ (\%)$**:** Vekting av kriterium, definert i kap. 3.7.4

	Kriterie	Vekt (i %)								
1	Prediksjonsnøyaktighet									
2	Data- og modellkvalitet									
3	Funksjonalitet									
	Alternativ	**Score**	**Sum**	**Score**	**Sum**	**Score**	**Sum**		**Totalscore**	
1	De klassiske algoritmene + dagens løsning									
2	Dynamic Ensemble Selection (DES) + dagens løsning									

Figur 12 Modellen for MCDA-analyse. Denne modellen er egenutviklet i regnearket, Excel. Formålet med modellen er å regne ut totalscoren for de ulike alternativene. Totalscoren baseres på formel 3.2.

Modellen som er presentert i figur 12 er utformet for å fylle inn verdier i de gule og blå områdene. De gule feltene er ment for å inneholde vektprosentene som er definert i evalueringskriteriene, i dette tilfelle under kapittel 3.7.4, med forbehold om at ingen verdier kan ha verdien 0. De blå feltene, derimot, skal inneholde den begrunnende scoren på en skala fra 1 til 5. Denne scoren reflekterer i hvilken grad de ulike alternativene tilfredsstiller kriteriene. Ved å fylle ut disse verdiene, vil denne modellen automatisk beregne den totale scoren som blir fylt inn automatisk i det grønne feltet for alternativet.

Resultatene av MCDA-analysen vil bli presentert i kapittel 4.7.

Kapittel 4

Resultater

I resultatkapittelet presenteres funnene fra analysene som er gjennomført i tråd med den metodologien som er beskrevet i kapittel 3. Dette kapittelet er delt inn i sju seksjoner, hvor de tre første seksjonene presenterer resultatene av datasettet for kolorektal kreft. Dernest blir tilsvarende seksjoner presentert for datasettet for hode- og halskreft. Kapittelet avslutter med resultatene over mulighetsstudie, MCDA-analyse, som har blitt gjennomført. Resultatene vil bli nøye analysert og vurdert for å gi et helhetlig bilde av de observerte mønstrene, og for å kunne gi svar på de presenterte forskningsspørsmål. I de følgende tabellene vil fargekoordineringen være basert på algoritmetypene som er brukt. De klassiske algoritmene er markert med grønt, algoritmene gitt av LazyPredict er markert med gult, og DES-algoritmene er markert med blått.

4.1 Forundersøkelser av datasettet kolorektal kreft

Denne seksjonen presenterer resultater fra utførte forundersøkelser for datasettet om kolorektal kreft. Det presenteres dermed innledende analyser som kommer frem i steg 4 og 5 av dataforberedelse i arbeidsflyten, som er illustrert i figur 9. Formålet med forundersøkelser er å utforske og forstå datasettet som danner grunnlaget for videre forskning og analyse.

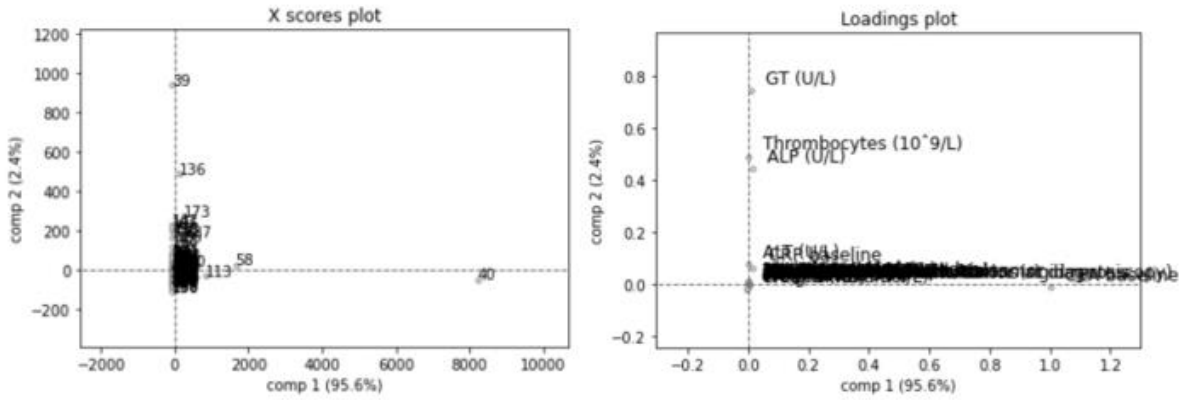
4.1.1 Visualisering

PCA-plott

For å illustrere sammenhenger og varianser i et datasett ble PCA-analyse utført, også kjent som prinsipalkomponentanalyse. PCA-analysen for det sentrerte datasettet er presentert i figur 13 som score- og loadingplot ved hjelp av innebyggede funksjoner i hoggorm-pakke. Disse plottene er primært like, hvor score plott tar for seg spredningen av observasjonene (pasient-ID) langs de to første prinsipalkomponentene (PC1 og PC2) i motsetning til at loadingplot viser hvordan de forskjellige variablene bidrar til prinsipalkomponentene. Sentrert betyr at datasettet er sentrert mot null, ved at gjennomsnittet av hver variabel trekkes fra alle observasjonene i datasettet. Dette gjør at hver variabel har et gjennomsnitt

på null. Ved å sentrere datasettet vil det være enklere å oppdage observasjoner med høy varians, og dermed være med på finne potensielle ekstremverdier.

Ved å stille score- og loadingplot side ved side, kan man se sammenhengen mellom observasjonene (pasient-IDer) og variablene i datasettet. Figuren presenterer også at komponent 1 (PC 1) forklarerer 95.6 % av den totale variansen i kontrast til at komponent 2 (PC 2) forklarer 2.4 % av variansen i datasettet.

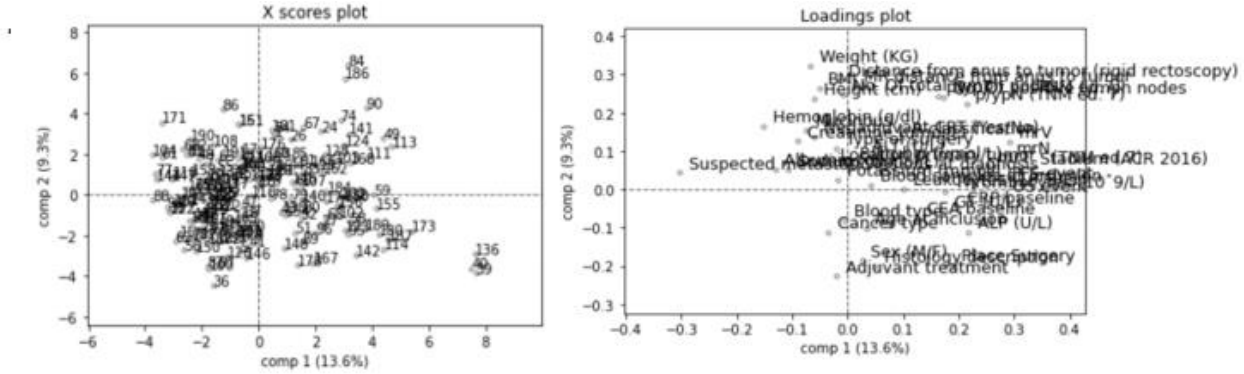


Figur 13: Score- og loadingplot for det sentrerte kolorektal datasettet

Ved første øyekast av det sentrete plottet, fremgår det tydelig at enkelte observasjoner (pasient-IDer) viser høye PCA-score langs en komponent og dermed plassert langt unna den sentrale klyngen av observasjoner i score plottet. Dette gjelder i hovedsak pasienter med indeksene: 39, 40, 58 og 136. En mulig forklaring for disse observasjonene kan være tilstedeværelsen av mulige ekstremverdier (*eng. outliers*) i datasettet. Spesielt pasient 40 kan bli betraktet som en ekstremverdi, siden hele 95.6% av hele variansen er forklart langs PC1. På samme måte kan enkelte variabler ha høye PCA-verdier og dermed plassert langt i fra den sentrale klyngen med variabler i datasettet. I tillegg kan man observere at pasientnummer 39 har en høy verdi for variabelen «GT(U/L)», og pasientnummer 40 har en høy verdi for variabelen «CEA-baseline». Den sistnevnte variabelen, «CEA-baseline», kan ikke observeres tydelig i figuren på grunn av andre variabelnavn som er overlappende.

På lik linje som den forrige figuren presenterer figur 14 score- og loadingplot for PCA-analysen utført på samme datasett, men etter å ha standardisert datasettet. Standardisering av datasettet betyr at hver variabel transformeres til ha gjennomsnitt på null og standardavvik på en. Dette gjøres ved å tekke i fra gjennomsnittet på variablene for hver verdi og dividere

med standardavviket (Raschka & Mirjalili, 2019). Ved standardisering vil store forskjeller i verdier være redusert, og ekstremverdier vil ikke synes like godt ved visualisering.



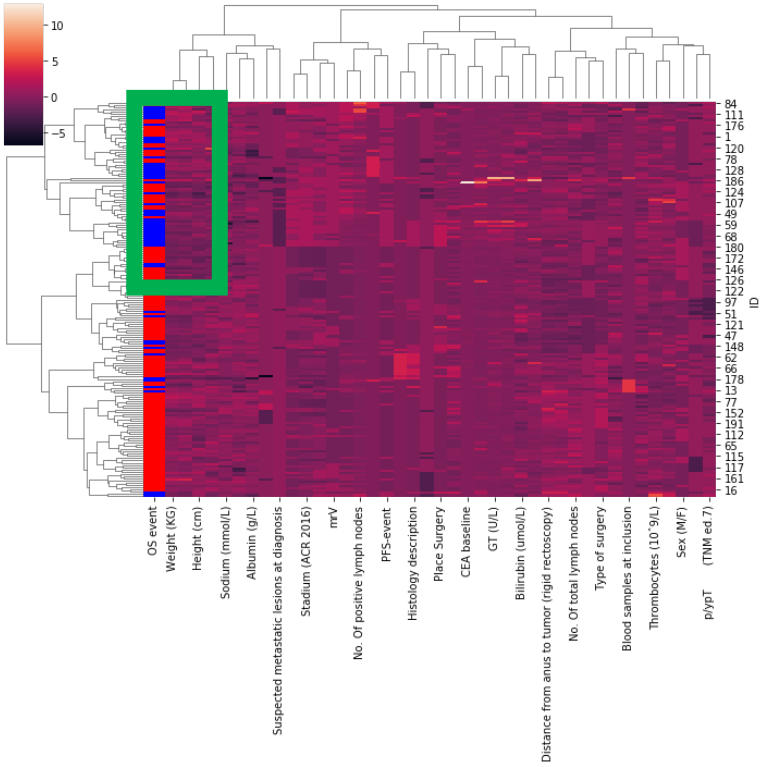
Figur 14: Score- og loadingplot for det standardiserte kolorektal datasettet

Figur 14 informerer om at den total forklarte variansen er redusert for komponent 1 til 13.6% i forhold til hvor mye det ble forklart når datasettet ikke var standardisert. Komponent 2 har derimot økt sin totale forklaring med 9.2% i forhold til 2.4%. Sammenlignet med figur 13, ligger alle observasjoner og variabler mer samlet. Dette kan skyldes av at når datasettet standardiseres, vil alt av data i datasettet ha samme vekt og ekstremverdier vil dermed ikke være like fremtredende som på figur 13. Selv om det ikke er like klare skiller mellom observasjonene i figur 14, er det tydelig at observasjonene 39, 40 og 136 er plassert litt utenfor resten av gruppen i score plottet.

Clustermap

Clustermap-plottet av OxyTarget-datasettet, som er vist i figur 15, demonstrerer at det er to tydelige grupper av pasienter som kan klassifiseres som enten positiv eller negativ for overlevelse (OS-event). De røde stripene i første kolonne helt til venstre i figuren, representerer pasienter med negativ OS-event, som betyr de overlevde. De blå stripene, derimot, representerer pasienter med positiv OS-event, som betyr de ikke overlevde. I en ideell situasjon ville alle de røde stripene være konsentrert i det nederste delen av dendrogrammet og alle de blå stripene i det øvre dendrogrammet som er merket med grønt. Imidlertid er dette ikke tilfelle for OxyTarget-datasettet som inneholder pasienter med tykktarmskreft. Det er tydelig at i øvre kategori, merket med grønt, har større blanding med røde pasienter. Dette kan tyde på at det er pasienter som er veldig like hverandre som har fått to forskjellige utfall i løpet av oppfølgningsperioden på 5 år.

På bakgrunn av at datasettet inneholder mange variabler, har kun de viktigste variablene som lager grupperinger med hverandre blitt presentert i figuren. Fire mindre grupper av variablene har blitt presentert, Dette tyder på at variablene i hver gruppe har en sammenheng med hverandre og kan ha en mulig betydning for utfallet av pasientens tilstand. I figuren kan det også observeres noen hvite stripper langs variablene «CEA baseline» og «GT (U/L)» ved pasient 124 og 186. Dette kan være en mulig tolkning av at observasjonen indikerer på potensielle ekstremverdier. Tidligere i kapittelet kunne man observere i PCA plottene at variabelen «CEA baseline» kan ligge til grunn for utslag av ekstremverdier i datasettet. Dette funnet kan relateres til resultatene i figuren med clustermap.



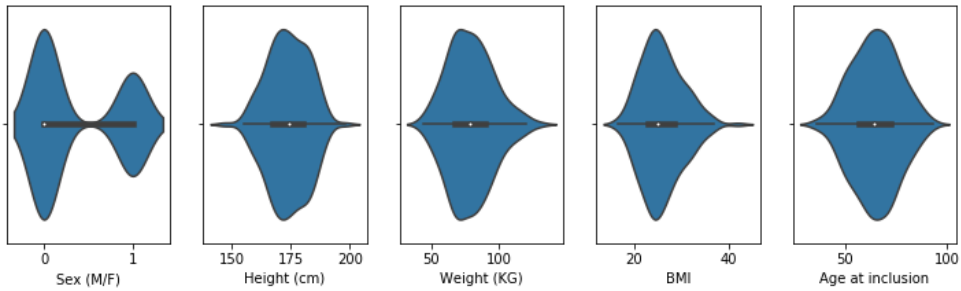
Figur 15: Clustermap av pasientprøvene for kolorektal datasettet med etikett i første kolonne som identifiserer OS verdi

Clustermap tydeliggjør at enkelte pasientprøver vil ha utfordringer med å bli riktig klassifisert av modeller. Årsaken til dette kan være at det mange prøver som tilhører en klasse, men som har blitt plassert i gruppen for den motsatte klassen. Figuren bekrefter også at det er ubalanse i datasettet med ujevn klassefordeling. Dette kan potensielt påvirke modellene negativt som trenes opp av datasettet. I stedet for å bruke fargeetiketter for å finne observasjoner med OS-event, viser figur B.3 i vedlegg B.3 clustermap basert på observasjonene med PFS-event.

Violinplot

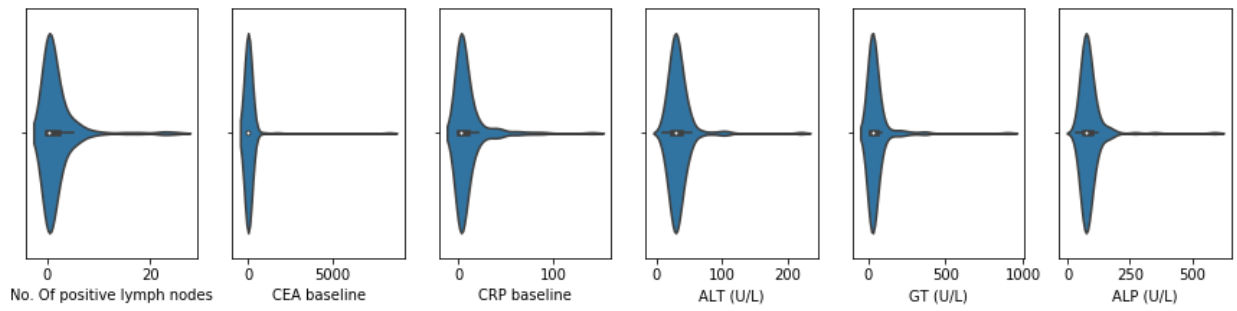
En visualisering av datasettet og en deteksjon av mulige avvik fra gjennomsnittet ble utført ved hjelp av violinplot. Violinplot kan gi verdifull informasjon om datasettets egenskaper som kan være nyttige for analyse og modellering. Datasettet OxyTarget består av totalt 42 variabler etter preprosesseringen, og en figur som viser violinplot for hver av disse variablene er presentert i vedlegg B.4.

Deler av figuren i vedlegg B.4 er presentert i figur 16 og 17. Figur 16 viser en violinplot av personkarakteristikkene til pasientene i datasettet, altså kjønn, høyde, vekt, BMI og alder. Figuren viser at det er flere menn enn kvinner i datasettet, med kategori 0 langs x-aksen. Høyde, vekt og BMI er normalfordelt i datasettet, og aldersgruppen er også normalfordelt, med de fleste pasientene i alderen mellom 50 og 85 år.



Figur 16: Violinplot av personkarakteristikkene til pasientene i datasettet. Dette gjelder for kjønn, høyde, vekt, BMI og alder.

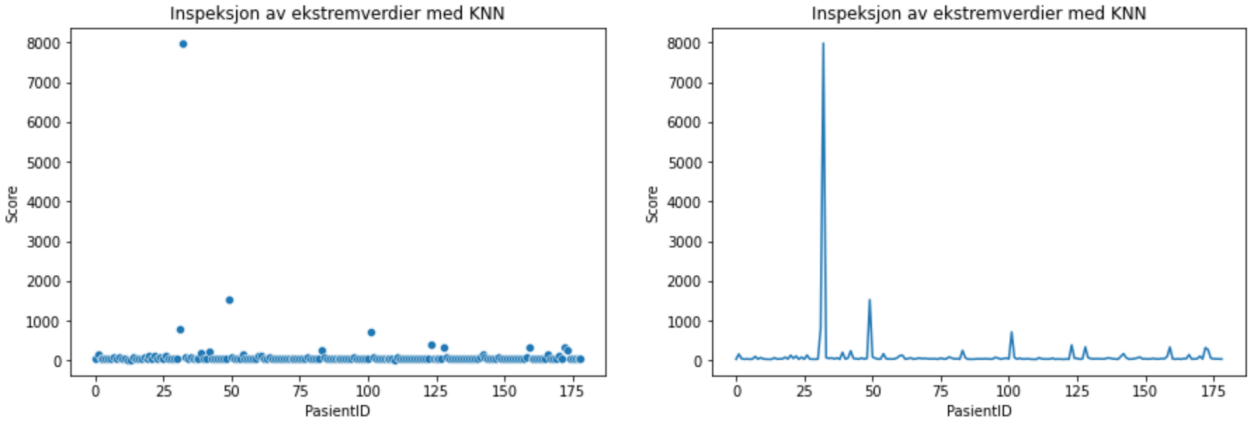
Nedenfor presenteres figur 17 som viser de mest markante variablene observert ved hjelp av violinplot. Samtlige av de seks variablene i figuren viser avvik fra gjennomsnittet og antyder tilstedeværelse av ekstremverdier i datasettet. Spesielt skiller variabelen CEA-baseline seg ut, der det ser ut til å være en eller flere observasjoner med verdi nærmere 5000, når denne variabelen normalt sett har verdier langt mindre enn dette.



Figur 17: Violinplot for de mest markante variablene som ble observert, kolorektal kreft

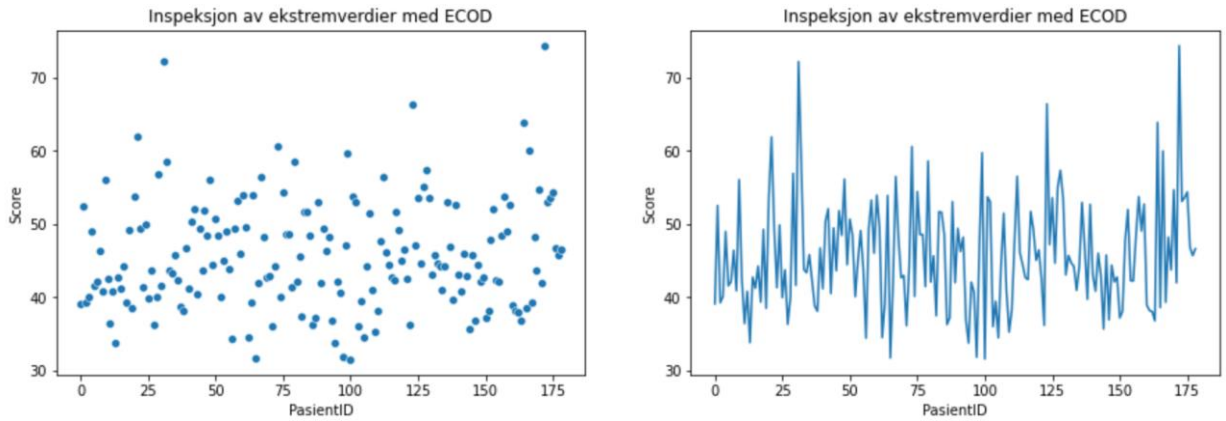
4.1.2 Inspeksjon av ekstremverdier

Figur 18 viser to grafer som presenterer resultatene av en PyOD-analyse av potensielle ekstremverdier i datasettet for tykktarmskreft. Begge grafene viser de samme resultatene, men på to forskjellige måter. Grafen til venstre viser resultatene som et scatterplot, mens den til høyre presenterer resultatene som et linjediagram. Analysen er basert på KNN (K-Nærmeste Naboer), der y-aksen representerer scorene for ekstremverdi (*eng. outliers score*) gitt av analysen, og x-aksen viser pasientene i datasettet.



Figur 18: Et scatterplot (V.S) og et linjediagram (H.S) av ekstremverdiene gjort av en PyOD-analyse med KNN for datasettet kolorektal kreft

Som beskrevet i metodekapittelet, steg-5 i kapittel 3.4.2, ble det utført to analyser for inspeksjon av ekstremverdier med to forskjellige algoritmer. Hensikten var å finne konsistente observasjoner som ble identifisert av begge algoritmer, og på den måten gi et mer solid grunnlag for å avgjøre hvilke observasjoner som kan utelates fra datasettet. Figur 19 viser resultatene fra den andre analysen, som ble utført med ECOD-algoritmen (Empirical Cumulative Distribution Functions).



Figur 19: Et scatterplot (V.S) og et linjediagram (H.S) av ekstremverdiene med en PyOD-analyse med ECOD for datasettet kolorektal kreft

Figur 18, generert med KNN, viser tydelig tilstedeværelsen av noen få observasjoner i datasettet som skiller seg markant fra de øvrige observasjonene, og kan antas som mulige ekstremverdier. Av særlig interesse er en særegen observasjon som skiller seg ut med en ekstremverdiscore på nær 8000. I figur 19 med ECOD observeres en mindre varians, som gjør det noe mer krevende å identifisere mulige ekstremverdier. Imidlertid er det to observasjoner som skiller seg ut med en ekstremverdiscore over 70.

En sammenfatning av de øverste observasjonene med høyest ekstremverdiscore fra analysene, presenteres i tabell 29. De observasjonene som ble oppdaget av begge algoritmene, er markert i tabellen med grønn farge.

Tabell 29: En oversikt over hvilke pasienter som er identifisert som ekstremverdier for kolorektal datasettet. Observasjoner som har blitt identifisert av både modellene med KNN og ECOD er markert med grønn farge.

Algoritme	**PasientID**
KNN	39 – 40 – 58 – 95 – 113 – 136 – 141 – 173 – 186 – 187
ECOD	26 – 39 – 40 – 84 – 90 – 111 – 136 – 178 – 180 – 186

Som det fremgår av tabellen over, er observasjonene med pasient-ID 39, 40, 136 og 186 blitt identifisert som mulige ekstremverdier av modellene som baserer seg på KNN- og ECOD-algoritmene. Tabell 30 viser ekstremverdiscorene for disse observasjonene fra hver modell og mulige årsaker til at observasjonene er identifisert som ekstremverdier. En lignende tabell er presentert i vedlegg B.5 som tar for seg de resterende observasjonene fra tabell 29.

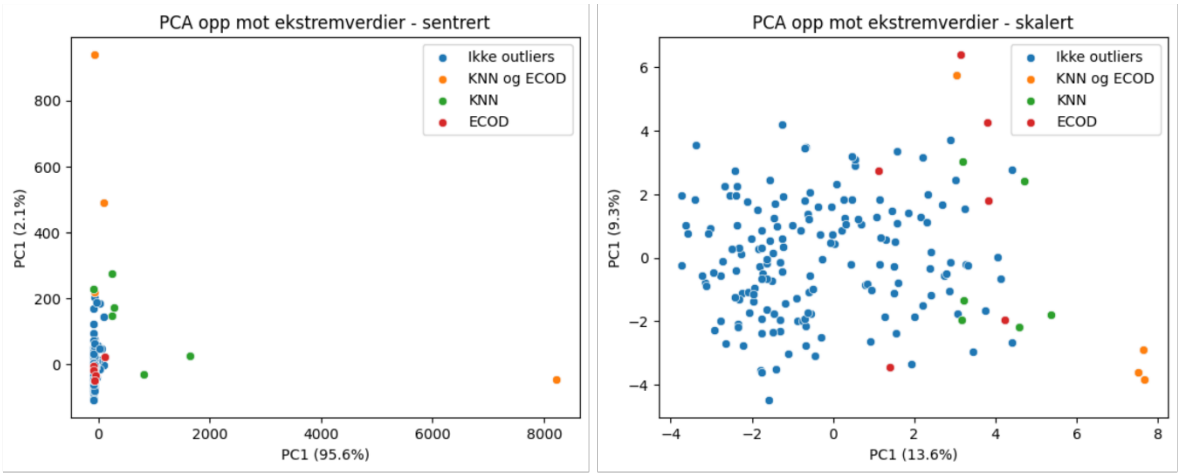
Tabell 30: Oversikt over de identifiserte ekstremverdiene med KNN- og ECOD-score (ekstremverdiscore) samt mulige årsaker til dette resultatet

PasientID	**KNN - score**	**ECOD - score**	**Mulige årsaker**
39	808.56	72.19	Variablene ALT (U/L), ALP(U/L), GT (UL) er veldig høy i forhold til hvor normalen ligger
40	7982.32	58.60	Høy CEA baseline og ALP(U/L) verdi
136	390.52	66.40	CEA baseline, Thrombocytes (10^9/L) og GT (UL) har høy verdi i forhold til hvor normalen ligger
186	325.13	74.36	Høy ALT (U/L) og GT (UL) verdi

Basert på oversikten over mulige årsaker, kan man observere sammenhengen med violinplottet presentert i figur 17 i kapittel 4.1.1. I violinplottet ble det observert at variablene «CEA-baseline», «ALT (U/L)», «ALP (U/L)» og «GT (U/L)» hadde noen prøver med høye verdier for disse variablene. Dette stemmer overens med observasjonene som ble identifisert som mulige ekstremverdier, og som viser høye verdier på de samme variablene.

4.1.3 Analyse av ekstremverdier i forhold til PCA

Figur 20 illustrerer PCA-plott med fargelegging av observasjoner presentert i tabell 29 som tar for seg mulige ekstremverdier i datasettet. To grafer er illustrert i figuren: en til venstre side som viser det sentrerte datasettet og en til høyre side som viser det standardiserte datasettet. De fire observasjonene som ble identifisert som ekstremverdier av både KNN og ECOD er fargekodet oransje i PCA-plottet. De resterende seks observasjonene som ble foreslått av KNN er indikert med grønt, og de resterende seks foreslått av ECOD er fargekodet med rødt i plottet.



Figur 20: PCA-plott med fargelegging av ekstremverdier uten standardisering (V.S) og med standardisering (H.S)

I henhold til kapittel 4.1.1, kan det fastslås at den sentrerte PCA-analysen forklarer en større andel av variasjonen sammenlignet med den standardiserte. Som en konsekvens av dette, kan det observeres en enda mer uttalt deteksjon av ekstreme verdier i en ikke-standardisert PCA-plott. Det er tydelig at observasjoner markert med fargen oransje avdekker potensielle ekstremverdier, da de skiller seg ut fra de andre observasjonene i datasettet. De grønne observasjonene har også noe større varians enn hvor de generelle ligger. De observasjonene som ble identifisert av ECOD, markert med rødt, ser ut til å være godt integrert med resten av observasjonene i datasettet. Hvilket indikerer at deres status om ekstremverdier er ikke

nødvendigvis overbevisende. Det kan observeres at PCA-plottet med standardisert data ikke resulterer i like tydelige resultater som det tilsvarende plottet generert med sentrerte data, slik det fremgår av venstre plott i figuren. Likevel, er det mulig å identifisere enkelte observasjoner som skiller seg betydelig ut fra resten av datasettet, og som dermed kan indikere forekomsten av ekstremverdier. Dette er spesielt tilfelle for de oransje markeringene på plottet.

I tabell 31 blir en oversikt over PasientID til de forskjellige observasjonene presentert sammen med deres respektive fargekoder som er vist i PCA-plottet. Videre gir tabellen en detaljert sammenlikning mellom observasjonene som ble klassifisert som potensielle ekstremverdier i analysene med ECOD og KNN, og de som kan identifiseres som like kun ved å betrakte PCA-plottet uten standardisering.

Tabell 31: Sammenhengen mellom KNN, ECOD og PCA med tanke på ekstremverdier, samt hvilke fargekoder de ulike ekstremverdiobservasjonene har fått i PCA-plottet

KNN	ECOD	PCA
39	39	39
40	40	40
136	136	136
186	186	186
58	26	58
95	84	95
113	90	113
141	111	141
173	178	173
187	180	187

Både tabell 31 og figur 20 viser tydelig til at de fire observasjonene med indeksene 39, 40, 136 og 186 som ble identifisert som ekstremverdier av både KNN og ECOD, også skiller seg ut i PCA-plottet. Videre viser det seg at de resterende seks observasjonene som KNN identifiserte, også er mulig å betrakte som potensielle ekstremverdier av å kun se på det sentrerte PCA-plottet. Basert på disse resultatene ble alle observasjonene med oransje og grønn fargekode i tabellen eliminert fra datasettet før modellering. De røde observasjonene forble i datasettet, da ekstremverdiscoren var nær grensen til å havne i gruppen med resten av observasjoner og fordi PCA-plottet indikerte at de var godt plassert sammen med resten av observasjone

4.2 Kolorektal kreft med OS som responsvariabel

I denne seksjonen presenteres resultatene fra maskinlæringsmodellene som ble trent på datasettet av kolorektal pasienter, hvor OS-event ble brukt som responsvariabel. Som beskrevet i metodekapittelet, referer OS-event til total overlevelse, hvor pasienter som forble i live innen oppfølgingsperioden på fem år, fikk tildelt klasse 0. Dersom pasienten døde i oppfølgingsperioden, ble klasse 1 tildelt.

4.2.1 Evaluering av modeller på testdata

Som nevnt i metoden ble modellene i oppgaven hyperparameteroptimalisert. Hver modell med den beste kombinasjonen fra optimaliseringene ble trent med 4-foldet kryssvalideringsprosess med hele tusen repetisjoner. Tabell 32 presenterer de gjennomsnittlige testresultatene for modellene.

Tabell 32: Oversikt over de gjennomsnittlige testresultatene på test/valideringsdata for kolorektal med OS-event

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random forest	0.79 ± 0.04	0.51 ± 0.12	0.87 ± 0.03	0.43 ± 0.13	0.67 ± 0.06
Logistisk regresjon	0.76 ± 0.06	0.55 ± 0.11	0.83 ± 0.04	0.39 ± 0.14	0.69 ± 0.07
QDA	0.80 ± 0.05	0.60 ± 0.11	0.86 ± 0.04	0.48 ± 0.14	0.73 ± 0.07
GaussianNB	0.78 ± 0.07	0.63 ± 0.10	0.84 ± 0.07	0.48 ± 0.14	0.73 ± 0.07
Nearest centroid	0.76 ± 0.06	0.61 ± 0.09	0.82 ± 0.05	0.45 ± 0.13	0.74 ± 0.07
KNORA-E	0.79 ± 0.04	0.51 ± 0.13	0.87 ± 0.03	0.43 ± 0.03	0.67 ± 0.07
KNORA-U	0.79 ± 0.04	0.50 ± 0.13	0.87 ± 0.03	0.42 ± 0.14	0.67 ± 0.07
DES-P	0.79 ± 0.04	0.50 ± 0.13	0.87 ± 0.03	0.43 ± 0.14	0.67 ± 0.07

I tabellen over er det tydelig at modellen med QDA kommer bedre ut enn random forest og logistisk regresjon for alle fem performance metrics. F1-positiv er relativ lav for alle modeller enn F1-negativ. Dette indikerer at modellene er mer sikre i å predikere klasse 0 enn klasse 1. Dette er et resultat som kunne forventes, da OS-event er en responsvariabel som er veldig skjevfordelt. Det er en betydelig mulighet for at en modell vil være i stand til å tilpasse seg den overpresenterende klassen og dermed prestere bedre på den enn den underpresenterende klassen.

De to algoritmene er merket med gult, viser de gjennomsnittlige testresultatene for algoritmene som har blitt inkludert etter forslag fra LazyPredict pakken. I følge LazyPredict skal GaussianNB gjøre det respektabelt for datasettet med kolorektal kreft. Dette kommer tydelig frem i resultattabellen sammenlignet med random forest og logistisk regresjon. Basert

på resultatene er GaussianNB ganske lik QDA, men gjør det noe bedre i F1-score som tyder på at modellen har klart å tilpasse klasse 1 bedre enn QDA og de andre modellene. MCC-scoren for GaussianNB er lik MCC-scoren for QDA som sammen topper med høyest MCC-score av modellene i tabellen. Dette skyldes av at sann positiv (eng. true-positive) som tar for seg antall korrekte klassifisert av klasse 1 er noe høyere for disse algoritmene som vist i figur 21 og dermed unngår falsk negativ (eng. false negative) i større grad.

Klassifiseringsalgoritmen nearest centroid ble foreslått av LazyPredict for datasettet hode- og halskreft. Denne algoritmen gjør det også ganske bra for kolorektal datasettet med nøyaktigheter som kan måle med resultatene til GaussianNB og QDA. Figur 21 med confusion matrix viser til at nearest centroid håndterte FN i like god grad som modellen med QDA.

Algoritmene merket i blått presenterer algoritmene fra DESlib-pakken. Testresultatene til KNORA-E, KNORA-U og DES-P gir relativ like poengscore. Som det var for random forest og logistisk regresjon, presterer de dårligere når det kommer til prediksjon av positiv klasse. Sammenlignet med de fem andre modellene, gjør modellene med DES-algoritmer svakere enn GaussianNB, QDA og nearest centroid. I forhold til random forest og logistisk regresjon, presterer modellene ganske lik random forest og noe bedre enn logistisk regresjon sett bort i fra at logistisk regresjon gjør det bedre for den positive klassen.

Den totale modelleringstid *(eng. run time)* for modellene med random forest, logistisk regresjon, QDA, GaussianNB og nearest centroid var på 1 time, 3 minutter, og 31 sekunder. For KNORA-E, KNORA-U og DES-P var modelleringstid derimot på 6 timer, 10 minutter, og 51 sekunder. Det er verdt å merke seg at tiden for de klassiske ML-modellene også inkluderer modeller med algoritmene SVC og KNN. Tiden for DES-modellene inkluderte også for tiden av modeller med algoritmene META-DES, MCB og OLA. Resultatene på disse modellene er utelatt fra resultatkapittelet, men inkludert i vedlegg C.1.

Figur 21 presenterer confusion matrix over alle de åtte algoritmene for testdata. Det er enkelt å se at modellene har vanskeligheter med å klassifisere pasienter fra den positive klassen riktig. Det kommer tydelig frem at mange av algoritmene har flere predikerte i falsk negativ (*eng. false negativ*e) enn antall riktig predikert i sann positiv (*eng. true positive*). Det er kun logistisk regresjon, QDA, GaussianNB og nearest centroid som har flere sanne positive enn falske negative og dermed håndterer positive klassen bedre enn de andre. Det kommer også

tydelig frem at datasettet er skjevfordelt med flere observasjoner i klasse 0 (69.98%) enn klasse 1 (30.32%). Et interessant resultat fra confusion matrix er at både random forest og modellene med DES-algoritmer er best på å unngå FP (falsk positiv) kategorien.

Figur 21: Confusion matrix over alle de åtte algoritmene på testdata, kolorektal kreft med OS-event

4.2.2 Evaluering av modeller på treningsdata

På lik linje som testresultatene for modellene med OS-event som responsverdi, presenterer tabell 33 de gjennomsnittlige resultatene for treningsdatasettet

Tabell 33: Oversikt over de gjennomsnittlige resultatene på treningsdata for kolorektal med OS-event

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Logistisk regresjon	0.95 ± 0.02	0.90 ± 0.05	0.96 ± 0.02	0.86 ± 0.06	0.92 ± 0.03
QDA	0.97 ± 0.01	0.94 ± 0.02	0.98 ± 0.01	0.93 ± 0.03	0.95 ± 0.02
GaussianNB	0.85 ± 0.04	0.75 ± 0.04	0.89 ± 0.05	0.65 ± 0.06	0.84 ± 0.03
Nearest centroid	0.79 ± 0.02	0.67 ± 0.03	0.85 ± 0.02	0.53 ± 0.04	0.78 ± 0.02
KNORA-E	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
KNORA-U	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
DES-P	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Resultatene i tabellen tyder på at både random forest og algoritmene fra DESlib-biblioteket kan gjøre feilaktige klassifiseringer. En fellesnevner for disse fire modellene er at de er av typen ensemble, som benytter flere beslutningstrær og kombinerer disse for å øke prediksjonsnøyaktigheten. Ved trening av beslutningstrær vil deler av treningsdatasettet bli brukt, og hvis observasjonene er riktig klassifisert for minst ett tre, vil det være mulig å oppnå en prediksjonsnøyaktighet på 1. Dette betyr nødvendigvis ikke at modellene vil være i stand til å gjøre feilfrie prediksjoner med testdata.

GaussianNB og nearest centroid, som ble foreslått av LazyPredict kommer dårligst ut av modellene på treningsdata. De er spesielt svake på klassifisering av den positive klassen. Selv om det virker som disse modellene presterer dårlig på treningsdata, betyr ikke det nødvendigvis at de er ineffektive modeller. Det kan faktisk hende at de har klart å generalisere problemet bedre enn andre modeller, og de som har oppnådd en nøyaktighet nær 1 er overtilpasset til treningsdataen. Det er modellene med GaussianNB og nearest centroid som har minst forskjell mellom trening og testresultatet. Dette betyr at muligheten for overtilpasning finner sted i mye mindre grad for modellene enn for de resterende.

Logistisk regresjon og QDA viser seg å utføre godt på treningsdata. Det ser ut til at de klarer å klassifisere den positive klassen bedre enn GaussianNB og nearest centroid. Basert på testresultatene fra tabell 32 og treningsresultatene ser det ut til at GaussianNB og QDA kommer best ut av de genererte modellene. Begge algoritmene har sitt felleskap i at de følger

Bayes-teoremet og følger en gaussisk fordeling. Forskjellen mellom disse ligger i at GaussianNB antar at alle kategorier/klasser i datasettet har en felles gaussisk fordeling i motsetning til QDA som antar at hver kategori har hvert sitt gaussisk fordeling.

4.2.3 Feil klassifiserte pasienter i testdata

Nedenfor følger tabell 34 som presenterer en matrise med pasienter som har blitt feilaktig klassifisert på valideringsdata under kryssvalidering. Matrisen er utformet ved å trekke de 30 mest utfordrende observasjoner fra hver modell og deretter tatt hvilke observasjonsindekser som er til felles mellom modellene. Det var totalt 13 observasjoner som var til felles og det er disse som presenteres i tabell 34. Vedlegg C.2 inneholder en oversikt over de 30 mest utfordrende prøver som har vært vanskelig å bli klassifisert i riktig kategori fra hver modell. Modellene ble som nevnt tidligere generert med tusen repetisjoner, og derfor består matrisen av antall ganger observasjonen har blitt feilklassifisert av tusen mulige.

Tabell 34: En matrise over antall tilfeller av feilaktig klassifisering av 1000 mulige i testdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med kolorektal kreft med responsen OS.

ID	Klasse	KNORAE	KNORAU	DESP	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
153	1	1000	1000	1000	1000	1000	1000	994	1000	999.25
176	1	1000	998	1000	1000	948	1000	985	1000	991.38
178	1	1000	998	1000	1000	993	997	964	979	991.38
28	1	1000	1000	1000	1000	934	996	990	1000	990
37	1	1000	1000	1000	1000	779	1000	998	1000	972.13
57	1	1000	1000	1000	1000	776	1000	994	1000	971.25
156	0	824	828	854	820	908	759	989	1000	872.75
Gjennomsnitt		974.9	974.9	979.1	974.3	905.4	964.6	987.7	997	969.73

Observasjonene 153, 176, 178 og 28 ser ut til å være de mest utfordrende pasientene i datasettet som har problemer med å bli klassifisert i riktig kategori av modellene. En nærmere undersøkelse av datasettet avdekket at disse observasjonene har en lav verdi for variabelen «CRP-baseline». Verdien disse observasjonene har fått for variabelen er veldig avvikende fra gjennomsnittsverdien på variabelen. Variabelen har et gjennomsnitt på 83.35, men disse observasjonene har fått en «CRP-verdi» under 10. I tillegg til dette har også observasjonene fått en verdi som er avvikende fra gjennomsnittsverdien på 50.58 for variabelen «GT (U/L)». Dette kan være en av årsakene til at modellene har store utfordringer med å klassifisere disse observasjonene i riktig klasse.

Pasientobservasjon 178 er et interessant resultat i forhold til de andre i tabellen. Denne observasjonen kan knyttes til ekstremverdier som ble presentert i kapittel 4.1.3, hvor observasjonen ble identifisert som en mulig ekstremverdi med ECOD. Selv om ekstremverdiscoren var på grensen til å ikke være en ekstremverdi, ble det bestemt å beholde observasjonen. Imidlertid indikerer resultatene i matrisen på at observasjonen kan være en verdi som burde ha vært eliminert. En nærmere titt på datasettet avdekket at observasjonen har krefttypen *tubular adenoma*. Sammenlignet med andre pasienter som har samme utfall og har krefttypen *tubular adenoma*, hadde pasient 178 betydelig mindre verdier for variablene «GT (U/L)» og «ALP (U/L)» som har en gjennomsnittsverdi på 33.38. Dette kan muligens være en av årsakene til at modellene har problemer med å klassifisere pasient 178, og være en av grunnen til at ECOD slo ut prøven som en potensiell ekstremverdi.

Basert på analysen av resultatene fra tabellen, ble det funnet at modellen med logistisk regresjon presterer bedre for observasjonene 37 og 57 som er merket i gult. Det er imidlertid ikke klart hvorfor logistisk regresjon gir bedre resultater for disse observasjonene enn de andre modellene. Videre undersøkelser av datasettet viste at verdiene for variablene «CEA baseline», «CRP baseline» og «Bilirubin (umol/L)» var avvikende fra gjennomsnittsverdiene for disse variablene. Det antydes at disse avvikene kan ha hatt en innvirkning på modellenes evne til å klassifisere observasjonene riktig.

Pasient 156 skiller seg ut fra mange av de andre observasjonene i matrisen ved å ha en avvikende «CRP»-verdi fra gjennomsnittet. I tillegg har pasient 156 høye verdier for variablene «Thrombocytes (10^9/L)» og «No. Of total lymph nodes», som kan ha bidratt til at modellene har hatt utfordringer med å klassifisere observasjonen riktig. Felles for observasjonene i matrisen utenom pasient 156 ligger i at observasjonene tilhører klasse 1 som har vist seg å være en klasse som er utfordrende å bli klassifisert riktig tidligere i seksjonen. En annen fellesnevner for disse observasjonene er at variabelen «GT (U/L)» har en betydelig avvikende verdi fra gjennomsnittet.

4.2.4 Feil klassifiserte pasienter i treningsdata

I likhet med matrisen for feilklassifiserte pasienter i testdata, presenterer tabell 35 en matrise med pasienter i treningsdata som har vist seg å være like utfordrende å klassifisere. I motsetning til testdata vil maksimalt antall feilklassifisert være 3000 for treningsdata. Dette

skyldes av at modellen kjøres med repeated k-fold med 4 splitt og 1000 repetisjoner noe som vil betyr at treningssettet vil bestå av 3 splitt multiplisert med 1000 repetisjoner.

KNORA-E, KNORA-U, DES-P og random forest hadde ingen vanskelige pasienter og klarte å klassifisere pasientene i riktig klasse uten problemer. Dette var forventet ettersom treningsresultatene i tabell 33 viste at modellene fikk nøyaktighetsytelsene på 1, altså feilfriklassifisering. Siden disse ikke hadde noen vanskelige pasienter, ble matrisen formet med hensyn på de resterende algoritmene. Alle vanskelige observasjoner fra treningsdata for hver modell ble trukket ut som en liste og deretter ble matrisen utformet ved trekke ut hvilke pasienter som har vært vanskelig i alle modeller. Tabell C.4 i vedlegg C.3 presenterer tabellen med liste over hvilke pasientobservasjoner som var vanskelig for hver modell.

Tabell 35: En matrise over antall tilfeller av feilaktig klassifisering av 3000 mulige i treningsdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med kolorektal kreft med responsen OS.

ID	Klasse	Logistisk regresjon	QDA	Nearest centroid	GaussianNB
1	0	51	9	2961	922

Matrisen presenterer kun observasjon 1 som den eneste pasienten som var utfordrende å klassifisere i riktig klasse. Ved å betrakte antall feilklassifiserte i forhold til det maksimale antallet på 3000, viser det seg at nearest centroid-algoritmen opplever størst utfordring. Ved en nærmere inspeksjon av datasettet, kan det observeres at variabelen "CEA baseline" har en betydelig lavere verdi enn gjennomsnittet for den samme variabelen. Dette kan antyde at nearest centroid-algoritmen kan ha problemer med å korrekt klassifisere observasjoner med en lav "CEA baseline"-verdi.

4.2.5 Viktige og mindre viktige variabler

Tabell 36 angir en oversikt over variabler som er klassifisert som viktige og mindre viktige for modellene basert på algoritmene KNORA-E, QDA og GaussianNB. Det skal påpekes at ettersom det ikke eksisterer store avvik i betydningen av variabler mellom algoritmene, er de gjenværende algoritmene inkludert i Vedlegg C.4. Det skal også noteres at både tabellen nedenfor og tabellene i vedlegg C.4 kun presenterer de fem mest betydningsfulle og minst betydningsfulle variablene blant totalt 40 variabler i det prosesserte datasettet som ble brukt til modellering. Sorteringen har blitt gjennomført ved å plassere de minst betydningsfulle

variablene øverst i listen, og deretter ordne dem i synkende rekkefølge etter betydning, med de mest betydningsfulle variablene nederst i listen.

Tabell 36: En oversikt over viktige og mindre viktige variabler i datasettet og hvordan variablene påvirker testytelsen. Testytelsen baseres på MCC-scoren.

KNORA-E			QDA			GaussianNB		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
Type of surgery	0.439051	0.000000	MR d. from anus to tumor	0.498793	0.000000	Cancer type	0.502505	0.000000
Place Surgery	0.438328	0.000723	Sodium (mmol/L)	0.492041	0.006751	mrN	0.499473	0.003032
mrN	0.436069	0.002260	ALP (U/L)	0.491241	0.000800	GT (U/L)	0.498230	0.001243
Albumin (g/L)	0.435189	0.000879	Height (cm)	0.490736	0.000505	Mucinous	0.495370	0.002860
p/ypN (TNM ed. 7)	0.435090	0.000100	Sex (M/F)	0.488745	0.001992	ALP (U/L)	0.492826	0.002544
….	….	….	….	….		….	….	….
Stadium (ACR 2016)	0.402483	0.006806	Bilirubin (umol/L)	0.446652	0.000261	Histology descriptio n	0.463906	0.000722
Leukocyte s (10^9/L)	0.400791	0.001692	CRP baseline	0.444204	0.002448	Suspecte d metastati c l.d	0.459086	0.004819
Suspecte d metastati c l.d	0.399205	0.001586	No. Of positive lymph nodes	0.442658	0.001546	Distance from anus to tumor (rigid.r)	0.454499	0.004587
Adjuvant treatment	0.397469	0.001736	Adjuvant treatment	0.437673	0.004984	No. Of positive lymph nodes	0.447704	0.006795
mrT (TNM ed.7)	0.395481	0.001987	R classificati on	0.432797	0.004876	R classificati on	0.438211	0.009493

Tabellen tydeliggjør at det ikke oppstår store endringer i MCC-nøyaktigheten når en variabel blir eliminert fra modellene. Jo mer betydningsfull en variabel er, desto større er reduksjonen i nøyaktighet når variabelen blir fjernet fra datasettet. En sammenligning av nøyaktighetene mellom den øverste og den nederste raden i tabellen avslører at forskjellen i nøyaktighet mellom elimineringen av en mindre betydningsfull og en betydningsfull variabel i datasettet er mindre enn 6,6%. Selv om dette kan betraktes som en markant forskjell av noen, antyder tabellen også at det ikke er noen variabler som er til felles. Det er altså utfordrende å identifisere én eller flere variabler som er betydningsfulle for alle de presenterte modellene, ettersom betydningen av variablene varierer fra modell til modell.

Det er viktig å bemerke at lignende tabeller som viser viktige og mindre viktige variabler, ble også utført for de andre modellene som ble studert. Disse tabellene blir imidlertid ikke presentert senere i kapittelet. De viktige og mindre viktige variablene for modellene som undersøker henholdsvis kolorektal kreft med PFS som responsvariabel, og hode- og halskreft med OS og DFS som responsvariabler, er inkludert som vedlegg D.4, F.4 og G.4. Årsaken til at disse tabellene ikke ble inkludert i hoveddelen av kapittelet var at resultatene ikke var like omfattende som forventet og at oppgaven ikke hadde hovedfokus på dette aspektet.

4.3 Kolorektal kreft med PFS som responsvariabel

I dette kapittelet presenteres resultatene for kolorektal datasettet med PFS som responsvariabel. PFS, en forkortelse for «Progression-Free Survival», som forteller om en pasient vil få tilbakefall av kolorektal kreft etter en behandling. Tilbakefall av sykdommen blir observert i et tidsrom på fem år, og pasientene blir kategorisert i to klasser: klasse 0 og klasse 1. Klasse 0 indikerer verken tilbakefall eller død, og klasse 1 indikerer enten død, lokalt eller metastatisk tilbakefall.

4.3.1 Evaluering av modeller på testdata

På lik linje som seksjon 4.2, ble datasettet for kolorektal med responsvariabelen PFS trent med 4-foldet kryssvalideringsprosess med tusen repetisjoner. Tabell 37 gir en oversikt over de gjennomsnittlige testresultatene for modellene. Disse verdiene har en dynamisk karakter, og kan forventes å endre seg ved hver iterasjon av modellkjøring.

Tabell 37: Oversikt over de gjennomsnittlige testresultatene på test-/valideringsdata for kolorektal med responsen PFS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.76 ± 0.06	0.63 ± 0.10	0.82 ± 0.04	0.47 ± 0.13	0.72 ± 0.06
Logistisk regresjon	0.73 ± 0.06	0.63 ± 0.09	0.79 ± 0.05	0.43 ± 0.13	0.71 ± 0.06
QDA	0.76 ± 0.06	0.66 ± 0.09	0.81 ± 0.05	0.48 ± 0.13	0.73 ± 0.07
GaussianNB	0.74 ± 0.07	0.65 ± 0.09	0.79 ± 0.06	0.44 ± 0.13	0.72 ± 0.07
Nearest centroid	0.74 ± 0.06	0.67 ± 0.08	0.78 ± 0.05	0.46 ± 0.12	0.73 ± 0.06
KNORA-E	0.74 ± 0.06	0.62 ± 0.09	0.80 ± 0.05	0.44 ± 0.13	0.71 ± 0.06
KNORA-U	0.76 ± 0.06	0.63 ± 0.09	0.82 ± 0.04	0.47 ± 0.13	0.73 ± 0.06
DES-P	0.76 ± 0.06	0.63 ± 0.09	0.82 ± 0.04	0.47 ± 0.13	0.73 ± 0.06

Ved å ta utgangspunkt i hver enkel performance metrics i tabellen, kan man observere at accuracy (nøyaktigheten) er lik for fire av klassifiseringsalgoritmene som har blitt presentert.

Disse fire klassifiseringsalgoritmene er random forest, QDA, KNORA-U og DES-P der scoren ligger på 0.76 med en margin på 0.06 Dette kan skyldes av liknende egenskaper på hvordan de håndterer data og foretar beslutninger. I tillegg kan en lignende observasjon bli gjort for de andre performance metricsene, som F1-negativ og MCC. I disse tilfellene gir de samme klassifiseringsalgoritmene lik poengscore, med en liten forskjell på 0.01. F1-negativ er den performance metricsen som gir generell høy score med et gjennomsnitt på 0.80. En høy F1-negativ score indikerer at modellen er i stand til å identifisere og minimere antall falske positive (*eng. false-positive)* prediksjoner, det vil si feilaktig klassifisering av en positiv klasse som negativ. Dette er også et resultat som er mulig å observere i figur 22 som tar for seg confusion matrix av testresultatene.

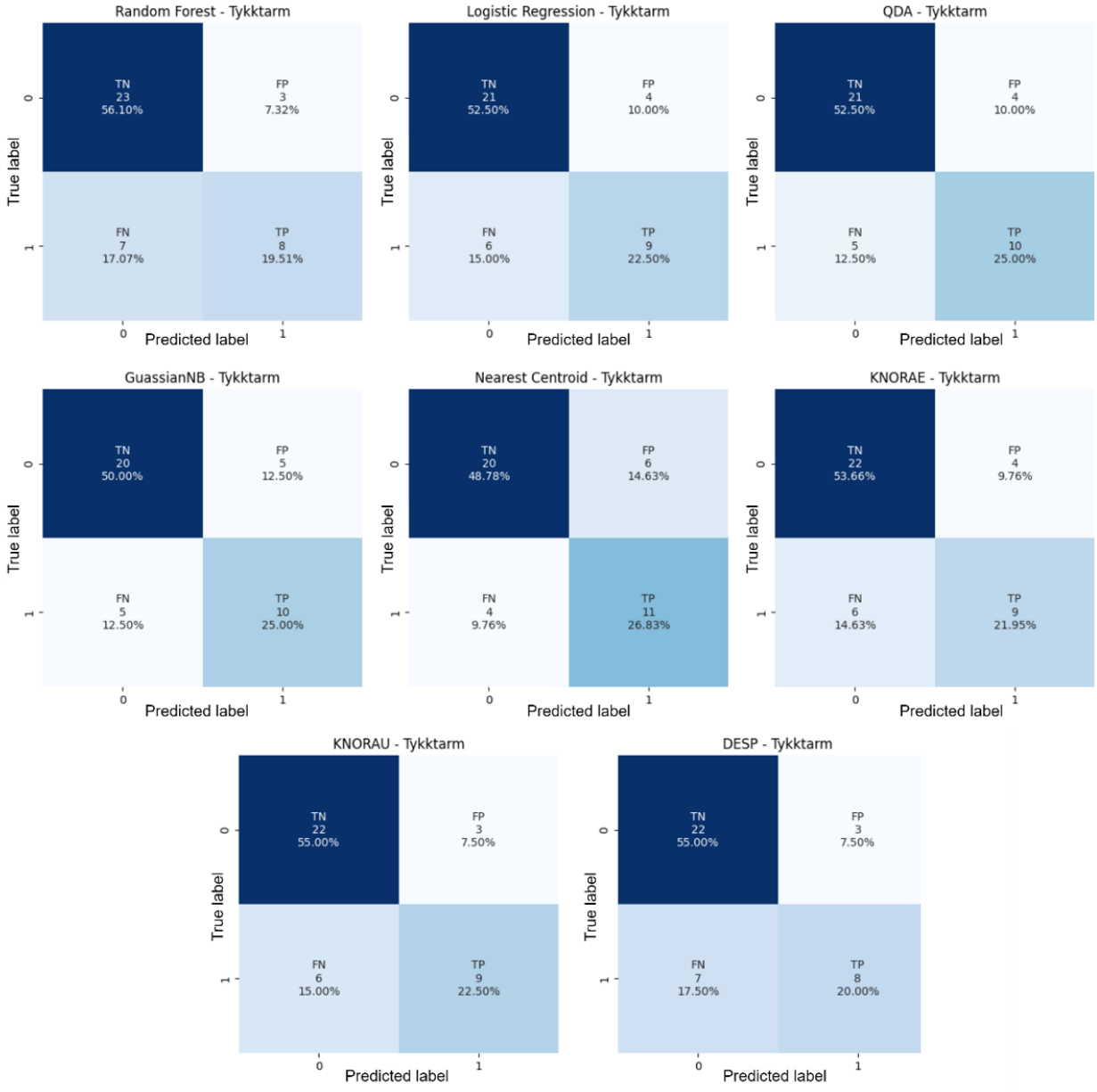
Av de generelle maskinlæringsalgoritmene, kan man se at modellen med QDA presterer best med høyest MCC-måling på 0.48. Ifølge LazyPredict er GaussianNB en respektabel algoritme for datasettet med kolorektal kreft, men for responsen PFS kan man observere at nearest centroid presterer bedre. Til tross for at nearest centroid er en algoritme som ble foreslått av LazyPredict for hode- og halskreft datasettet. I lys av ytelsesmålingene som er utført, kan det observeres at nearest centroid har relativt sett bedre resultater på flere av prestasjonsmålene, spesielt på F1-positiv, Matthews korrelasjonskoeffisient (MCC) og ROC-AUC. En mulig årsak til dette kan være knyttet til de høyere verdiene for sann positiv, som indikerer det riktige antallet positive klassifikasjoner av klasse 1, sammenlignet med de andre algoritmene som ble testet.

Testresultatene til modellene med DES-algoritmer, viser at de har relativt like og jevne prestasjoner sammenlignet med de øvrige algoritmene. Et lite unntak er logistisk regresjon. Sammenlignet med de resterende modellene, scorer DES-algoritmene enten like bra eller bedre for Matthews Correlation Coefficient (MCC). Dette kan skyldes av at disse algoritmene er i stand til å ta hensyn til variasjonene i dataene bedre enn de andre modellene. Ved å se på alle algoritmene for DES, kan man konkludere med at både KNORA-U og DES-P scorer best for DESlib-pakken.

I analysen av modelleringstiden fremkommer det at random forest, logistisk regresjon, QDA, GaussianNB og nearest centroid ble modellert på totalt 56 minutter og 35 sekunder. I kontrast til at KNORA-E, KNORA-U og DES-P hadde en lengre modelleringstiden på 8 timer, 53 minutter, og 57 sekunder. Det er verdt å merke seg at tiden for de klassiske ML-modellene også

inkluderer modeller som algoritmene SVC og KNN. Modelleringstiden for algoritmene fra DESlib-pakken inkluderte også modeller som META-DES, MCB og OLA. Resultatene på disse modellene er utelatt fra resultatkapittelet, men inkludert i vedlegg D.1.

Figur 22 presenterer confusion matrix for alle de åtte algoritmene på testdata. Som for confusion matrix på modellene med OS, er det enkelt å se at den positive klassen har vanskeligheter med å bli klassifisert i riktig kategori. Modellene med random forest og DES-algoritmer ser ut til å være best for å unngå FP kategorien.



Figur 22: Confusion matrix over alle de åtte algoritmene på testdata, kolorektal kreft med responsen PFS

4.3.2 Evaluering av modeller på treningsdata

I likhet med testresultatene for modellene med PFS-event som responsverdi, presenterer tabell 38 resultatene fra treningsdatasettet.

Tabell 38: Oversikt over de gjennomsnittlige resultatene på treningsdata for kolorektal med responsen PFS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Logistisk regresjon	0.88 ± 0.02	0.83 ± 0.03	0.91 ± 0.02	0.75 ± 0.05	0.86 ± 0.03
QDA	0.92 ± 0.01	0.89 ± 0.02	0.94 ± 0.01	0.84 ± 0.03	0.91 ± 0.02
GaussianNB	0.80 ± 0.03	0.73 ± 0.03	0.84 ± 0.04	0.57 ± 0.05	0.78 ± 0.02
Nearest centroid	0.76 ± 0.02	0.70 ± 0.02	0.80 ± 0.02	0.51 ± 0.04	0.76 ± 0.02
KNORA-E	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
KNORA-U	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
DES-P	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

En observasjon av tabellen viser at random forest, merket i grønt, og algoritmene fra DESlib-biblioteket, merket i blått, utfører feilfriklassifiseringer. Disse algoritmene tilhører ensemble-modelltypen, som bruker flere beslutningstrær og kombinerer disse for å forbedre modellytelsen. For å trene beslutningstrær, vil en del av treningsdatasettet bli brukt. Dersom observasjonene er riktig klassifisert for minst ett tre, kan modellen oppnå en prediksjonsnøyaktighet på 1. Dette betyr nødvendigvis ikke at modellene er perfekte. Dette er et resultat som kan indikerer potensiell overtilpasning, spesielt når forskjellen mellom testresultatene i kapittel 4.3.2 og treningsresultatet er så stort for modellene. Overtilpasning kan oppstå av ulike årsaker, som blant annet feilaktige valg av modellparametere eller ikke representative treningsdata.

I analysen gjort av LazyPredict, ble GaussianNB foreslått som den algoritmen som er best egnet for dette datasettet. Det er midlertid viktig å påpeke at den algoritmen er en av de som kommer dårligst ut av modellene på treningsdata. Sammen med nearest centroid, er de spesielt svake på klassifisering av den positive klassen. Selv om det virker som disse modellene presterer dårlig på treningsdata, betyr ikke det nødvendigvis at de er ineffektive modeller. Det kan faktisk hende at de har klart å generalisere problemet bedre enn andre modeller, og unngå potensiell overtilpasning av modeller, da modellene har minst forskjell mellom trening og testresultatet.

Logistisk regresjon og QDA har vist seg å gjøre det relativt bra på treningsdata. Disse modellene ser ut til å kunne tilpasse seg bedre når det kommer til å klassifisere den positive klassen. Basert på testresultatene fra tabell 37 og treningsresultatene ser det ut til at QDA kommer best ut av de genererte modellen. QDA har fått respektable nøyaktighetsmålinger på både test og treningsdata, samt viser indikasjon på at modellen ikke er overtilpasset med lite forskjell mellom trening og testresultat.

4.3.3 Feil klassifiserte pasienter i testdata

Tabell 39 presenteres som en matrise med feil klassifiserte pasienter fra valideringsdata, også kjent som testdata. Utformingen av matrisen er inspirert på lik måte som den tidligere presenterte matrisen i kapittel 4.2.3, hvor de 30 vanskeligste observasjoner fra hver modell blir presentert. Deretter identifiseres hvilke observasjoner som er felles mellom modellene. I dette tilfelle, var det totalt 12 observasjoner som var til felles, og disse blir presentert i tabell 39 i stigende rekkefølge basert på gjennomsnittet. Vedlegg D.2 inneholder listen over de 30 mest utfordrende prøver som har vært vanskelig å bli klassifisert i riktig kategori.

Tabell 39: En matrise over antall tilfeller av feilaktig klassifisering av 1000 mulige i testdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med kolorektal kreft med responsen PFS.

ID	Klasse	KNORAE	KNORAU	DESP	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
81	1	1000	1000	1000	1000	998	1000	1000	1000	999.75
179	1	1000	1000	1000	1000	1000	1000	996	1000	999.50
153	1	1000	1000	1000	1000	1000	1000	993	1000	999.13
143	1	1000	1000	1000	1000	1000	1000	991	1000	998.88
129	1	1000	1000	1000	999	998	1000	987	1000	998
164	1	1000	1000	1000	1000	990	1000	993	1000	997.88
5	1	1000	1000	1000	1000	987	1000	992	1000	997.38
57	1	992	1000	996	990	998	1000	995	1000	996.38
124	0	998	1000	1000	1000	974	986	995	1000	994.13
176	1	982	994	990	990	1000	1000	987	1000	992.89
37	1	998	1000	1000	1000	902	1000	1000	1000	987.50
86	1	934	980	974	965	1000	1000	995	1000	981
Gjennomsnitt		992	997.8	996.7	995.3	987.3	998.8	993.7	1000	994.8

Oversikten viser at modellene hadde betydelige utfordringer med å klassifisere pasientene 81, 179 og 153 i riktig kategori. En fellesnevner for disse pasientene er at de hadde en gjennomsnittscore på omtrent 999 tilfeller hvor algoritmene klassifiserte feil. Både CEA-baselinescore og CRP-baselinescore for pasientene 81, 179 og 153 ligger på omtrent 1. En nærmere gjennomgang av datasettet avslører at gjennomsnittsscoren for «CEA-baseline»

ligger på omtrent 80.3 og CRP-baseline ligger på omtrent 9.1. Dette kan derfor være en av årsakene til de hyppige feilaktige klassifiseringene for disse pasientene.

I matrisen over pasientobservasjonene, er det kun pasient 37, som skiller seg ut fra resten av observasjonene når det kommer til type kreftsvulst (*eng. cancer type*). En detaljert analyse av datasettet viser at pasienten har primærsvulsten i tarmen (tubular adenoma), mens alle de andre pasientene har den vanligste formen for svulst som utgår fra kjertelvev (adenocarcinoma). I tillegg kan man avdekke at denne pasienten er den eneste av de topp feil klassifiserte pasientene som har *transanal endoscopic* i motsetning til at de andre pasientene har «APR» eller «UAR» som «type of surgery». Dette gir en indikasjon på at denne type behandling gir utforinger for maskinlæringsalgoritmene i å klassifisere pasientene.

En annen interessant observasjon gjelder pasient 124. Ut ifra tabell 39 observeres det at denne pasienten ikke har like høy rangering som pasientene 81, 179 og 153 når det gjelder gjennomsnittlig score for feilaktig klassifisering. Ved nærmere undersøkelse av datasettet avdekkes det at pasient 124 har klassetilhørighet 0, mens de andre pasientindeksene i matrisen har klasse 1. En interessant observasjon ved pasient 124 er at den kan knyttes til et resultat som ble presentert med clustermap i kapittel 4.1.1. I clustermap-plottet var det mulig å observere noen hvite striper ved pasient 124, som kunne indikere en potensiell ekstremverdi. Selv om pasienten ikke ble identifisert som en ekstremverdi i testene som ble utført med PyOD, kan dette resultatet fra matrisen indikere at pasient 124 faktisk er en ekstremverdi som ikke har blitt oppdaget av testene presentert i kapittel 4.1.2.

Blant modellene som blir evaluert, viser resultatene at KNORA-U og DES-P presterer omtrent likt med hensyn til antall feilaktig klassifiserte algoritmer etter tusen iterasjoner for de presenterte prøvene. Imidlertid kan man avdekke at KNORA-E er bedre til å klassifisere pasienter siden den algoritmen kun avdekker syv av pasientene med tusen feil klassifiserte pasienter. Når det gjelder de klassiske maskinlæringsalgoritmene kan man observere at logistisk regresjon presterer best sammenlignet med random forest og QDA. En mulig forklaring på dette kan være at random forest er en ensemblemetode som kombinerer flere individuelle beslutningstrær, og det kan være komplekst å finne riktig balanse mellom antall trær og deres dybde for å unngå overtilpasning. Logistisk regresjon, derimot, antar en enkel lineær sammenheng mellom prediksjon og respons.

4.3.4 Feil klassifiserte pasienter i treningsdata

I likhet med matrisen som viser feil klassifiserte pasienter i testdata, presenterer tabell 40 en matrise for pasienter i treningsdata som har vært utfordrende å klassifisere korrekt. DES-algoritmene (KNORA-E, KNORA-U og DES-P) og random forest hadde ikke vanskeligheter med å klassifisere pasientene i riktig klasse, dermed ble matrisen formet med hensyn til de gjenværende algoritmene som hadde erfart vanskeligheter. Tabell D.4 i vedlegg D.3 inneholder en liste over pasientobservasjoner som var utfordrende å klassifisere for hver modell i treningssettet.

Tabell 40: En matrise over antall tilfeller av feilaktig klassifisering av 3000 mulige i treningsdata Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med kolorektal kreft med responsen PFS.

ID	Klasse	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
5	1	2569	2911	2967	2998	2861.25
1	0	1081	2252	1514	3000	1961.75
13	0	8	27	32	52	29.75
179	1	21	21	21	21	21
176	1	2	2	2	2	2
Gjennomsnitt		736.2	1042.6	907.2	1214.6	975.15

Blant de fem presenterte prøvene er det kun pasientobservasjon 5 som er spesielt utfordrende å klassifisere. Resultatene viser at alle maskinlæringsalgoritmene presentert i matrisen har en relativ høy feilklassifiseringsrate på omtrent 2800 tilfeller. En grundigere undersøkelse av datasettet avslørte en mulig årsak til denne høye scoren, nemlig at kun denne pasienten ble behandlet for type APR i motsetning til de fire andre pasientene. I tillegg hadde pasient 5 en veldig lav verdi for variabelen «CEA-baseline», «GT (U/L)» og «ALP (U/L)» sammenlignet med gjennomsnittsverdien på variablene. Gjennomsnittsverdien for «CEA-baseline» er på omtrent 80.3, «GT (U/L)» er omtrent 50.5 og «ALP (U/L)» er omtrent 85.8, mens for pasient 5 er «CEA-baseline» lik 1, «GT (U/L)» lik 24 og «ALP (U/L)» lik 68. Dette kan være en av årsakene til at observasjonen har blitt feilklassifisert en rekke ganger av modellene basert på logistisk regresjon, GaussianNB, QDA, og nearest centroid. Imidlertid viser modellen at logistisk regresjon presterer litt bedre, som kan tyde på at denne klassifiseringsalgoritmen har klart å tilpasse seg denne sammenhengen og derfor kan klassifisere observasjonene mer korrekt.

Ved en grundigere analyse av datasettet kan det observeres at to av pasientene som er inkludert i matrisen, har klassetilhørighet 0 for progresjonsfri overlevelse (PFS-event). Disse pasientene er 1 og 13. Dette faktum kan være betydningsfull faktor som bidrar til forståelsen av hvorfor maskinlæringsalgoritmene feilaktig klassifiserer pasientene. Disse pasientene har også lav verdi for variabelen «CEA baseline» som kan være en av årsakene til at modellene kan ha hatt utfordringer med klassifisering av pasientene.

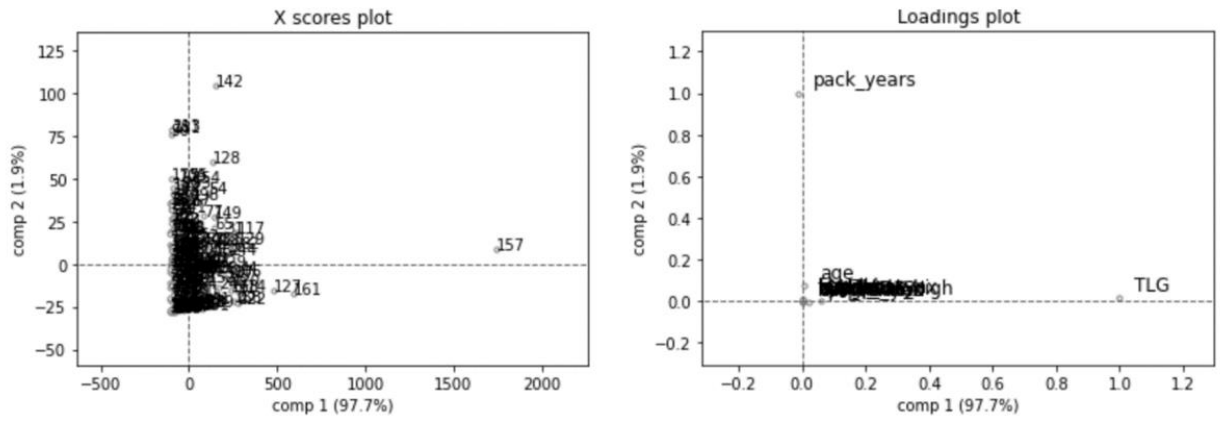
4.4 Forundersøkelser av datasettet hode- og halskreft

I denne delen presenteres resultater fra utførte forundersøkelser av hode- og halskreft datasettet. Dette innebærer presentasjon av innledende analyser som er fremkommet i steg 4 og 5 av dataforberedelse i arbeidsflyten. Formålet med denne seksjonen er å gi et solid grunnlag og en helhetlig forståelse av datasettet før videre analyser utføres. Resultatene fra forundersøkelsene gir en innledende innsikt i karakteristikker ved datasettet, inkludert variabler og sammenhenger mellom dem.

4.4.1 Visualisering

PCA-plott

I likhet med metoden for forundersøkelser av kolorektal kreft datasettet, blir en prinsipalkomponentanalyse (PCA-analyse) brukt for å illustrere sammenhenger og varianser for hode- og halskreft. Resultatene fra PCA analysen er presentert som score- og loadingplot i figur 23, og dette gjelder for datasettet fra OUS som har blitt brukt for å trene modellene. PCA-plottet tar for seg det sentrerte datasettet. Disse plottene gir en visuell presentasjon av struktur til datasettet, hvor score plott visualiserer spredningen av observasjonene (pasient-ID) langs de to første prinsipalkomponentene (PC1 og PC2), og loadingplottet viser bidraget fra variablene til de forskjellige prinsipalkomponentene.

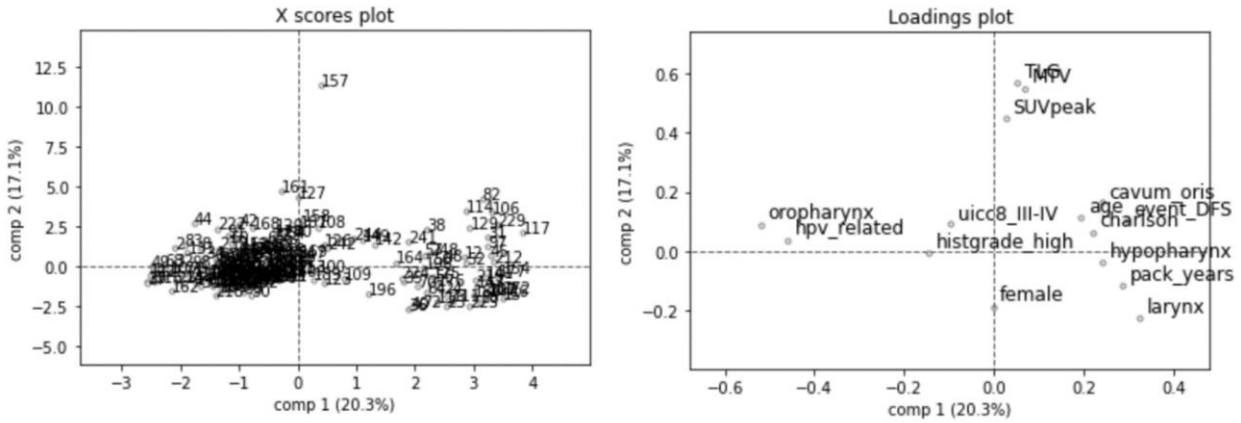


Figur 23: PCA-plott for det sentrerte datasettet om hode- og halskreft

Ved å sette plottene opp mot hverandre, kan man få bedre forståelse og se sammenhengene mellom pasientobservasjoner og variabler. Figuren presenterer at komponent 1 (PC1) utgjør en betydelig del av den totale variasjonen i datasettet, nemlig 97.7 %, mens komponent 2

(PC2) presenterer en marginal andel på 1.9 % av variansen. I figuren kan det også avdekkes avvik for spesielt to av pasientene, 142 og 157. I dette tilfellet kan det observeres at disse pasientene, er assosiert med variablene «pack_years» og «TLG» i loadingsplottet. Det antas at disse avvikene kan tilskrives ekstremverdier for disse variablene, grunnet disse ligger langt unna klyngen med de andre observasjonene.

Figur 24 nedenfor presenterer et skalert PCA-plott på samme datasett. Det kan observeres at både pasient-IDene og variablene har blitt normalfordelt og spenner over store deler av x-aksen. Basert på score plottet kan man umiddelbart legge merke til at pasient-ID 157 er den eneste pasienten som skiller seg ut fra mengden. Ved en nærmere observasjon av loadingplottet kan det identifiseres at enkelte pasienter kan skille seg ut av resten av mengden, basert på en avvikende verdi for variablene «SUVpeak», «MTV» og «TLG».



Figur 24: PCA-plott for det standardiserte hode- og halskreft datasettet

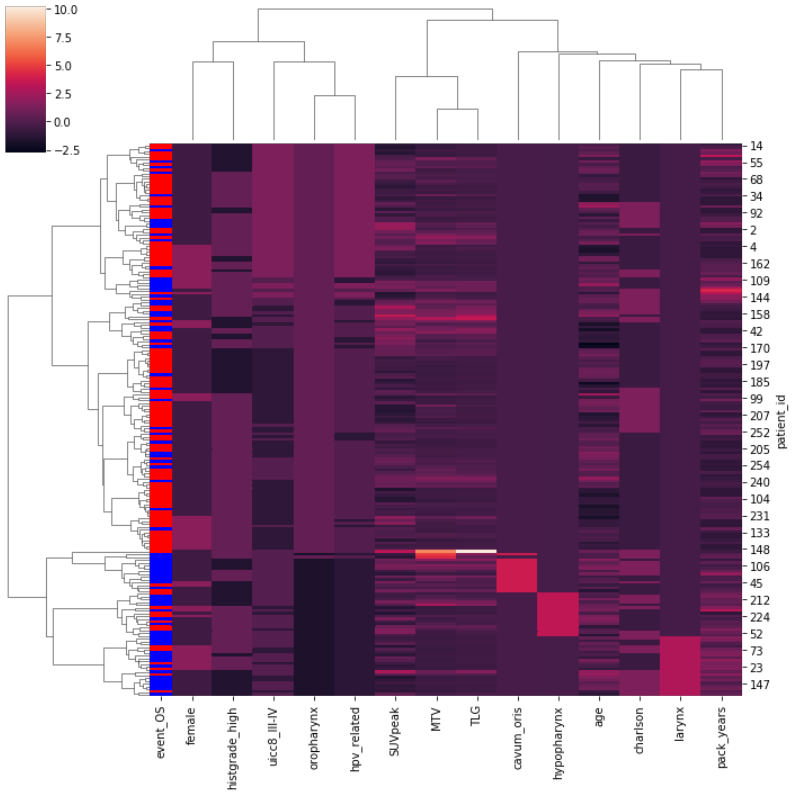
En interessant observasjon i figuren er at det danner to grupper i scoreplottet. Ved nærmere undersøkelser av korresponderende variabler i loadingplottet, kan det trekkes en sammenheng mellom de to gruppene og primærsvulsten til pasientene. Store deler av datasettet består av pasienter som har primærsvulst i oropharynx (munnsvelget), og det er tydelig at den venstre gruppen i scoreplot referer til disse. Datasettet består også av pasienter med primærsvulst i cavum oris (munnhulen), strupesvelg (hypopharynx) og larynx (strupehode). Andelen pasienter i datasettet med en av disse primærsvulstene er lavere enn andelen pasienter med primærsvulst i munnsvelg. Ved en nærmere inspeksjon av loadingplottet er det tydelig at den høyre gruppen i scoreplot er assosiert med den andre gruppen som ikke har primærsvulst i munnsvelget.

Lignende PCA-analyse har blitt utført på det eksterne datasettet fra MAASTRO klinikken i Nederland. Resultatene på disse er inkludert i vedlegg E.1.

Clustermap

På lik linje som clustermap-plottet for OxyTarget-datasettet, presenterer figur 25, et clustermap-plott for hode- og halskreftdatasettet. De røde stripene representerer pasienter med negativ OS-event, som betyr overlevde, mens de blå stripene representerer pasienter med positiv OS-event, som betyr de ikke overlevde. En tydelig observasjon langs de stripete fargekodene viser at det er dannet tre forskjellige grupper i plottet, til tross for at det kun er to kategorier for OS-event. Dette kan indikere at det er en gruppe som skiller seg fra de andre på grunn av ulike faktorer. En mulig forklaring kan være at gruppene har blitt dannet basert på svulsttyper. Som det ble presentert i tabell 22 i kapittel 3.6.1 inneholder datasettet pasienter med fire forskjellige svulsttyper, og det er mulighet for at dette har ført til dannelse av tre ulike grupper blant dem.

I denne figuren er det også tydelig at variablene i datasettet er organisert i tre forskjellige grupperinger. Dette kan indikere at variablene i hver gruppe er relatert til hverandre og kan ha en innvirkning på utfallet for en pasient. En spesiell observasjon er at variablene fra PET-parametere, inkludert MTV, TLG og SUVpeak danner en egen gruppe. Figuren bekrefter også at det er ubalanse i datasettet med ujevn klassefordeling. Videre observasjon av figur 25 viser til lyse striper ved variablene MTV, TLG og SUVpeak ved pasient 106 og 140. Dette er et resultat som kan referere til potensielle ekstremverdier i datasettet. I stedet for å bruke farge etiketter for å finne observasjoner med OS-event, viser figur E.3 i vedlegg E.2 clustermap basert på observasjonene med DFS-event. Vedlegget inneholder også clustermap over MAASTRO datasettet.

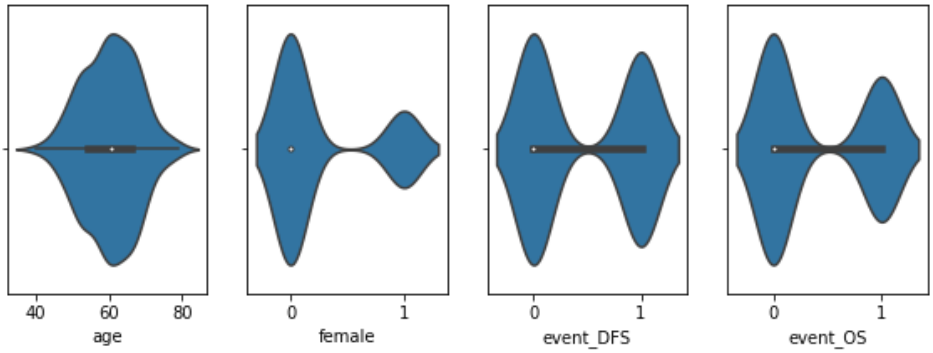


Figur 25: Clustermap av hode- og halskreft datasettet

Violinplot

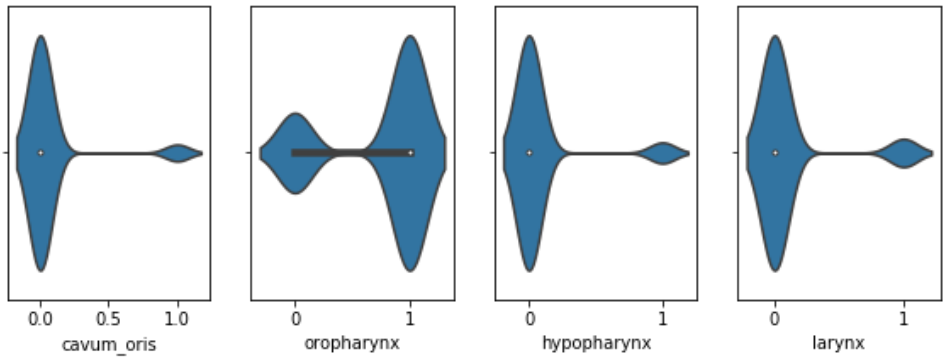
Som nevnt tidligere kan violinplot gi verdifull informasjon om datasettets egenskaper som kan være nyttige for analyse og modellering. Datasettet for hode- og halskreft består av totalt 16 variabler etter preprosesseringen. Dette inkluderer også responsvariablene OS og DFS. En visualisering av alle variablene i form av violinplot er inkludert i vedlegg E.3. Videre er det utført en tilsvarende analyse på MAASTRO datasettet, og resultatet er også inkludert i samme vedlegg.

Nedenfor presenteres tre figurer som tar for seg violinplot for noen utvalgte variabler i datasettet fra Oslo universitetssykehus. Figur 26 viser et violinplot av personkarakteristikkene til pasientene i datasettet, altså alder og kjønn, samt responsvariablene OS og DFS. Resultatene indikerer at aldersfordelingen i datasettet er normalfordelt, med de fleste pasientene i alderen mellom 50 og 70 år. Kjønnsfordelingen er derimot skjevfordelt, med mange flere mannlige (kategori 0) enn kvinnelige pasienter (kategori 1). Når det gjelder responsvariablene, kan man observere at DFS er relativt jevnfordelt og OS har en mer skjevfordelt fordeling.



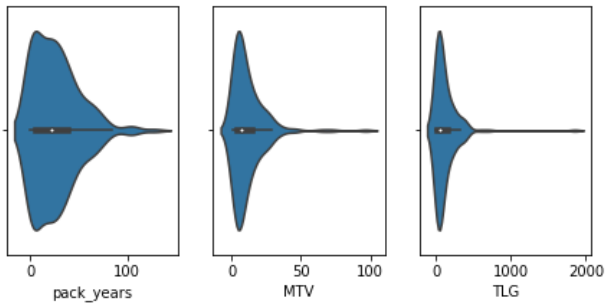
Figur 26: Violinplot over fordelingen av personkarakteristikkene til pasientene i hode- og halskreft datasettet. Dette gjelder alder, kjønn og responsvariablene generell overlevelse (OS) og sykdomsfri overlevelse (DFS).

Nedenfor presenteres figur 27 som viser et violinplot over svulsttypene i datasettet. Det fremgår tydelig at det er en overvekt av prøver med primærsvulst i oropharynx (munnsvelg) i forhold til de andre svulsttypene.



Figur 27: Violinplot over fordelingen av de ulike svulsttyper for hode- og halskreft datasettet.

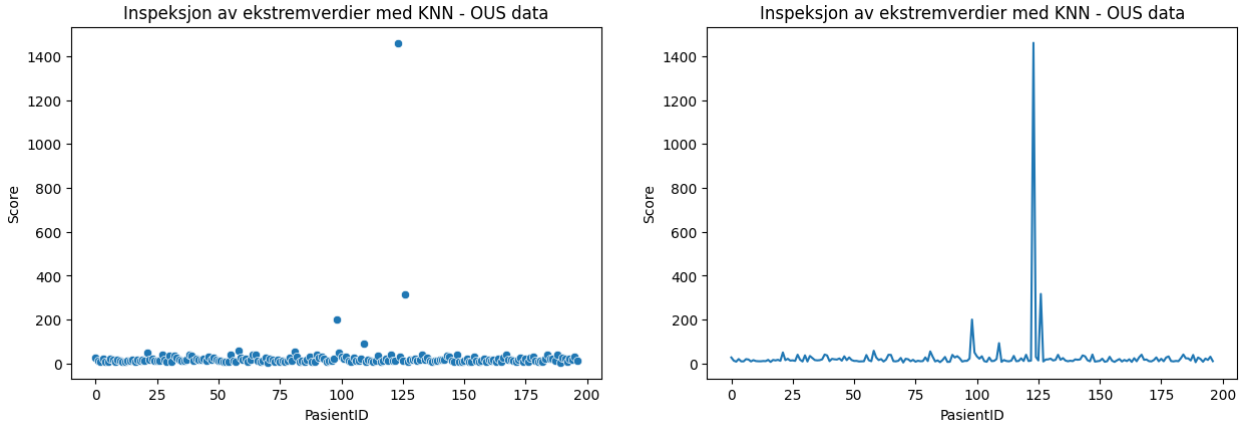
Figur 28 viser de mest markante variablene observert ved hjelp av violinplot. Samtlige av de tre variablene i figuren viser avvik fra gjennomsnittet og antyder tilstedeværelse av ekstremverdier i datasettet. Spesielt skiller variabelen TLG seg ut, der det ser ut til å være en eller flere observasjoner med verdi nærmere 2000, når denne variabelen normalt sett har verdier langt mindre enn dette.



Figur 28: Violinplot som viser avvik fra gjennomsnittet og antyder tilstedeværelse av ekstremverdier i hode- og halskreft datasettet.

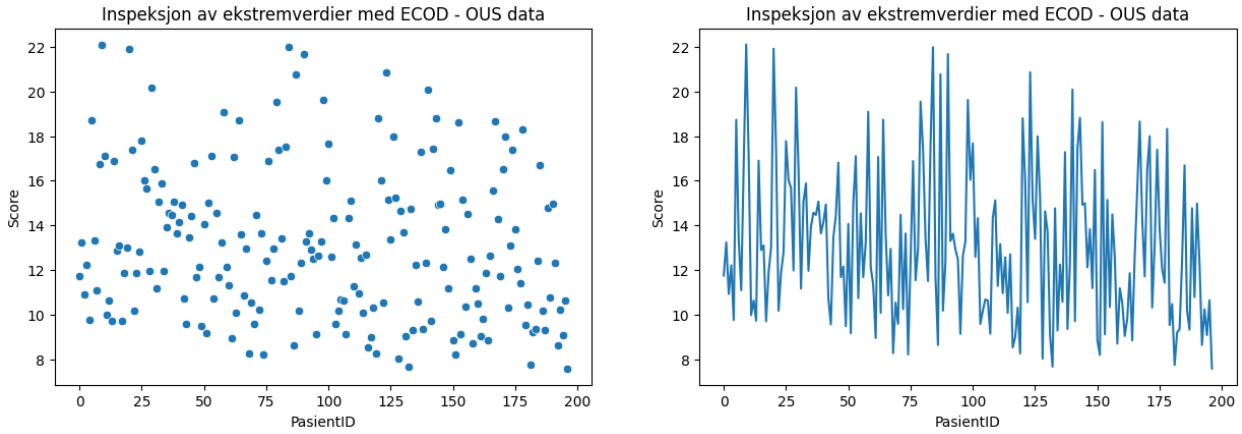
4.4.2 Inspeksjon av ekstremverdier

Figur 29 viser to grafer som presenterer resultatene av en PyOD-analyse av potensielle ekstremverdier i datasettet fra OUS for hode- og halskreft. Begge grafene viser de samme resultatene, men på to forskjellige måter. Det venstre diagrammet viser resultatene som et scatterplot, og det til høyre presenterer resultatene som et linjediagram. Analysen er basert på KNN (k-nærmeste naboer), der y-aksen representerer ekstremverdiscore av analysen, og x-aksen viser pasientene i datasettet.



Figur 29: Et scatterplot (V.S) og et linjediagram (H.S) av ekstremverdiene gjort av en PyOD-analyse med KNN for datasettet hode- og halskreft

Som nevnt tidligere i kapittel 3.4.2, ble det utført to analyser for inspeksjon av ekstremverdier med to forskjellige algoritmer. Figur 30 viser resultatene fra den andre analysen, som ble utført med ECOD-algoritmen (Empirical Cumulative Distribution Functions).



Figur 30: Et scatterplot (V.S) og et linjediagram (H.S) av ekstremverdiene gjort av en PyOD-analyse med ECOD for datasettet hode- og halskreft

Figur 29 med KNN viser tydelig at det er noen observasjoner i datasettet som skiller seg ut som ekstremverdier i forhold til resten av observasjonene. Tre observasjoner skiller seg spesielt ut fra resten av klyngen. For ECOD er variansen mindre, som øker kompleksiteten i å identifisere mulige ekstremverdier kun ved å se på figuren.

Tabell 41 gir en oppsummering av de fremste observasjonene med høyest ekstremverdiscore fra analysene. Observasjoner som er identifisert av begge algoritmene, er fremhevet med grønn farge i tabellen.

Tabell 41: En oversikt over hvilke pasienter som er identifisert som ekstremverdier for hode- og halsdatasettet. Observasjoner identifisert av begge modellene er markert med grønt.

Algoritme	**PasientID**
KNN	127 – 142 – 157 – 161
ECOD	15 – 30 – 111 – 114 – 117 – 127 – 157

Som det fremgår av tabellen over, er observasjonene med pasient-ID 127 og 167 blitt identifisert som mulige ekstremverdier av begge analysene. Tabell 42 viser ekstremverdiscorene for disse observasjonene for hver modell samt mulige årsaker til at observasjonene er identifisert som ekstremverdier. En lignende tabell er presentert i vedlegg E.4 som tar for seg de resterende observasjonene fra tabell 41.

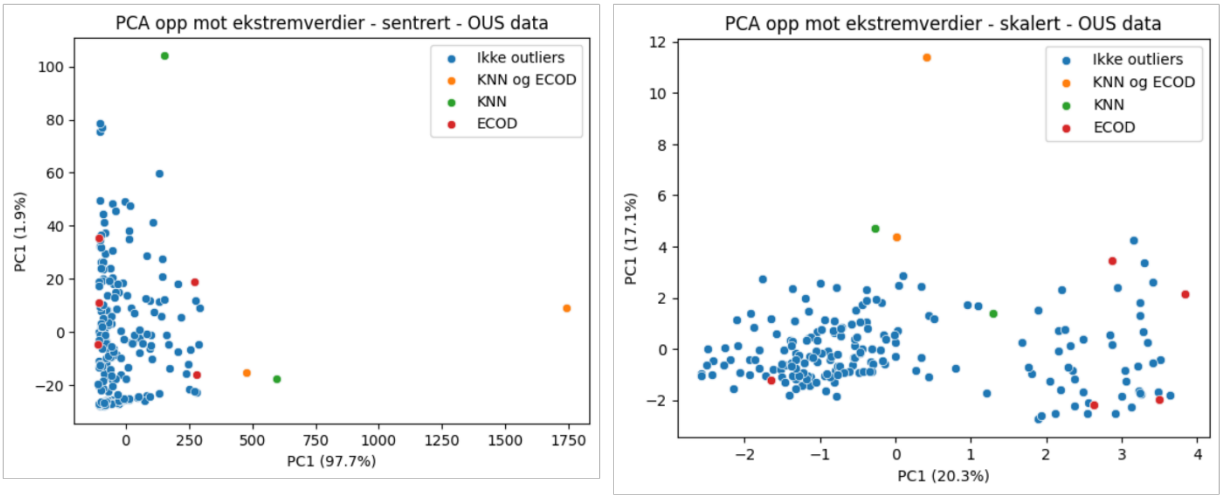
Tabell 42: Oversikt over de identifiserte ekstremverdiene med KNN- og ECOD-score (ekstremverdiscore) samt mulige årsaker til dette resultatet.

PasientID	**KNN - score**	**ECOD - score**	**Mulige årsaker**
127	200.53	20.19	Høy verdi for variabelen TLG.
157	1460.63	20.86	Variablene MTV og TLG er relativ høye i forhold til normalen. Dette ble også synliggjort i PCA-plottet, referer til kap. 4.4.1.

4.4.3 Analyse av ekstremverdier i forhold til PCA

Figur 31 illustrerer PCA-plott med fargelegging av observasjoner presentert i tabell 41 som tar for seg mulige ekstremverdier i datasettet. I figuren vises to figurer, en til venstre på sentrert data, og en til høyere på standardisert data. De to observasjonene som ble identifisert som ekstremverdier av både KNN og ECOD er fargekodet oransje i PCA-plottet. De resterende to

observasjonene anbefalt av KNN, og de resterende fem fra ECOD, er henholdsvis fargekodet grønt og rødt i plottet.



Figur 31: PCA-plott sammenlignet med ekstremverdier, både skalert og ikke-skalert (datasett fra OUS)

Av de to grafene i figuren, viser den venstre grafen tydeligere resultater på mulige ekstremverdier i datasettet. PCA-plottet indikerer klart at de observasjonene som er merket med oransje og grønn fagkode, kan være ekstremverdier, ettersom de skiller seg ut fra de andre observasjonene i datasettet. De røde observasjonene, som ble identifisert av ECOD til å være ekstremverdier, ser ut til å være godt blandet med resten av observasjonene, noe som indikerer at de nødvendigvis ikke er ekstremverdier. Det høyre PCA-plottet med standardisert data gir også indikasjoner på at de observasjonene merket i oransje er ekstremverdier.

Tabell 43 presenterer en oversikt over hvilke observasjoner (PasientID) som representerer de ulike fargekodene i PCA-plottet. I tillegg gir tabellen en oversikt over observasjonene som er felles mellom analysene utført med ECOD og KNN, samt observasjonene som kan betraktes som potensielle ekstremverdier ved å kun se på PCA-plottet og PCA-analysen for datasettet uten standardisering.

Tabell 43: Sammenhengen mellom KNN, ECOD og PCA med tanke på ekstremverdier, samt hvilke fargekoder de ulike ekstremverdiobservasjonene har fått i PCA-plottet.

KNN	ECOD	PCA
127	127	127
157	157	157
142	15	128
161	30	142
-	111	161
-	114	181
-	117	213

I tabellen og i figur 31 er det åpenbart at de to observasjonene som både KNN og ECOD identifiserte som ekstremverdier, også er tydelige i PCA-plottet. Videre kan de resterende to observasjonene som ble identifisert av KNN også bli betraktet som potensielle ekstremverdier i forhold til PCA-plottet. Basert på disse resultatene ble alle observasjoner med oransje og grønn fargekode eliminert fra datasettet før modellering. De røde observasjonene forble i datasettet, ettersom ekstremverdiscoren var nær grensen til å havne i gruppe med resten av observasjonene, og PCA-plottet indikerte at de var godt blandet med de øvrige observasjonene.

4.5 Hode- og halskreft med OS som responsvariabel

Seksjonen presenterer resultatene fra maskinlæringsmodellene som ble trent på datasettet bestående av pasienter med hode- og halskreft, der OS ble brukt som responsvariabel. Som beskrevet i metodekapittelet, referer OS-event til total overlevelse, hvor pasienter som forble i live ved slutten av en femårs oppfølgingsperiode ble tildelt klasse 0. Dersom pasienten døde i oppfølgingsperioden, ble klasse 1 tildelt.

4.5.1 Evaluering av modeller på testdata

På lik linje som modellene for kolorektal kreft, ble modellene for hode- og halskreft trent med hyperparameteroptimalisering. Hver modell med den beste kombinasjonen fra optimaliseringene ble trent med 4-foldet kryssvalidering med hele tusen repetisjoner. Tabell 44 presenterer de gjennomsnittlige testresultatene for modellene.

Tabell 44: De gjennomsnittlige testresultatene på test-/valideringsdata for hode- og halskreft med responsen OS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.76 ± 0.05	0.65 ± 0.08	0.81 ± 0.04	0.48 ± 0.11	0.73 ± 0.06
Logistisk regresjon	0.75 ± 0.06	0.66 ± 0.08	0.80 ± 0.05	0.48 ± 0.12	0.73 ± 0.06
QDA	0.74 ± 0.05	0.65 ± 0.08	0.80 ± 0.04	0.45 ± 0.12	0.72 ± 0.06
GaussianNB	0.72 ± 0.05	0.59 ± 0.11	0.78 ± 0.04	0.39 ± 0.13	0.68 ± 0.06
Nearest centroid	0.75 ± 0.05	0.67 ± 0.08	0.80 ± 0.04	0.47 ± 0.11	0.73 ± 0.06
KNORA-E	0.73 ± 0.05	0.61 ± 0.08	0.80 ± 0.04	0.42 ± 0.11	0.70 ± 0.06
KNORA-U	0.74 ± 0.06	0.64 ± 0.08	0.79 ± 0.05	0.44 ± 0.12	0.71 ± 0.11
DES-P	0.73 ± 0.05	0.63 ± 0.07	0.79 ± 0.04	0.43 ± 0.11	0.71 ± 0.05

I tabellen fremgår det tydelig at random forest, logistisk regresjon og QDA viser en jevn ytelse for alle fem performance metrics. GaussianNB, som tidligere viste god ytelse for kolorektal

kreft, viser seg å ha den laveste ytelsen av alle de åtte modellene på datasettet for hode- og halskreft. Selv om LazyPredict tidligere påpekte at nearest centroid som den best egnede algoritmen for hode- og halskreft datasettet, viser ytelsen ikke noe nevneverdig forbedring sammenlignet med de andre klassiske ML-modellene, bortsett fra GaussianNB. F1-scoren for positive klasser indikerer at nearest centroid gjør det noe bedre enn de andre modellene, men scoren skiller seg ikke betydelig fra de andre til å trekke en konklusjon om at nearest centroid er klart bedre.

Det er av betydning å bemerke at LazyPredict foreslår algoritmer ved å trene modeller med normale *(eng. default)* hyperparameter og presenterer den modellen som viser best ytelse. Dette kan bety at nearest centroid presterer bedre når modellene kun bruker standard hyperparametere, men andre algoritmer kan vise seg å gi bedre resultater når det utføres hyperparameteroptimalisering. Dette kan være en av grunnene som bidrar til at de klassiske ML-algoritmene random forest, logistisk regresjon og QDA viser jevn ytelse sammen med nearest centroid.

De algoritmene som er merket med blå i tabellen representerer algoritmene fra DESlib-pakken. Testresultatene for KNORA-E, KNORA-U og DES-P viser relativt like poengscore, og presterer også jevnt med de øvrige algoritmene, med unntak av GaussianNB. I sammenligning med de resterende modellene scorer DES-algoritmene dårligere for MCC. Dette skyldes av at sann positiv (*eng. true-positive*) som tar for seg antallet korrekt klassifiserte for klasse 1 er noe lavere for disse modellene. Blant DES-algoritmene viser KNORA-U den høyeste poengscoren og presterer bedre på F1-positiv score og MCC.

Totalt sett har modellene problemer med å klassifisere den positive klassen. Dette indikerer at modellene er mer sikre i å predikere klasse 0 enn klasse 1. På samme måte som for modellene med kolorektal kreft med OS som respons, er datasettet med hode- og halskreft også påvirket av en skjevfordelt klassefordeling. Dermed vil muligheten være stor for at modellene vil klare å tilpasse den overpresenterende klassen og dermed presentere bedre enn den underpresenterende klassen.

I henhold til den utførte analysen, var den totale modelleringstiden *(eng. run time)* for modellene med random forest, logistisk regresjon, QDA, GaussianNB og nearest centroid på 1 time, 58 minutter, og 56 sekunder. Tiden inkluderer også modelleringstiden for SVC og KNN

som er inkludert i vedlegg F.1. På den annen side var modelleringstiden for DES-algoritmene på 11 timer, 49 minutter, og 21 sekunder. Tiden for DES-algoritmene inkluderer også modelleringstiden for META-DES, MCB og OLA. Resultatene på de tre sistnevnte algoritmene er utelatt fra resultatkapittelet, men inkludert i vedlegg F.1.

Figur 32 presenterer confusion matrix for alle de åtte algoritmene på testdata. Som for kolorektal kreft med responsvariabelen OS, er det enkelt å se at den positive klassen har vanskeligheter med å bli korrekt klassifisert, og at datasettet er dominerende av den negative klassen (61.42%) i forhold til den positive (38.58%). Av modellene klarer random forest og KNORA-E å unngå FP kategorien i større grad enn de andre, men ikke i like stor grad som de presenterte modellene med kolorektal datasettet.

Figur 32: Confusion matrix for alle de åtte algoritmene på testdata, hode- og halskreft med OS-event

4.5.2 Evaluering av modeller på treningsdata

Tabell 45 presenterer de gjennomsnittlige resultatene fra treningsdatasettet for modellene med OS event som responsvariabel.

Tabell 45: De gjennomsnittlige resultatene på treningsdata for hode- og halskreft med responsen OS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.88 ± 0.02	0.83 ± 0.03	0.91 ± 0.01	0.74 ± 0.03	0.86 ± 0.02
Logistisk regresjon	0.81 ± 0.02	0.73 ± 0.03	0.85 ± 0.01	0.59 ± 0.04	0.79 ± 0.02
QDA	0.76 ± 0.02	0.68 ± 0.03	0.81 ± 0.02	0.49 ± 0.04	0.74 ± 0.02
GaussianNB	0.73 ± 0.02	0.61 ± 0.06	0.80 ± 0.01	0.42 ± 0.04	0.70 ± 0.03
Nearest centroid	0.76 ± 0.02	0.68 ± 0.03	0.81 ± 0.02	0.49 ± 0.04	0.75 ± 0.02
KNORA-E	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
KNORA-U	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
DES-P	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

DES-algoritmene presterer med prediksjonsnøyaktighet på 1 som betyr at modellene klassifiserer feilfri på treningsdatasettet. Modellene med DES-algoritmer er ensemble modeller og har tendens til å trene flere beslutningstrær og hvis observasjonene er riktig klassifisert for minst ett tre, vil det være mulig å få en prediksjonsnøyaktighet på 1. Dette betyr nødvendigvis ikke at modellene vil klare å predikere feilfri med testdata som også ble bekreftet tidligere i kapittelet med testresultatene til modellene. Random forest som også kategoriseres ensemble modeller gir noe lavere nøyaktighet på alle nøyaktighetsscorene enn DES-algoritmene, men likevel presterer den jevnt med DES-modellene.

Utenom DES-modellene presterer alle de resterende modellen svakt på F1-positiv og MCC-score. Det er verdt å merke seg at modellene har vanskeligheter med å klassifisere den positive klassen, noe som er likt med resultatene fra test-datasettet. GaussianNB og nearest centroid, som ble foreslått av LazyPredict, og QDA gir de laveste resultatene på treningsdata. Selv om det virker som disse modellene presterer dårlig på treningsdata, betyr ikke det nødvendigvis at de er ineffektive modeller. Modellene kan faktisk ha klart å generalisere problemet bedre enn de andre modellene, inkludert DES-modellene som har en presisjonsnøyaktighet på 1. Det er modellene med de fem klassiske som har minst forskjell mellom trening og testresultatet. Dette kan bety at muligheten for overtilpasning finner sted i mye mindre grad for modellene enn for modellene med DES-algoritmer.

Basert på testresultatene fra tabell 44 og treningsresultatene ser det ut til at random forest og QDA kommer best ut av de presenterte modellene. Begge modellene skiller seg ut spesielt med respektable målinger for både MCC og F1-positv som er to målinger som har vist seg å være utfordrende å prestere godt på det aktuelle datasettet. Modellene har som også nevnt, mindre forskjell mellom trening og testresultatene som også forteller hvor godt modellen har tilpasset seg uten mulighet for overtilpasning. .

4.5.3 Feil klassifiserte pasienter i testdata

Nedenfor følger tabell 46 som presenterer en matrise med vanskelige pasienter fra valideringsdata. Matrisen er utformet ved å identifisere de 30 mest utfordrende observasjonene for hver modell, og deretter finne observasjonene som er felles for alle modellene. Det var totalt 13 observasjoner som var felles, og disse presenteres i tabell 46 i stigende rekkefølge basert på gjennomsnittet. Vedlegg F.2 inneholder oversikten som har blitt brukt til å forme matrisen i figuren med de 30 mest utfordrende prøver som har vært vanskelig å bli klassifisert i riktig kategori fra hver modell.

Tabell 46: En matrise over antall tilfeller av feilaktig klassifisering av 1000 mulige i testdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med hode- og hals datasettet med responsen OS.

ID	Klasse	KNORAE	KNORAU	DESP	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
26	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
74	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
105	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
124	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
166	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
171	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
189	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
195	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
209	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
243	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
13	1	1000	1000	990	998	999	1000	1000	1000	998.38
216	1	1000	990	990	999	1000	1000	1000	1000	997.38
241	0	940	1000	1000	1000	1000	1000	1000	1000	992.5
Gjennomsnitt		995.4	999.2	998.5	999.8	999.9	1000	1000	1000	999.09

De fleste observasjonene i matrisen, med unntak av observasjon 241, har primærsvulsten i munnsvelg (oropharynx). Pasientobservasjon 241, derimot, har primærsvulst i munnhulen (cavum oris) og er den eneste observasjonen som tilhører klasse 0 i matrisen. Pasienten er altså den eneste pasienten i matrisen som er i live, og en av kun to pasienter i hele datasettet som tilhører klasse 0 med primærsvulst i munnhulen.

Et nærmere dypdykk i datasettet tyder på at variabelen «TLG - Total Lesion Glycolysis» har enten høyere eller lavere verdi enn gjennomsnittet på 95.97 for alle pasientprøver i matrisen utenom observasjonsprøvene 195 og 209. Dette kan indikere at modellene har problemer med å generalisere og finne riktig klasse når en observasjon har en «TLG»-verdi som er veldig avvikende fra gjennomsnittet.

Pasientprøvene 195 og 209 har en verdi på 0 for variabelen «pack_years». Dette betyr at pasientene ikke har røyket innen en gitt periode på antall år med 20 sigaretter per dag. «TLG»-verdien for disse observasjonene er nær liggende gjennomsnittsverdien for «TLG»-variabelen. En nærmere undersøkelse av datasettet tyder på det er en sammenheng mellom pasienter med en «pack_years» på 0 og variabelen «TLG». Faktisk viser det seg at pasienter med en «pack_years»-verdi på 0 har enten høy eller lav «TLG»-verdi i datasettet. Dette kan være en av grunnene til at observasjonen 195 og 209 har vanskeligheter med å bli klassifisert av samtlige modeller.

4.5.4 Feil klassifiserte pasienter i treningsdata

På samme måte som matrisen med feil klassifiserte pasienter fra testdata, presenterer tabell 47 en matrise som viser en oversikt over pasienter fra treningsdata som har vært utfordrende å bli riktig klassifisert. KNORA-E, KNORA-U og DES-P hadde ingen vanskeligheter med å klassifisere pasientene i riktig klasse. Siden disse algoritmene ikke hadde noen vanskelige pasienter i treningsdata, ble matrisen formet ut fra de gjenværende algoritmene. Matrisen er strukturert på lik linje som de tidligere i oppgaven. Vedlegg F.3 inneholder en tabell som viser en oversikt over hvilke pasientobservasjoner som var vanskelig for hver modell i treningssettet.

Tabell 47: En matrise over antall tilfeller av feilaktig klassifisering av 3000 mulige i treningsdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med hode- og hals datasettet med responsen OS.

ID	Klasse	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
253	0	1070	3000	2929	70	3000	2013.8
13	1	597	696	698	698	698	677.4
243	1	133	151	151	151	151	147.4
241	0	4	12	12	12	12	10.4
Gjennomsnitt		451	964.75	947.50	232.75	965.25	712.25

Av de fire presenterte prøvene, er det kun observasjon 253 som er mest utforende å klassifisere. Modellene med logistisk regresjon og nearest centroid har størst problem med observasjonen. En nærmere undersøkelse av datasettet avdekket at pasienten har en høy verdi for variabelen «pack_years». Pasienter i datasettet som hadde tilsvarende egenskaper som pasient 253, hadde et positivt utfall for OS. Pasientobservasjon 253 hadde derimot et negativt utfall for OS og kan dermed være en av grunnene til at observasjonen har blitt feilklassifisert en rekke ganger med logistisk regresjon, random forest og nearest centroid. Modellen med GaussianNB presterer derimot bedre for observasjonen. Dette kan bety at GaussianNB har klart å tilpasse denne sammenhengen.

Pasientobservasjon 241 ble identifisert som en vanskelig pasient for valideringsdataene, og i henhold til datasettet er denne observasjon preget av en primærsvulst i munnhulen (cavum oris). Pasienten er også den eneste pasienten i matrisen som fortsatt lever, og er en av de to mulige tilfellene i hele datasettet som har primærsvulst i munnhulen og tilhører klasse 0. Dette kan delvis forklare hvorfor modellene hadde problemer med å klassifisere riktig et par ganger av 3000 mulige tilfeller.

En nærmere undersøkelse av datasettet avdekket ikke like klare indikasjoner som i de forgående undersøkelsene på hvorfor observasjonene 13 og 243 ble klassifisert feil. Eneste avvik fra variabelens gjennomsnittsverdi var «TLG»- og «MTV»-verdien. «MTV» har et gjennomsnitt på 10.53. Observasjon 13 hadde betydelig høyere verdi enn gjennomsnittet for variablene, og observasjon 243 hadde en mindre verdi for variablene enn gjennomsnittsverdien. Dette kan være en av grunnene til at modellene har hatt problemer med å klassifisere observasjonene i riktig klasse, og hvorfor observasjon 13 er mer utfordrende enn observasjon 243.

4.5.5 Evaluering av modeller på MAASTRO data

Nedenfor presenteres tabell 48 som viser resultatene fra eksterne datasett fra MAASTRO-klinikken i Nederland for hver modell som er trent med datasettet fra OUS.

Tabell 48: De gjennomsnittlige resultatene på MAASTRO datasettet for hode- og hals med responsen OS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.65 ± 0.02	0.72 ± 0.02	0.51 ± 0.03	0.30 ± 0.04	0.63 ± 0.02
Logistisk regresjon	0.61 ± 0.02	0.70 ± 0.02	0.45 ± 0.04	0.22 ± 0.06	0.59 ± 0.02
QDA	0.63 ± 0.01	0.72 ± 0.01	0.47 ± 0.03	0.27 ± 0.03	0.61 ± 0.01
GaussianNB	0.65 ± 0.03	0.71 ± 0.07	0.55 ± 0.03	0.31 ± 0.05	0.64 ± 0.02
Nearest centroid	0.63 ± 0.01	0.72 ± 0.01	0.43 ± 0.01	0.27 ± 0.03	0.60 ± 0.01
KNORA-E	0.61 ± 0.03	0.67 ± 0.04	0.51 ± 0.04	0.21 ± 0.07	0.60 ± 0.03
KNORA-U	0.64 ± 0.02	0.72 ± 0.02	0.50 ± 0.04	0.28 ± 0.05	0.62 ± 0.02
DES-P	0.64 ± 0.02	0.72 ± 0.02	0.50 ± 0.04	0.28 ± 0.05	0.62 ± 0.02

Samtlige modeller presterer bedre for F1-positiv enn F1 -negativ, altså den positive klassen er enklere for modellene. Sammenlignet med resultatene fra valideringsdata i tabell 44, er dette totalt motsatt, der valideringsdata hadde større problemer med F1-positiv enn F1-negativ. Dette er et resultat som kan skyldes av ulik klassefordeling mellom MAASTRO datasettet og OUS-datasettet. Som presentert i kapittel 3.6.1 består MAASTRO datasettet av flere positive klasser enn negative mens OUS datasettet består av flere prøver med den negative klassen enn den positive klassen.

Alle de åtte modellene i tabellen presterer veldig jevnt. Av disse er det KNORA-E, logistisk regresjon, QDA og nearest centroid som skiller seg ut som dårligst. Både KNORA-E og logistisk regression har prestert dårligst i henhold til MCC-verdi. QDA, logistisk regression og nearest centroid har størst problemer med den negative klassen i henhold til hvordan disse har presentert for F1-negativ med en nøyaktighetsytelse under 0.5. Av disse fire er det logistisk regresjon som kommer dårligst ut med lav ytelsesverdi for F1-negativ og MCC.

For alle de åtte klassifiseringsalgoritmene, er confusion matrix for MAASTRO-datasettet presentert i vedlegg F.5. I tillegg gir vedlegg F.6 en detaljert oversikt over prestasjonen til modellene som er trent med algoritmene SVC, KNN, META-DES, OLA og MCB

4.6 Hode- og halskreft med DFS som responsvariabel

Kapittelet presenterer resultatene for modellene trent med hode- og halskreft datasettet med DFS som responsvariabel. DFS referer til sykdomsfri overlevelse (*eng. Disease-Free Survival)*. En pasient som forble i live og verken opplevde regional, lokalt eller metastatisk tilbakefall fikk tildelt kategori 0. Dersom en av hendelsene hadde inntruffet innen oppfølgingsperioden på fem år, hadde pasienten fått tildelt kategori 1.

4.6.1 Evaluering av modeller på testdata

Tabell 49 presenterer det gjennomsnittlige testresultatet på modellene som er trent med 4-foldet kryssvalidering med tusen repetisjoner. Modellene ble på lik linje som modellene som ble presentert tidligere hyperparameteroptimalisert.

Tabell 49: De gjennomsnittlige testresultatene på test-/valideringsdata for hode- og hals med responsen DFS.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.70 ± 0.06	0.63 ± 0.08	0.75 ± 0.05	0.40 ± 0.12	0.69 ± 0.06
Logistisk regresjon	0.68 ± 0.06	0.62 ± 0.08	0.72 ± 0.05	0.34 ± 0.12	0.67 ± 0.06
QDA	0.69 ± 0.06	0.63 ± 0.08	0.74 ± 0.05	0.38 ± 0.12	0.69 ± 0.06
GaussianNB	0.68 ± 0.06	0.58 ± 0.11	0.74 ± 0.05	0.35 ± 0.13	0.66 ± 0.07
Nearest centroid	0.69 ± 0.06	0.64 ± 0.08	0.73 ± 0.05	0.38 ± 0.12	0.69 ± 0.06
KNORA-E	0.66 ± 0.05	0.58 ± 0.08	0.71 ± 0.05	0.31 ± 0.12	0.65 ± 0.06
KNORA-U	0.66 ± 0.06	0.60 ± 0.07	0.69 ± 0.06	0.31 ± 0.11	0.65 ± 0.06
DES-P	0.66 ± 0.06	0.60 ± 0.07	0.70 ± 0.05	0.31 ± 0.11	0.65 ± 0.06

Totalt sett presterer modellen med random forest best av de åtte modellene i tabellen. MCC – scoren på 0.4 er spesielt merkbar, ved at den skiller seg ut positivt fra de andre modellene i tabellen. Nearest centroid og QDA presterer også godt, og er de som gir tøffest motstand til resultatene til random forest. Nearest centroid er algoritmen som ble foreslått av LazyPredict for hode- og halskreft og det er tydelig at modellen presterer respektabelt. Som nevnt tidligere foreslår LazyPredict algoritmer ved å lage modeller med normale *(eng. default)* hyperparameter og presenterer modellen som kommer best ut. Dette kan bety at nearest centroid kommer best ut når modellene kun tar i bruk de normale hyperparameterne, men at andre kan presentere bedre når det utføres hyperparameteroptimalisering. Dette kan være en grunn til at random forest gjør det bedre og QDA presterer tett opp med nearest centroid.

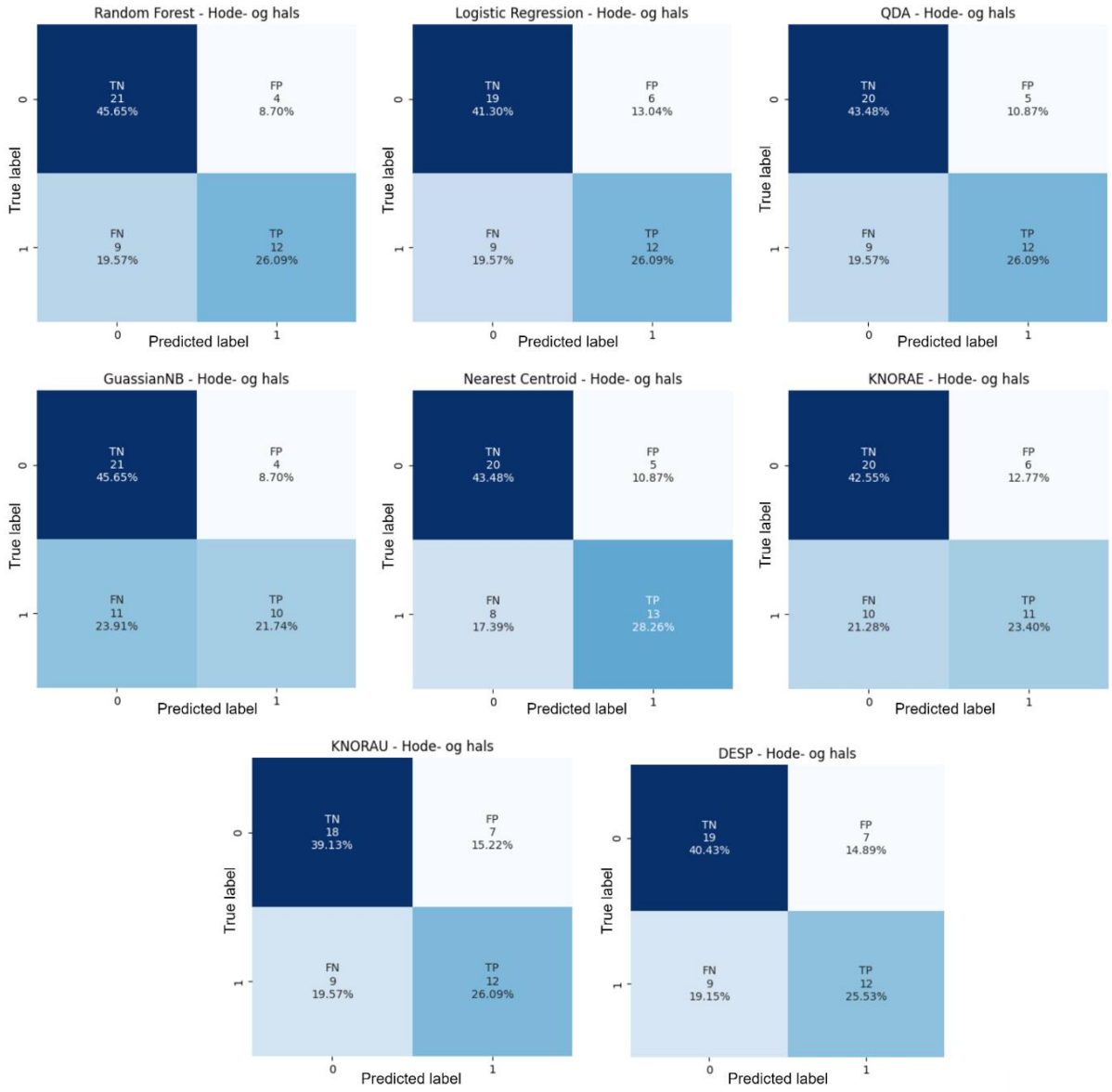
Av alle modellene i tabellen presterer algoritmene fra DESlib-pakken dårligst. Testresultatene til KNORA-E, KNORA-U og DES-P gir relativt like poengscore for alle nøyaktighetsmålingene og

presterer dårligst sammenlignet med hvordan de klassiske har prestert. MCC-nøyaktigheten for modeller med DES-algoritmer avviker med hele 9 prosentpoeng fra den beste algoritmen.

På lik linje som modellene med OS, sliter også modellene for DFS med å klassifisere den positive klassen korrekt. Dette kan gjenspeiles i modellenes nøyaktighet for F1-positiv og F1-negativ. Dette fremgår også tydelig i figur 33, som viser confusion matrix over alle de åtte algoritmene på testdataen.

Den totale modelleringstiden *(eng. run time)* for de klassiske ML-algoritmene var på 1 time, 24 minutter, og 54 sekunder. Tiden inkluderer også modelleringstiden for SVC og KNN som er inkludert i vedlegg G.1. For DES-algoritmene var modelleringstiden derimot på 9 timer, 44 minutter, og 56 sekunder. Dette inkluderte modelleringstiden for META-DES, MCB og OLA. Resultatene på disse er også inkludert i vedlegg G.1.

Figur 33 presenterer confusion matrix over alle de åtte algoritmene på test data. Modellene med random forest og GuassianNB ser ut til å være best til å håndtere FP kategorien best. Tidligere presenterte modeller viste at modeller med DES-algoritmer hadde gode resultater når det gjaldt FP, men det samme gjald ikke for modellene med DFS som responsvariabel på hode- og halsdatasettet.



Figur 33: Confusion matrix over alle de åtte algoritmene på test data, hode- og halskreft med DFS-event.

4.6.2 Evaluering av modeller på treningsdata

Tabell 50 presenterer resultatene over modellytelsene på treningssettet.

Tabell 50: De gjennomsnittlige resultatene på treningsdata for hode- og hals med responsen DFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.78 ± 0.02	0.72 ± 0.03	0.81 ± 0.01	0.55 ± 0.04	0.77 ± 0.02
Logistisk regresjon	0.72 ± 0.02	0.67 ± 0.03	0.76 ± 0.02	0.44 ± 0.05	0.71 ± 0.02
QDA	0.71 ± 0.02	0.65 ± 0.03	0.75 ± 0.02	0.41 ± 0.04	0.70 ± 0.02
GaussianNB	0.69 ± 0.03	0.59 ± 0.08	0.75 ± 0.02	0.38 ± 0.05	0.67 ± 0.03
Nearest centroid	0.71 ± 0.02	0.66 ± 0.02	0.75 ± 0.02	0.41 ± 0.04	0.70 ± 0.02
KNORA-E	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
KNORA-U	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
DES-P	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Som det var tilfellet for modellene med kolorektal datasettet og hode- og halsdatasettet med OS som responsvariabel, oppnår DES-algoritmene en prediksjonsnøyaktighet på 1. Som tidligere påpekt, er DES-algoritmer ensemble modeller som trener flere beslutningstrær. Hvis observasjonene er riktig klassifisert for minst ett tre, kan det resultere i en prediksjonsnøyaktighet på 1. Det bør imidlertid bemerkes at dette ikke nødvendigvis betyr at modellene vil klare å predikere testdata uten feil.

Random forest som også er en ensemble modell gir noe lavere nøyaktighet på alle nøyaktighetsmålingene enn DES-algoritmene, og er den som gir tøffest motstand. Algoritmen presterer respektabel for alle nøyaktighetsmålingene i forhold til modellene med de andre klassiske ML-algoritmene. MCC, F1-negativ og ROC-AUC for random forest utmerker seg med spesielt høy nøyaktighet.

GaussianNB, nearest centroid og QDA kommer dårligst ut på ut på treningsdata. Modellene har svake resultater på nøyaktighetstypene MCC og F1-positiv. Som nevnt for treningsresultatene for de andre modellene i oppgaven , betyr nødvendigvis ikke det presenterte resultatet at modellen er svak eller dårlig. Ved å sammenligne testresultatene opp mot treningsresultatet, har de klassiske modellene minst forskjell. Dette kan bety at muligheten for overtilpasning finner sted i mye mindre grad og at modellene har klart å generalisere problemet bedre enn DES-modeller.

4.6.3 Feil klassifiserte pasienter i testdata

Nedenfor følger tabell 51 som presenterer en matrise med feil klassifiserte pasienter fra valideringsdata. Matrisen er utformet i samsvar med tidligere presenterte matriser ved å vise de 30 utfordrende observasjoner fra hver modell og deretter identifisere hvilke observasjoner som er felles mellom modellene. Det var totalt 10 observasjoner som var til felles og det er disse som presenteres i tabell 51 etter stigende rekkefølge basert på gjennomsnittet. Vedlegg G.2 inneholder listen over de 30 mest utfordrende pasientene som matrisen er utledet fra.

Tabell 51: En matrise over antall tilfeller av feilaktig klassifisering av 1000 mulige i testdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med hode- og hals datasettet med responsen DFS.

ID	Klasse	KNORAE	KNORAU	DESP	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
43	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
92	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
189	1	1000	1000	1000	1000	1000	1000	1000	1000	1000
241	0	1000	1000	1000	1000	1000	1000	1000	1000	1000
105	1	1000	1000	1000	998	1000	1000	1000	1000	999.75
26	1	1000	996	1000	1000	999	1000	1000	1000	999.38
209	1	1000	998	998	1000	995	1000	1000	1000	998.88
195	1	996	988	998	1000	999	1000	1000	1000	997.63
139	1	996	988	994	1000	1000	1000	1000	1000	997.25
133	1	1000	984	992	1000	1000	1000	1000	1000	997
Gjennomsnitt		999.2	995.4	998.2	999.8	999.3	1000	1000	1000	999.98

Observasjonene merket med rødt er også klassifisert som vanskelige av modellene som anvendte OS som responsvariabel. Mulige årsaker for hvorfor disse kan være vanskelige er dekket i kapittel 4.5.3.

En grundig undersøkelse av pasientobservasjonene 43, 92, 139 og 133 avdekket at variabelen «TLG - Total Lesion Glycolysis» hadde verdier som avvek fra gjennomsnittet for «TLG»-variablene. Dette var også en av grunnene som ble diskutert for enkelte av observasjonene presentert i kapittel 4.5.3. Det kan tenkes at modellene hadde problemer med å generalisere og klassifisere riktig når en observasjon hadde en «TLG»-verdi som avvek sterkt fra gjennomsnittet. Pasientene 43, 92, 139 og 133 ble ikke inkludert i matrisen for OS i kapittel 4.5.3. Årsaken til at disse observasjonene ikke ble inkludert i matrisen over vanskelige pasienter, kan skyldes av at disse pasientene befant seg lenger ned på listen over vanskelige pasienter. Dermed kunne observasjonene ha falt utenfor topp 30 med OS, og dermed ikke blitt inkludert ved utforming av matrisen.

4.6.4 Feil klassifiserte pasienter i treningsdata

Tabell 52 presenterer en matrise for pasienter i treningsdata som har vært utfordrende å bli klassifisert i riktig klasse. DES-algoritmene hadde ingen vanskelige pasienter og er derfor ikke inkludert i matrisen. Vedlegg G.3 inneholder en tabell som viser en oversikt over hvilke pasientobservasjoner som var vanskelig for hver modell i treningssettet.

Tabell 52: En matrise over antall tilfeller av feilaktig klassifisering av 3000 mulige i treningsdata. Oversikten tar for seg pasientprøver som har vært utforende for modellene trent med hode- og hals datasettet med responsen DFS.

ID	Klasse	Random forest	Logistisk regresjon	QDA	GaussianNB	Nearest centroid	Gjennomsnitt
253	0	1335	2994	37	2951	2995	2062.4
11	0	26	814	2184	2308	2313	1529
247	0	53	1102	9	1428	1431	804.6
13	1	649	659	748	730	697	696.6
243	1	135	142	142	142	142	140.6
8	0	18	113	237	21	79	93.6
241	0	21	21	21	21	21	21
Gjennomsnitt		319.57	835	482.57	1085.86	1096.86	763.97

Matrisen presenterer totalt syv pasienter som har vanskeligheter med å bli klassifisert i riktig kategori. Alle observasjoner som er markert med gult, ble også identifisert som vanskelige pasienter i kapittel 4.5.4. Pasient 241 ble også identifisert som en utfordrende pasient for valideringsdata i kapittel 4.6.3. I kapittel 4.5.4 er det allerede blitt dekket mulige årsaker til hvorfor disse observasjonene kan være feilaktig klassifisert.

Paisentobservasjon 11 er den eneste pasienten i matrisen som har svulst i strupehodet (larynx). I datasettet er det totalt 21 pasienter med primærsvulst i strupehode. Ved å trekke ut pasienter med tilsvarende egenskaper som observasjon 11, var det en tydelig sammenheng i datasettet. Pasienter som hadde tilsvarende egenskaper som observasjon 11 hadde et positivt utfall for DFS, og observasjon 11 derimot hadde et negativt utfall for DFS. Dette kan forklare hvorfor modellen ikke klarte å korrekt klassifisere observasjon 11. Av de modellene som ble presentert i matrisen, synes det som om modellen med random forest har klart å finne et skille mellom observasjon 11 og tilsvarende observasjoner, og dermed klart å håndtere dette problemet bedre enn de andre modellene presentert i matrisen.

Observasjonene 247 har en høy «TLG»-verdi og «MTV»-verdi som er avvikende fra gjennomsnittsverdien for variablene. Pasienten er en kvinne og en av de eldste i datasettet. Dette kan være grunnene til at pasient 247 har hatt problemer med enkelte av modellene. Det er tydelig at modellene med random forest og QDA har hatt mindre problemer med pasienten. Denne analysen indikerer at modellen har oppnådd suksess i å identifisere riktige sammenhenger for å skille pasienter med tilsvarende egenskaper som pasient 247. Pasient 8 har også en høy «TLG»- og «MTV»-verdi som er veldig avvikende fra gjennomsnittsverdien for variablene. Til forskjell fra pasient 247 er pasient 8 en yngre mann. Dette kan være en av

faktorene som har bidratt til at flere klassifiseringsalgoritmer har mindre problemer med å klassifisere denne pasienten. Imidlertid har modellen med QDA større problemer med pasient 8 enn pasient 247. Dette kan bety at QDA har klart å tilpasse seg bedre for observasjoner med lignende egenskaper som pasient 247 enn pasient 8.

4.6.5 Evaluering av modeller på MAASTRO data

Tabell 53 som presenterer hvordan det eksterne datasettet fra MAASTRO-klinikken i Nederland har presentert på de samtlige modellene som er trent på datasettet fra OUS med DFS som responsvariabel.

Tabell 53: Presenterer de gjennomsnittlige resultatene på MAASTRO data for hode- og hals med DFS som responsvariabel.

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
Random Forest	0.71 ± 0.02	0.79 ± 0.01	0.53 ± 0.04	0.37 ± 0.05	0.66 ± 0.02
Logistisk regresjon	0.68 ± 0.02	0.78 ± 0.02	0.45 ± 0.04	0.32 ± 0.06	0.62 ± 0.02
QDA	0.69 ± 0.01	0.78 ± 0.01	0.46 ± 0.02	0.33 ± 0.03	0.63 ± 0.01
GaussianNB	0.68 ± 0.04	0.75 ± 0.09	0.52 ± 0.03	0.33 ± 0.05	0.65 ± 0.02
Nearest centroid	0.69 ± 0.01	0.78 ± 0.01	0.45 ± 0.02	0.35 ± 0.03	0.63 ± 0.01
KNORA-E	0.66 ± 0.03	0.74 ± 0.03	0.48 ± 0.05	0.26 ± 0.07	0.62 ± 0.03
KNORA-U	0.69 ± 0.02	0.78 ± 0.02	0.49 ± 0.05	0.34 ± 0.05	0.64 ± 0.03
DES-P	0.69 ± 0.03	0.78 ± 0.02	0.49 ± 0.05	0.34 ± 0.06	0.64 ± 0.02

I likhet med MAASTRO datasettet på modellene trent med OS som responsvariabel, viser modellene trent med DFS som responsvariabel bedre for F1–positiv sammenlignet med F1–negativ. Det indikerer at modellen lettere kan forutsi korrekt for observasjoner med positive klassen. Dette er et resultat som er totalt motsatt av resultatene som ble observert på valideringsdata. Dette kan skyldes av ulik klassefordeling mellom MAASTRO- og OUS-datasettet, som beskrevet i kapittel 4.5.3. I kapittel 3.6.1 ble det presentert at MAASTRO datasettet har flere prøver med positive klasser enn negative, i kontrast til at OUS datasettet har flere negative prøver enn positive.

Modellen med random forest presterer best av alle de åtte modellene for MAASTRO datasettet. Modellen presterer best i samtlige nøyaktighetsmålinger, og F1-negativ, MCC og ROC-AUC måler betydelig bedre enn de resterende modellene i tabellen. De resterende av de klassiske ML-algoritmer presterer ganske jevnt. Det er kun GaussianNB og nearest centroid som skiller seg med henholdsvis høyere måling for F1-negativ og MCC.

Av DES-algoritmene presterer KNORA-U og DES-P ganske jevnt med de klassiske ML-algoritmene. Modellene har til og med bedre målinger for samtlige performance metrics enn hvordan enkelte av ML-algoritmene har prestert. Modellene gjør blant annet bedre enn modellene med logistisk regresjon og QDA. Av DES-modellene gir modellen med KNORA-E den svakeste ytelsen ved vurdere målingene den har gitt for F1-positiv og MCC.

I vedlegg G.5 presenteres en confusion matrix for alle de åtte algoritmene på MAASTRO-datasettet. I tillegg finnes en oversikt over prestasjonene til modellene som bruker algoritmene SVC, KNN, META-DES, OLA og MCB i vedlegg G.6.

4.7 Resultater av MCDA-analyse

For å foreta en akademisk og informativ sammenligning om hvorvidt Dynamic Ensemble Selection (DES) algoritmene er bedre egnet til å predikere enn de klassiske maskinlæringsalgoritmene kan dette bli basert på resultatene fra MCDA-analysen. Resultatene i denne analysen tar utgangspunkt i funnene fra test- og treningsdataene og confusion matrix som ble presentert innledningsvis i kapittel 4. I tillegg vil denne analysen bruke modellen som ble forklart i delkapittel 3.7.5 til å beregne totalscoren og gi en indikasjon på hvilket alternativ som kan være best for denne studien.

Som nevnt i kapittel 3.7.2, blir to alternativer anvendt for å evaluere og etablere et sammenlikningsgrunnlag for å finne det mest hensiktsmessige beslutningsstøtteverktøyet innen kreftmedisin. Det første alternativet undersøker anvendelsen av de klassiske maskinlæringsalgoritmene, mer spesifikt random forest, logistisk regresjon, QDA, samt GaussianNB og nearest centroid. Det andre alternativet ser på bruken av Dynamic Ensemble Selection fra DESlib-pakken, og vurderer KNORA-E, KNORA-U og DES-P som potensielle beslutningsstøtteløsninger. Begge alternativene har en fellesnevner i at de skal bli brukt som en integrert del av den eksisterende beslutningsprosessen, som baserer seg på kliniske og etiske retningslinjer samt pasientenes behov og ønsker.

4.7.1 Konkrete kvalitative scorer

Under følger figur 34 som tar for seg modellen som ble presentert i delkapittel 3.7.5 med verdiscorene for alternativene og vekting i prosent av kriteriene som er satt.

	Kriterie	Vekt (i %)								
1	Prediksjonsnøyaktighet	35 %								
2	Data- og modellkvalitet	40 %								
3	Funksjonalitet	25 %								
	Alternativ	**Score**	**Sum**	**Score**	**Sum**	**Score**	**Sum**		**Totalscore**	
1	De klassiske algoritmene + dagens løsning	3	1,05	3	1,2	4	1		3,25	
2	Dynamic Ensemble Selection (DES) + dagens løsning	2	0,7	2	0,8	2	0,5		2	

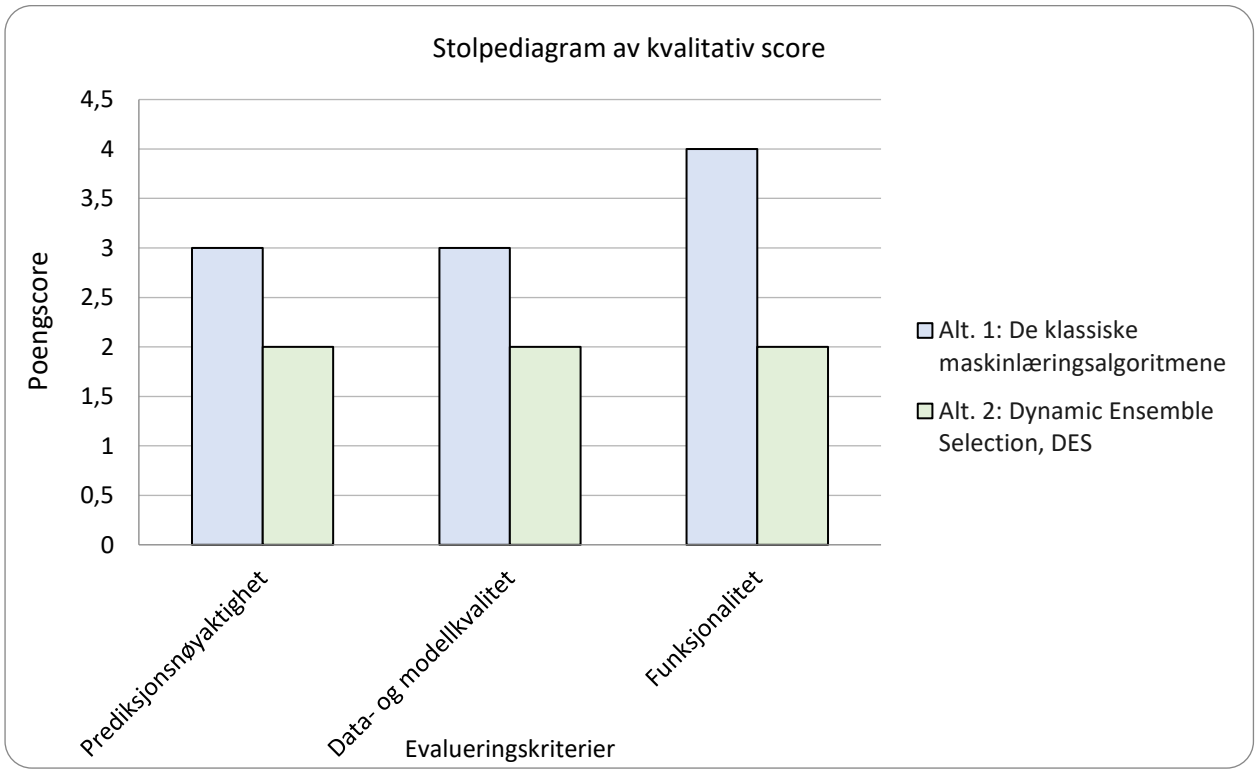
Figur 34: Den utfylte modellen for MCDA-analysen med verdiene hentet fra evalueringskriteriene og begrunnelsene gitt i kapittel 4.7.2.

De gule feltene er ment for å inneholde vektprosentene for de definerte evalueringskriteriene, som i dette tilfelle er beskrevet under kapittel 3.7.3. De blå feltene, derimot, skal inneholde den begrunnende scoren på en skala fra 1 til 5. Dette gjøres for å

evaluere i hvilken grad de ulike alternativene tilfredsstiller kriteriene. I det grønne feltet finner man den samlede totalscoren som har blitt beregnet ut ved hjelp av modellen som er basert på den presenterte formelen 3.2.

$$Totalscore = \sum_{i=0}^{n} (\text{score}_\text{i} \text{ x vekting}_\text{i} \ (\%))$$

Stolpediagrammet i figur 35 gir en omfattende oversikt og et bedre sammenlikningsgrunnlag av hvor godt de forskjellige alternativene oppfyller de gitte kriteriene.



Figur 35: Sammenlikning av kvalitative scorene.

4.7.2 Begrunnelse av gitt kvalitativ score

Prediksjonsnøyaktighet

Gjennom en helhetlig tilnærming og vurdering av alternativenes generelle prestasjoner for performance metrics i de ulike datasett med varierte responsvariabler, kan man konkludere med at alternativ 1 scorer best for datasettene. Generelt sett kan man observere at de klassiske algoritmene gjør det bedre for de fleste tilfellene enn DES-algoritmene for testdata/valideringsdata. Det er viktig å merke seg at fokuset på prediksjonsnøyaktighet

vanligvis ligger på testdata, da dette representerer ukjente data og gir dermed en indikasjon på hvordan algoritmene vil prestere på nye pasienter. Treningsdata kan ikke nødvendigvis gi en nøyaktig refleksjon av hva som kan forventes av en modell når den brukes på ukjent data.

Ved å evaluere nøyaktighetsscoren (accuracy) til testdata, viser resultatene at alternativ 1 har en gjennomsnittligscore som overgår DES-algoritmene. Likevel er det verdt å merke at i to tilfeller, oppnår DES-algoritmene marginalt bedre score enn de klassiske algoritmene, og det er for datasettet knyttet til kolorektal kreft med OS- og PFS-hendelser. Til tross for dette, er scoreforskjellene ikke vesentlige og har liten innvirkning på den generelle poengsummen for å vurdere i hvilken grad alternativet oppfyller kriteriet.

En nærmere undersøkelse av performance metricsene, spesielt F1-positiv, F1-negativ og ROC-AUC, indikerer at alternativ 1 presterer generelt bedre enn alternativ 2. Dette kan skyldes av at DES-algoritmene er alle såkalte ensemblemetoder, som betyr at de kombinerer flere modeller for å oppnå gode resultater. Imidlertid kan enkelte datasett eller problemstillinger være mer utfordrende for ensemblemetoder (Wang et al., 2011). Videre er KNORA-E og KNORA-U algoritmene basert på ideen om å la modeller stemme over beslutninger for å øke nøyaktigheten. Det kan være tilfeller der modellene ikke er enige om beslutning, og dette kan føre til performance metricsene påvirkes negativt.

Basert på denne informasjonen om ytelsesberegninger og evaluering av nøyaktighet, er det besluttet å tildele alternativ 1 en score på **3**, mens alternativ 2 har blitt tildelt en score på **2**. Selv om alternativ 1 oppnår bedre resultater enn alternativ 2, er det viktig å bemerke at begge modellene har utfordringer med å klassifisere den positive klassen, også kjent som F1-positiv. Ingen av alternativene gir perfekt nøyaktighet, noe som kan vurderes som en betydelig ulempe, spesielt når F1-positiv tar for seg utfallet om død eller ikke-død, tilbakefall eller ikke-tilbakefall, avhengig av responsvariabelen.

<u>Data- og modellkvalitet</u>

Alternativ 1 gjør det generelt bedre for dette kriteriet når man tar utgangspunkt i de datasettene som ble brukt i denne oppgaven. Som presentert, begge alternativene benytter seg av samme type datasett som består av manglende verdier (NaN-values). Nærmere bestemt hadde datasettet for kolorektal kreft og hode- og halskreft henholdsvis 6334 og 104

manglende verdier. En fellesnevner for begge datasettene er at de har begrenset data og informasjon. Datasettene består henholdsvis 192 og 197 pasienter, noe som resulterer i relativt små datasett og begrenset mengde data som kan brukes til testing. Det bør bemerkes at Alternativ 1 generelt sett er mer egnet til å håndtere datasett som inneholder lite data og manglende verdier.

Basert på det store avviket av manglende verdier i testdata og de få pasientene man hadde i datasettene, kan man basert på resultatene om antall feil klassifiserte pasienter konkludere om at Alternativ 1 som er de klassiske algoritmene presterer bedre enn Alternativ 2 – DES-algoritmene. Dette kan skyldes i stor grad det store antallet manglende verdier og det begrensende antallet pasienter i datasettene som ble brukt i studien. Til tross for disse utfordringene, ser det ut til at de klassiske algoritmene har klart å generalisere og finne viktige sammenhenger selv når datasettet ikke er optimalt, og dermed er i stand til å klassifisere færre pasienter feil enn DES-algoritmene.

Generelt sett kan man lese av de feil klassifiserte pasientene for både testdata, at de klassiske maskinlæringsalgoritmene (random forest, QDA og logistisk regresjon) har færre tilfeller for feilaktig klassifisering enn DES-algoritmene. I helheten kan man konkludere med at for datagrunnlag, så scorer alternativ 1 bedre enn alternativ 2. Selv om datasettet er begrenset og lite informasjon om pasientene ligger tilgjengelig, kan man observere ved hjelp av resultatene for feil klassifiserte pasienter at de klassiske algoritmene gjør det vesentlig bedre. Derfor ble alternativ 1 gitt en score på **3**, og alternativ 2 ble tildelt en score på **2**. Således, konkluderes det at generelle algoritmer egner seg bedre for de gitte datasett.

<u>Funksjonalitet</u>

Alternativ 1 scorer best på funksjonalitet, da alternativet bidrar til å spare tid og ressurser, og gjør alternativet enklere å bruke. Implementeringen av algoritmer som ikke støtter funksjoner eller andre metoder som kan optimalisere og effektivisere arbeidet kan forsinke prosessen med å finne svar og hjelpe legen med å ta en beslutning.

Alternativ 2 har et stort potensial, men i helsesektoren holder det ikke med en slik løsning, da effektiviteten ikke kan holde i takt med de andre alternativet. I tillegg krever implementeringen av alternativ 2 større ressurskapasitet, da genereringstiden for modellen er mye lenger for alternativ 2 enn alternativ 1. Dette går på konsekvensene av hvor effektivt

alternativet kan vurderes. For større ressurskapasitet og lengre tid på å få et svar, går på konsekvensene at behandlingsforløpet blir tregere.

Alternativ 2 hadde også problemer med bruk av cross_val_score funksjonen ved hyperparameteroptimalisering. Dette er en funksjon som er brukt i eksemplene i dokumentasjonen i Optuna og er anbefalt å bruke for å evaluere modellens ytelse med kryssvalidering (Akiba et al., 2019). Alternativet støtter ikke funksjonen for kryssvalidering ved hyperparameter optimalisering, og rapporterte feilmeldinger om at datasettet innehold manglende verdier (NaN-values), selv om det ikke var tilfelle. Løsningen på problemet var å rekonstruere programmet med en for-løkke og ta i bruk Repeated Stratified K-Fold for kryssvalidering. Dette resulterte i at hyperparameter optimalisering for modellene i alternativ 2 ble satt på vent og tok lenger tid å finne en løsning. Dette var et problem som ikke var tilfelle for alternativ 1, og var dermed et problem som svekket alternativ 2 ytterligere.

Grunnet den tidkrevende modelleringsprosessen og utfordringene med integrasjonen av funksjonen cross_val_score() for evaluering og støtten av Optuna (hyperparameter), er alternativ 2 ansett som mindre effektiv og tildelt en poengsum på **2**. Alternativ 1 som anses effektiv har blitt tildelt poengsum på **4**.

4.7.3 Rangering av alternativene

I denne rangeringen vil ikke dagens løsning bli tatt i betraktning under vurderingen, da oppgaven tar sikte på å evaluere forskjellige alternative løsninger som kan gjøre beslutningen sikrere og enklere. Som MCDA-analysen innledet med, vil de mulige alternative løsningene ikke bli ansett som en erstatning for den nåværende beslutningsprosessen som tar for seg kliniske og etiske retningslinjer som helsepersonalet arbeider etter, samt pasientens interesse og behov. På grunnlag av dette vil rangeringen og konklusjonen ta utgangspunktet i de gitte scorene og helhetsinntrykket av en kombinasjon av dagens løsning med ett av mulige alternativene.

Alternativ 1 – De klassiske maskinlæringsalgoritmene

Et godt alternativ for de datasettene som denne oppgaven jobber ut ifra. Det scorer best i alle kriteriene; prediksjonsnøyaktighet, datakvalitet og funksjonalitet, som gjør at dette alternativet total sett anses som den beste løsningen for helsepersonellet. Alternativet gir

gode score for nøyaktighet, modelleringstiden er mye kortere altså algoritmene skriver ut et svar raskt samtidig som den er mer nøyaktig. En annen fordel med dette alternativet er at det er færre tilfeller hvor de klassiske algoritmene feilaktig klassifiserer pasientene. I tillegg aksepterer de enkeltstående algoritmene ulike funksjoner og metoder som kan brukes for å optimalisere og effektivisere eksempelvis modelleringstiden eller gi mer konkrete svar. Dette vil bli ansett som en bonusfaktor siden dette kan være med på å fortsatt analysere små datasett, som i dette tilfelle, eller gjøre prosessen noe raskere samtidig som man holder en høy nøyaktighet. Alt i alt et alternativ som kan vurderes som et beslutningsstøtteverktøy for helsepersonell.

Alternativ 2 – Dynamic Ensemble Selection, DES-algoritmer

Oppfyller problemstillingens ønske om å ha et alternativ som kan brukes som et beslutningsstøtte for dagens løsning, men skuffer blant annet da den reelle ressursbesparelsen ikke er så effektiv som forventet. En ulempe med denne metoden er hva algoritmene scorer på nøyaktighet og antall feil klassifiserte pasienter. Ifølge kliniske ekspertisen, Hanne Osnes-Ringen, er dette to viktige kriterier for helsepersonell, for om algoritmene ikke klarer å tilfredsstille de mest essensielle og nøkkelfaktorene for problemstillingen så er det et alternativ som burde unngås å ta i bruk (Osnes-Ringen, 2023). For at et alternativ skal anses som et beslutningsstøtteverktøy, burde den løsningen ha gode resultater for de ulike målingene (performance metrics), samt kunne klassifisere pasientene korrekt.

En pålitelig respons er av avgjørende betydning for helsepersonell når de bruker algoritmer til å hjelpe dem i deres arbeid. Selv om det kan ta lengre tid å få et svar, er det av største betydning at svaret kan stoles på og at det er nøyaktig. Pålitelighet er spesielt viktig i helsesektoren hvor det kan ha stor innvirkning på pasientens liv og helse. Derfor må algoritmene være pålitelige og levere nøyaktige resultater for å kunne gi helsepersonell tillit til å ta viktige beslutninger.

Nedenfor presenteres rangeringen etter MCDA-analysen:

Nr. 1: Alternativ 0 + Alternativ 1, Dagens løsning + de klassiske maskinlæringsalgoritmene

Nr. 2: Alternativ 0 + Alternativ 2, Dagens løsning + Dynamic Ensemble Selection (DES)

Det er viktig å merke seg at ulike algoritmer og teknikker vil respondere forskjellig og gi ulike svar på forskjellige typer data og problemer. Det kan derfor være lurt å øke bevisstheten over at i denne oppgaven vil alternativ 1 være den beste løsningen som et beslutningsstøtteverktøy, men om datasettet er av høyere kvalitet, tar for seg et større omfang og/eller tar for seg andre type data utenom kreftpasienter så kan alternativ 2 være bedre. I tillegg bør det bemerkes at alternativ 1, til tross for navnet «de klassiske algoritmene», kun omhandler fem spesifikke algoritmer, nemlig random forest, logistisk regression, QDA, samt GuassianNB og nearest centroid som ble foreslått av LazyPredict. Det betyr ikke nødvendigvis at alternativ 1 presenterer alle algoritmene som faller seg naturlig under denne kategorien. Tilsvarende gjelder for DES-algoritmene, Derfor er det viktig å være klar over at valg av ulike algoritmer innenfor samme kategori kan føre til ulike resultater og konklusjoner, spesielt i en MCDA-analyse.

Kapittel 5

Diskusjon

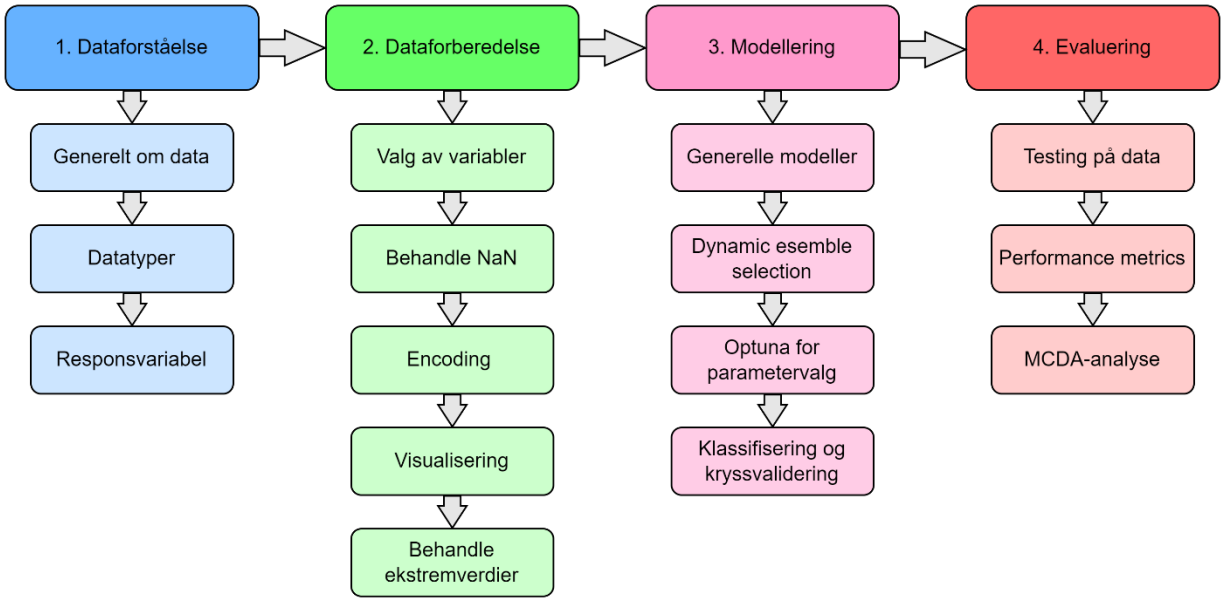
Formålet med masteroppgaven har vært å utforske bruk av maskinlæringsteknikker, spesielt Dynamic Ensemble Selection (DES), for å forbedre prediksjonen av kreftpasienter ved hjelp av ulike helsedata. En grundig analyse av DES-modellene og noen klassiske modeller ble utført for å evaluere deres muligheter til å utvikle medisinske applikasjoner som kan brukes som beslutningsstøtteverktøy for helsepersonell. Denne analysen ble gjort på to ulike datasett med forskjellige behandlingsutfall. Det første datasettet tok for seg kolorektal kreft med generell overlevelse (OS) og progresjonsfri overlevelse (PFS), og det andre datasettet tok for seg hode- og halskreft med generell overlevelse (OS) og sykdomsfri overlevelse (DFS). I det følgende kapittelet diskuteres oppgavens resultater og eventuelle utfordringer knyttet til oppgaven, samt foreslå mulige løsninger på disse problemene.

5.1 CRISP-DM som metodologi

CRISP-DM-prosessen er ikke bare relevant for prosjekter som arbeider med data. Den kan også brukes som en overordnet modell for å planlegge og administrere andre type datanalyseprosjekter. Det er verdt å merke seg at selv om CRISP-DM-prosessen gir en god ramme for data mining-prosjekter, er det fortsatt viktig å ha kunnskap om dataanalysemetodene som brukes, og å ha tilstrekkelig kompetanse for å gjennomføre prosjektet på en vellykket måte.

Som forklart i delkapittel 2.4 av boken *CRISP-DM 1.0: Step-by-step data mining guid,* kan man forstå at metoden består av seks faser. I denne oppgaven ble kun fire av fasene tatt i bruk, som illustrert i figur 36, altså forretningsforståelse og implementering ble utelukket. Dette gjør at metoden ikke har fulgt metoden fra A til Å, og grunnen til det er den manglende informasjonen. Dersom man hadde fått informasjon og behov som sykehusene trenger rundt problemstillingen, kunne man brukt behovsanalysen til å inkludere forretningsforståelse. Og som en del av fase seks som er implementering, kunne man lagd en pilotversjon av et Dashboard som kan eksempelvis vise noe av resultatene. Dette faller utenfor formålet med denne oppgaven, og dermed kan man se på dette i videre arbeid. Ved å utelate implementering kan man gå glipp av verdifulle tilbakemeldinger fra brukere og virkeligheten,

og dermed ikke oppnå ønskede resultater.



Figur 36: Arbeidsflyten, den presenterte arbeidsmetoden gitt i kapittel 3.4.

Fordelen med å bruke CRISP-DM er at metoden gir en strukturert tilnærming som kan hjelpe til med å organisere arbeidet på en hensiktsmessig måte, og ikke minst for å kunne overføre arbeidsflyten til videre arbeid. Gjennom denne prosessen har det dannet seg en større forståelse og oppmerksomhet rundt hvor effektivt arbeidet blir ved å etablere en arbeidsflyt tidlig i oppgaven. Dette gjør at alt blir likestilt og har det samme utgangspunktet. Samtidig gir det et klart rammeverk for å definere problemet, forstå dataene, lage og evaluere modeller og implementere resultatene.

En annen fordel med denne metoden er at den legger til rette for en iterativ tilnærming til maskinlæring, hvor resultatene fra en fase kan brukes til å justere og forbedre neste fase. Som for eksempel når man jobber med dataforståelsen før modelleringen, så sørger man for at man filtrerer ut og kvitter seg med de dataene som er irrelevante og som vil lage støy for modellene. Dette gjør at datasettet vil være bedre rustet og gir mer pålitelige og ønskede svar når man beveger seg til neste fase.

Selv om CRISP-DM-prosessen kan tilpasses og brukes til forskjellige typer data mining-prosjekter, er det viktig å merke seg at prosessen ikke er en magisk formel for å løse alle datarelaterte problemer. Det er fortsatt viktig å ha en dyp forståelse av problemstillingen og dataene man jobber med, og å ha kunnskap om relevante dataanalyseverktøy og -teknikker.

Til slutt er det verdt å merke seg at CRISP-DM-prosessen er en levende prosessmodell som stadig utvikles og forbedres. Selv om prosessen har vært i bruk i mange år, er det fortsatt muligheter for å tilpasse og forbedre modellen for å gjøre den enda mer effektiv og relevant for moderne prosjekter.

5.2 Datasett

Begge datasettene som har blitt brukt i oppgaven er av ulike grunner ikke det beste datasettet som er egnet for å lage maskinlæringsmodeller. For å generere gode ML-modeller er det viktig å ha et datasett med god kvalitet, samt være representativ for formålet modellen skal brukes til. Det er flere faktorer som er med på å avgjøre om et datasett er representativ for formålet og om den er av god kvalitet. Størrelse, pålitelighet, representativ og balanse er noen av mange viktige faktorer som man burde ta stilling til ved vurdering av et datasett. Følgende underkapittel diskuterer datasettene i oppgaven og hvilke utfordringer disse kan følge med ved generering av ML-modeller.

5.2.1 Kolorektal kreft

Datasettet bestod av 192 pasientprøver som fikk behandling for kolorektal kreft i tidsperioden fra 2013 til 2017. Det er verdt å merke seg at det er flere år siden dette datasettet ble samlet, og det er dermed en mulighet for at det kan være noen endringer som kan ha skjedd i mellomtiden som kan påvirke påliteligheten til dataene. I likhet med resten av verden, har Norge vært igjennom den velkjente Covid-19-pandemien. Denne pandemien har hatt en betydning innvirkning på samfunnet, og det kan ligge en mulighet for at diverse forhold har endret seg etter den tid. Noen av forholdene som har endret seg er blant annet behandlings- og ventetiden, hvor den har blitt lengre under og etter pandemien sammenlignet med 2019 (Helsedirektoratet., 2023). Dette er en kilde til usikkerhet som kan påvirke påliteligheten til datasettet når det gjelder dets nåværende relevans og gyldighet.

Størrelsen på datasettet er heller ikke positivt. Jo større datasettet er, desto bedre klarer modeller å tilpasse seg og dermed klare å generalisere godt til nye data. Som nevnt i det teoretiske rammeverket, registrerte kreftregisteret at 1687 kvinner og 1517 menn ble påvist for kolorektal kreft i 2021 (Kreftregisteret, 2022). Modellene i oppgaven som er trent med OxyTarget datasettet vil dermed ha problemer med å representere norgesbefolkningen. Datasettet har også ubalanse i datasettet med skjevfordelte responsvariabler. Prøver med

OS-evnet er veldig ubalansert og et datasett som er veldig ubalansert, har tendens til å favorisere den dominerende klassen og dermed presterer dårlig på den underrepresenterte klassen. Dette ble observert med resultatene fra modellene i oppgaven, og dette er et resultat som ikke er positivt.

Som presentert i kapittel 3.5.1 består datasettet av hele 6334 manglende verdier som tilsvarer 32.99%. Å erstatte alle disse manglende verdiene med representative verdier kan føre til at datasettet blir mindre presist, da dette kommer til å utgjøre en stor del av datasettet og dermed ikke være det samme som ha de reelle verdiene. På bakgrunn av dette, ble variabler med veldig mange manglende verdier eliminert som beskrevet i kapittel 3.5.1, og dette kan ha vært med på å få vekk viktige variabler som ellers kunne ha vært en viktig betydning for modellene.

5.2.2 Hode- og halskreft

Hode- og halsdatasettet fra OUS besto av 197 pasientprøver som ble behandlet for hode- og halskreft i perioden 2007-2013. Som for kolorektal datasettet ble dette datasettet samlet for noen år tilbake og dermed være påvirket av endring i flere forhold som kan gjøre datasettet noe mindre pålitelig.

Helsedirektoratet presenterer i et handlingsprogram at det årlig diagnostiseres 800 nye tilfeller med hode- og hals kreft (Helsedirektoratet., 2020). På lik linje som for kolorektal, viser statistikken til at det vil være mange med hode- og halskreft i samfunnet, og en modell som er trent med 197 pasienter vil dermed støtte på store utfordringer til å representere norgesbefolkningen. Datasettet er også ubalansert med skjevfordelte responsvariabler. Som for modellene med kolorektal, ble det også observert at dette har vært et problem som har virket negativt på modellene med hode- og halskreftspasienter.

Datasettet hadde totalt 104 manglende verdier fordelt på variablene *«hpv_related»* og *«uicc8_III_IV»*. Pasientene med manglende verdier for variablene ble som beskrevet i kapittel 3.6.2 utdelt en egen kategori. Dette kan være med på å påvirke modellene til finne viktige sammenhenger som ellers ville vært mulig om datasettet ikke hadde manglede verdier.

5.3 Hyperparameteroptimalisering (Optuna)

For å finne de beste kombinasjonene av hyperparametere, ble det i denne oppgaven undersøkt hvordan det åpne kodebiblioteket Optuna fungerer for formålet. I forhold til den tradisjonelle metoden grid search har denne oppgaven erfart Optuna som et veldig effektivt og tidsbesparende verktøy for å finne de beste kombinasjonene med hyperparametere. Den er også enkel å sette opp, og det ligger godt med eksempler i dokumentasjonen som kan brukes som støtte ved uklarhet og oppsett. Grid search er en metode som prøver ut alle mulige kombinasjoner og foreslår den beste kombinasjonen (Pedregosa et al., 2011). Optuna derimot har en adaptiv algoritme som lærer av tidligere evaluering, og dermed justerer parameteren slik at den kan utforske områder som har større potensial for å oppnå bedre resultatkombinasjoner. På denne måten kan Optuna effektivt navigere gjennom det enorme søkeområdet for hyperparameterkombinasjoner og finne en optimal kombinasjon raskere og mer effektivt.

Optuna har en parameter som heter «trials» som spesifiserer antall kombinasjoner som skal prøves under hyperparameteroptimaliseringen. Denne parameteren gir en betydelig tidsbesparelse ved å begrense antall kombinasjoner som man mener kan være nok for å finne den beste kombinasjonen. I oppgaven ble det også observert at Optuna er i stand til å generere informative visualiseringer som viser resultatene fra hvert forsøk. Disse visualiseringene gir en enkel forståelse av hvordan forskjellige hyperparameterkombinasjoner påvirker ytelsen. I kapittel 3.4.3 blir en slik visualisering presentert, som er generert av innebygde funksjoner i Optuna.

Den største utfordringen med Optuna har vært å bruke den sammen med DES-modeller. Som nevnt i MCDA-analysen, er det anbefalt å bruke «cross_val_score» fra scikit-learn sammen Optuna. Denne teknikken brukes for å beregne og evaluere score som også ble brukt i denne oppgaven, men modellene med DES-algoritmer hadde problemer med bruk av denne funksjonen. Modellene med DES støttet derimot ikke funksjonen «cross_val_score», og rapporterte feilmeldinger om at datasettet inneholdt manglende verdier (NaN-values), når dette ikke var tilfelle. Dette skapte flere utfordringer, spesielt når feilmeldingene ikke var spesifikke og gjorde det vanskelig å identifisere problemet. Etter flere forsøk på å finne årsaken til problemet ble det oppdaget at modellene ikke støttet "cross_val_score"

funksjonen. For å løse problemet ble funksjonen erstattet med "Repeated Stratified K-Fold" for kryssvalidering, som ble brukt til å beregne modellenes score ved hyperparameteroptimalisering.

5.4 Evaluering av resultater

Datasettet for kolorektal kreft ble undersøkt med fokus på responsvariablene, generell overlevelse (OS) og progresjonsfri overlevelse (PFS). Datasettet for hode- og halskreft, derimot, ble gitt med responsvariablene OS, sykdomsfri overlevelse (DFS) og lokalt eller regionalt tilbakefall (LRC). Av disse ble det laget modeller med OS og DFS. Følgende underkapittelet diskuterer modellenes prestasjoner av de presenterte datasettene og deres tilhørende responsvariabel, og hvordan modellene med DES-algoritmer har presentert i forhold til de klassiske algoritmene.

5.4.1 Kolorektal kreft med responsvariabelen OS-event

Modellen som hadde høyest gjennomsnittlig ytelse for kolorektal datasettet med OS-event var QDA og GaussianNB, med en MCC på 0.48. Sammenlignet med de andre nøyaktighetsytelsene, gir MCC målingene som presentert i kapittel 2.2.6 en mer robust og rettferdigmåling da den tar hensyn til ubalanse i data. Som forklart i kapittel 3.5.1, er det også kjent at datasettet er ubalansert med flere prøver med den negative klassen enn den positive. Av modellene med QDA og GaussianNB presterer modellen med GaussianNB bedre for nøyaktighetsmålingen F1-positiv. F1- positiv har vist seg å være en nøyaktighetsytelse som er vanskelig å gjøre det bra på, da datasettet er ubalansert, og har dermed problemer med å klassifisere den underpresterende klassen i datasettet.

En av utfordringene som både DES-algoritmene og klassiske algoritmene møter når de forsøker å predikere nøyaktige målinger for dette datasettet, er mangfoldet av variabler. Mange av variablene har blitt eliminert vekk, som forklart i kapittel 3.5.2. Dette kan ha ført til at viktige variabler også er blitt fjernet, som kunne ha avgjørende for modellene i å finne viktige sammenhenger. Tiltak som kan tas for å være flinkere til å eliminere de riktige variablene, er mange som blant annet vurdere påvirkningen av elimineringen. Dette kan være ved at man gjør det som ble presentert under «viktige og mindre viktige variabler» i kapittel 4.2.5, ved å ta en og en variabel å se hvor stor påvirkning dette har for oppgaven. Videre består datasett av mange variabler i forhold til antall prøver, og dette kan forstyrre modellene

i å finne viktige sammenhenger og varians mellom pasientprøvene. Det er derfor viktig å ta hensyn til kompleksiteten av variabler når man bygger en prediksjonsmodell for slike datasettet.

En mulighet som kan testes ut for dette datasettet, er feature engineering og se om dette kan hjelpe modellene med å prestere bedre. Feature engineering referer til utledning av nye variabler basert på tilgjengelig data. Gitt at datasettet består av et omfattende antall variabler, gir det rom for muligheter for å utlede nye og relevante variabler. Dette kan være avgjørende betydning for å oppdage andre sammenhenger som kan bidra til å øke prediksjonsytelse til modellene.

5.4.2 Kolorektal kreft med responsvariabelen PFS-event

For kolorektal datasettet med responsvariabelen PFS, også kjent som progresjonsfri tilbakefall, hadde QDA høyest gjennomsnittlig ytelse med en MCC på 0.48. Andre algoritmer som oppnådde tilsvarende MCC-måling var random forest, KNORA-E, KNORA-U. Funnet antyder at DES-algoritmer, sammenlignet med klassiske algoritmer, kan oppnå like gode resultater. Dette kan forklares av at DES-algoritmer tar hensyn til variasjon og usikkerhet i dataene, og dermed kan de være bedre egnet til å håndtere komplekse sammenhenger og høydimensjonale datasett.

På samme måte som for modellene med OS-event, kan modellene hatt utfordringer med å finne viktige sammenhenger i datasettet i betydning av den har få prøver og mange variabler. I forhold til modellene med OS, har samtlige modeller med PFS fått høyere prediksjonsnøyaktighet på F1-positiv. Dette kan være av at datasettet er mer banalsnert for PFS-event enn OS-event.

5.4.3 Hode- og halskreft med responsvariabelen OS-event

For modellene med OS som responsvariabel ble det presentert i resultatkapittelet at de klassiske ML-modellene med random forest, logistisk regresjon og QDA kommer best ut. Alle disse tre modellene fikk respektable målinger på de gjennomsnittlige nøyaktighetsytelsene. Av disse tre nådde modellene med random forest og logistisk regresjon best resultat for MCC med en ytelse på 0.48.

De samme modellene testet på det eksterne datasettet fra MAASTRO klinikken viste seg å ha en drastisk fall i MCC målingen for samtlige modeller. Modellen med random forest var den beste med en måling på 0.3. Den store forskjellen mellom datasettet gitt av OUS og MAASTRO ligger i klasseforskjellen. Som allerede nevnt inneholdt OUS datasettet, flere prøver av den negative klassen enn den positive, i motsetning til at MAASTRO datasettet hadde flere prøver med den positive enn den negative klassen. Dette kan ha ligget i grunn for at modellen har hatt en nedgang i sin prediksjonsytelse. Det må også settes i betraktningen at pasienter er forskjellige fra hverandre, og forskjellen vil være enda større mellom pasienter i Norge og pasienter i Nederland. Det kan blant annet være forskjeller i genetiske sammetninger, livstil, kosthold, miljøfaktorer, tilgjengelighet av helsetjenester og behandling (Syse, 2014). Dette kan muligens sees i sammenheng med at modellene som er trent med OUS er mer robust og pålitelig for klassifiseringsproblemer med norske pasienter enn andre.

5.4.4 Hode- og halskreft med responsvariabelen DFS-event

Den klassiske ML-modellen, random forest, presterte best totalt sett for responsvariabelen DFS. Modellen hadde en MCC nøyaktighet på 0.4 som er betydelig bedre enn hvordan de andre modellene hadde prestert. De samme modellene testet på det eksterne datasettet fra MAASTRO klinikken viste seg å ha en fall i MCC måling for samtlige modeller. På lik linje som for modellene med OS, var det fortsatt modellene med random forest som presterte best for det eksterne datasettet med en MCC måling på 0.37 som er veldig nær nøyaktigheten fikk valideringsdata.

Både OS og MAASTRO datasettet hadde en mer balansert i klassefordeling for DFS enn OS. Selv om balansen mellom klassene er jevnere dominerer fortsatt den negative klassen for OUS datasettet, og den positive klassen dominerer i MAASTRO datasettet. Samtlige modeller presterer ganske jevnt for både valideringsdata og MAASTRO for målingene nøyaktighet, MCC og ROC-AUC. Dette kan skyldes av at modellene har klart å finne viktige sammenhenger når klassene er mer balansert, og derfor presterer ganske jevnt på begge datasettet enn hvordan modellene med OS hadde presentert.

Modellen med KNORA-U presenterer best av alle DES-modellene, men den er fortsatt slått av den klassiske modellen med random forest. DES-modellene hadde de dårligste målingene på valideringsdata i forhold til de resterende modellene, men for det eksterne datasettet

presenterte KNORA-U og DES-P bedre nøyaktighet for MCC enn de andre utenom modellene med random forest og nearest centroid. Totalt sett indikerer resultatene at KNORA-U kommer best ut av DES-modellene, men modellen med random forest gjør det best av alle.

5.4.5 Algoritmenes utførelse på tvers av datasettene

De presenterte algoritmene har vist varierende respons i forhold til begge datasettene, samt tilhørende responsvariablene. Generelt kan det observeres at klassiske algoritmer, særlig random forest og QDA, har oppnådd gode resultater for begge datasettene. På kolorektal datasettet har GaussianNB fungert som en av de beste algoritmene, og for hode- og halsdatasettet var modellene med GaussianNB blant de dårligste. Dette resultatet antyder at GaussianNB ikke nødvendigvis er en egnet algoritme for å håndtere på tvers av ulike helsedata.

Basert på litteraturen *Python data science handbook: Essential tools for working with data,* kan det også bemerkes at en av svakhetene med GaussianNB er at den forutsetter normalfordeling for de forklarende variablene (VanderPlas, 2016). Som det er vist i violinplottene for de ulike datasettene, kan man se at hode- og halskreft datasettet har en begrenset mengde normalfordelte variabler. Av de 14 variablene som er inkludert i datasettet (uten responsvariablene), er 11 av dem kategoriske variabler. Siden disse variablene ikke kan antas som kontinuerlige, kan de heller ikke være normalfordelt, og dermed ansees som en annen årsak til at GaussianNB presterer dårligere for hode- og halskreft datasettet enn kolorektal datasettet.

Av DES-algoritmene presterte alle ganske likt for kolorektal datasettet med OS og PFS, samt for hode- og halsdatasettet med OS som responsvariabel. MCC-nøyaktigheten var ganske jevn og lik for disse algoritmene. Imidlertid var det en nedgang i MCC-nøyaktigheten for modellene som brukte DES-algoritmer når DFS var responsvariabelen for hode- og halskreft. Dette kan skyldes av at modellene ikke har klart å identifisere klare sammenhenger i treningssettet for å skille mellom klassene, ettersom datasettet består av få prøver og har en mer normalfordelt klassefordeling enn ved bruk av OS. Ved å observere resultatene til MAASTRO-data med DES-modellene, antyder resultatene at algoritmene KNORA-U og DES-P hadde best ytelse på tvers av datasettene. Disse to DES-algoritmene hadde dermed jevne resultater på tvers av datasettene i oppgaven.

5.5 LazyPredict

I tillegg til å benytte DES-algoritmer og klassiske ML-algoritmer, ble det i denne oppgaven også inkludert algoritmer som er blitt anbefalt av LazyPredict for å oppnå gode resultater på datasettet. Som presentert i kapittel 3.7.1, ble GaussianNB foreslått som den beste algoritmen for kolorektal datasettet, i motsetning til at nearest centroid ble foreslått som den beste for hode- og halskreftdatasettet.

Resultatene for modellene GaussianNB har vist seg å gjøre det bra for kolorektal datasettet, spesielt for modellen med OS som responsvariabel. GaussianNB var en av de best med en MCC nøyaktighet på 0.48 på valideringsdata. For modellene med PFS som responsvariabel presterte modellen med GaussianNB noe svakere, og var en av de som hadde laves MCC score på 0.44. Modellene med nearest centroid på kolorektal datasettet, presenterte ganske jevnt med de beste modellene for datasettet. Modellen presenterte til og med bedre enn GaussianNB for PFS som responsvariabel.

Modellene med nearest centroid for hode- og halskreftdatasettet viste seg å ha gode resultater på valideringsdataene. Som nevnt tidligere i kapittelet, oppnådde random forest de beste resultatene for modellene for både OS og DFS. Imidlertid viste nearest centroid også gode resultater på valideringsdataene og var en tett utfordrer til random forest. GaussianNB som ble foreslått av LazyPredict for kolorektal datasettet, presterte derimot dårligst av samtlige modeller med OS som responsvariabel og var en av de dårligste for DFS. På MAASTRO datasettet presenterte modellene med GaussianNB og nearest centroid ganske jevnt med de resterende modellene for både OS og DFS.

Selv om LazyPredict har foreslått at GaussianNB og nearest centroid skal gjøre det best på datasettene, viser store deler av resultatene at det ikke er tilfelle sammenlignet med de andre modellene i oppgaven. En av de viktigste årsakene til dette er at LazyPredict bruker standard hyperparametere når den lager modeller og presenterer den best av disse. Dette betyr at modellene som LazyPredict foreslår kan være best når modeller ikke optimaliseres, men det er mulig at andre modeller kan gjøre det betydelig bedre når de optimaliseres. Dette kan ligge i grunn for at enkelte modeller i oppgaven har vist seg å prestere bedre enn de foreslåtte modellene med LazyPredict.

I praksis har LazyPredict vist seg å være veldig enkel i bruk og bruker relativt kort tid på å presentere en oversikt over hvilke modeller som presterer bedre og dårligere for et datasett. Denne oppgaven har erfart at LazyPredict har sine fordeler, der man kan teste ut ulike tilnærminger, lære nye algoritmer og være et nyttig verktøy for de som ønsker å utforske ulike modellvalg.

Det oppgaven har erfart som negativt med LazyPredict er mulighet den har for parameteroptimalisering og validering. Som nevnt tidligere, tillater ikke pakken å optimalisere parametere. LazyPredict bruker også en enkel valideringsteknikk ved å splitte datasettet inni i treningssett og testsett. Dette vil da være med på å ikke gi like pålitelige resultater som ved kryssvalidering. Om LazyPredict tillater for optimalisering av hyperparametere samt endrer sin valideringsteknikk, vil pakken være mer pålitelig og verdifull for mange

5.6 Overtilpasning

Når validerings-/testresultatene sammenlignes med treningsresultatene indikerer enkelte modeller som overtilpasset *(eng. overfitting)*. For alle fire tilfellene med, OS-event for kolorektal, PFS-event for kolorektal, OS-event for hode- og halskreft, og DFS-event for hode- og halskreft, så viser treningsresultatene til modeller med DES-algoritmer er overtilpasset. Dette kan observeres ved at modellene har fått en nøyaktighet på 1 for alle nøyaktighetsmålinger. Når resultatene sammenlignes med testresultatene, kan det observeres at modellene presterer noe dårligere, da modellene hadde fått noen lavere nøyaktighetsytelse på testdata.

I tillegg til DES-modellene, viser også random forest på kolorektal datasettet med OS-event at den er overtilpasset. En fellesnevner for modellene med DES-algoritmene og random forest er at disse er ensemble-modeller, og tar i bruk beslutningstrær i sine algoritmer. Dette er en sammenheng som kan trekkes i linje med en litteratur som har diskutert om at modeller bygget opp med beslutningstrær har tendenser til å overtilpasse treningsdata, spesielt når antall beslutningstrær økes (Kumar & Jain, 2020). Dette skyldes av at ensemble-modeller trener beslutningstrær med deler av treningsdatasettet, og dersom modellen inneholder mange beslutningstrær, er det stor sannsynlighet for at en observasjon blir klassifisert av minst ett tre. Dette kan resultere i en høy prediksjonsnøyaktighet for treningsdata, men dette

kan også føre til at modellen tilpasses for støy i dataen og dermed ha dårligere ytelse på nye data.

I oppgaven ble antall bestemmelsestrær (n_estimators) bestemt av Optuna etter hyperparameteroptimalisering. Oversikt over disse er presentert i vedlegg A. En mulig tilnærming for å redusere overtilpasning i ensemble modeller kan ligge i å begrense antall beslutningstrær i søksområdet ved hyperparameteroptimalisering. Dette kan deretter brukes til å evaluere om det gir en effekt på resultatene og en reduksjon i overtilpasning. Selv om dette kan være en løsning på problemet, viser resultatene av modellene med random forest på hode- og halskreft datasettet at modellene ikke er overtilpasset, selv med flere beslutningstrær enn modellene med random forest for kolorektal kreft. Dermed kan det også ligge andre faktorer som påvirker ensemble modellene til å være overtilpasset.

En mulig forklaring kan være at variasjonen i treningsdataen er for lav, noe som gjør det vanskelig for modellene å finne klare mønstre i datasettet og dermed bli overtilpasset på treningsdataen. Som en følge av dette vil modellene ha vanskeligheter med å finne avgjørende skiller og mønstre i testdata, og dette kan føre til at modellene presterer dårlig på testdata. Dette problemet skyldes ofte av at datasettet er relativt lite, noe som også er tilfellet i denne oppgaven med kolorektal og hode- og hals datasettene.

Resultatene i denne oppgaven viser at modeller med DES-algoritmer har en tendens til å overtilpasse mer enn modeller med random forest og andre klassiske ML-algoritmer. Dette er tydelig da alle modellene med DES-algoritmer hadde en prediksjonsnøyaktighet på 1 for treningsdataene. Modellene med DES-algoritmer kan være ekstra sårbare for overtilpasning sammenlignet med modeller med random forest, fordi DES bruker dynamisk tilnærming for å velge individuelle modeller som inkluderes i ensemblet for hver prediksjon. Dette kan føre til at DES-modellene tilpasser seg støy i datasettet i større grad og dermed presenterer dårlig på nye data.

5.7 Evaluering av MCDA-analyse

Formålet med MCDA-analysen var å gjøre en grundig seleksjonsprosess for å finne ut hvilken av de to presenterte alternativene i oppgaven som vil være formålstjening i dagens og fremtidens virksomhet. MCDA-analyse er en metode for å ta beslutninger når det er flere kriterier som må tas i betraktning samtidig. MCDA kan være en verdifull analysemetode i evalueringsfasen av CRISP-DM-prosessen for flere grunner.

For det første kan MCDA hjelpe til med å definere kriteriene som skal brukes for å evaluere modellen. Ved å bruke MCDA-metoder kan man identifisere de viktigste kriteriene for å evaluere modellen, og definere dem på en klar og objektiv måte. Dette kan bidra til å sikre at evalueringen er grundig og systematisk.

For det andre kan MCDA bidra til å kvantifisere og veie ulike kriterier. Ved å bruke MCDA kan man kvantifisere ulike kriterier og vekte dem i henhold til deres relative betydning. Dette kan bidra til å redusere subjektiviteten og øke objektiviteten i evalueringen.

I sum kan bruk av MCDA-metoder i evalueringsfasen av CRISP-DM-prosessen bidra til å sikre at evalueringen er grundig, objektiv og systematisk. Ved å kvantifisere og vekte ulike kriterier kan man også ta beslutninger som er mer informerte og nøyaktige, og som bidrar til å optimalisere ytelsen til modellen eller løsningen som evalueres.

Avgrensingene i MCDA har vært med på å snevre inn analysen. Dette har sørget for at de delene som skulle analyseres har blitt analysert på en grundig måte. Det man derimot står i fare for når man gjør avgrensninger, er å ikke inkludere aspekter som har påvirkning. Resultat man står igjen med kan derfor være misvisende eller uklart.

For det første hadde man ikke et grunnlag å basere analysen på, som var en stor utfordring. En behovsanalyse som tar for seg sykehusets behov og ønsker hadde gjort analysen mye mer kvalitetssikret. I henhold til *Konseptvalg med flermålsanalyse som verktøy*, vil det være en betydelig usikkerhet knyttet til utførelsen av analyser dersom det er begrenset informasjonsgrunnlag tilgjengelig (Reinertsen, 2014). Informasjonsgrunnlaget representerer et viktig steg i gjennomføringen av analyser, og dermed kan en mangel på informasjon føre til feilaktige konklusjoner. I slike tilfeller anbefales det å utføre intervjuer for å øke mengden tilgjengelig informasjon som vil gjøre analysen mer troverdig (Reinertsen, 2014).

Dette var en grunnene til at en klinisk ekspert ble etterspurt og brukt i denne analysen for å sette de ulike evalueringskriteriene. Selv om en klinisk ekspert kom med anbefalinger på evalueringskriteriene og dets vekting, kan dette kanskje være basert på litt kjønn og andre preferanser. Dette gjør at MCDA-analysen kan anses som subjektiv og ikke presenterer alle sykehusene i Norge. Imidlertid er informanten en overlege med dyp innsikt i sitt fagfelt, og dermed vil hennes synspunkter og meninger bli vurdert ut fra et helseperspektiv.

Ved å ikke ha med kvantitativ data, blir resultatene og konklusjonene som blir gjort av MCDA-analysen svekket grunnet manglende metode. En mulig løsning for dette avviket kunne ha vært tilgjengeliggjøring av økonomiske data relatert til sykehusdrift. Da ville man kunne styrke konklusjonene som ble gjort underveis i analysen. Videre ville dette muliggjøre bruk av ulike simuleringer, eksempelvis Monte Carlo-simulering, som er en beregningsmetode for usikkerhetsanalyse (Mooney, 1997). Gjennom bruk av slike teknikker ville det være mulig å beregne estimater og tildele poeng til de ulike kriteriene for alternative løsninger. Dette ville gi et mer presist bilde av usikkerhetene og kompleksitetene som er involvert i beslutningsprosessen og ville bidra til å gjøre beslutningene mer velbegrunnede.

5.8 Sammenlikning av klassiske ML- og DES-algoritmer

For å kunne svare på forskningsspørsmålene burde en diskusjon og sammenligning av algoritmene bli gjort. Som nevnt i det teoretiske rammeverket, kapittel 2.3.3, har de ulike maskinlæringsalgoritmene sine fordeler og ulemper. Følgende underkapittelet setter algoritmene opp mot hverandre, og diskuterer svakhetene og mulighetene som har blitt avdekket til å være til stede.

5.8.1 Klassiske ML-algoritmer

I denne oppgaven har det vært begrenset fokus i fem utvalgte klassiske maskinlæringsalgoritmer, nærmere bestemt random forest, logistisk regresjon, QDA, samt de to anbefalingene gitt av LazyPredict; nearest centroid og GaussianNB. Det er derfor verdt å påpeke at begrensningen til kun fem algoritmer gjør at konklusjonen av både oppgaven og den utførte MCDA-analysen kan bli noe misvisende, da det finnes et stort antall klassiske algoritmer som ikke er inkludert i analysen og oppgaven.

Selv om de klassiske algoritmene er det alternativet som ble anbefalt av MCDA-analysen, og de viste seg å være effektive i ulike testscenarioer i denne oppgaven, bør det legges noe oppmerksom i at dette ikke nødvendigvis vil gjelde i alle tenkelige situasjoner. De klassiske har enkelte særtrekk, slik som deres lineære natur, som gjør at de kan oppnå bedre ytelse enn DES-algoritmene i visse sammenhenger. Det er verdt å merke seg at de klassiske algoritmene også kan fungere godt med begrensede og mindre omfattende datasett.

En viktig faktor som bør tas med i betraktningen i denne oppgaven, er at de klassiske algoritmene ga relativ gode resultater for akkurat de datasettene som denne oppgaven brukte. Det er imidlertid viktig å erkjenne at for andre datasett med større variasjon og en betydelig større prøvepopulasjon, kan resultatene avvike helt fra det som har blitt observert her. Dette kan skyldes at ytelsen til maskinlæringsalgoritmene i stor grad er avhengig av kvantiteten og kvaliteten på datasettene som blir brukt. Jo større datasett de klassiske ML-algoritmene har tilgang til, desto mer pålitelige og validerte resultater vil de sannsynligvis gi. Imidlertid vil modelleringstiden øke naturligvis, og det kan også være flere tilfeller av feilaktig klassifisering.

5.8.2 DES-algoritmer

Oppgaven har som formål å evaluere om Dynamic Ensemble Selection (DES) algoritmer fra DESlib-biblioteket kan oppnå like gode eller bedre resultater enn klassiske maskinlæringsalgoritmer, som beskrevet i oppgaveinnledningen. DESlib-biblioteket er et relativt nytt bibliotek som ble ansett for å ha potensial til å prestere godt ved studiets start. Basert på utført arbeid og presenterte resultater, indikerer modellene med DES-algoritmer at de ikke oppnår det evaluerte potensialet.

Selv om resultatene ikke var helt som forventet, betyr det ikke nødvendigvis at DES-algoritmene fra DESlib pakken bør utelukkes som alternativ for å utvikle medisinske applikasjoner. Det kan ligge flere faktorer i denne oppgaven som har forhindret i å vurdere det fulle potensialet for modellene med DES-algoritmer. DES-algoritmer skiller seg fra de klassiske ML-algoritmene, ved at de benytter dynamisk tilnærming til å velge individuelle modeller som inkluderes i ensemblet for hver prediksjon. Dette er en tilnærming som kan føre til at modellene tilpasser støy i datasettet i større grad, og dermed føre til overtilpasning og prestere dårligere på ny data. Modeller med DES-algoritmer er spesielt sårbar for dette,

når det brukes små datasett. Som presentert tidligere i kapittelet, er dette et tilfelle i denne oppgaven med to datasett som inneholder få prøver og resultater som indikerer at modellene er overtilpasset. Algoritmene kan ha vist større potensial og vært effektivt om datasettene i oppgaven hadde vært noe større og komplekse, slik at modellene hadde vært mer robuste mot støy i dataene.

En annen faktor som kan ha satt påvirkning på modellene, kan ligge i at datasettene som har brukt er ubalansert som nevnt tidligere. Ensemble-læringsalgoritmer er ofte godt egnet for problemer med ubalanserte datasett. Imidlertid, når datasettene som brukes i oppgaven er så lite som de er med få tilgjengelige prøver, kan dette føre til at modellene sliter med å identifisere den underrepresenterte klassen og dermed levere dårligere prediksjonsytelse på valideringsdata. Modeller med DES-algoritmer bruker altså komplekse teknikker, og kan derfor ha utfordringer med å finne riktige mønster og sammenhenger som den kan bruke for å skille to klasser, når modellene har få prøver i datasettene som den kan trenes på.

Det er viktig å merke seg at DESlib-pakken fortsatt er under utvikling med oppgavens utførelse, og det ligger potensialet for at algoritmene i biblioteket har mulighet til å prestere bedre i tiden som kommer. Selv om studiet resulteres i at algoritmene ikke er optimal, er det fortsatt rom for flere undersøkelser og studier for å vurdere hvilke potensiale algoritmer fra DESlib biblioteket kan gi for å utvikle medisinske applikasjoner. Veien videre og områder som kan studeres videre på er diskutert i kapittel 5.9.

5.8.3 Klassiske ML-algoritmer vs. DES-algoritmer

For å sammenligne utførelsen og ytelsen av klassiske maskinlæringsalgoritmer med DES-algoritmer, kan flere faktorer bli tatt i betraktning. Ytelse, nøyaktighet, og generell utførelse av de ulike oppgavene som er testet og presentert i resultatkapittelet kan være noen av disse faktorene. Det bør midlertid bemerkes at i denne oppgaven ble modellene evaluert på tre ulike kriterier: først og fremst deres ytelse på testdata, dernest antallet falske positive (FP) og falske negative (FN) resultater de generer, og til slutt deres prestasjon på treningsdata.

Dynamic Ensemble Selection (DES) er en teknikk innen maskinlæring som innebærer å kombinere flere individuelle modeller for å danne en sterke modell som kan gjøre mer nøyaktige prediksjoner. DES bruker en dynamisk tilnærming til ensemble-læring som involverer en seleksjonsprosess for å bestemme hvilke modeller som skal inkluderes i

ensemblet basert på deres ytelse på tidligere prediksjoner. Dette gjør det mulig å tilpasse ensemblet til endringer i dataene over tid, og dermed forbedre prediksjonsevnen.

En av svakhetene med DES er imidlertid at det krever betydelige ressurser og tid for å bygge og optimalisere en robust ensemble-modell. Basert på resultatene som ble presentert i denne oppgaven, kan man legge merke til at DES-algoritmene bruker mye lenger tid på modelleringen enn de klassiske maskinlæringsalgoritmene. Dette skyldes, ifølge DESlib-pakken, at DES innebærer å teste og velge en rekke forskjellige modellkombinasjoner, og optimalisere parameterne for hver modell. Dette kan være en kompleks og tidkrevende prosess.

På den annen side kan klassiske algoritmer være enklere å implementere og trene enn DES. Disse algoritmene er ofte mindre komplekse og krever mindre tid og ressurser for å bygge en modell. Dette gjør dem til et mer praktisk valg i situasjoner der tids- og ressursbegrensninger er viktige faktorer. En annen faktor som kan påvirke valget mellom DES og klassiske algoritmer, er størrelsen på datasettet. En fellesnevner for begge alternativene er utfordringen de har hatt med begrenset datasettet. Datasett har vært lite innholdsrik på informasjon og bestående av en del manglende verdier. Dette kan ha gitt utfordringer for DES-algoritmene som kan kreve store og komplekse datasett for å oppnå høy ytelse. Hvis datasettet er for lite, kan det være vanskelig eller umulig å få nok variasjon i modellene for å bygge en effektiv DES-modell. På den annen side kan klassiske algoritmer være mer robuste og fungere bedre på mindre datasett.

Generelt sett kan det observeres at modellene som bruker DES-algoritmene og random forest har en tendens til å håndtere falske positive (FP) godt, basert på de presenterte confusion matrix-ene i forbindelse med testresultatene. Modellene i oppgaven skal ha et mål om å ha lavest mulig antall for både FP- og falske negative (FN). FP vil bety at en pasient blir predikert som syk når pasienten ikke syk og FN betyr at en pasient har blitt klassifisert som ikke syk når pasienten faktisk er syk. Imidlertid presterte samtlige modeller med DES-algoritmer og random forest dårlig når det gjelder falske negative (FN). Som tidligere diskutert, kan dette skyldes at datasettet består av flere negative enn positive prøver. Dette kan ha ført til at modellene har tilpasset seg til den negative klassen i større grad. Til tross for at modellene har dårlig ytelse på FN, er det fortsatt positivt at de håndterer FP godt. Dette betyr bare at modellene fokuserer mindre på FP som er positivt. Dette kan ha en samfunnsmessig fordel

ved å redusere muligheten for feil diagnose, feil behandling og økte kostnader (Dresselhaus et al., 2002). Feil diagnose kan for eksempel føre til unødvendige behandlinger og bekymringer for pasienten, samt gi falsk trygghet til helsepersonell.

I sum kan DES være en verdifull teknikk for å forbedre prediksjonsytelsen i store og komplekse datasett. Imidlertid krever DES betydelige ressurser og tid, og i noen situasjoner kan det være mer praktisk å bruke klassiske algoritmer som random forest eller logistisk regresjon. Disse algoritmene kan gi like gode eller bedre resultater når tidsbruk er et kriterium, eller når datasettet er for lite til å støtte en effektiv DES-modell. Det er viktig å påpeke at DES-algoritmen er fortsatt under utvikling. De klassiske algoritmene kan dermed være generelt sett bedre egnet og oppnå bedre resultater i de fleste klassifiseringsproblemer. Samtidig ligger det også potensialet for at DES-algoritmene har mulighet til å prestere bedre i tiden som kommer.

5.9 Veien videre

Selv om denne oppgaven har gitt verdifulle innsikt i bruk av kunstig intelligens for å utvikle mediniske applikasjoner, er det fremdeles flere utfordringer som gjenstår å løse, samt som kan utforskes og forberedes. Som diskutert tidligere i kapittelet, har datasettene i oppgaven, ikke vært helt optimale. Ettersom de består av et lite antall prøver og kan potensielt ikke representere pasientpopulasjonen til enhver tid. For å trekke en grundig og klar konklusjon ville det vært ideelt å gjennomføre en tilsvarende studie, med datasett bestående av et betydelig større og mer variert datasett. Et større datasett vil også være med å teste og evaluere det fulle potensialet som ligger i modellene som bruker DES-algortimer.

Samarbeid mellom sykehus og legekontorer i Norge for å opprette en felles database kan bidra til å samle et større og mer relevant datasett for kreftrelatert data. Denne løsningen kan sikre et komplett datasett med representasjon av et mangfold av mennesker med ulik bakgrunn, alder, kjønn, geografisk lokalisering, og andre relevante variabler. Løsningen krever høy grad av sikkerhet og personvern for å overholde gjeldende helselover og regler i Norge, slik at all data er trygg og beskyttet. I den norske helselovgningen kan man finne et regelverk for tilgang og bruk av helseopplysninger, men det nåværende regelverket tar lite hensyn til særskilte egenskaper ved maskinlæringsprosjekter (Prop. 112 L (2020– 2021)). Det kan dermed være et behov for noe tilpasning og justering. Et samarbeid med den norske

helseetaten kan være en ideell løsning, da det vil muliggjøre banebrytende forskning og utvikling av nye og innovative løsninger innen kreftbehandling.

I en diskusjon med kliniske eksperten, Hanne Osenes-Ringen, og biveilederen, Natalia Knust, fra Helsedirektoratet ble det klart at fra et medisinsk ståsted er det viktigere for dem at resultatene er korrekte og pålitelige for å kunne ha tillit til resultatene. Å arbeide med større datasett vil øke påliteligheten av modellene og representere pasientgruppen i større grad. Dette vil øke tilliten hos ekspertene og helsepersonell. Dette vil igjen føre til økt interesse for videre forskning innen fagfeltet som kan muliggjøre utviklingen av beslutningsverktøy, som kan hjelpe helsepersonell i å ta mer informerte beslutninger om kreftbehandling.

På lang sikt kan en forbedret datasettet være av nytte for å optimalisere prognosen og overlevelsesraten til kreftpasienter ved å tilby mer presise og pålitelige prediksjoner. Dette kan videre bruks som et grunnlag for utvikling av prediksjonsmodeller som kan identifisere effektiv kreftbehandling for en pasient. Hvis modellene er validert og resultatene er til å stole på, kan helsepersonell benytte seg av disse modellene som beslutningstøtte ved behandling av nye kreftpasienter.

På enda lengre sikt vil det være en mulighet å prøve og kombinere maskinlæring med skyteknologi. Skybaserte tjensteplattformer som Google Cloud Platform (GCP) tilbyr et bredt spekter av tjenester innen databehandling, database, kunstig intelligens og mye mer (Google, u.å). Med potensialet til å lagre all dataen i GCP, da den har kapabel til å håndtere store mengder med data, kan den brukes til å trene modeller i GCP i reell sanntid. Modellene vil forbedre seg kontinuerlig med ny lagret data og tilpasse seg pasientenes endrede tilstand over tid. Videre kan GCP brukes til å utvikle et dashbord som kan fungere som en felles beslutningsplattform for norske sykehus og legesenter. Et velkjent prosjekt som benytter cloud- og ML-teknologi er McLaren-prosjektet, som ble gjennomført i samarbeid med Deloitte (Deloitte., u.å). Prosjektet hadde som mål å utvikle en skybasert løsning som simulerer og predikerer ulike scenarier i et løp, og gir anbefalinger til føreren om hvilke valg som bør tas i sanntid.

I delene som har blitt fulgt, har det blitt diskutert hvordan studien kan utvides og bygges videre når et større datasett er tilgjengelig. Det er imidlertid verdt å merke seg at, selv med de datasettene som ble brukt i oppgaven er det fortsatt en rekke muligheter for videre

forskning innenfor samme tema. Det kan være interessant å presentere matrisene i kapitlene 4.2.3, 4.3.3, 4.5.3 og 4.6.3 med predikert sannsynlighet (*eng. predicted class probabilities*) i stedet for predikert klasse. Matrisene i kapitlene viser antall ganger prøver har blitt klassifisert av tusen mulige på testdata, og mange av de tusen prediksjonene kan ha vært nær grenseverdien på 0.5. Dette indikerer usikkerhet i hvilken klasse prøven tilhører. Å generere matriser som viser gjennomsnittet av predikert sannsynlighet, vil gi en bedre forståelse av usikkerheten i modellen og åpne for videre undersøkelser.

I DESlib pakken finnes det fortsatt mange andre algoritmer som ikke er utprøvd i denne oppgaven. Det kan hende at en av disse kan gjøre det bedre med de samme datasettene enn hvordan modellene i denne oppgaven har prestert. DESlib-biblioteket er fortsatt i utviklingsfasen, som tidligere diskutert, og det ligger mulighet for at algortimene kan bli enda bedre enn hvordan de har prestert under gjennomførelsen av denne oppgaven.

Kapittel 6

Konklusjon

I denne oppgaven har hovedfokuset vært å besvare problemstillingen *«Hvordan predikerer Dynamic Ensemble Selection på helsedata om kreftpasienter i forhold til andre klassiske maskinlæringsalgoritmer, og hvilke av de vurderte algoritmene kan brukes for å støtte beslutningstaking innen kreftmedisinsk sammenheng?»* For å utforske problemstillingen har oppgaven formulert tre forskningsspørsmål som tar sikte på å oppnå en grundigere forståelse av problemstillingen og bidra til å avdekke relevante data og innsikt:

1. I hvilken grad skiller nøyaktigheten for modellene som bruker DES eller klassiske algoritmer seg fra hverandre?
2. Hvilke muligheter og utfordringer er til stede ved benyttelse av DES-algoritmer?
3. Kan bruken av MCDA-analyse tilrettelegge implementering av maskinlæringsmetoder for å støtte medisinsk beslutningstaking?

Studien har utviklet modeller som benytter fem klassiske maskinlæringsalgoritmer og tre DES-algoritmer for å avdekke resultater som kan besvare forskningsspørsmålene. For å trekke klare konklusjoner, ble to forskjellige krefttyper med hvert sitt datasett benyttet. Modellene ble trent på datasettene fra Oslo Universitetssykehus som omhandlet kolorektal kreft med responsene generell overlevelse (OS) og progresjonsfri overlevelse (PFS), og hode- og halskreft med responsene OS og sykdomsfri overlevelse (DFS). Deretter ble modellene for hode- og halskreft også testet på et eksternt datasett fra MAASTRO-klinikken i Nederland.

For kolorektal datasettet var GaussianNB-modellen den beste for OS-responsvariabelen, med MCC og F1-score på henholdsvis 0.48 og 0.63 for testdata under kryssvalidering med tusen repetisjoner. For modellene med PFS for kolorektal kreft oppnådde modellen med QDA den beste ytelsen for MCC med en nøyaktighetsytelse på 0.48. Selv om oppgaven har tatt for seg fem ulike nøyaktighetsmålinger, representerer MCC målingene en strengere og bedre evaluering av modellene, da den tar hensyn til ubalanse i datasettene. Derfor har MCC blitt brukt som en av de viktigste faktorene til vurdering av modellene.

For modellene med OS for hode- og halskreft, oppnådde random forest-modellen best ytelse med en MCC-nøyaktighet på 0.48. Den samme algoritmen oppnådde en MCC på 0.30 på det eksterne datasettet fra MAASTRO klinikken. For modellene med DFS for hode- og halskreft var random forest den beste algoritmen med en MCC-score på 0.40, og en nøyaktighet på 0.37 på det eksterne datasettet.

Resultatene indikerer at modellene som bruker DES ikke oppnår høyere prestasjoner enn modellene med klassiske algoritmer. DES-algoritmer bruker en dynamisk tilnærming for å velge individuelle modeller i sitt ensemble for hver prediksjon. Dette kan være en utfordring for datasett med få prøver, som er mer sårbare for støy og kan føre til overtilpasning av modellene på treningsdata. Dette har vært tilfelle for alle modellene som bruker DES-algoritmer, og de har oppnådd en nøyaktighetsytelse på 1 for treningsdata. Resultatene viser også at modellene som bruker DES-algoritmer har betydelig lengre modelleringstid enn klassiske modeller. I tillegg har studien avdekket utfordringer knyttet til implementering av hyperparameteroptimalisering, som kan være en tidkrevende prosess. Det er verdt å merke seg at DESlib-pakken fortsatt er under utvikling, og at det er potensialet for at algoritmene i biblioteket kan forbedres i tiden som kommer, noe som kan føre til kortere modelleringstid.

Basert på MCDA-analysen kan man også konkludere med at alternativ 1 som omhandler de klassiske algoritmene utfører det relativt bedre enn DES-algoritmene, og spesielt med en kombinasjon av den eksisterende beslutningsprosessen. Om datasettet er av høyere kvalitet, omfatter flere data og/eller tar for seg andre type data, kan alternativ 2 om DES-algoritmene være et bedre valg. I tillegg bør det bemerkes at begge alternativene kun tar for seg et begrenset antall algoritmer. Derfor kan valg av andre evalueringskriterier og algoritmer føre til ulike resultater og konklusjoner, spesielt i en MCDA-analyse.

Fra de diskusjonene som har blitt gjort, kan materialene og metode som er benyttet kritiseres på noen områder. De har visse svakheter og unøyaktigheter som bør tas i betraktning når resultatene avveies. Til tross for dette, har metoden vist å være hensiktsmessig og vært forenlig med formålet den har hatt. Med økt kunnskap kan man muligens overvinne tradisjonelle utfordringer ved hjelp av ny teknologi, og samtidig realisere potensialet for å forbedre eksisterende beslutningsprosesser. Til slutt er det viktig å påpeke at bruken av maskinlæring i denne oppgaven er ment som et supplerende beslutningsstøtteverktøy innenfor kreftbehandling, og ikke som en erstatning for nåværende praksis.

Referanseliste

Abdi, H. & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2 (4): 433-459.

Agrawal, T. (2021). *Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient*. Bangalore: Springer.

Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. (2019). *Optuna: A next-generation hyperparameter optimization framework*. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, In KDD.

Alpaydin, E. (2020). *Introduction to machine learning*: MIT press.

Anaconda. (2020). Anaconda Software Distribution. *Anaconda Documentation*, Vers. 2-2.4.0.

Atewan, T. A. (2022). *Time Series Analysis with Python Cookbook*: Packt Publishing.

Auffarth, B. (2020). *Artificial Intelligence with Python Cookbook: Proven recipes for applying AI algorithms and deep learning techniques using TensorFlow 2. x and PyTorch 1.6*: Packt Publishing Ltd.

Baltussen, R., Youngkong, S., Paolucci, F. & Niessen, L. (2010). Multi-criteria decision analysis to prioritize health interventions: Capitalizing on first experiences. *Health policy*, 96(3): 262-264.

Belorkar, A., Guntuku, S. C., Hora, S. & Kumar, A. (2020). *Interactive Data Visualization with Python*: Packt Publishing Ltd.

Blumberg, B., Cooper, D. & Schindler, P. (2014). *EBOOK: Business Research Methods*: McGraw Hill.

Boyd, D. & Crawford, K. (2012). Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society*, 15.

Bruce, P., Bruce, A. & Gedeck, P. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python*: O'Reilly Media.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS Inc.*

Chen, D. Y. (2022). *Pandas for everyone: Python data analysis, 2nd Edition*. 2 utg.: Addison-Wesley Professional.

Chicco, D. & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21.

Cohen, M. X. (2022). *Practical Linear Algebra for Data Science: From Core Concepts to Applications Using Python*: O'Reilly Media, Inc.

Cruz, R. M. O., Hafemann, L. G., Sabourin, R. & Cavalcanti, G. D. C. (2020). DESlib: A Dynamic ensemble selection library in Python. *Journal of Machine Learning Research*, 21: 1-5.

Dahouda, M. K. & Joe, I. (2021). *A deep-learned embedding technique for categorical features encoding*. IEEE Access. Tilgjengelig fra: https://ieeexplore.ieee.org/abstract/document/9512057 (lest 15.03.22).

Dalland, O. (2000). *Metode og oppgaveskriving for studenter*: Gyldendal.

Deloitte. (u.å). *How Formula 1 technology is shaking up manufacturing sector*. Deloitte: Deloitte. Tilgjengelig fra: https://www2.deloitte.com/uk/en/pages/impact-report-2019/stories/supplycycle.html (lest 01.05.23).

Deshpande, P. C., Skaar, C., Brattebø, H. & Fet, A. M. (2020). Multi-criteria decision analysis (MCDA) method for assessing the sustainability of end-of-life alternatives for waste plastics: A case study of Norway. *Science of the Total Environment*, 719.

Dougherty, J. & Ilyankou, I. (2021). *Hands-on data visualization: Interactive Storytelling From Spreadsheets to Code*: O'Reilly Media, Inc.

drawio. (u.å). *Security-first diagramming for teams.* drawio. Tilgjengelig fra: https://www.drawio.com/ (lest 03.03.23).

Dresselhaus, T. R., Luck, J. & Peabody, J. W. (2002). The ethical problem of false positives: a prospective evaluation of physician reporting in the medical record. *Journal of medical ethics*, 28 (5): 291-294.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B. & Tabona, O. (2021). A survey on missing data in machine learning. *J Big Data*, 8 (1): 140.

Engesæth, L. J. S. (2022). *Predicting patient outcome using radioclinical features selected with RENT for patients with colorectal cancer*: Norwegian University of Life Sciences, Ås. Tilgjengelig fra: https://hdl.handle.net/11250/3036071 (lest 06.03.23).

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2): 179-188.

Folkehelseinstituttet. (2020). *Kreft i tykktarm og endetarm*. Tilgjengelig fra: https://www.helsebiblioteket.no/innhold/artikler/pasientinformasjon/kreft-i-tykktarm-og-endetarm (lest 06.02.23).

Fosse, E., Vallevik, V. B. & Zaka, A. (2020). *Bruk av helseopplysninger for å lette samarbeid, læring og bruk av kunstig intelligens i helse- og omsorgstjenesten*: Helse- og omsorgsdepartementet. Tilgjengelig fra: https://www.regjeringen.no/no/dokumenter/bruk-av-helseopplysninger-for-a-lette-samarbeid-laring-og-bruk-av-kunstig-intelligens/id2740599/?expand=horingsnotater (lest 11.02.23).

Foxwell, H. J. (2020). *Creating Good Data: A Guide to Dataset Structure and Data Representation*: Apress.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*: O'Reilly Media, Inc.

Google. (u.å). *Google Cloud documentation*: Google. Tilgjengelig fra: https://cloud.google.com/docs (lest 01.05.23).

Google Colaboratory. (u.å). *Velkommen til Colab!* Tilgjengelig fra: https://colab.research.google.com/#scrollTo=Nma_JWh-W-IF.

Grimsrud, T. K., Jakobsen, E., Johannesen, T. B., Larsen, I. K., Larønningen, S., Møller, B., Nygård, M., Robsahm, T. E., Seglem, A. H. & Aagnes, B. (2021). *Kreft i Norge - hva sier tallene?*: Kreftregisteret. Tilgjengelig fra: https://www.kreftregisteret.no/globalassets/cancer-in-norway/2020/cin-2020-special_issue.pdf (lest 15.01.23).

Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S. & Smith, N. J. (2020). Array programming with NumPy. *Nature*, 585 (7825): 357-362. doi: 10.1038/s41586-020-2649-2.

Helsedirektoratet. (2020). *Hode-/halskreft – handlingsprogram*: Helsedirektoratet. Tilgjengelig fra: https://www.helsedirektoratet.no/retningslinjer/hode-hals-kreft-handlingsprogram#apiUrl (lest 22.04.23).

Helsedirektoratet. (2023). *Sykehusopphold - ventetid*: Helsedirektoratet. Tilgjengelig fra: https://www.helsedirektoratet.no/statistikk/kvalitetsindikatorer/sykehusopphold/gjennomsnittlig-ventetid-i-somatisk-helsetjeneste (lest 07.05.23).

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(3): 90-95. doi: 10.1109/MCSE.2007.55.

Jafari, R. (2022). *Hands-On Data Preprocessing in Python: Learn how to effectively prepare data for successful data analytics*: Packt Publishing Ltd.

Khan, S. I. & Hoque, A. (2020). SICE: an improved missing data imputation technique. *J Big Data*, 7 (1): 37.

Kreftforeningen. (2023). *Hode- og halskreft*: Helsenorge. Tilgjengelig fra: https://www.helsenorge.no/sykdom/kreft/hode-og-halskreft/ (lest 23.08.23).

Kreftforeningen. (u.å). *Behandling av kreft*: Kreftforeningen. Tilgjengelig fra: https://kreftforeningen.no/om-kreft/behandling/ (lest 01.04.23).

Kreftregisteret. (2021). *Hode- og halskreft*: Kreftforeningen. Tilgjengelig fra: https://kreftforeningen.no/om-kreft/kreftformer/hode-og-halskreft/ (lest 08.02.23).

Kreftregisteret. (2022). *Tykk- og endetarmskreft*: Kreftregisteret. Tilgjengelig fra: https://www.kreftregisteret.no/Temasider/kreftformer/Tykk--og-endetarmskreft/ (lest 07.11.22).

Kumar, A. (2016). *Learning predictive analytics with Python*: Packt Publishing Ltd.

Kumar, A. & Jain, M. (2020). Ensemble Learning for AI Developers: Learn Bagging Stacking, and Boosting Methods with Use Cases. *Apress*.

Larsen, I. K., Ursin, G., Weiderpass, E. & Nystad, E. (2014). *Kreft i Norge*: Folkehelseinstituttet. Tilgjengelig fra: https://www.fhi.no/nettpub/hin/ikke-smittsomme/kreft/ (lest 15.01.23).

Larsen, I. K., Grimsrud, T. K., Jakobsen, E., Johannesen, T. B., Larønningen, S., Møller, B., Nygård, M., Robsahm, T. E., Seglem, A. H. & Aagnes, B. (2022). *Cancer in Norway 2021-*

Cancer incidence, mortality, survival and prevalence in Norway. Cancer Registry of Norway: Cancer Registry of Norway. Tilgjengelig fra: https://www.kreftregisteret.no/globalassets/cancer-in-norway/2021/cin_report.pdf (lest 15.01.23).

Larsen, I. K., Møller, B., Johannesen, T. B., Robsahm, T. E., Grimsrud, T. K., Larønningen, S., Seglem, A. H., Hestad, J. J., akobsen, E. J. & Ursin, G. (2023). *CancerinNorway2022-Cancerincidence,mortality,survivalandprevalencein Norway*: CancerRegistryofNorway (lest 28.01.23).

Laudon, K. C. & Traver, C. G. (2018). *E-commerce 2018: Business, technology, society* 14 utg.: Pearson

McKinney, W. (2010). *Data structures for statistical computing in python*. Proceedings of the 9th Python in Science Conference.

Meld.St. nr.34 (2015-2016). *Verdier i pasientens helsetjeneste. Melding om prioritering*: Helse- og omsorgsdepartementet. Tilgjengelig fra: https://www.regjeringen.no/no/dokumenter/meld.-st.-34-20152016/id2502758/ (lest 30.03.23).

Midtfjord, A. D. (2018). *Prediksjon av behandlingsutfall for hode-og halskreft ved bruk av radiomics av PET/CT-bilder*: Norwegian University of Life Sciences, Ås. Tilgjengelig fra: http://hdl.handle.net/11250/2570202 (lest 15.03.23).

Moan, J. M., Amdal, C. D., Malinen, E., Svestad, J. G., Bogsrud, T. V. & Dale, E. (2019). The prognostic role of 18F-fluorodeoxyglucose PET in head and neck cancer depends on HPV status. *Radiotherapy and Oncology*, 140: 54-61.

Mooney, C. Z. (1997). *Monte carlo simulation*: Sage.

NOU 1997: 20. *Omsorg og kunnskap!— Norsk kreftplan*: Sosial- og helsedepartement Tilgjengelig fra: https://www.regjeringen.no/no/dokumenter/nou-1997-20/id141003/?ch=4 (lest 02.02.23).

NOU 2023: 4. *Tid for handling. Personellet i en bærekraftig helse- og omsorgstjeneste*: Regjeringen, helse- og omsorgstjeneste. Tilgjengelig fra: https://www.regjeringen.no/no/dokumenter/nou-2023-4/id2961552/?ch=13 (lest 14.02.23).

Nygård, M. (u.å). *Alkohol og tobakk hovedårsak til hode og halskreft*: Kreftregisteret. Tilgjengelig fra: https://www.kreftregisteret.no/Generelt/Nyheter/Alkohol-og-tobakk-og-ikke-HPV-smitte-er-hovedarsak-til-hode-og-halskreft/ (lest 08.02.23).

Oliveira, M. D., Mataloto, I. & Kanavos, P. (2019). Multi-criteria decision analysis for health technology assessment: addressing methodological challenges to improve the state of the art. *The European Journal of Health Economics*, 20(6): 891-918. doi: 10.1007/s10198-019-01052-3.

Osnes-Ringen, H. (2023). *Kommunikasjon med klinisk ekspert, spesialist i ortopedisk kirurgi og overlege Hanne Osnes-Ringen i Helsedirektoratet (20.04.23)*.

OUS. (2023). *Tykk- og endetarmskreft*: Oslo universitetssykehus, OUS. Tilgjengelig fra: https://oslo-universitetssykehus.no/behandlinger/tykk-og-endetarmskreft (lest 06.01.23).

Owen, L. (2022). *Hyperparameter Tuning with Python: Boost your machine learning model's performance via hyperparameter tuning*: Packt Publishing Ltd.

Pandala, S. R. (2022). *Lazy Predict*. Tilgjengelig fra: https://github.com/shankarpandala/lazypredict (lest 12.02.23).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of machine Learning research*, 12: 2825-2830.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. & Dubourg, V. (u.å). *Linear and Quadratic Discriminant Analysis*: Scikit-learn. Tilgjengelig fra: https://scikit-learn.org/stable/modules/lda_qda.html (lest 12.02.23).

Per, J. & Claes, W. (2006). *Benchmarking k-nearest neighbour imputation with homogeneous Likert data*: Empir Software Eng. Tilgjengelig fra: https://link.springer.com/article/10.1007/s10664-006-9001-9 (lest 24.03.23).

Plotnikova, V., Dumas, M. & Milani, F. (2020). Adaptations of data mining methodologies: A systematic literature review. *PeerJ Computer Science*, 6: e267. doi: 10.7717/peerj-cs.267.

Prop. 112 L (2020– 2021). *Endringer i helsepersonelloven og pasientjournalloven (bruk av helseopplysninger for å lette samarbeid, læring og bruk av kunstig intelligens i helse- og omsorgstjenesten mv.)*: Helse- og omsorgsdepartementet.

Qayyum, A., Qadir, J., Bilal, M. & Al-Fuqaha, A. (2020). Secure and robust machine learning for healthcare: A survey. *IEEE Reviews in Biomedical Engineering*. doi: 10.1109/RBME.2020.3013489.

Raschka, S. & Mirjalili, V. (2019). *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*: Packt Publishing Ltd.

Rasheed, K., Qayyum, A., Ghaly, M., Al-Fuqaha, A., Razi, A. & Qadir, J. (2022). Explainable, trustworthy, and ethical machine learning for healthcare: A survey. *Computers in Biology and Medicine*.

Reinertsen, K. M. (2014). *Konseptvalg med flermålsanalyse som verktøy*. Norwegian University of Life Sciences, Ås. Tilgjengelig fra: http://hdl.handle.net/11250/218114 (lest 28.03.23).

Rolstadås, A., Olsson, N., Johansen, A. & Langlo, J. (2020). Praktisk prosjektledelse: fra idé til gevinst (2. utg.). *Fagbokforlaget*.

Røe, K. (2018). *The OxyTarget Study – Merging Functional MRI and Circulating Biomarkers for Biopsy-Free Detection of Chemoradiotherapy Resistant Rectal Cancer*. Tilgjengelig fra: https://forskningsprosjekter.ihelse.net/prosjekt/2013002 (lest 20.01.2023).

Saleh, H. (2018). *Machine Learning Fundamentals: Use Python and scikit-learn to get up and running with the hottest developments in machine learning*: Packt Publishing Ltd.

Siekelova, A., Podhorska, I. & Imppola, J. J. (2021). *Analytic hierarchy process in multiple–criteria decision–making: a model example*. SHS web of conferences (90): EDP Sciences.

Souza, M. A., Cavalcanti, G. D. C., Cruz, R. M. O. & Sabourin, R. (2018). Online local pool generation for dynamic classifier selection: an extended version. *arXiv*.

Syse, A. (2014). *Sosiale helseforskjeller i Norge*: Folkehelseinstituttet Tilgjengelig fra: https://www.fhi.no/nettpub/hin/samfunn/sosiale-helseforskjeller/ (lest 05.05.23).

Tomic, O., Graff, T., Liland, K. H. & Næs, T. (2019a). hoggorm: a python library for explorative multivariate statistics. *The Journal of Open Source Software*, 4(39). doi: 10.21105/joss.00980.

Tomic, O., Fehlner, A., Liland, K. H., Graff, T. & Rimal, R. (2019b). *hoggormplot*. Tilgjengelig fra: https://github.com/olivertomic/hoggormPlot (lest 19.01.23).

Van Rossum, G. & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*: O'Reilly Media, Inc.

Walker, M. (2020). *Python Data Cleaning Cookbook: Modern techniques and Python tools to detect and remove dirty data and extract key insights*: Packt Publishing Ltd.

Wang, G., Hao, J., Ma, J. & Jiang, H. (2011). *A comparative assessment of ensemble learning for credit scoring*: Expert systems with applications. Tilgjengelig fra: https://www.sciencedirect.com/science/article/abs/pii/S095741741000552X?casa_token=QsioUaKqs8gAAAAA:cX0tYvqtKQcSUxIr7c7oqhZoyuifET2PlTIKDzSmYImVlNJTuxoiq_YF4pnycKLawmFS1QUDEDU (lest 28.04.23).

Waskom, M., Botvinnik, O., Ostblom, J., Gelbart, M., Lukauskas, S., Hobson, P., Gemperline, D. C., Augspurger, T., Halchenko, Y. & Cole, J. B. (2017). mwaskom/seaborn: v0.8.1 (September 2017). *Zenodo*. doi: 10.5281/zenodo.883859.

Wilke, C. O. (2019). *Fundamentals of data visualization: a primer on making informative and compelling figures*: O'Reilly Media.

Yang, L. (2018). *Encoding Categorical Features*. Tilgjengelig fra: https://towardsdatascience.com/encoding-categorical-features-21a2651a065c (lest 24.01.23).

Zhao, Y., Nasrullah, Z. & Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20: 1-7.

Vedlegg A

Optimalisering av hyperparametere

I dette vedlegget presenteres resultatene fra Optuna som tar for seg optimalisering av hyperparametere. Alt av optimaliseringer har blitt modellert med 200 trials i Optuna. Alle modeller har blitt optimalisert utenom GuassianNB. Basert på at GuassianNB har et begrenset antall hyperparametere som kan justeres og ikke alltid gir betydelige forbedringer i ytelsen, ble den utelatt fra modellene.

A.1 Parametere

Tabell A.1 presenterer hvilke parametere som har blitt optimalisert med hjelp av Optuna, som er et optimaliseringsverktøy. Tabellen inneholder oversikt over algoritmene som parameterne tilhører, beskrivelse av parameterne, samt de ulike verdiene til hyperparameterne som har blitt undersøkt.

Tabell A.1: Oversikt over parameternavn, optimaliseringsverdier.

Parameternavn	Algoritmer	Beskrivelse	Hyperparameterverdier
n_estimators	Random forest KNORA-E DES-P META-DES MCB OLA	Antall bestemmelsestrær som skal brukes for modellen. Jo flere desto kompleks blir modellen	Heltall mellom 1 og 210
max_depth	Random forest	Kontrollerer dybden av bestemmelsestrærne. Bestemmer altså antall nivåer trærne kan ha	[None,2,4,6,8,10]
Criterion	Random forest	Måler kvaliteten på beslutningen gitt av et nivå på et tre	[gini, entropy]
gamma	Support Vector Classification	Styrer beslutningsgrensens form og mengde bøyning	[scale, auto, 0.1, 0.3, 0.5, 0.7]
c	Logistic Regression Support Vector Classification	Setter styrken på reguleringen	Heltall mellom 1 og 100
penalty	Logistic Regression	Regulering for å unngå overtilpasning	[l1, l2]

solver	Logistic Regression Support Vector Classification	Valg av numerisk algoritme	[liblinear, saga]
kernel	Support Vector Classification	Kjernefunksjon som skal brukes i algoritmen	[linear, poly, rbf, sigmoid]
reg_param	Quadratic Discriminant Analysis	Mengde regularisering som brukes	Desimaltall mellom 0 og 1
n_neighbors	K-Neighbors Classifier	Antall nærmeste naboer	Heltall mellom 1 og 10
algorithm	K-Neighbors Classifier	Algoritme for å finne nærmeste nabo	[auto, ball_tree, kd_tree, kd_tree]
weights	K-Neighbors Classifier	Vekt funksjon som brukes til beregning ved klassifisering	[uniform, distance]
leaf_size	K-Neighbors Classifier	Hvor stor bladene skal være for å finne nærmeste nabo	Heltall mellom 10 og 50
metric	Nearest centroid	Hvilken metrikk som brukes for å estimere avstand	[euclidean, manhattan]
pool_classifiers	KNORA-E KNORA-U DES-P META-DES MCB OLA	En liste med klassifikatorer. Dette som bygger ensemblet	[Random forest, BaggingClassifier, [Random forest , BaggingClassifier]]
k	KNORA-E KNORA-U DES-P META-DES MCB OLA	Antall nærmeste naboer til å estimere kompetansen til baseklassifikatorene	Heltall mellom 2 og 15
knn_metric	KNORA-E KNORA-U DES-P META-DES MCB OLA	Hvilken metrikk som brukes av KNN for å estimere avstand. NB: mahalanobis støttet ikke for hode- og hals	[minkowski, mahalanobis]
vote	KNORA-E KNORA-U DES-P META-DES	Hvordan modellprediksjoner kan kombineres for endelige prediksjoner	[soft, hard]

meta_classifiers	META-DES	Sekundær klassifikator som kombinerer resultatene fra de ulike pool klassifikatorene	[Random forest, GaussianNB, Logistic regression, Gradient boosting classifier]

A.2 Parameterkombinasjon av klassiske algoritmer

Tabellene A.2 - A.4 viser de beste hyperparameterkombinasjonene over de klassiske algoritmene: random forest, logistisk regresjon, QDA og nearest centroid.

Tabell A.2: Hyperparameterkombinasjoner for random forest.

RANDOM FOREST

Datasett	n_estimators	criterion	Max_depth
Tykktarm (OS)	181	Gini	None
Tykktarm (PFS)	88	Entropy	None
Hode -og Hals (OS)	186	Entropy	4
Hode -og Hals (DFS)	209	Gini	2

Tabell A.3: Hyperparameterkombinasjoner for logistisk regresjon.

Logistisk regresjon

Datasett	C	Penalty	Solver
Tykktarm (OS)	9	L2	Saga
Tykktarm (PFS)	1	L1	Saga
Hode -og Hals (OS)	173	L2	Liblinear
Hode -og Hals (DFS)	1	L1	Saga

Tabell A.4: Hyperparameterkombinasjoner for QDA og nearest centroid.

QUADRATIC DISCRIMINANT ANALYSIS (QDA)	
Datasett	**Reg_param**
Tykktarm (OS)	0.19390547006449
Tykktarm (PFS)	0.69673348231131
Hode -og Hals (OS)	0.95366884925301
Hode -og Hals (DFS)	0.98646579256011

NEAREST CENTROID	
Datasett	**Reg_param**
Tykktarm (OS)	euclidean
Tykktarm (PFS)	euclidean
Hode -og Hals (OS)	euclidean
Hode -og Hals (DFS)	euclidean

A.3 Parameterkombinasjon av klassiske algoritmer (ekstra)

Tabellene A.5 og A.6 inneholder de optimale hyperparameterkombinasjonene for henholdsvis SVC og KNN. Disse to algoritmene, selv om de ikke ble inkludert i den primære analysen i oppgaven, er presentert som vedlegg for de som ønsker å undersøke ytelsen til disse algoritmene.

Tabell A.5: Hyperparameterkombinasjoner for SVC.

SUPPORT VECTOR CLASSIFICATION			
Datasett	**C**	**gamma**	**Kernel**
Tykktarm (OS)	2	0.1	rbf
Tykktarm (PFS)	40	scale	Rbf
Hode -og Hals (OS)	5	0.1	Sigmoid
Hode -og Hals (DFS)	27	0.3	Linear

Tabell A.6: Hyperparameterkombinasjoner for KNN.

K-Neighbors Classifier				
Datasett	**n_neighbors**	**algorithm**	**Weights**	**leaf_size**
Tykktarm (OS)	5	ball_tree	uniform	37
Tykktarm (PFS)	10	kd_tree	distance	41
Hode -og Hals (OS)	8	kd_tree	uniform	10
Hode -og Hals (DFS)	8	kd_tree	uniform	33

A.4 Parameterkombinasjon av DES-algoritmer

Tabellene A.7 - A.9 viser de beste hyperparameterkombinasjonene over DES-algoritmene i oppgaven: KNORA-E, KNORA-U og DES-P.

Tabell A.7: Hyperparameterkombinasjoner for KNORA-E.

KNORA - E

Datasett	n_estimators	Pool classifiers	K	Voting	knn_metric
Tykktarm (OS)	171	[Random forest, Bagging classifier]	11	soft	minkowski
Tykktarm (PFS)	112	[Random forest, Bagging classifier]	10	soft	minkowski
Hode -og Hals (OS)	76	[Random forest, Bagging classifier]	7	hard	minkowski
Hode -og Hals (DFS)	164	[Random forest, Bagging classifier]	2	hard	minkowski

Tabell A.8: Hyperparameterkombinasjoner for KNORA-U.

KNORA - U

Datasett	n_estimators	Pool classifiers	K	Voting	knn_metric
Tykktarm (OS)	108	Random forest	6	hard	minkowski
Tykktarm (PFS)	146	Random forest	13	soft	minkowski
Hode -og Hals (OS)	179	Random forest	11	hard	minkowski
Hode -og Hals (DFS)	115	Random forest	14	soft	minkowski

Tabell A.9: Hyperparameterkombinasjoner for DES-P.

DES-P

Datasett	n_estimators	Pool classifiers	K	Voting	knn_metric
Tykktarm (OS)	167	Random forest	13	soft	minkowski
Tykktarm (PFS)	122	Random forest	4	hard	minkowski
Hode -og Hals (OS)	91	Random forest	15	hard	minkowski
Hode -og Hals (DFS)	160	Random forest	12	soft	minkowski

A.5 Parameterkombinasjon av DES-algoritmer (ekstra)

Tabellene A.10 - A.12 presenterer de beste hyperparameterkombinasjonene for META-DES, MCB og OLA som er tre andre algoritmer som blitt undersøkt på, men som ikke er inkludert i selve analysen av oppgaven. Dette er lagt til som vedlegg for de som ønsker å se hvordan disse har prestert. Det er verdt å merke seg at MCB og OLA tilhører kategorien DCS (Dynamic Classifier Selection) og ikke DES (Dynamic Ensemble Selection) i DESlib-pakken. Forskjellen mellom disse ligger i måten modellene velger hvilke klassifikatorer som skal brukes. DES velger en undergruppe av klassifikatorer som gir best ytelse for hele test datasettet, og DCS velger en eller flere klassifikatorer som gir best prediksjonsytelse for hver individuell test datasettet (Cruz et al., 2020).

Tabell A.10: Hyperparameterkombinasjoner for META-DES.

META-DES

Datasett	n_estimators	Pool classifiers	Meta classifiers	K	Vote	knn_metric
Tykktarm (OS)	117	Random forest	GuassianNB	9	soft	minkowski
Tykktarm (PFS)	181	Random forest	Logistic regression	7	hard	minkowski
Hode -og Hals (OS)	179	Random forest	Random forest	9	soft	minkowski
Hode -og Hals (DFS)	133	Bagging classifier	Gradient boosting	9	hard	minkowski

Tabell A.11: Hyperparameterkombinasjoner for MCB.

MCB

Datasett	n_estimators	Pool classifiers	K	knn_metric
Tykktarm (OS)	156	[Random forest, Bagging classifier]	2	minkowski
Tykktarm (PFS)	186	[Random forest, Bagging classifier]	12	minkowski
Hode -og Hals (OS)	188	[Random forest, Bagging classifier]	6	minkowski
Hode -og Hals (DFS)	128	[Random forest, Bagging classifier]	12	minkowski

Tabell A.12: Hyperparameterkombinasjoner for OLA.

OLA

Datasett	n_estimators	Pool classifiers	K	knn_metric
Tykktarm (OS)	39	[Random forest, Bagging classifier]	6	minkowski
Tykktarm (PFS)	141	[Random forest, Bagging classifier]	15	minkowski
Hode -og Hals (OS)	145	[Random forest, Bagging classifier]	14	minkowski
Hode -og Hals (DFS)	75	[Random forest, Bagging classifier]	3	minkowski

Vedlegg B

Kolorektal kreft – Forundersøkelser

I vedlegget presenteres resultater fra forundersøkelser av kolorektal datasettet som ikke har blitt inkludert i selve oppgaven.

B.1 Manglende verdier

Som beskrevet i kapittel 3.5.3 ble noen variabler eliminert manuelt basert på vurderinger av hvor liten verdi variabelen hadde for datasettet. Tabell B.1 gir en oversikt over hvilke variabler som ble eliminert og begrunnelsen for elimineringen.

Tabell B.1: En detaljert beskrivelse av de manuelt eliminerte variabler i kolorektal datasettet

Variabeler	**Begrunnelse for eliminering**
Date of inclusion	Dato er en variabel som er unik for hver pasient, og som dermed ikke tilfører vesentlig verdi ved trening av modeller.
Symptoms	En variabel som omfatter symptomer kan være av begrenset nytte da symptomene kan variere fra pasient til pasient. I tillegg består datasett av et begrenset antall prøver (samples), noe som øker risikoen for at kvaliteten på datasettet reduseres når det er mange unike verdier i variabelen.
Date of referral to specialist	Dato er en variabel som er unik for hver pasient, og som dermed ikke tilfører vesentlig verdi ved trening av modeller for denne oppgaven. Imidlertid kan denne variabelen være vikitig i andre sammenhenger som feature engineering
Date primary biopsy	Dato er en variabel som er unik for hver pasient, og som dermed ikke tilfører vesentlig verdi. I forbindelse med feature engineering vil det være av betydning å beholde variabelen. Grunnen er at den kan benyttes til å beregne

	tidsintervaller mellom biopsien til operasjon eller tilbakefall. I denne oppgaven har det ikke blitt utført feature engineering, og variabelen har derfor blitt fjernet.
Biopsy histology ID	Variabelen som angir arkiverte ID-nummer i skyehus er unik for hver pasient og gir dermed ikke nødvendigvis stor verdi ved trening av modeller. Imidlertid kan denne variabelen være vikitig i andre sammenhenger, eksempelvis når det er behov for å spore pasienter.
Date MR w/diagnose	Dato er er unik for hver pasient og vil dermed ikke være viktig ved trening av modeller i denne oppgaven. Ved feature engineering vil det dermot være viktig å beholde variablen. I denne oppgaven har det ikke blitt utført feature engineering, derfor har variablen blitt fjernet.
Date other radiology	Dato er en variabel som er unik for hver pasient, og som dermed ikke tilfører vesentlig verdi. I forbindelse med feature engineering vil det være av betydning å beholde variabelen. Grunnen er at den kan benyttes til å beregne tidsintervaller mellom biopsien til operasjon eller tilbakefall.
Histology reference no.	Variabelen som angir arkiverte ID-nummer i skyehus er unik for hver pasient og gir dermed ikke nødvendigvis stor verdi ved trening av modeller. Imidlertid kan denne variabelen være vikitig i andre sammenhenger, eksempelvis når det er behov for å spore pasienter.
Date surgery	Dato er en variabel som er unik for hver pasient, og som dermed ikke tilfører vesentlig verdi ved trening av modeller.
Other exams	Unike kommentarer og et høyt antall manglende verdier (*NaN-values*).

Date metastatic disease	Et høyt antall manglende verdier (NaN-values) som gjør at datoene har begrenset verdi for trening av modeller.
Date local recurrence	Selv om variabelen inneholder viktig infomasjon, er det en stor mengde manglende verdier som gjør den uegnet for bruk i trening. Totalt sett er det 183 manglende verdier av 192 mulige.
Last registered alive	Den samme informasjon blir presentert i form av OS-event.
Further follow up	Variabelen som inneholder unike kommentarer for hver pasient er ikke relevant, da den ikke gir verdifull informasjon for prediksjonsformål.

Tabell B.2 viser en oversikt over hvilke variabler som har blitt eliminert fra datasettet etter testen som ble diskutert i kapittel 3.5.2 ble utført. Denne oversikten er et utklipp fra en Python-celle. Alle variabler under den røde streken ble eliminert vekk fra datasettet.

Tabell B.2: Viser oversikt over antall manglende verdier i hver variabel. Alt under det rød streken er eliminert fra datasettet

	Variabel	Antall NaN
0	Blood samples at inclusion	1
1	Height (cm)	1
2	Weight (KG)	1
3	BMI	1
4	Biopsy histology ID	1
5	Cancer type	1
6	mrV	1
7	Date surgery	1
8	Adjuvant treatment	1
9	Last registered alive	1
10	Further follow up	1
11	PFS-event	3
12	OS event	3
13	Creatinine (umol/L)	4
14	Hemoglobin (g/dl)	5
15	mrN	6
16	Distance from anus to tumor (rigid rectoscopy)	8
17	MR distance from anus to tumor	10
18	Sodium (mmol/L)	11
19	Potassium (mmol/L)	11
20	Stadium (ACR 2016)	12
21	mrT (TNM ed.7)	13
22	CEA baseline	13
23	CRP baseline	13
24	Leukocytes (10^9/L)	15
25	Blood type	15
26	Bilirubin (umol/L)	20
27	MSI	22
28	ALT (U/L)	24
29	Place Surgery	28
30	ALP (U/L)	28
31	Thrombocytes (10^9/L)	29
32	Type of surgery	30
33	GT (U/L)	30
34	Histology reference no.	31
35	R classification	33
36	Albumin (g/L)	33
37	Histology description	37
38	Mucinous	38
39	p/ypT (TNM ed.7)	42
40	No. Of positive lymph nodes	42
41	No. Of total lymph nodes	42
42	p/ypN (TNM ed. 7)	45
43	Differentiation	49
44	AST (U/L)	50

	Variabel	Antall NaN
45	Smallest distance to CRM (mm)	51
46	Neutrophils (10^9/L)	55
47	Lymphocytes (10^9/L)	55
48	Monocytes (10^9/L)	55
49	Eosinophils (10^9/L)	55
50	Basophils (10^9/L)	55
51	Calcium (total) (mmol/L)	56
52	mrN bowel wall	58
53	Distance tumor-distal margin (mm)	59
54	Distance tumor-proximal margin (mm)	59
55	pNerve	59
56	pV	62
57	LDH (U/L)	75
58	Carbamide	86
59	Time until PFS event	111
60	Date start RT	115
61	RT schedule	115
62	ESR (mm/h)	116
63	Date chemotherapy	117
64	Type of chemotherapy	117
65	Date RT finish	120
66	Date post-CRT MR	120
67	Description post-CRT MR	120
68	TRG (CAP/AJCC)	127
69	TRG (Bateman)	127
70	MORS	129
71	Time until OS event	131
72	mrT4 subtype	132
73	KRAS	144
74	NRAS	144
75	BRAF	144
76	Location metastasis	151
77	Date metastatic disease	154
78	Metastatic organ	154
79	Comment adjuvant treatment	164
80	Included in other study	165
81	Comment	165
82	Comments pathology	166
83	Palliative therapy	170
84	Chloride (mmol/L)	170
85	Other exams	171
86	Other cancer	174
87	Date local recurrence	179
88	Type of local recurrence	180

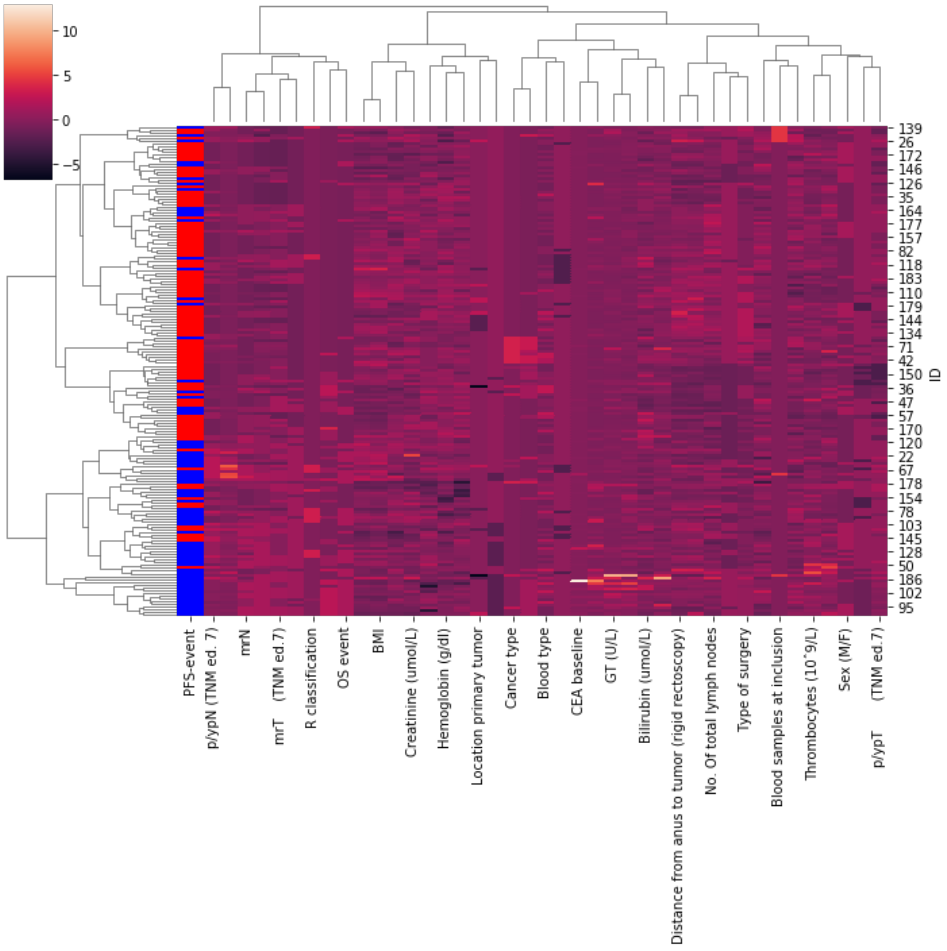
B.2 LabelEncoding

Tabell B.3: Resultater over LabelEncoding på kolorektal datasettet

Variabler	**Kategorier**
Sex (M/F)	Mann: 0 Kvinne: 1
Blood samples at inclusion Suspected metastatic lesions at diagnosis Neoadjuvant CRT (Yes/No) Adjuvant treatment	 Yes: 0 No: 1
Location primary tumor	Anus: 0 Duodenum: 1 Rectosigmoid: 2 Rectum: 3 Sigmoid: 4
Cancer type	Adenocarcinoma: 0 Mucinous Adenocarcinoma: 1 Neuroendocrine carcinoma: 2 Squamous cell carcinoma: 3 Tubular adenoma: 4
Place Surgery	Ahus: 0 DNR: 1 MISSING: 2 Ullevål: 3
Type of surgery	APR: 0 Hartmann: 1 LAR: 2 Laser treatmen: 3 MISSING: 4 Polypectomy: 5 Transanal endoscopic microsurgery: 6 UAR: 7
Histology description	Adenocarcinoma: 0 Fibrose: 1 Fibrosis: 2 MISSING: 3 Serrated adenoma: 4 Signet ring cellular adenoma: 5 Tubular adenoma: 6
Mucinous	Ja: 0 MISSING. 1 Nei: 2
Blood type	0+: 0 AB-: 5 0-: 1 B+: 6 A+: 2 B-: 7 A-': 3 Missing: 8 AB +: 4

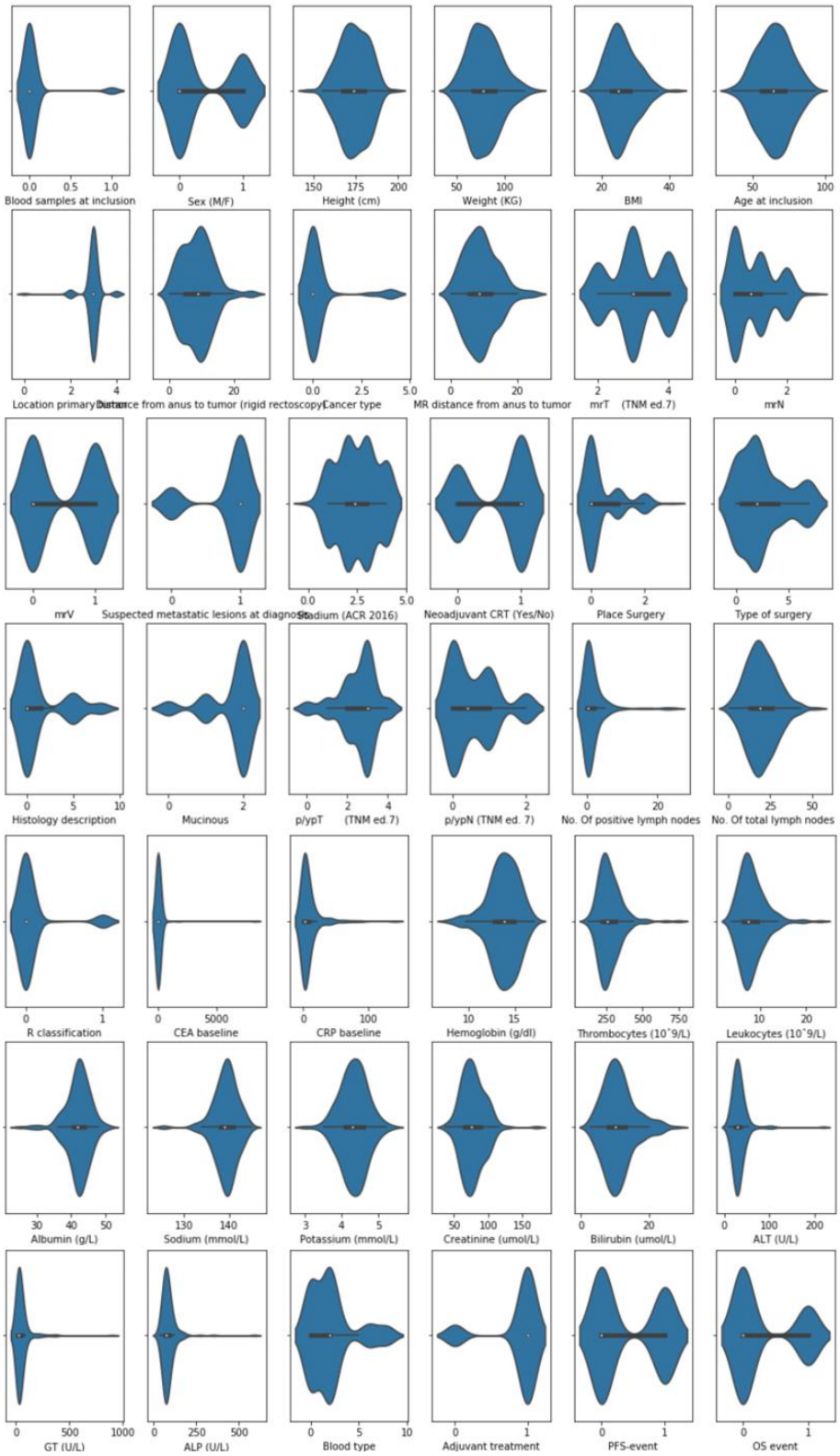
B.3 Clustermap

Figur B.1 presenterer clustermap-plottet av OxyTarget-datasettet. De røde stripene representerer pasienter med negativ PFS-event, som betyr at pasientene er verken død eller fått tilbakefall. De blå stripene representerer pasienter med positiv PFS-event, som betyr at pasientene enten har død eller fått lokalt eller metastatisk tilbakefall.



Figur B.1: Clustermap på kolorektal datasett med PFS

B.4 Violinplot



Figur B.2: Violinplot over variablene i kolorektal datasett

B.5 Ekstremverdier

I kapittel 4.1.2, som omhandler inspeksjon av ekstremverdier, ble det presentert en tabell (B.4) som gir en oversikt over hvilke observasjoner som kunne være mulige ekstremverdier. Fire av observasjonene ble identifisert av både KNN og ECOD, og i tabell 30 i kapittel 4.1.2 ble det presentert ekstremverdiscoren samt en oversikt over mulige årsaker til at disse ble slått ut. Tabellene B.5 og B.6 presenterer tilsvarende informasjon for de gjenværende observasjonene som ble identifisert av henholdsvis KNN og ECOD.

Tabell B.4: En oversikt over hvilke pasienter som er identifisert som ekstremverdier for kolorektal datasettet

Algoritme	**Pasient-ID**
KNN	39 – 40 – 58 – 95 – 113 – 136 – 141 – 173 – 186 – 187
ECOD	26 – 39 – 40 – 84 – 90 – 111 – 136 – 178 – 180 – 186

Tabell B.5: Oversikt over de gjenværende identifiserte ekstremverdier med KNN, kolorektal

PasientID	**KNN - score**	**Mulige årsaker**
58	1530.02	Høy CEA baseline verdi i forhold til normalen
95	254.95	Høy ALT (U/L) verdi i forhold til normalen
113	717.65	Høy CEA baseline i forhold til normalen
141	341.34	Høy Thrombocytes (10^9/L) verdi
173	338.42	Høye verdier for CEA baseline, GT (UL) og ALP(U/L)
187	271.00	Høye verdier for CEA baseline, GT (UL) og ALP(U/L)

Tabell B.6: Oversikt over de gjenværende identifiserte ekstremverdier med ECOD, kolorektal

PasientID	**ECOD - score**	**Mulige årsaker**
26	61.87	Den eldste pasienten (93 år) i datasettet. Lave verdier for CEA baseline, ALT (U/L) og GT (U/L). Høye verdier for Creatinine (umol/L) og MR distance from anus to tumor
84	60.55	Lav verdi for CEA baseline og høy No. Of positive lymph nodes verdi
90	58.57	Overvektig pasient (112 kg) og veldig lav CEA baseline. GT (U/L) og ALP (U/L) har også lavere verdi enn hvor normalen ligger.
111	59.72	Overvektig pasient (120 kg) og lav CEA baseline
178	63.87	Lave verdier for CEA baseline, Albumin (g/L), GT (U/L) og ALP (U/L)
180	59.96	Høy Leukocytes (10ˆ9/L) verdi og veldig høy CEA baseline

Vedlegg C

Kolorektal kreft med OS-event

Vedlegget presenterer diverse resultater som ikke ble presentert i selve oppgaven for modeller med generell overlevelse (OS) som responsverdi på datasettet med kolorektal kreft. Vedlegget inneholder også noen resultater som presenterer mer detaljerte og omfattende oversikter av resultater som har blitt presentert i selve oppgaven.

C.1 Evaluering av modeller på utelatte algoritmer

Tabell C.1 som presenterer testresultatene for modellene som ble utelatt fra kapittel 4.2. På lik linje som modellene som ble presentert i kapittel 4, ble modellene presentert i tabellen under hyperparameteroptimalisert og kjørt med 4-foldet kryssvalidering med tusen repetisjoner.

Tabell C.1: De resterende gjennomsnittlige testresultatene for modellene som ikke ble inkludert i oppgaven, kolorektal - OS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.79 ± 0.05	0.55 ± 0.12	0.86 ± 0.03	0.44 ± 0.14	0.69 ± 0.07
KNN	0.78 ± 0.04	0.43 ± 0.14	0.86 ± 0.03	0.37 ± 0.15	0.64 ± 0.06
META-DES	0.79 ± 0.04	0.51 ± 0.13	0.87 ± 0.03	0.42 ± 0.14	0.67 ± 0.07
MCB	0.79 ± 0.04	0.50 ± 0.13	0.86 ± 0.03	0.41 ± 0.14	0.67 ± 0.07
OLA	0.79 ± 0.04	0.50 ± 0.13	0.87 ± 0.03	0.42 ± 0.14	0.67 ± 0.07

Tabell C.2 presenterer de gjennomsnittlige resultatene over modellytelsen for treningssettet.

Tabell C.2: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, kolorektal - OS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.99 ± 0.01	0.98 ± 0.01	0.99 ± 0.00	0.97 ± 0.02	0.98 ± 0.01
KNN	0.83 ± 0.02	0.60 ± 0.06	0.89 ± 0.01	0.55 ± 0.06	0.72 ± 0.03
META-DES	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
MCB	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
OLA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

C.2 Vanskelige pasienter i testdata, topp 30

Tabell C.3 gir en oversikt over pasient-ID for de 30 mest vanskeligste pasientobservasjonene. Tabellen er sortert i synkende rekkefølge etter antall feilklassifiserte observasjoner for hver pasient, og observasjonene som er felles for de øverste 30 er fremhevet med farger i tabellen. Disse observasjonene ble benyttet for å generere matrisen beskrevet i kapittel 4.2.3.

Tabell C.3: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på testdata, kolorektal – OS.

KNORA-E	KNORA-U	DES-P	Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
57	37	176	69	153	37	52	43
21	29	178	153	51	153	47	153
153	69	29	57	78	57	37	124
78	153	69	37	69	44	57	80
51	89	83	51	132	176	153	37
37	28	153	178	178	67	28	2
178	78	89	176	2	178	156	78
28	57	78	89	89	28	13	57
176	51	162	28	22	51	96	50
89	21	51	29	26	13	2	176
162	178	37	21	52	83	67	156
69	176	21	78	163	45	21	28
29	162	57	162	83	78	176	29
83	83	28	83	71	52	51	79
139	139	24	24	176	89	170	67
24	24	139	139	91	71	71	52
50	50	50	50	28	163	44	1
124	124	22	124	79	22	178	96
22	12	124	12	124	184	124	69
12	22	12	22	156	189	50	71
84	84	26	84	126	156	45	91
163	163	84	26	21	47	64	154
26	26	163	140	80	69	132	12
140	140	140	163	149	64	10	148
138	137	166	138	37	17	22	31
137	166	137	137	67	25	25	178
166	156	138	44	57	74	17	21
44	44	44	166	44	137	80	85
156	138	156	156	29	79	154	64
67	103	67	103	24	149	140	140

C.3 Vanskelige pasienter i treningsdata

Tabell C.4 presenterer en liste over hvilke pasienter som har vært vanskelig å bli klassifisert i riktig kategori for treningsdata. Modellen med GuassianNB hadde flest problemer og utgjorde 19 pasienter totalt. Tabellen er sortert etter størst vanskelighetsgrad. Som presentert i kapittel 4.2.4 hadde KNORA-E, KNORA-U, DES-P og random forest ingen vanskelige pasienter. Det var kun observasjon 1 som var til felles og det var også denne observasjonen som ble presentert i matrisen i kapittel 4.2.4.

Tabell C.4: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på treningsdata, kolorektal – OS.

Logistic regression	QDA	GuassianNB	Nearest centroid
2	10	2	2
1	13	**1**	**1**
178	**1**	10	12
5	-	5	13
190	-	4	10
-	-	185	178
-	-	13	188
-	-	9	176
-	-	184	-
-	-	188	-
-	-	189	-
-	-	6	-
-	-	190	-
-	-	191	-
-	-	182	-
-	-	181	-
-	-	7	-
-	-	192	-
-	-	176	-

C.4 Viktige og mindre viktige variabler

Tabell C.5 og C.6 presenterer en oversikt over viktige og mindre viktige variabler for alle de resterende algoritmene som ikke ble presentert i selve oppgaven. Tabellene viser kun de fem mest betydningsfulle variablene og de fem minst betydningsfulle variablene blant totalt 40 variabler i det prosesserte datasettet som ble brukt til modellering.

Tabell C.5: Oversikt over viktige og mindre viktige variabler på de klassiske modellene, kolorektal – OS.

Random forest			Logistic regression			Nearest centroid		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
MR distance from anus to tumor	0.443106	0.000000	Histology description	0.410243	0.000000	Stadium (ACR 2016)	0.473450	0.000000
No. Of positive lymph nodes	0.44 0523	0.002583	Cancer type	0.409027	0.001216	mrV	0.469740	0.003710
Mucinous	0.440118	0.000405	BMI	0.408848	0.000179	CEA baseline	0.460238	0.009502
ALP (U/L)	0.440099	0.000019	No. Of total lymph nodes	0.408242	0.000606	mrN	0.458057	0.002181
mrN	0.439939	0.000160	Neoadjuvant CRT (Yes/No)	0.405289	0.002953	Sex (M/F)	0.456701	0.001356
….	….	….	….	….		….	….	….
Bilirubin (umol/L)	0.414001	0.000569	CRP baseline	0.376685	0.001152	Potassium mmol/L	0.438989	0.000796
R classification	0.413568	0.000432	p/ypN (TNM ed. 7)	0.375650	0.001035	Blood type	0.438177	0.000812
Suspected metastatic l.d	0.408413	0.005156	Suspected metastatic l.d	0.362552	0.013098	Histology description	0.432213	0.005965
CEA baseline	0.400154	0.008259	Adjuvant treatment	0.333604	0.028948	No. Of positive lymph nodes	0.431506	0.000707
Adjuvant treatment	0.397690	0.002464	R classification	0.286818	0.046786	Adjuvant reatment	0.429744	0.001762

Tabell C.6: Oversikt over viktige og mindre viktige variabler på DES-modellene, kolorektal – OS.

KNORA-U			DES-P		
Variable	**MCC**	**Endring**	**Variable**	**MCC**	**Endring**
ALP (U/L)	0.440920	0.000000	MR distance from anus to tumor	0.440326	0.000000
mrN	0.439738	0.001181	CRP baseline	0.437326	0.003000

Sex (M/F)	0.437417	0.002321	No. Of positive lymph nodes	0.436812	0.000514
mrV	0.434640	0.002777	ALP (U/L)	0.436123	0.000689
Type of surgery	0.434421	0.000219	Thrombocytes (10^9/L)	0.435566	0.000557
….	….	….	….	….	….
p/ypT (TNM ed.7)	0.408379	0.001191	Leukocytes (10ˆ9/L)	0.407071	0.006043
Age at inclusion	0.407820	0.000559	p/ypT (TNM ed.7)	0.406828	0.000243
CEA baseline	0.405499	0.002321	Adjuvant treatment	0.405803	0.001025
Stadium (ACR 2016)	0.396626	0.008873	CEA baseline	0.402991	0.002812
Adjuvant treatment	0.392162	0.004465	Stadium (ACR 2016)	0.401648	0.001343

Vedlegg D

Kolorektal kreft med PFS-event

Vedlegget presenterer diverse resultater som ikke ble presentert i selve oppgaven for modeller med progresjonsfri overlevelse (PFS) som responsverdi på datasettet med kolorektal kreft. Vedlegget inneholder også noen resultater som presenterer mer omfattende oversikter av resultater som har blitt presentert i selve oppgaven.

D.1 Evaluering av modeller på utelatte algoritmer

Tabell D.1 presenterer testresultatene for modellene som ble utelatt fra kapittel 4.3. På lik linje som modellene som ble presenter i kapittel 4, ble modellene presentert i tabell D.1 hyperparameteroptimalisert og kjørt med 4-foldet kryssvalidering med 1000 repetisjoner.

Tabell D.1: De resterende gjennomsnittlige testresultatene for modellene som ikke ble inkludert i oppgaven, kolorektal – PFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.75 ± 0.06	0.66 ± 0.09	0.80 ± 0.05	0.46 ± 0.13	0.73 ± 0.06
KNN	0.73 ± 0.05	0.51 ± 0.11	0.81 ± 0.03	0.41 ± 0.13	0.66 ± 0.06
META-DES	0.75 ± 0.06	0.63 ± 0.09	0.81 ± 0.04	0.47 ± 0.13	0.72 ± 0.06
MCB	0.74 ± 0.06	0.61 ± 0.09	0.80 ± 0.05	0.43 ± 0.13	0.70 ± 0.06
OLA	0.75 ± 0.06	0.63 ± 0.10	0.81 ± 0.04	0.47 ± 0.13	0.72 ± 0.06

Tabell D.2 presenterer de gjennomsnittlige resultatene over modellytelsen for treningssettet. Det er tydelig at modellen har hatt feilfriklassifisering på treningsdatasettet. Dette betyr nødvendigvis ikke at modellene er perfekte. Det kan hende at modellen har overtilpasset og dermed ha dårlig generaliseringsytelse på ny data.

Tabell D.2: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, kolorektal PFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
KNN	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
META-DES	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
MCB	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
OLA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

D.2 Vanskelige pasienter i testdata, topp 30

Tabell D.3 gir en oversikt over pasient-ID for de 30 mest vanskeligste pasientobservasjonene. Tabellen er sortert i synkende rekkefølge etter antall feilklassifiserte observasjoner for hver pasient, og observasjonene som er felles for de øverste 30 er fremhevet med farger i tabellen. Disse observasjonene ble benyttet for å generere matrisen beskrevet i kapittel 4.3.3.

Tabell D.3: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på testdata, kolorektal – PFS.

KNORA-E	KNORA-U	DES-P	Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
5	5	5	43	2	176	17	1
129	164	153	179	86	28	37	79
81	43	89	143	143	129	67	129
153	179	124	120	51	57	47	57
143	89	43	89	153	5	81	37
164	129	37	81	176	37	179	86
29	143	29	69	179	81	96	85
179	153	179	37	120	47	2	67
120	69	120	29	129	153	57	47
124	29	129	164	57	67	124	31
43	37	143	153	81	164	86	5
37	124	69	124	69	143	120	143
89	57	164	5	164	179	153	96
57	120	81	129	5	86	164	81
71	81	57	91	178	51	28	2
91	176	176	176	124	1	5	176
69	91	91	57	67	17	46	43
176	86	86	78	132	124	143	124
47	67	71	86	89	178	51	153
67	51	51	51	37	41	176	164
2	71	85	84	33	13	129	107
86	78	78	71	78	71	21	179
51	47	140	85	126	10	140	28
85	85	84	140	79	120	71	29
13	84	67	52	1	96	85	60
78	140	47	162	39	78	41	170
84	2	2	2	47	148	45	149
46	46	46	178	13	46	132	41
52	52	52	46	29	45	170	145
140	55	55	55	43	69	149	71

D.3 Vanskelige pasienter i treningsdata

Tabell D.4 presenterer en liste over hvilke pasienter som har vært vanskelig å bli klassifisert i riktig kategori med treningsdata. Modellen med GuassianNB hadde flest problemer og utgjorde 20 pasienter totalt. Tabellen er sortert etter størst vanskelighetsgrad. Som presentert i kapittel 4.3.4 hadde KNORA-E, KNORA-U, DES-P og Random forest ingen vanskelige pasienter. Observasjoner som var ti felles er markert med fargekoder, og det var disse observasjonene som ble inkludert i utformingen av matrisen i kapittel 4.3.4.

Tabell D.4: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på treningsdata, kolorektal – PFS.

Logistic regression	QDA	GuassianNB	Nearest centroid
5	5	5	2
2	1	2	1
1	13	1	5
1900	10	10	12
179	179	13	13
10	176	4	179
13	-	185	178
188	-	179	176
176	-	12	188
178	-	190	-
7	-	6	-
189	-	184	-
-	-	7	-
-	-	188	-
-	-	191	-
-	-	183	-
-	-	181	-
-	-	189	-
-	-	176	-
-	-	17	

D.4 Viktige og mindre viktige variabler

Tabellene D.5, D.6 og D.7 presenterer en oversikt over viktige og mindre viktige variabel for alle de modellene som er presentert i kapittel 4.3. Tabellene viser kun de fem mest betydningsfulle variablene og de fem minst betydningsfulle variablene blant totalt 40 variabler i det prosesserte datasettet som ble brukt til modellering.

Tabell D.5: Oversikt over viktige og mindre viktige variabler på DES-modellene, kolorektal – PFS.

KNORA-E			KNORA-U			DESP		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
mrV	0.492996	0.000 000	Adjuvant treatment	0.462596	0.000000	No. Of positive lymph nodes	0.495879	0.000000
ALP (U/L)	0.492834	0.000162	Neoadjuvant CRT (Yes/No)	0.459036	0.003560	ALP (U/L)	0.488631	0.007248
Neoadjuvant CRT (Yes/No)	0.490513	0.002320	No. Of positive lymph nodes	0.456770	0.002266	Bilirubin (umol/L)	0.487330	0.001301
Mucinous	0.488114	0.002400	Location primary tumor	0.455356	0.001414	Neoadjuvant CRT (Yes/No)	0.485496	0.001834
Creatinine (umol/L)	0.486742	0.001371	Age at inclusion	0.455251	0.000106	Mucinous	0.481578	0.003917
….	….	….	….	….	….	….	….	….
Histology description	0.458386	0.000584	Height (cm)	0.419570	0.002240	Hemoglobin (g/dl)	0.457296	0.000280
Stadium (ACR 2016)	0.457016	0.001370	Sex (M/F)	0.414997	0.004573	Weight (KG)	0.452585	0.004711
Sodium (mmol/L)	0.443226	0.013791	R classification	0.405456	0.009541	No. Of total lymph nodes	0.451755	0.000830
Hemoglobin (g/dl)	0.441205	0.002020	Sodium (mmol/L)	0.403820	0.001636	Sodium (mmol/L)	0.442357	0.009399
R classification	0.431782	0.009424	Hemoglobin (g/dl)	0.402518	0.001301	R classification	0.437439	0.004918

Tabell D.6: Oversikt over viktige og mindre viktige variabler på de klassiske modellene, kolorektal – PFS.

Random forest			Logistic regression			QDA		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
No. Of positive lymph nodes	0.507447	0.000000	Potassium (mmol/L)	0.465003	0.000000	ALP (U/L)	0.502746	0.000000
mrT (TNM ed.7)	0.493994	0.013453	Thrombocytes (10^9/L)	0.457859	0.007144	Blood samples at inclusion	0.496999	0.005747
mrV	0.487242	0.006752	Mucinous	0.453012	0.004847	Leukocytes (10^9/L)	0.490965	0.006033
ALP (U/L)	0.486810	0.000432	Sex (M/F)	0.447565	0.005448	Location primary tumor	0.489569	0.001396
GT (U/L)	0.486284	0.000526	CEA baseline	0.447109	0.000456	Creatinine (umol/L)	0.488504	0.001065
….	….	….	….	….	….	….	….	….
Stadium (ACR 2016)	0.448033	0.002820	Adjuvant treatment	0.408184	0.004497	R classification	0.460287	0.004664
R classification	0.445158	0.002875	Stadium (ACR 2016)	0.405160	0.003024	Weight (KG)	0.456934	0.003354
Sodium (mmol/L)	0.445110	0.000047	p/ypN (TNM ed. 7)	0.402208	0.002952	Suspected metastatic l.d	0.456306	0.000627
Hemoglobin (g/dl)	0.443418	0.001692	R classification	0.372259	0.029950	Sex (M/F)	0.454786	0.001520
Suspected metastatic l.d	0.441594	0.001824	Suspected metastatic l.d	0.366388	0.005871	p/ypT (TNM ed.7)	0.450666	0.004120

Tabell D.7: Oversikt over viktige og mindre viktige variabler på modellene foreslått av LazyPredict, kolorektal – PFS.

GuassianNB			Nearest centroid		
Variable	**MCC**	**Endring**	**Variable**	**MCC**	**Endring**
ALP (U/L)	0.478526	0.000000	mrV	0.468526	0.000000
mrN	0.471136	0.007391	mrT (TNM ed.7)	0.468094	0.000432
mrT (TNM ed.7)	0.470918	0.000218	Thrombocytes (10^9/L)	0.465762	0.002332
GT (U/L)	0.468637	0.002281	ALP (U/L)	0.465291	0.000471
Blood samples at inclusion	0.465062	0.003576	Sex (M/F)	0.462480	0.002810
….	….	….	….	….	….
p/ypT (TNM ed.7)	0.426035	0.000515	No. Of positive lymph nodes	0.447224	0.000167

Weight (KG)	0.425424	0.000611	No. Of total lymph nodes	0.444850	0.002374
Suspected metastatic l.d	0.423701	0.001723	ALT (U/L)	0.443816	0.001034
Albumin (g/L)	0.421299	0.002401	R classification	0.437662	0.006154
R classification	0.412046	0.009253	Albumin (g/L)	0.426943	0.010720

Tabellene er rangert etter de mindre viktige på toppen og de viktige på bunnen. Jo viktigere variabel desto mindre blir nøyaktigheten når variabelen droppes av datasettet. En variabel som virker å være viktig i samtlige modeller er variablene *«R Classification»*.

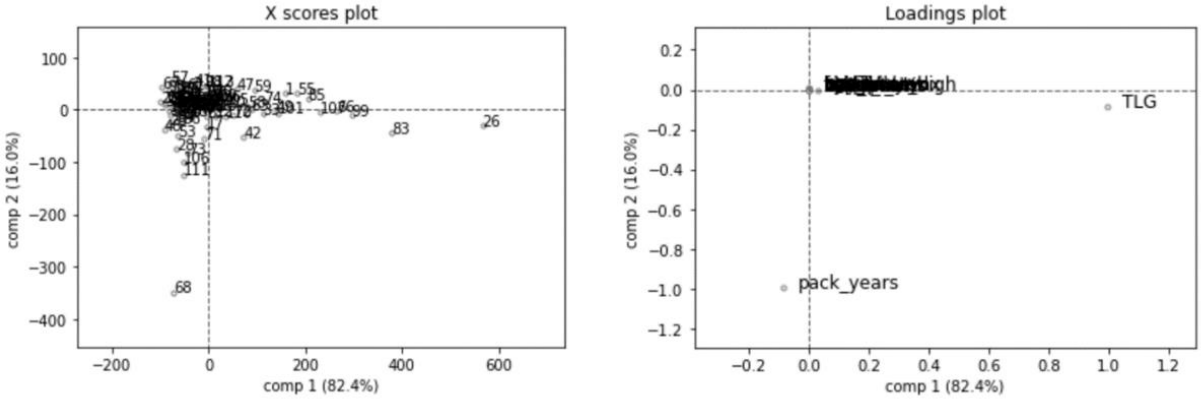
Vedlegg E

Hode og halskreft – Forundersøkelser

I vedlegget presenteres resultater fra forundersøkelser av hode- og halsdatasettet som ikke har blitt inkludert i selve oppgaven.

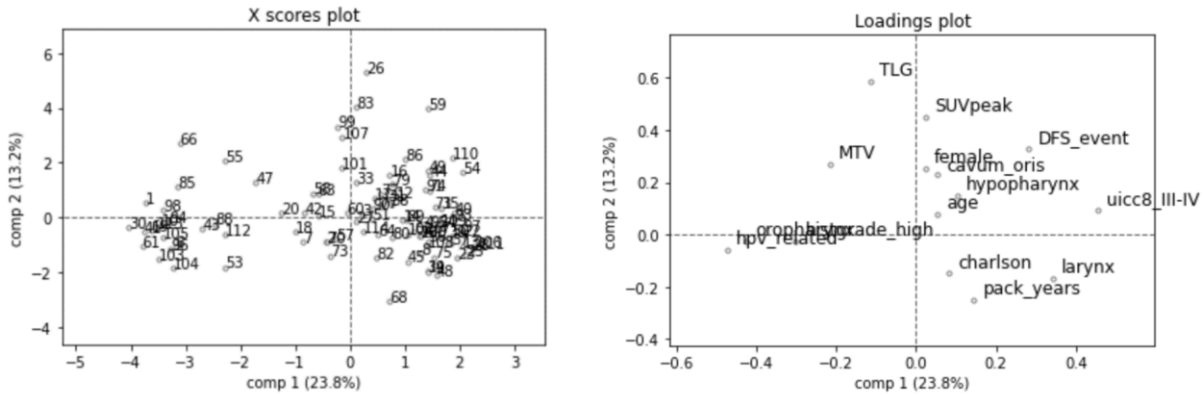
E.1 PCA av MAASTRO-datasett

Resultatet på PCA-analysen på sentrert MAASTRO datasettet, er presentert som score- og loadingplot i figur E.1. Det er tydelig at det er noen pasienter som skiller seg ut. Pasient-ID 26, 68 og 83 skiller seg ut fra resten av klyngen. Ved å legge begge diagrammene opp mot hverandre, kan det trekkes en sammenheng mellom de nevnte pasientprøvene og variablene *«pack_years»* og *«TLG»*. Et nærmere dypdykk i datasettet avklarte at pasient 68 hadde en høy verdi for variabelen *«pack_years»,* mens pasient 26 og 83 hadde en høy verdi for variablene *«TLG»*. Dette er et resultat som kan indikerer at pasient 26, 68 og 83 er mulige ekstremverdier i datasettet.



Figur E.1: Score- og loadingplot for det sentrerte MAASTRO datasettet.

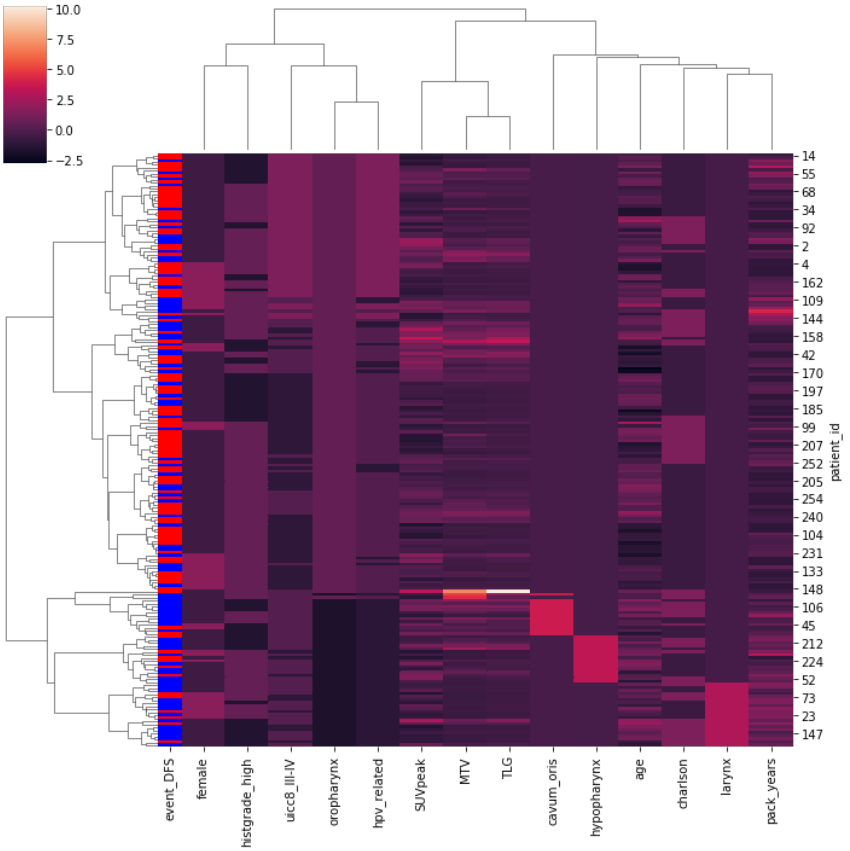
Figur E.2 nedenfor presenterer PCA-plott på samme datasett, men skalert i motsetning til det som ble presentert over. Det kan observeres at både pasient-Idene og variablene har blitt normalfordelt og spenner over store deler av x-aksen. På lik linje som figur 24 i kapittel 4.4.1, danner scoreplot to grupper, selv om det ikke like stor skille som den i kapittel 4.4.1. Dette kan sette i kontekst med at datasettet består av pasienter med primærsvulst i forskjellige områder, og at dette danner gruppen i scoreplottet.



Figur E.2: Score- og loadingplot for det standardiserte MAASTRO datasettet.

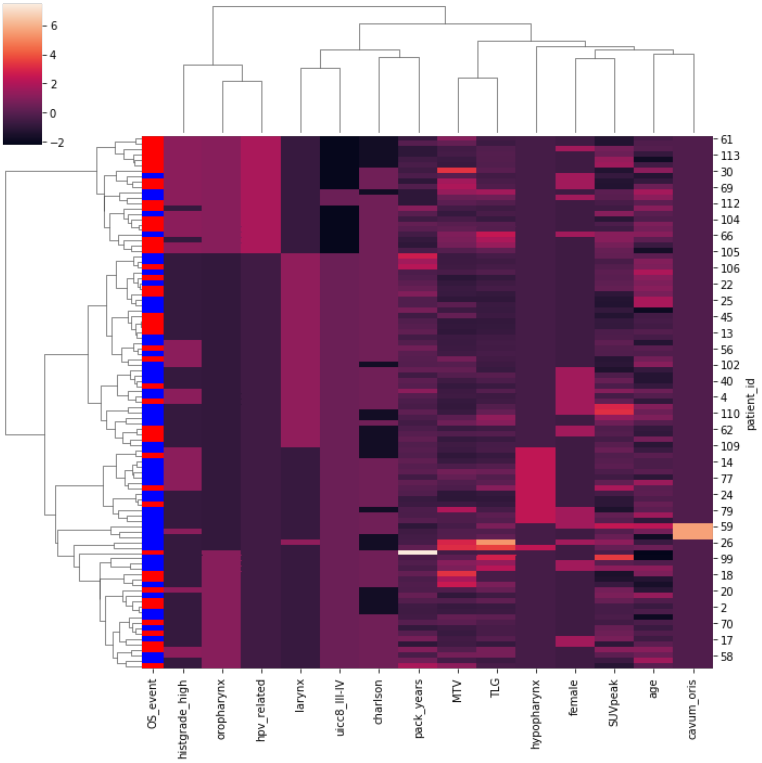
E.2 Clustermap

Figur E.3 presenterer clustermap-plottet av OxyTarget-datasettet. De røde stripene representerer pasienter med negativ PFS-event, som betyr at pasientene har verken død eller fått tilbakefall. De blå stripene representerer pasienter med positiv PFS-event, som betyr at pasientene enten har død eller fått lokalt eller metastatisk tilbakefall.

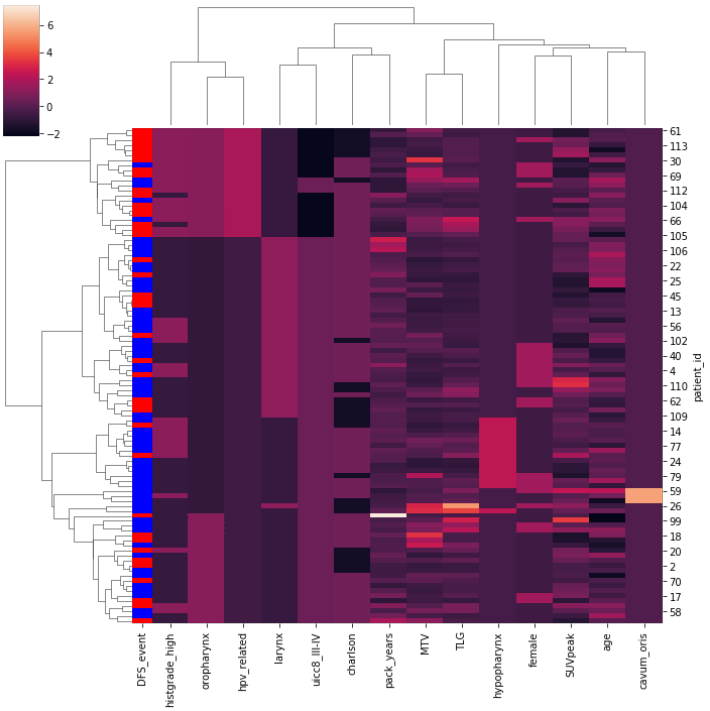


Figur E.3: Clustermap på hode- og hals datasett med DFS.

Figur E.4 og E.5 presenterer tilsvarende clustermap-plott for MAASTRO datasettet.



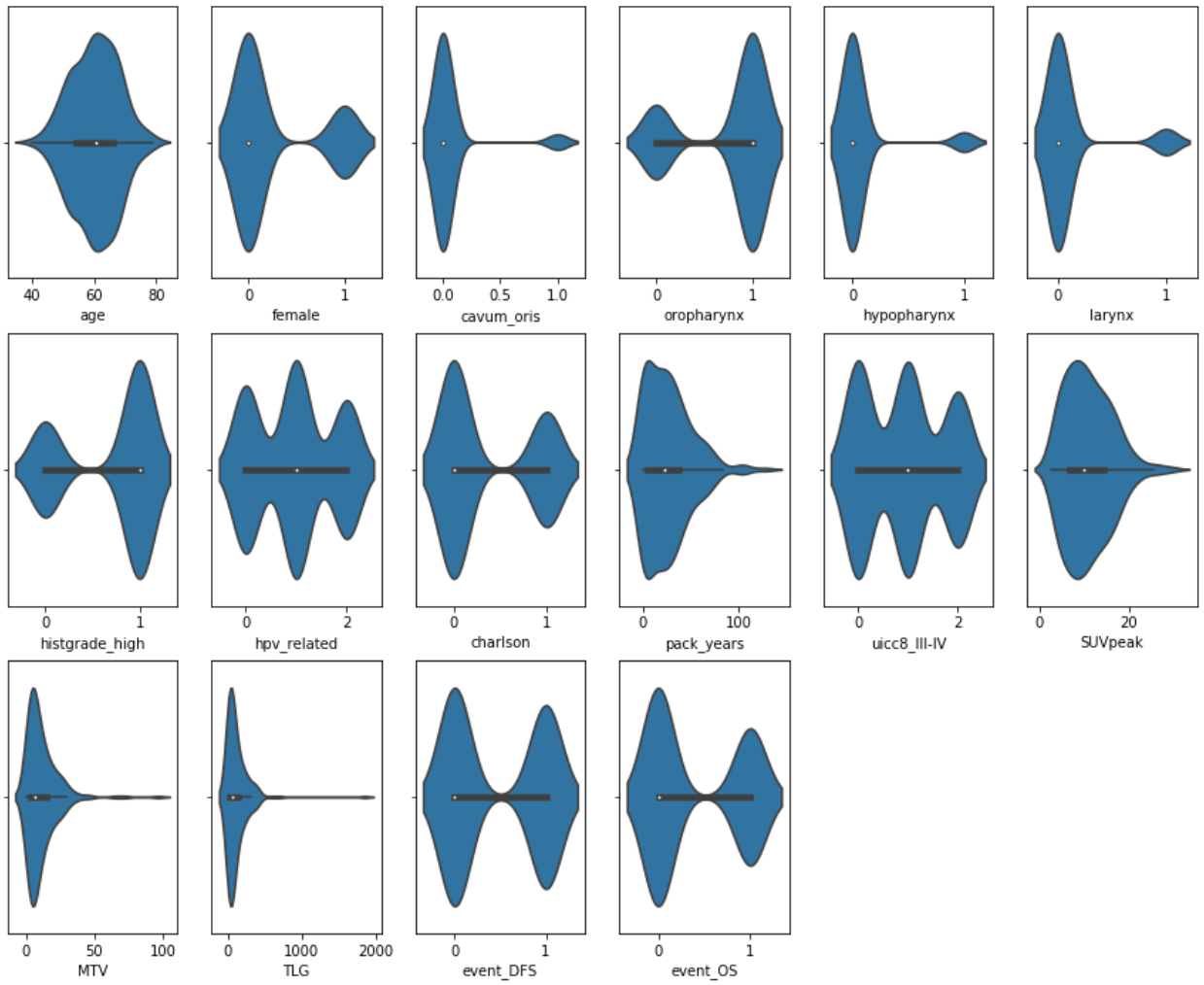
Figur E.4: Clustermap på MAASTRO datasett med OS.



Figur E.5: Clustermap på MAASTRO datasett med DFS.

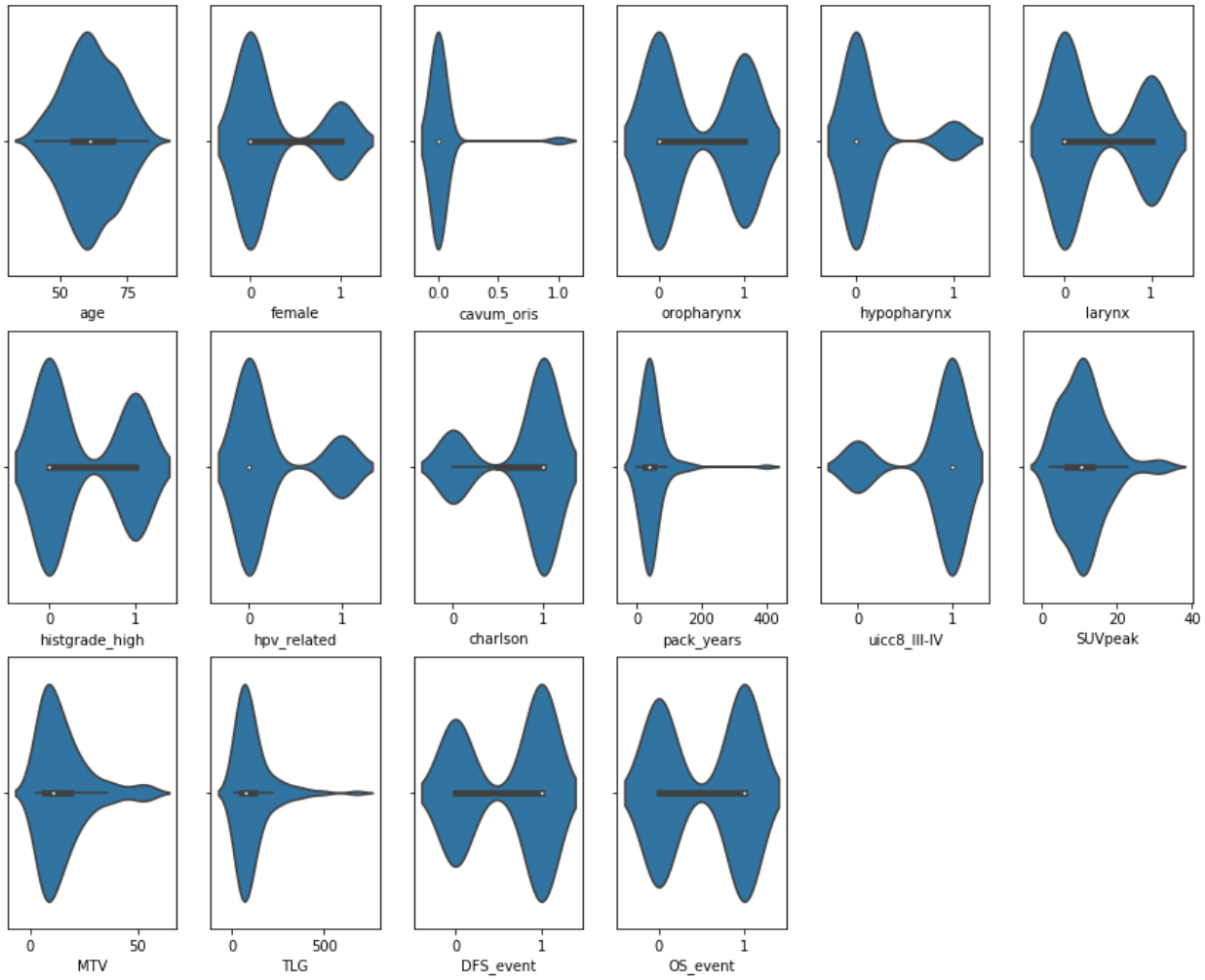
E.3 Violinplot – OUS og MAASTRO

Figur E.6 presenterer violinplot over alle variablene i datasettet fra OUS.



Figur E.6: Violinplot over variablene i hode- og halsdatasett.

Figur E.7 presenterer violinplot over alle variablene i datasettet fra MAASTRO klinikken.



Figur E.7: Violinplot over variablene i MAASTRO datasett.

E.4 Ekstremverdier

I kapittel 4.4.2, inspeksjon av ekstremverdier, ble tabell 41 presentert som inneholder en oversikt over hvilke observasjoner som ble slått ut som mulige ekstremverdier. To av observasjonene ble identifisert med både KNN og ECOD, og tabell 42 i kapittel 4.1.2 presenterte ekstremverdiscoren og oversikt over mulige årsaken til at disse ble slått ut. Tabell E.1 og E.2 er tilsvarende tabeller som presenterer de gjenværende observasjonene som ble identifisert av KNN og ECOD.

Tabell E.1: Oversikt over de gjenværende identifiserte ekstremverdier med KNN, hode- og hals

PasientID	KNN - score	Mulige årsaker
142	1530.02	Høy verdi for variablene pack_years og TLG.
161	254.95	Høy verdi for variabelen TLG.

Tabell E.2: Oversikt over de gjenværende identifiserte ekstremverdier med ECOD, hode- og hals

PasientID	ECOD - score	Mulige årsaker
15	61.87	En av de eldste i datasettet med primærsvulst i larynx (strupehode). Andel prøver med larynx er lite i forhold til andre svulsttyper. Kan ha en betydning med alder og svulsttype.
30	60.55	Veldig lav MTV verdi.
111	58.57	Høy verdi for variabelen pack_years.
114	59.72	Høy verdi for variabelen TLG.
117	63.87	Høy verdi for variabelen TLG.

Vedlegg F

Hode og halskreft – OS

Vedlegget presenterer diverse resultater som ikke ble presentert i selve oppgaven for modeller med generell overlevelse (OS) som responsverdi på datasettet med Hode- og hals. Vedlegget inneholder også noen resultater som presenterer mer omfattende oversikter av resultater som har blitt presentert i selve oppgaven

F.1 Evaluering av modeller på utelatte algoritmer

Tabell F.1 presenterer testresultatene for modellene som ble utelatt fra kapittel 4.5. På lik linje som modellene som ble presenter i kapittel 4 ble, modellene presentert i tabell F.1 hyperparameteroptimalisert og kjørt med 4-foldet kryssvalidering med 1000 repetisjoner.

Tabell F.1: De resterende gjennomsnittlige testresultatene for modellene som ikke ble inkludert i oppgaven, hode- og hals - OS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.69 ± 0.06	0.59 ± 0.08	0.75 ± 0.05	0.34 ± 0.12	0.67 ± 0.06
KNN	0.71 ± 0.05	0.53 ± 0.10	0.79 ± 0.04	0.37 ± 0.12	0.66 ± 0.05
META-DES	0.73 ± 0.05	0.63 ± 0.08	0.79 ± 0.04	0.43 ± 0.11	0.71 ± 0.06
MCB	0.73 ± 0.06	0.63 ± 0.08	0.78 ± 0.05	0.43 ± 0.12	0.71 ± 0.06
OLA	0.74 ± 0.05	0.64 ± 0.08	0.79 ± 0.04	0.44 ± 0.12	0.71 ± 0.06

Tabell F.2 presenterer de gjennomsnittlige resultatene over modellytelsen for treningssettet.

Tabell F.2: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, hode- og hals-OS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.66 ± 0.03	0.54 ± 0.04	0.72 ± 0.02	0.27 ± 0.06	0.63 ± 0.03
KNN	0.76 ± 0.02	0.61 ± 0.04	0.83 ± 0.01	0.48 ± 0.05	0.71 ± 0.02
META-DES	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
MCB	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
OLA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Et interessant resultat er hvordan SVC presterer for valideringsdata i forhold til treningsdata. Det er tydelig at SVC for høyere nøyaktighet på samtlige målinger for valideringsdata enn med treningsdata. Dette kan skilles av at modellen kanskje er undertilpasset. En modell som er undertilpasset skal i prinsippet presentere dårlig på valideringsdata, noe som ikke er tilfelle i resultatene som har blitt presentert.

F.2 Vanskelige pasienter i testdata, topp 30

Tabell F.3 gir en oversikt over pasient-ID for de 30 mest vanskeligste pasientobservasjonene. Tabellen er sortert i synkende rekkefølge etter antall feilklassifiserte observasjoner for hver pasient, og observasjonene som er felles for de øverste 30 er fremhevet med farger i tabellen. Disse observasjonene ble benyttet for å generere matrisen beskrevet i kapittel 4.5.3.

Tabell F.3: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på testdata, hode- og halskreft med respons OS

KNORA-E	KNORA-U	DES-P	Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
169	105	189	70	154	35	154	35
216	241	26	195	26	40	43	209
34	166	105	154	105	39	26	13
243	240	154	124	241	57	241	39
171	169	240	43	253	73	171	57
189	243	209	166	124	124	170	70
209	35	171	243	43	166	169	73
240	34	166	169	42	169	166	120
170	253	169	224	243	170	124	124
74	36	243	189	195	241	39	166
105	189	35	170	216	216	13	170
124	124	34	171	189	11	216	171
13	170	253	241	166	26	209	40
26	74	170	74	169	34	164	72
70	209	74	26	123	36	105	241
195	13	124	34	74	72	89	11
43	26	241	105	34	105	34	26
154	171	72	35	171	198	100	36
166	70	195	209	209	13	242	216
42	154	70	216	13	171	61	164
253	195	43	13	213	209	195	224
35	43	13	240	36	243	189	253
72	224	224	57	151	61	144	61
67	42	36	42	240	74	62	74
241	216	216	72	121	100	123	198
224	72	42	123	73	111	74	111
158	67	67	39	168	123	243	195
168	57	168	253	198	195	60	189
39	168	57	144	56	224	55	243
125	40	40	36	39	189	77	105

F.3 Vanskelige pasienter i treningsdata

Tabell F.4 gir en oversikt over pasient-ID for de 30 mest vanskeligste pasientobservasjonene. Tabellen er sortert i synkende rekkefølge etter antall feilklassifiserte observasjoner for hver pasient, og observasjonene som er felles for de øverste 30 er fremhevet med farger i tabellen. Disse observasjonene ble benyttet for å generere matrisen beskrevet i kapittel 4.5.4.

Tabell F.4: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på treningsdata, hode- og halskreft med responsen OS

Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
253	**253**	**253**	11	**253**
13	**13**	11	**13**	11
243	247	247	**243**	247
241	**243**	**13**	**253**	**13**
240	250	**243**	242	**243**
242	11	**241**	8	**241**
-	252	**240**	244	8
-	15	242	**241**	**240**
-	**241**	-	15	-
-	8	-	**240**	-
-	**240**	-	16	-
-	5	-	-	-
-	244	-	-	-

F.4 Viktige og mindre viktige variabler

Tabellene F.5, F.6 og F.7 presenterer en oversikt over viktige og mindre viktige variabel for alle de modellene som er presentert i kapittel 4.5. Tabellene viser alle de 14 variablene i datasettet. Tabellene er rangert etter de mindre viktige på toppen og de viktige på bunnen. Jo viktigere variabel desto mindre blir nøyaktigheten når variabelen droppes av datasettet.

Tabell F.5: Oversikt over viktige og mindre viktige variabler for DES-modellene, hode- og halskreft med responsen OS

KNORA-E			KNORA-U			DESP		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
SUVpeak	0.460444	0.000000	SUVpeak	0.454726	0.000000	SUVpeak	0.451908	0.000000
hypopharynx	0.454010	0.006433	age	0.430606	0.024120	TLG	0.441808	0.010099
TLG	0.450130	0.003880	TLG	0.427252	0.003354	hypopharynx	0.440875	0.000934
histgrade_high	0.444313	0.005818	hypopharynx	0.422681	0.004570	larynx	0.439387	0.001488
cavum_oris	0.440402	0.003911	larynx	0.422474	0.000208	cavum_oris	0.436182	0.003205
larynx	0.439632	0.000769	histgrade_high	0.420343	0.002131	age	0.434384	0.001798
age	0.436796	0.002837	oropharynx	0.413833	0.006510	oropharynx	0.429599	0.004785
uicc8_III-IV	0.431462	0.005334	cavum_oris	0.413597	0.000236	histgrade_high	0.427192	0.002407
oropharynx	0.426966	0.004496	uicc8_III-IV	0.408046	0.005551	uicc8_III-IV	0.426240	0.000953
charlson	0.426044	0.000922	female	0.407210	0.000836	charlson	0.422703	0.003536
female	0.424137	0.001907	hpv_related	0.405147	0.002063	female	0.416933	0.005770
hpv_related	0.418703	0.005435	charlson	0.394721	0.010426	hpv_related	0.413809	0.003124
MTV	0.397947	0.020756	pack_years	0.374228	0.020493	MTV	0.389266	0.024543
pack_years	0.372506	0.025441	MTV	0.369898	0.004330	pack_years	0.368841	0.020425

Tabell F.6: Oversikt over viktige og mindre viktige variabler på de klassiske modellene, hode- og halskreft med responsen OS

Random forest			Logistic regression			QDA		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
TLG	0.498256	0.000000	charlson	0.490238	0.000000	cavum_oris	0.460124	0.000000
SUVpeak	0.493914	0.004342	age	0.480534	0.009704	histgrade_high	0.459422	0.000702
hypopharynx	0.487386	0.006528	larynx	0.478940	0.001594	TLG	0.457465	0.001957
age	0.482471	0.004915	TLG	0.476079	0.002861	hypopharynx	0.454693	0.002772
cavum_oris	0.482093	0.000378	histgrade_high	0.476059	0.000020	female	0.452866	0.001827
larynx	0.481673	0.000420	MTV	0.475737	0.000322	charlson	0.451860	0.001006
histgrade_high	0.478071	0.003602	hypopharynx	0.475233	0.000504	oropharynx	0.450883	0.000976
charlson	0.472613	0.005458	female	0.474728	0.000506	MTV	0.450822	0.000061
female	0.469077	0.003536	cavum_oris	0.473132	0.001595	larynx	0.447169	0.003653
Oropharynx	0.467703	0.001374	oropharynx	0.471645	0.001488	age	0.440189	0.006980
hpv_related	0.466371	0.001332	SUVpeak	0.464420	0.007225	uicc8_III-IV	0.439906	0.000283
uicc8_III-IV	0.457305	0.009066	uicc8_III-IV	0.434107	0.030313	hpv_related	0.435177	0.004729
MTV	0.456668	0.000637	pack_years	0.410010	0.024097	SUVpeak	0.426910	0.008267

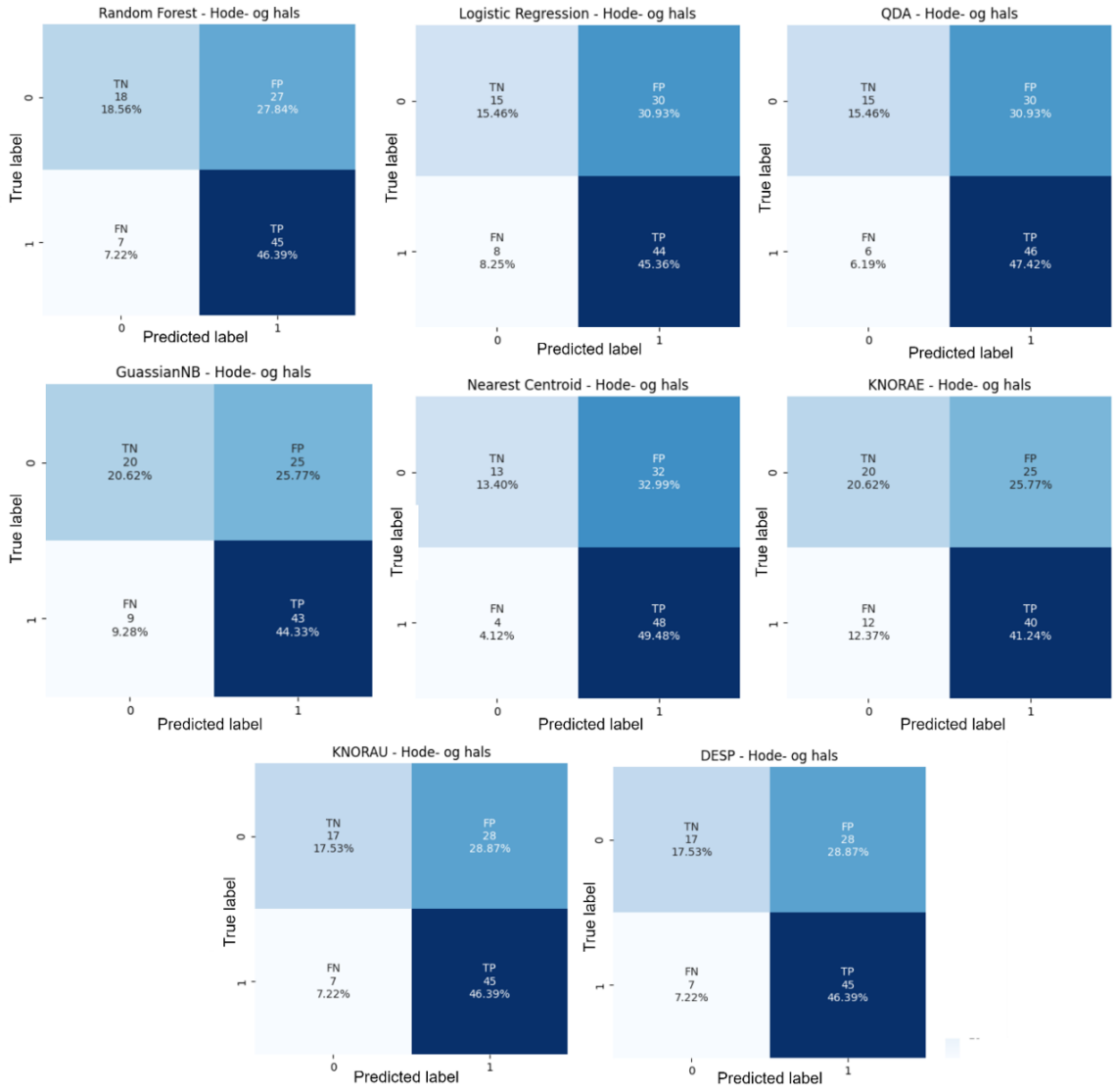
Tabell F.7: Oversikt over viktige og mindre viktige variabler på modellene foreslått av LazyPredict, hode- og halskreft med responsen OS

GuassianNB			Nearest centroid		
Variable	**MCC**	**Endring**	**Variable**	**MCC**	**Endring**
cavum_oris	0.414661	0.000000	TLG	0.490628	0.000000
hypopharynx	0.400192	0.014469	hypopharynx	0.480811	0.009817
pack_years	0.390413	0.009779	histgrade_high	0.474451	0.006360
age	0.389164	0.001249	cavum_oris	0.473495	0.000957
female	0.388763	0.000402	uicc8_III-IV	0.473455	0.000039
uicc8_III-IV	0.388309	0.000453	larynx	0.472485	0.000971
SUVpeak	0.387988	0.000321	oropharynx	0.472000	0.000485
larynx	0.385410	0.002578	female	0.466850	0.005150
TLG	0.385009	0.000401	SUVpeak	0.466374	0.000476
oropharynx	0.384702	0.000307	charlson	0.464330	0.002044
hpv_related	0.382776	0.001926	MTV	0.464180	0.000150
histgrade_high	0.382625	0.000151	hpv_related	0.461265	0.002915
charlson	0.377912	0.004712	age	0.444459	0.016806
MTV	0.360388	0.017525	pack_years	0.419665	0.024793

Modellene med DES-algoritmer viser seg å ha en markant forskjell når variabelen pack_years droppes. Modellene faller ned med nesten 9% i MCC nøyaktighet når variabelen fjernes fra datasettet.

F.5 Confusion matrix - MAASTRO

Figur F.1 presenterer confusion matrix over alle de åtte modellene fra kapittel 4.5.5 på MAASTRO datasettet. Sammenlignet med confusion matrix for valideringsdata i figur 32 i kapittel 4.5.1 kommer det tydelig frem at det eksterne datasettet har betydelig større andel av prøver med den positive klassen med totalt 42 observasjoner. Matrisen viser også til et oversiktlig bilde over at den negative klassen har flere antall feilpredikerte enn antall riktig predikert av 45 observasjonsprøver.



Figur F.1: Confusion matrix over alle de åtte algoritmene på MAASTRO datasettet med responsen OS.

F.6 Utelatte modeller på MAASTRO

Nedenfor følger tabell F.8 som presenterer testresultatene på MAASTRO datasettet for modellene som ble utelatt fra kapittel 4.5.

Tabell F.8: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, MAASTRO datasettet med responsen OS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.56 ± 0.03	0.65 ± 0.04	0.38 ± 0.06	0.09 ± 0.06	0.54 ± 0.04
KNN	0.61 ± 0.04	0.63 ± 0.05	0.59 ± 0.02	0.22 ± 0.07	0.60 ± 0.03
META-DES	0.64 ± 0.02	0.72 ± 0.02	0.50 ± 0.04	0.28 ± 0.05	0.62 ± 0.02
MCB	0.62 ± 0.03	0.70 ± 0.03	0.48 ± 0.04	0.23 ± 0.06	0.60 ± 0.03
OLA	0.64 ± 0.02	0.72 ± 0.02	0.50 ± 0.04	0.28 ± 0.05	0.62 ± 0.02

Vedlegg G

Hode og halskreft – DFS

Vedlegget presenterer diverse resultater som ikke ble presentert i selve oppgaven for modeller med sykdomsfri overlevelse (DFS) som responsverdi på datasettet med Hode- og hals. Vedlegget inneholder også noen resultater som presenterer mer omfattende oversikter av resultater som har blitt presentert i selve oppgaven

G.1 Evaluering av modeller på utelatte algoritmer

Nedenfor følger tabell G.1 som presenterer testresultatene for modellene som ble utelatt fra kapittel 4.6. På lik linje som modellene som ble presenter i kapittel 4 ble, modellene presentert i tabell G.1 hyperparameteroptimalisert og kjørt med 4-foldet kryssvalidering med 1000 repetisjoner.

Tabell G.1: De resterende gjennomsnittlige testresultatene for modellene som ikke ble inkludert i oppgaven, hode- og halskreft med responsen DFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.66 ± 0.06	0.59 ± 0.08	0.71 ± 0.05	0.32 ± 0.12	0.65 ± 0.06
KNN	0.66 ± 0.05	0.54 ± 0.09	0.73 ± 0.04	0.32 ± 0.12	0.64 ± 0.05
META-DES	0.65 ± 0.06	0.60 ± 0.07	0.68 ± 0.06	0.29 ± 0.12	0.64 ± 0.06
MCB	0.65 ± 0.06	0.60 ± 0.07	0.69 ± 0.06	0.30 ± 0.12	0.65 ± 0.06
OLA	0.66 ± 0.06	0.60 ± 0.07	0.69 ± 0.06	0.31 ± 0.12	0.65 ± 0.06

Tabell G.2 presenterer de gjennomsnittlige resultatene over modellytelsen for treningssettet.

Tabell G.2: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, hode- og halskreft med responsen DFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.72 ± 0.02	0.67 ± 0.04	0.76 ± 0.02	0.44 ± 0.05	0.71 ± 0.03
KNN	0.71 ± 0.02	0.61 ± 0.04	0.77 ± 0.02	0.43 ± 0.05	0.70 ± 0.02
META-DES	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
MCB	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
OLA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

G.2 Topp 30 vanskelige pasienter - testdata

Tabell G.3 presenterer en oversikt med pasient-ID over de topp 30 pasientobservasjoner som er vanskelige å bli klassifisert riktig. Tabellen er sortert etter pasienter med høyest antall

feilklassifiserte på topp og synkende nedover. Alle de felles observasjonene av topp 30 er fargelagt i tabellen, og det var nettopp disse observasjonene som ble brukt for å generere matrisen i kapittel 4.6.3.

Tabell G.3: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på testdata, Hode- og hals – DFS

KNORA-E	KNORA-U	DES-P	Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
133	74	43	224	243	189	13	35
43	105	154	74	139	92	115	140
105	35	26	154	140	195	164	168
26	189	72	92	124	139	170	11
110	110	241	166	92	111	195	39
241	241	110	169	189	11	61	61
209	154	115	189	194	39	92	73
115	115	92	194	241	70	100	92
92	92	74	209	115	61	124	139
35	70	70	241	110	73	140	124
74	43	105	195	66	43	144	154
70	169	169	35	36	164	166	43
189	72	35	115	43	36	169	166
154	66	66	110	74	124	189	111
169	209	189	140	168	140	194	189
66	26	195	43	171	35	209	194
136	36	253	66	253	154	241	209
34	253	224	70	105	166	43	70
139	224	209	34	133	241	26	241
195	34	139	139	150	26	243	26
170	195	34	198	154	170	139	198
72	139	133	171	26	194	123	195
253	133	36	150	195	209	39	170
224	170	170	136	34	115	110	164
243	164	216	133	121	105	105	105
166	168	164	26	198	243	89	120
150	198	136	124	166	120	133	133
216	216	168	72	73	133	136	136
168	120	198	243	209	136	150	150
140	136	120	105	164	150	171	171

G.3 Vanskelige pasienter i treningsdata

Tabell G.4 presenterer en liste over hvilke pasienter som har vært vanskelig å bli klassifisert i riktig kategori med treningsdata. Modellen med logistisk regresjon hadde flest problemer og utgjorde totalt 15 pasienter. Tabellen er sortert etter størst vanskelighetsgrad. Som

presentert i kapittel 4.6.4 hadde modellene med DES-algoritmer ingen vanskelige pasienter. Observasjoner som var til felles er markert med fargekoder, og det var disse observasjonene som ble inkludert i utformingen av matrisen i kapittel 4.6.4.

Tabell G.4: Oversikt over topp 30 pasienter som er vanskelig å bli klassifisert riktig på treningsdata, Hode- og hals – DFS

Random forest	Logistic regression	QDA	GuassianNB	Nearest centroid
253	253	253	11	253
13	247	11	13	11
243	11	247	8	247
247	13	13	243	13
11	252	243	242	243
241	250	8	253	8
8	243	241	241	252
242	8	240	244	241
252	5	-	247	-
240	241	-	15	-
15	240	-	16	-
-	2	-	240	-
-	239	-	-	-
-	15	-	-	-
-	254	-	-	-

G.4 Viktige og mindre viktige variabler

Tabellene G.5, G.6 og G.7 presenterer en oversikt over viktige og mindre viktige variabel for alle de modellene som er presentert i kapittel 4.6. Tabellene viser alle de 14 variablene i datasettet. Tabellene er rangert etter de mindre viktige på toppen og de viktige på bunnen. Jo viktigere variabel desto mindre blir nøyaktigheten når variabelen droppes av datasettet.

Tabell G.5: Oversikt over viktige og mindre viktige variabler på DES-modellene, hode- og halskreft med responsen DFS

KNORA-E			KNORA-U			DESP		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
SUVpeak	0.353780	0.000000	SUVpeak	0.344198	0.000000	SUVpeak	0.348175	0.000000
uicc8_III-IV	0.317868	0.035912	hypopharynx	0.318481	0.025717	hpv_related	0.319725	0.028450
charlson	0.315766	0.002101	cavum_oris	0.317636	0.000845	charlson	0.319050	0.000675
oropharynx	0.314008	0.001759	oropharynx	0.317139	0.000497	cavum_oris	0.315566	0.003484
larynx	0.310183	0.003825	charlson	0.309313	0.007826	oropharynx	0.315410	0.000156
hypopharynx	0.310108	0.000075	pack_years	0.307562	0.001752	uicc8_III-IV	0.314709	0.000701
cavum_oris	0.307290	0.002818	age	0.305712	0.001849	hypopharynx	0.314682	0.000026
hpv_related	0.303676	0.003614	uicc8_III-IV	0.305177	0.000535	age	0.312893	0.001789
histgrade_high	0.300144	0.003533	larynx	0.303392	0.001785	pack_years	0.304137	0.008757
female	0.299333	0.000810	histgrade_high	0.302203	0.001189	female	0.301958	0.002178
TLG	0.298551	0.000782	hpv_related	0.301917	0.000286	TLG	0.301619	0.000340
pack_years	0.298370	0.000181	female	0.301081	0.000835	larynx	0.299834	0.001785
age	0.297034	0.001336	TLG	0.292530	0.008551	histgrade_high	0.294642	0.005192
MTV	0.269518	0.027516	MTV	0.271889	0.020641	MTV	0.284864	0.009778

Tabell G.6: Oversikt over viktige og mindre viktige variabler på de klassiske modellene, hode- og halskreft med responsen DFS

Random forest			Logistic regression			QDA		
Variabel	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**	**Variabel**	**MCC**	**Endring**
SUVpeak	0.414335	0.000000	age	0.350912	0.000000	hpv_related	0.393546	0.000000
hypopharynx	0.411949	0.002385	MTV	0.348673	0.002239	cavum_oris	0.390809	0.002737
histgrade_high	0.411802	0.000147	hpv_related	0.348383	0.000290	hypopharynx	0.389308	0.001501
age	0.408051	0.003751	female	0.347624	0.000760	histgrade_high	0.389105	0.000204
hpv_related	0.407471	0.000580	SUVpeak	0.346882	0.000742	female	0.384530	0.004575
oropharynx	0.404772	0.002699	histgrade_high	0.345934	0.000948	oropharynx	0.384224	0.000306
female	0.403893	0.000880	hypopharynx	0.345791	0.000143	age	0.380813	0.003411
larynx	0.400448	0.003445	oropharynx	0.344103	0.001687	TLG	0.378194	0.002620
TLG	0.398753	0.001695	cavum_oris	0.342941	0.001163	larynx	0.377464	0.000730
cavum_oris	0.396776	0.001977	charlson	0.341103	0.001838	uicc8_III-IV	0.370554	0.006910
charlson	0.394537	0.002239	larynx	0.339129	0.001974	MTV	0.368937	0.001617
uicc8_III-IV	0.387559	0.006978	TLG	0.338890	0.000239	charlson	0.366510	0.002428
MTV	0.376305	0.011254	pack_years	0.326095	0.012795	SUVpeak	0.364875	0.001635
pack_years	0.367211	0.009094	uicc8_III-IV	0.311757	0.014338	pack_years	0.350609	0.014266

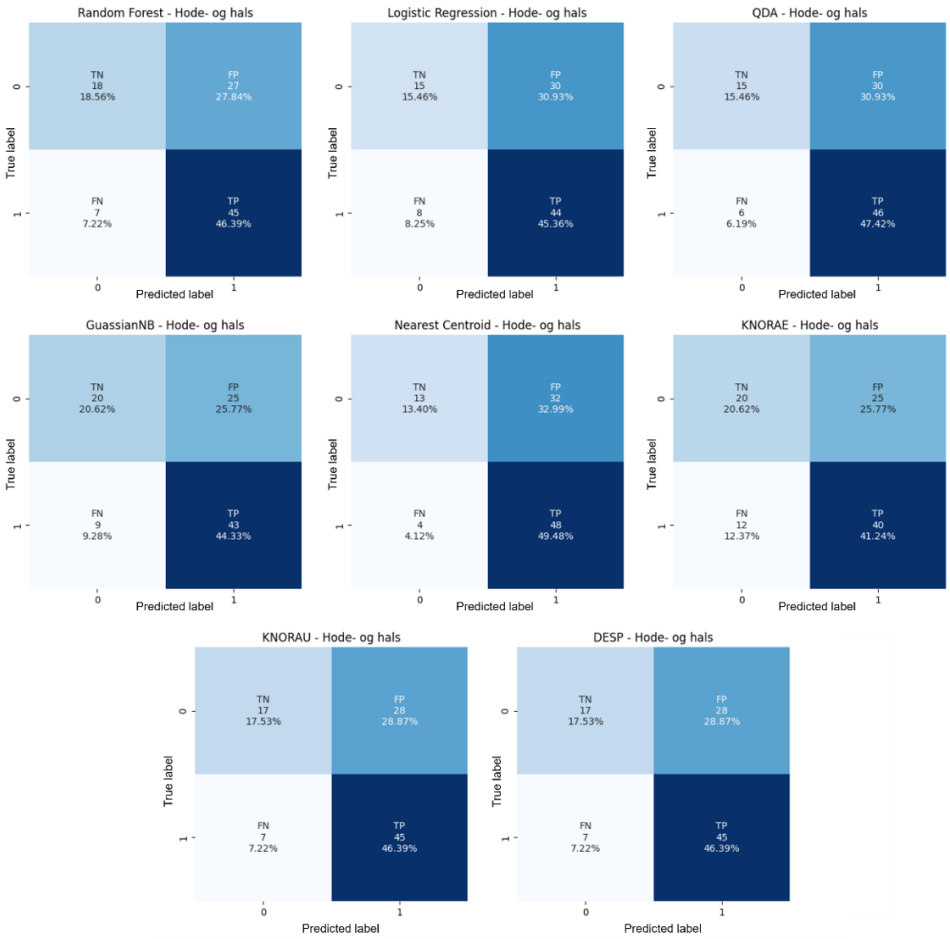
Tabell G.7: Oversikt over viktige og mindre viktige variabler på modellene foreslått av LazyPredict, Hode- og hals – DFS

GuassianNB			Nearest centroid		
Variable	**MCC**	**Endring**	**Variable**	**MCC**	**Endring**
cavum_oris	0.366660	0.000000	histgrade_high	0.396870	0.000000
hpv_related	0.359455	0.007205	hypopharynx	0.394580	0.002289
histgrade_high	0.350275	0.009180	hpv_related	0.394452	0.000128
hypopharynx	0.349733	0.000543	female	0.387018	0.007434
pack_years	0.348842	0.000890	TLG	0.384903	0.002116
uicc8_III-IV	0.348217	0.000625	cavum_oris	0.383349	0.001553
larynx	0.347655	0.000562	age	0.381706	0.001644
female	0.345054	0.002601	MTV	0.380177	0.001529
oropharynx	0.341180	0.003874	oropharynx	0.379830	0.000347
age	0.339500	0.001679	larynx	0.378567	0.001263
charlson	0.336717	0.002783	uicc8_III-IV	0.374162	0.004404
SUVpeak	0.325866	0.010851	SUVpeak	0.374144	0.000019
MTV	0.316735	0.009131	charlson	0.369720	0.004424
TLG	0.306442	0.010293	pack_years	0.341331	0.028389

Det kommer tydelig frem i tabellene at det ikke er store endringer i nøyaktigheten ved å droppe en variabel for modellene. For modellene med DES-algoritmer er det tydelig at variabelen SUVpeak er mindre viktig, mens MTV er en variabel som er viktig for modellene. Som for modellene med OS, ser det også til at modellene med DFS også har markante forskjeller i MCC nøyaktighet dersom enkelte variabler droppes ved modellering.

G.5 Confusion matrix – MAASTRO

Confusion matrix over alle de åtte modellene fra kapittel 4.6.5 på MAASTRO datasettet er presentert i figur G.1. I matrisen er det enkelt å gjenkjenne at det er flere positive prøver i datasettet sammenlignet med andel positive prøver det var i valideringsdatasettet. I matrisene kommer det også et klart bilde over at den negative klassen har størst problem med å bli klassifisert. Det er flere antall feilpredikerte enn antall riktig predikert av den negative klassen, altså kategori 0.



Figur G.1: Confusion matrix over alle de åtte algoritmene på MAASTRO datasettet med responsen DFS.

G.6 Utelatte modeller på MAASTRO

Nedenfor følger tabell G.8 som presenterer testresultatene på MAASTRO datasettet for modellene som ble utelatt fra kapittel 4.6.

Tabell G.8: De resterende gjennomsnittlige treningsresultatene for modellene som ikke ble inkludert i oppgaven, MAASTRO-datasettet med responsen DFS

	Accuracy	F1-positiv	F1-negativ	MCC	ROC-AUC
SVC	0.69 ± 0.02	0.78 ± 0.02	0.50 ± 0.05	0.35 ± 0.06	0.64 ± 0.03
KNN	0.68 ± 0.04	0.72 ± 0.05	0.60 ± 0.04	0.33 ± 0.08	0.66 ± 0.04
META-DES	0.65 ± 0.03	0.74 ± 0.03	0.44 ± 0.05	0.24 ± 0.08	0.60 ± 0.03
MCB	0.67 ± 0.03	0.76 ± 0.02	0.47 ± 0.05	0.29 ± 0.07	0.62 ± 0.03
OLA	0.69 ± 0.02	0.78 ± 0.02	0.49 ± 0.05	0.34 ± 0.06	0.64 ± 0.03

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway