UNIVERZA V MARIBORU
FAKULTETA ZA STROJNIŠTVO

Lucijano BERUS

RAZVRŠČANJE VZORCEV Z UPORABO INTELIGENTNIH METOD

Magistrsko delo
študijskega programa 2. stopnje
Strojništvo

Maribor, avgust 2017

Univerza v Mariboru

Fakulteta za strojništvo

RAZVRŠČANJE VZORCEV Z UPORABO INTELIGENTNIH METOD

Magistrsko delo

Študent(ka):	Lucijano BERUS
Študijski program:	študijski program 2. stopnje Strojništvo
Smer:	Računalniško inženirsko modeliranje
Mentor:	doc. dr. Simon KLANČNIK

Maribor, avgust 2017

Številka: S-BM0251
Datum in kraj: 27.06.2017, Maribor

Na osnovi 330. člena Statuta Univerze v Mariboru (Statut UM-UPB12, Uradni list RS, št. 29/2017) izdajam:

SKLEP O ZAKLJUČNEM DELU

LUCIJANU BERUSU, študentu magistrskega študijskega programa druge stopnje **STROJNIŠTVO,** smer **RAČUNALNIŠKO INŽENIRSKO MODELIRANJE**, se dovoljuje izdelati zaključno delo.

Tema zaključnega dela je pretežno s področja **Katedre za proizvodno strojništvo.**

Mentor:	**doc. dr. Simon Klančnik**
Somentor:	/
Zunanji delovni somentor:	/

Naslov zaključnega dela: **Razvrščanje vzorcev z uporabo inteligentnih metod**

Naslov zaključnega dela v angleškem jeziku: **Classification of patterns with use of intelligent methods**

Rok za izdelavo in oddajo zaključnega dela je: **27.06.2018.** Zaključno delo je potrebno izdelati skladno z »Navodili za pripravo magistrskega dela« in ga v treh izvodih oddati v pristojnem referatu članice. Hkrati se odda tudi izjava mentorja o ustreznosti zaključnega dela ter poročilo o preverjanju podobnosti z drugimi deli.

Pravni pouk: Zoper ta sklep je možna pritožba na Senat članice v roku 10 delovnih dni od dneva prejema sklepa.

UNIVERZA V MARIBORU FAKULTETA ZA STROJNIŠTVO

Dekan:

red. prof. dr. Bojan Dolšak

Obvestiti:
- kandidata,
- mentorja,
- odložiti v arhiv.

I Z J A V A

Podpisani _____, izjavljam, da:

- je diplomsko delo rezultat lastnega raziskovalnega dela,
- predloženo delo v celoti ali v delih ni bilo predloženo za pridobitev kakršnekoli izobrazbe po študijskem programu druge fakultete ali univerze,
- so rezultati korektno navedeni,
- nisem kršil-a avtorskih pravic in intelektualne lastnine drugih,
- soglašam z javno dostopnostjo diplomskega dela v Knjižnici tehniških fakultet ter Digitalni knjižnici Univerze v Mariboru, v skladu z Izjavo o istovetnosti tiskane in elektronske verzije zaključnega dela.

Maribor,_____ Podpis: _____

ZAHVALA

Zahvaljujem se mentorju dr. Simonu KLANČNIKU za pomoč in vodenje pri opravljanju magistrskega dela.

Zahvaljujem se tudi svojim staršem ter vsem znancem in prijateljem, ki so mi stali ob strani in me podpirali v času študija.

RAZVRŠČANJE VZORCEV Z UPORABO INTELIGENTNIH METOD

Ključne besede: umetna inteligenca, klasifikacija, strojno učenje, Parkinsonova bolezen, umetna nevronska mreža

UDK: **004.93.021(043.2)**

POVZETEK

Magistrsko delo obravnava področje umetne inteligence, strojnega učenja, razvrščanja kompleksnih vzorcev in metode določitve značilk. Predstavljeno je delovanje nekaterih najpogosteje uporabljenih razvrščevalnih algoritmov. Izdelan je bil algoritem za zaznavo Parkinsonove bolezni na podlagi zajetega zvočnega signala. Meritve zvoka so bile narejene na štiridesetih posameznikih. Od tega je bila polovica zdravih in polovica z Parkinsonovo boleznijo. Namen naloge je razviti robusten sistem za zaznavo prisotnosti Parkinsonove bolezni. Za izboljšanje natančnosti razvrščanja, so bile uporabljene različne tehnike določitve značilk (Pearsonov korelacijski koeficient, Khendallov korelacijski koeficient in Samoorganizacijske gruče) in topologije nevronskih mrež. S pomočjo usmerjene nevronske mreže, je bila dosežena 86,47 % natančnost razvrščanja. Omenjena natančnost je bila dosežena z uporabo redukcije značilk na podlagi Pearsonovega korelacijskega koeficienta.

CLASSIFICATION OF PATTERNS WITH USE OF INTELLIGENT METHODS

Key words: artificial intelligence, classification, machine learning, Parkinson's disease, artificial neural network

UDK: 004.93.021(043.2)

ABSTRACT

This Master's thesis discusses artificial intelligence, machine learning, classification of complex patterns and feature selection procedure. Some of the most used classification algorithms are introduced. Algorithm for the detection of Parkinson's disease based on sound measures has been made. Sound measurements of forty individuals were used as a dataset. Half of the individuals are healthy and half have the Parkinson's disease. Purpose of this thesis is to present robust system for Parkinson's disease detection. Few different feature selection techniques (Pearson's correlation coefficient, Khendall's correlation coefficient and Self-organizing maps) and neural network topologies have been used for improving classification accuracy. With the use of feed-forward neural network 86,47 % accuracy was achieved based on Pearson's correlation coefficient.

KAZALO VSEBINE

KAZALO SLIK

KAZALO PREGLEDNIC

1 UVOD

Sodobnega življenja brez računalnikov si ne moremo več predstavljati. Njihova vloga postaja vse pomembnejša. Računalniki so sposobni prevzeti naloge, ki so bile še do nedavnega v domeni ljudi. Kar pa ni nujno slabo, kajti pri stroju obstaja manjša verjetnost povzročitve napake. Stopnja ponovljivosti in natančnosti je višja. Ni padca pozornosti. Poleg tega človeku njegove kognitivne lastnosti pogosto ne zadoščajo, da bi se dovolj učinkovito in zanesljivo odločal v kompleksnih situacijah. Sposobnost računalnikov za obdelovanje in shranjevanje podatkov iz leta v leto narašča, tudi v smislu zapletenosti programov, ki jih izvajajo. Veča se tudi potreba po obdelavi velike količine podatkov in po samostojnem odločanju manjših in večjih sistemov. Ti sistemi morajo pogosto hitro reagirati na zunanje spremembe in ne morejo čakati na človekove odločitve. V zadnjem desetletju se je povečalo zanimanje za klasifikacijske algoritme in njihovo uporabo na področjih kmetijstva, ekonomije, strojništva, farmacije, kemije, medicine itd. Razcvet omenjenih algoritmov je omogočil predvsem razvoj računalnikov, ki so zmožni z bolj ali manj preprostimi operacijami obdelati veliko količino podatkov, obenem pa imajo na voljo prostor, kjer te podatke skladiščijo.

1.1 Opredelitev oz. opis problema, ki je predmet raziskovanja

Hiter življenjski tempo povečuje potrebo po strojih, ki so sposobni inteligentnih odločitev. Stroji so tako pripravljeni sprejemati odločitve, ki temeljijo na kompleksnem razmerju vhodnih parametrov. V magistrskem delu se bomo lotili problema klasifikacije kompleksnih vzorcev. Opisali bomo različne klasifikatorje in algoritme za izbiro značilk. Podrobneje bomo opisali nevronske mreže, katere bodo služile kot pripomoček za reševanje problema. Na podlagi zvočnih meritev se bomo lotili kompleksnega problema diagnosticiranja Parkinsonove bolezni (PB). V magistrskem delu bomo opisali nov način zaznave PB, in sicer na podlagi zvoka oz. glasilk. Do sedaj so PB diagnosticirali s pomočjo scintigrafijske možganske slike. Takšna diagnostika pa predstavlja veliki strošek za bolnišnice ter napor za bolnike, ki so običajno pripadniki starejših generacij.

1.2 Cilji in raziskovalne hipoteze magistrskega dela

Namen magistrskega dela je čim bolj podrobno spoznati različne klasifikatorje in jih tudi implementirati na testnem primeru določanja PB. Cilj naloge je, s pomočjo umetne inteligence oz. klasifikatorjev, čim uspešneje razvrstiti vzorce. Želimo določiti tudi ustrezno strukturo nevronske mreže in izločiti posamezne parametre zvoka, ki vsebujejo premalo informacijo prisotnosti PB. Hipoteza magistrske naloge je, da je PB mogoče diagnosticirati s pomočjo umetne inteligence.

1.3 Predpostavke in omejitve raziskave

Magistrsko delo bo slonelo na podlagi 1040-ih zvočnih posnetkov (26 zvočnih posnetkov na posameznika). V raziskavi je sodelovalo 40 oseb (20 jih boleha za PB, ostalih 20 pa predstavljajo zdravi posamezniki).Zvočni posnetki so predhodno obdelani s programsko opremo Praat, s pomočjo katere so pridobljeni akustični parametri. Za implementacijo nevronskih mrež bo uporabljen programski paket Matlab.

Struktura magistrskega dela je sledeča: drugo poglavje služi kot uvod v področje umetne inteligence. Tukaj se spoznamo s pojmom inteligentnega obnašanja v okolju, zgodovino umetne inteligence in nekaterih rešitev, ki jih je umetna inteligenca ponudila večjim svetovnim podjetjem. Tretje poglavje govori o strojnem učenju. Opisan je postopek učenja razvrščevalnega algoritma, določitve značilk, način ocenjevanja stopnje uspešnosti razvrščanja in kaj sploh pomeni dobro razvrščanje. Četrto poglavje predstavlja nekatere najpopularnejše in ponavadi najaplikativnejše razvrščevalne oz. klasifikacijske algoritme. Opisani so nekateri matematični postopki in način operiranja algoritmov za namen uspešne razvrstitve vzorcev. Sledi peto poglavje, kjer je prikazan problem razvrščanja vzorcev, in sicer problem zaznave Parkinsonove bolezni s pomočjo glasilk. Poglavje opisuje sestavo programskih kod, podaja rezultate ter jih primerja z rezultati drugih, neodvisnih avtorjev.

2 UMETNA INTELIGENCA

Človeški rod sam sebe imenuje »homo sapiens«, kar v neposrednem prevodu pomeni pametni človek. Termin nakazuje kolikšen pomen pripisujemo (nam prirojeni) inteligenci. Tako imenovano inteligenco povezujemo z miselnimi procesi, ki potekajo v cerebralnem korteksu. Ta ima eno izmed ključnih vlog pri procesih kot so čutenje (fizično in psihično), govor, pozornost, spomin, zavedanje in predvidevanje posledic. Človek in ostali sesalci imajo sposobnost hitrega učenja ter prilagajanja novonastalih vzorcev obnašanja. Plazilci na primer lahko oblikujejo nove vzorce vedenja oz. svoje obnašanje prilagodijo okolju šele v obdobju večjih generacij. Na podlagi anatomije človeka lahko sklepamo, da sama materija kot skupek atomov, ki sestavljajo naše možgane, pripelje do inteligentnih procesov. Ti potekajo v več nivojih. Inteligenca na nek način predstavlja sposobnost procesiranja in podajanja informacij v danem okolju.

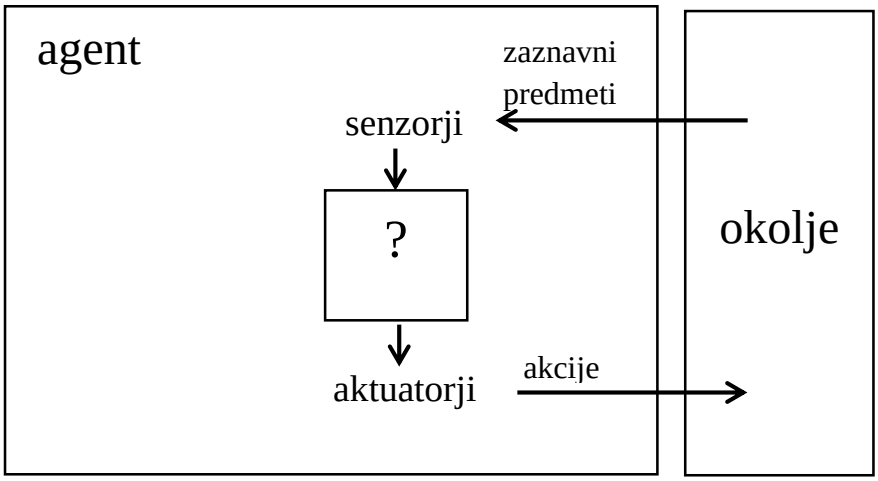
Učenje, znanje in inteligenca so močno povezani pojmi. Čeprav ne obstaja splošno veljavna definicija inteligence, jo lahko grobo opredelimo kot sposobnost prilagajanja okolju in sposobnost reševanja problemov. Že v sami definiciji se pojavita izraza učenje – prilagajanje. Za reševanje problemov je tako nujno potrebno znanje in njegova uporaba. Področje umetne inteligence se tako ukvarja z razvojem sistemov, ki se obnašajo inteligentno in so sposobni reševati relativno težke probleme. Pogosto temeljijo na oponašanju človekovega načina reševanja problemov. Umetna inteligenca pokriva področja strojnega učenja, računalniškega zaznavanja, predstavitev znanja, razumevanja naravnega jezika, avtomatskega sklepanja, logičnega programiranja, kvalitativnega modeliranja, igranja iger, hervinističnega reševanja problemov, robotike, kognitivnega modeliranja, avtomatskega sklepanja in dokazovanja izrekov [1].

Cilj umetne inteligence je razviti naprave in metode, ki se vedejo, kot da bi razpolagale z inteligenco oz. izdelati stroj, ki posnema človeško razmišljanje (z zavestjo in čustvi). Lahko jo opišemo tudi kot študij inteligentnih agentov, ki zaznavajo in sprejemajo podatke iz okolice ter s pomočjo implementirane funkcije izvršujejo akcije. Tako so inteligentni agenti postali glavna unifikacijska tema, ki povezuje zelo različna področja umetne inteligence. Z agenti lahko predstavimo algoritme iz različnih področij, kot so reševanje problemov z iskanjem, igranje iger, avtomatsko sklepanje, planiranje, verjetnostno sklepanje, klasifikacija, razpoznavanje itd. [2].

2.1 Inteligentni agenti

Inteligentni agenti predstavljajo strukturo, s katero lahko povezujemo različna področja znotraj umetne inteligence. Beseda agent izhaja iz latinske besede »agere«, ki pomeni napraviti, storiti. Pojem agent lahko razumemo tudi kot poskus povezovanja navidezno zelo različnih tem (igranje iger, sklepanje, planiranje), ki so predmet obravnave umetne inteligence. Agent je lahko katerakoli virtualna ali pa resnična entiteta, ki zaznava okolje in na to okolje deluje. Po navadi agenti predstavljajo napravo ali program, ki iz okolja s pomočjo senzorjev sprejemajo informacije. Agent na okolje deluje s pomočjo aktuatorjev.

Splošno shemo agenta prikazuje slika 2.1. Za njegove zaznavne vhode, katere dobi iz okolja, uporabljamo izraz zaznavni predmet. Zaporedje zaznavnih predmetov je popolna zgodovina vsega, kar je agent zaznal. V splošnem je lahko izbrana akcija agenta odvisna od celotnega zaporedja zaznavnih predmetov. Njegovo obnašanje je opisano s funkcijo, katera je odgovorna za preslikavo danega zaporedja zaznavnih predmetov v akcijo. Agente lahko opišemo tudi tako, da podamo kriterij učinkovitosti, okolje, aktuatorje in senzorje [2].



Slika 2.1:Agent deluje interaktivno z okolico [2]

Željene lastnosti inteligentnega agenta:

- **Avtonomnost**. Agent pri svojem posegu deluje samostojno in ne potrebuje posegov ustvarjalca. Sposoben je samostojno zaključiti aktivnosti, brez posredovanja človeka. Ključni element avtonomnosti je zmožnost prevzemanja pobude.
- **Poosebitev**. Agent ima sposobnost upoštevanja verjetnostnega značaja, npr. čustev.
- **Racionalnost**. Racionalni agent deluje racionalno v primeru, da za vsako možno zaporedje zaznavnih predmetov izbere akcijo, ki maksimira nek kriterij učinkovitosti.
- **Sposobnost sklepanja**. Agent zna reagirati na nepopolne specifikacije nalog in informacije iz okolja na podlagi predznanja o splošnih ciljih. Sklepanje je lahko preudarnega značaja (zahteva, da agent vsebuje simbolični model sklepanja) ali reaktivnega značaja (agent se simultano odziva na trenutno stanje).
- **Prilagodljivost**. Agent se je sposoben avtomatsko prilagoditi spremembam v okolju, ima sposobnost učenja in izboljšave na podlagi izkušenj.
- **Komunikativnost**. Agent je zmožen sodelovati pri komuniciranju z ljudmi in drugimi agenti. Namen komunikacije je pridobitev čim kvalitetnejših informacij, ki agentu pomagajo pri izpolnitvi izbranih ciljev.
- **Stalnost delovanja**. Agent je sposoben delovati daljše časovno obdobje, oz. nepretrgoma in ne samo enkratno.
- **Sodelovanje**. Agent ne uboga ukazov na slepo. Ima možnost sprotnega spreminjanja in zavračanja zahtev ter postavljanja vprašanj za razjasnitev. Sposoben je sodelovanja z ostalimi agenti, kar največkrat vključuje sporočila na visokem nivoju.
- **Fleksibilnost**. Na podlagi stanja okolja je agent zmožen samostojne izbire akcije. Agentove akcije niso določene z togim scenarijem.

Navedene lastnosti naj bi vseboval vsak inteligentni agent. Vendar se v praksi pojavljajo odstopanja od teh splošnih pravil. Pri komercialni uporabi tako opazimo različne stopnje vsebovanosti določenih lastnosti. Stopnja inteligentnosti agentov (v smislu prilagojenosti in učinkovitosti v danem okolju), pa je odvisna predvsem od zmožnosti njihovih razvijalcev, da implementirajo množico njihovih sposobnosti.

2.1.1 Vrste agentov

V nadaljevanju bodo opisane nekatere vrste agentnih programov, ki predstavljajo skoraj vse inteligentne sisteme [3]:

- **Preprosti odzivni agenti**. So najenostavnejša vrsta agenta. Aktivirajo se v primeru, da je posameznemu pogoju zadoščeno. Akcija agenta se izbere na osnovi tekočega zaznavnega predmeta, pri tem pa celotna zgodovina zaznav ni pomembna. Imajo zelo omejeno inteligenco.
- **Odzivni agenti, temelječi na modelu**. Takšen agent mora vzdrževati notranje stanje, ki je odvisno od zgodovine zaznavanja. Odzivni agenti, temelječi na modelu, se od preprostih odzivnih agentov razlikujejo samo v določitvi stanja. Le-to je sedaj odvisno tudi od predhodnega stanja in akcije agenta. Osveževanje stanja zahteva vključitev informacij o okolju in o posledicah agentovih akcij na okolje.
- **Agenti, temelječi na cilju**. Agent potrebuje ciljno informacijo. Takšni agenti se uporabljajo predvsem v primeru, ko je odločitev agenta odvisna od njegovega cilja. Agentni program, s pomočjo iskalnih strategij in planiranja, določi pot do cilja.
- **Agenti temelječi na koristi**. Ločimo za agenta koristna oz. ugodna ter nekoristna oz. neugodna stanja. Agenti se odločajo v skladu s funkcijo koristi. Ta preslika stanje (ali zaporedje stanj) v realno število. Število pa podaja stopnjo ugodnosti, na podlagi katere agent najde ustrezno pot do odločitve.
- **Učeči se agenti**. Element učinkovitosti je odgovoren za izbiro zunanjih akcij. Sprva element učinkovitosti predstavlja celoten agent. Učeči se element je odgovoren za tvorbo izboljšav. Povratna informacija agentovega delovanja je podana s strani ocenjevalca. Z njegovo pomočjo, učeči se element določi kako spremeniti element učinkovitosti, da bo agent bolje deloval. Eden izmed pomembnih elementov je tudi problemski generator. Ta je odgovoren za predlaganje akcij, ki vodijo do novih in poučnih izkušenj.

2.1.2 Okolje

Okolje predstavlja prostor agentovega delovanja in je vse, kar ni del agenta ter vse, kar agenta obkroža. Agent v okolju operira in prebiva ter nanj vpliva. Lahko je urejeno ali kaotično. Običajno agent operira v okolju, ki je kombinacija spodaj navedenih lastnosti.

Preglednica 2.1: Lastnosti okolja [4]

Lastnost okolja	Opis
Zaznavno in delno zaznavno.	Pravega agenta opredeljuje sposobnost zaznave okolja (posledično je tudi okolje opredeljeno zaznavno). Ponavadi so v celoti zaznavna okolja preprosta. Večina okolij v realnosti je delno zaznavnih.
Deterministično in stohastično.	Popolnoma deterministično okolje je tisto, pri katerem so vsa prihodnja stanja okolja določljiva s pomočjo sedanjega stanja in akcij agenta. Stohastično okolje je tisto, ki vsebuje določeno stopnjo negotovosti ali nepoznanih zunanjih vplivov.
Epizodično in sekvenčno.	Okolje je epizodično, če se katera izmed nalog agenta ne nanaša na preteklo učinkovitost ali pa v primeru, če naloga agenta ne more vplivati na njegovo prihodnjo učinkovitost . Sicer je okolje sekvenčno.
Statično in dinamično.	Statično okolje se skozi čas ne spreminja. Dinamično okolje je tisto, ki se s časom spreminja in v katerem ima agent možnost odzivanja ali neodzivanja na spremembe.
Diskretno in zvezno.	Diskretno okolje imana voljo končno število stanj.V zveznem pa okolju je število stanj neskončno..
Eno-agentno ali več-agentno.	Več-agentno okolje je tisto, v katerem agenti medsebojno sodelujejo in tekmujejo. V nasprotnem primeru, agent ostale vidi kot del stohastičnega okolja.

2.2 Zgodovina umetne inteligence

Temelje umetne inteligence so postavili številni filozofi in matematiki že pred nekaj stoletji. Koncepte umetne inteligence zasledimo že v grški mitologiji. Kot prvega, ki je zaslužen za postavitev temeljev umetne inteligence smatramo grškega filozofa Aristotela, ki je postavil temelje logike in deduktivnega sklepanja [5].

Že pred približno 400 leti so ljudje pisali o naravi misli. Hobbes, je poudaril, da je sklepanje simbolno tako kot govorjenje na glas ali odgovarjanje s svinčnikom in papirjem. Idejo simboličnega sklepanja so nadalje razvili Descartes, Pascal, Spinoza, Leibnitz in drugi pionirji filozofije mišljenja [6].

Leta 1950 je Alan Turing, ki je bil takrat vodja laboratorija na manchesterski univerzi, objavil članek, ki ponazarja imitacijsko igro. Članek je bil objavljen v filozofski reviji Mind in je kasneje služil kot podlaga za znani koncept Turingovega testa.

Za formalni začetek umetne inteligence se šteje konferenca v Dartmounthu v ZDA, leta 1956. Konferenco je organiziral docent matematike na Dartmounthu, John McCarthy. Predlagal je dvomesečni študij umetne inteligence na Dartmounth Collegeu. Namen študija naj bi bil popis in simulacija različnih vidikov učenja s pomočjo računalnika. McCarthy je leta 1958 zasnoval programski jezik Lisp, ki je postal najbolj razširjen jezik na področju umetne inteligence [2].

Arthur Samuel (IBM) je leta 1952 zgradil program Dama, ki se je učil in igral damo (igra s kartami). Program je dosegal primerljivo inteligenco z najboljšimi resničnimi igralci. Herb Gelernter (IBM) je leta 1959 napisal prvi program za dokazovanje teoremov diferencialne matematike [2,5].

Leta 1961 je James Slagle, v doktorski dizertaciji na MIT-ju in z uporabo programa Lisp, napisal prvi shematični integracijski program, ki je sposoben rešiti različne diferencialne probleme na stopnji univerzitetnega študenta [5].

Med leti 1970 in 1980 so se strokovnjaki na področju umetne inteligence posvečali predvsem ekspertnim sistemom, katerih cilj je bil zajeti znanje različnih strok. Namen tega je bil računalnik, ki bi lahko namesto njih opravljal zahtevne naloge (npr. na področju medicine). Med leti 1965 in 1983 sta Buchanan in Feigenboum razvijala projekt DENDRAL, ki je služil za analizo molekul v organski kemiji [6].

V 90-ih letih 20-ega stoletja je bilo veliko napredka na področju umetne inteligence. Napredek so povzročili predvsem vse bolj dostopni osebni računalniki in vse zmogljivejša strojna oprema. Inteligentni sistemi so bili sposobni razumevanja naravnega jezika oz. človeškega govora, prevajanja besedila, strojnega vida, virtualne resničnosti, mehkega sklepanja, planiranja, inteligentnega poučevanja itd. Leta 1984 sta Buchanan in Shortliffe razvila MYCIN. Gre za ekspertni sistem, ki je s pomočjo umetne inteligence prepoznaval bakterije, povzročiteljice različnih infekcij.

Konec 20-ega stoletja so bile razvite in komercialno dostopne »pametne igrače«. To so bili roboti, ki so posnemali obnašanje domačih živali. Leta 1997 je bila organizirana prva uradna tekma robotskega nogometnega prvenstva. Istega leta je NASA izvedla prvi pristanek avtonomnega robota na Marsu. Poimenovala ga je Sojourner [5].

Danes je umetna inteligenca splošno uveljavljen pojem. Njena uporaba je prisotna na več področjih, od zasnove internetnih strani, napovedovanja delnic, prepoznave predmetov, prepoznave izrazov in povezovanja s čustvenimi stanji ljudi do učenja na podlagi več milijonov primerov. Uporablja se tudi na področju internetne varnosti, zavarovalništva, na področju glasbe itd. Zato lahko z velikimi pričakovanji zremo v prihodnost. Pričakujemo lahko nadaljnji razvoj in prodor umetne inteligence v vsakdanje življenje ter njen doprinos h kvalitetnejšemu in varnejšemu življenju ljudi.

2.3 Uporaba umetne inteligence

Uporaba umetne inteligence je vse bolj v porastu. Če je bila še pred nekaj desetletji predvsem domena znanstveno fantazijskih filmov, jo dandanes lahko zasledimo v zelo različnih oblikah. Umetna inteligenca je dosegla nivo, kjer je preko raznih programskih jezikov splošno dosegljiva in uporabljiva v komercialne namene. V nadaljevanju navajamo nekaj primerov uporabe umetne inteligence znanih svetovnih podjetij.

Apple je na podlagi umetne inteligence zasnoval pseudo-inteligentno, digitalno in osebno asistentko SIRI. Uporabnikom Apple računalnikov pomaga najti informacije, dodati dogodke na koledar, pomaga pri pošiljanju sporočil itd. Gre za uporabniku prijazen računalniški modul. SIRI za svoje učenje uporablja tehnike strojnega učenja(postaja vedno bolj pametna in prilagojena potrebam) [7].

Amazon uporablja transakcijski modul, kar mu omogoča spopadanje z astronomskim številom transakcij in realizacijo velikih dobičkov. Algoritmi opisanega modula postajajo vse bolj dovršeni in s tem pripomorejo k uspešnejšemu napovedovanju potencialnih kupcev, glede na njihovo obnašanje na svetovnem spletu. V prihodnosti Amazon planira, da bi modul postal tako napreden, da bi potrošnikom izdelke poslal, še preden bi sami ugotovili, da jih potrebujejo oz. preden bi jih naročili[7].

Neflix ima (kot ponudnik video vsebin) oblikovan napovedovalni modul, ki uporabnikom svetuje ogled vsebin. Modul analizira na milijone video posnetkov in predlaga tiste, ki bi posameznika utegnile zanimati. Uporabnikom svetuje glede na njihove pretekle izbire in reakcije. Z večanjem baze video vsebin postaja omenjeni modul, v smislu napovedovanja, iz leta v leto uspešnejši [7].

Facebook koristi metode umetne inteligence za zaznavo in prepoznavo obrazov. Uporablja pa tudi inteligentni protiteroristični modul, ki prepozna propagandne teroristične slike ali video vsebine ter sovražna sporočila, ki so kakor koli povezana z terorizmom [7].

3 STROJNO UČENJE IN OSNOVE RAZVRŠČANJA

Strojno učenje je področje umetne inteligence in zajema tehnike, s katerimi se stroj (računalnik) nauči reševanja določenih specifičnih in ozko usmerjenih nalog. To mu omogočajo podatki ali izkušnje iz preteklih izvajanj. Skupna točka vseh tehnik strojnega učenja je, da se stroj uči na že rešenih podatkih, s pomočjo katerih se ustvari model. Podatki so združeni v podatkovne množice (»dataset«), ki so oblikovane kot preproste tekstovne datoteke ali podatkovne baze. Model se kasneje uporabi pri reševanju podobnih, a nepoznanih problemov.

V zadnjih letih je področje strojnega učenja v razcvetu. Njegova uporaba je razširjena v različnih okoljih in aplikacijah. Vse večja dostopnost podatkov omogoča sistemom strojnega učenja, da oblikujejo modele, ki temeljijo na veliki količini podatkov. Sama rast procesne moči računalniških sistemov pa ponuja analitično podlago sistemom strojnega učenja, za spopadanje s problemi [8].

Glede na tehnike učenja, delimo strojno učenje na [9]:

- **Nadzorovano učenje**: je praktično sinonim za razvrščanje in regresijo. Nadzorovano učenje uporabimo, kadar želimo, da se stroj nauči razvrščati (klasificirati) podatke, ali pa jim pripisovati številske vrednosti (regresija). Stroj se uči iz primerov (podatkovne množice) z že znanimi izhodi (odvisna spremenljivka). Izhodi posameznih elementov podatkovne množice, v primeru razvrščanja vzorcev, predstavljajo pripadnost posameznemu razredu.
- **Nenadzorovano učenje**: je sinonim za grupiranje. Nenadzorovano učenje uporabimo, ko želimo odkriti še neznane povezave in strukturo med podatki. Za izvedbo nenadzorovanega učenja ne potrebujemo izhodov podatkovne množice, ampak le vhode (neodvisne spremenljivke). Število gruč je lahko vnaprej določeno ali pa se odločitev o številu gruč prepusti stroju. Nenadzorovano učenje se pogosto uporablja tudi za preslikavo podatkovne množice (brez izhodov) v višji ali pa nižji prostor značilk.
- **Delno nadzorovano učenje**: gre za posebno vrsto učenja, kjer imajo nekateri objekti podane izhode, drugi pa ne. Na začetku se postavi model na podlagi objektov z podanimi izhodi. Nato se model prilagodi tako, da se upoštevajo še objekti, katerih izhodi niso podani. Prednost modela je predvsem v tem, da lahko izkoristimo vse podatke, ki so nam na voljo. Primer delno nadzorovanega učenja je iskanje anomalij, kjer nam ni poznano, kakšna so odstopanja od nekega povprečja.

- **Okrepitveno učenje**: je učenje, kjer se sistem uči na podlagi nagrajevanj ali kazni , odvisno od izidov učenja. Pri nagrajevanju lahko sodeluje človek, ali pa je podana ocenitvena funkcija. Človek v določenih primerih posreduje dodatne informacije o samih podatkih ali pa v iterativnem postopku poda mnenje o kvaliteti modela.

Poudarek strojnega učenja je na implementaciji metod, s katerimi stroj sam pride do spoznanj, brez vmesnega posredovanja človeka. Namesto eksplicitnega programiranja, pri katerem bi računalnik naučili, da reši izbrano nalogo skozi naše ukaze, se pri strojnem učenju osredotočimo predvsem na metode, ki same pridejo do ugotovitev. Stroj bo tako samostojno kreiral model, kateri mu bo omogočal določeno stopnjo uspešnosti. Stopnja uspešnosti je v splošnem odvisna od izbranega modela, nastavitev modela, velikosti in raznolikosti podatkovne množice, izbire značilk itd. Strojno učenje je zelo širok pojem. V sklopu magistrske naloge se bomo omejili na učenje razvrščevalnih oz. klasifikacijskih algoritmov (KA) na podlagi učnih primerov z podanimi značilkami.

3.1 Razvrščanje

Osrednja metoda strojnega učenja v okviru tega magistrskega dela, je metoda razvrščanja oz. klasifikacije, kjer stroj naučimo razvrščati objekte v vnaprej določene razrede. Z regresijo napovemo številske vrednosti (zvezne vrednosti), pri razvrščanju pa napovemo nominalne vrednosti (diskretne vrednosti). Razvrščanje se uporablja v primerih, kot so prepoznava vzorcev na slikah ali kamerah, preprečevanje nezaželene elektronske pošte, prepoznavanje pisave, prepoznavanje govora, preprečevanje prevar in diagnosticiranje bolezni. Če podatke razvrščamo v dva razreda, govorimo o binarni klasifikaciji, sicer govorimo o večrazredni klasifikaciji.

V nadaljevanju bomo za metodo klasifikacije uporabljali naslednjo definicijo objektov. En objekt je par $(\boldsymbol{x}_i, y_i)$, kjer je $\boldsymbol{x}_i$ vhod ali vektor vrednosti (značilk) tega objekta in y_i izhod ali dejanski razred objekta (skalarna vrednost). Podatkovna množica *X* je definirana kot množica vseh objektov. Iz slednje množice izberemo objekte za učenje razvrščevalnega algoritma.

$$X = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_p, y_p)\}$$

$$\boldsymbol{x}_i = (x_i^1, x_i^2, \dots, x_i^l) \tag{3.1}$$

$$y_i \in \{razred_1, razred_2, \dots, razred_k, \}$$

$$p = \check{s}tevilo\ objektov$$

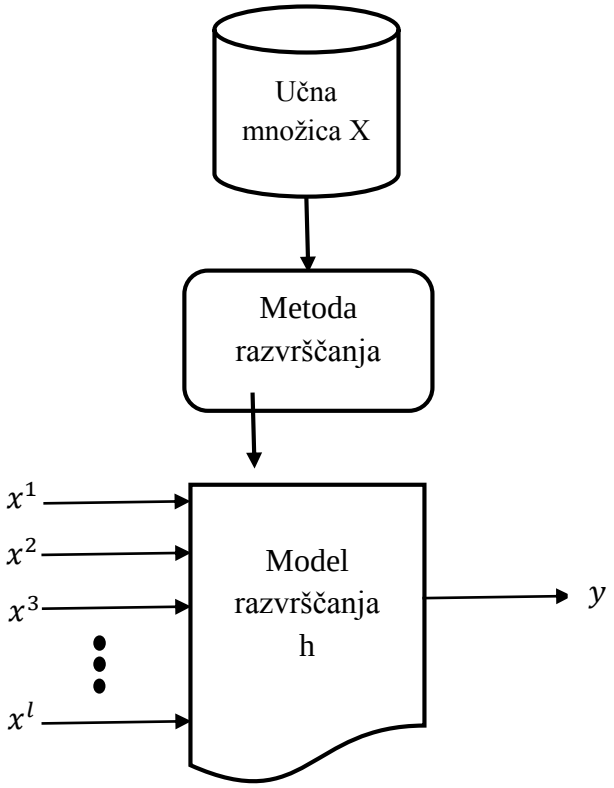
$$l = \check{s}tevilo\ značilk$$

$$k = \check{s}tevilo\ razredov$$

Metodološki cilj nadzorovanega učenja, in s tem tudi razvrščanja, je najti **razvrščevalni** ali **klasifikacijski model** h, ki popiše izbrano množico objektov. Model h je približek resnične nepoznane funkcije f. Ta nam je nepoznana iz različnih razlogov, kot so npr. premajhne količine podatkov, neupoštevanje vseh dejavnikov, nereprezentativnega vzorca, merilne negotovosti, šuma vsebovanega znotraj meritev itd.

$$y = f(\boldsymbol{x}) \approx h(\boldsymbol{x}) \tag{3.2}$$

Spodnja slika prikazuje postopek razvrščanja in formiranja modela klasifikacije h. Cilj učenja je, da zgradimo klasifikacijski model tako, da posnema sistem, ki je podlaga same domene učne množice oz. podatkovne množice. Po navadi je bolj natančno in učinkovito oponašanje tudi bolj uspešno.



Slika 3.1: Splošni proces razvrščanja

3.2 Določitev značilk

Določitev značilk postaja čedalje bolj pomembna, saj imamo v sodobnem svetu na voljo veliko podatkov, ki imajo v veliko primerih po več tisoč značilk. Poraja se vprašanje, kako izločiti značilke, kako izbrati le tiste, ki o našem problemu povedo največ. V procesu določitve izberemo le tiste, ki so za naš problem najpomembnejše. Izločimo pa tiste, ki ne nosijo dovolj informacij o danem problemu. Določitev značilk nam omogoča uspešnejše delovanje razvrščevalnega algoritma, saj z eliminacijo nepomembnih značilk zmanjšamo stopnjo prekomernega prilagajanja (»over-fitting«). Poleg tega z zmanjšanjem števila le-teh pridobimo na hitrosti razvrščevalnega algoritma, saj se razvrščevalnemu modelu, tekom treniranja, ni potrebno prilagajati v tolikšni meri, kot bi se moral pri večjem številu značilk. Določitev lahko poteka tudi z transformacijo prvotnih značilk dimenzijski prostor.

3.2.1 Pearsonov korelacijski faktor

Pearsonov korelacijski faktor je eden izmed najbolj uporabljenih korelacijskih faktorjev v statistiki. Uporablja se za merjenje odvisnosti med dvema spremenljivkama. Gre za parametrični test, kar pomeni, da stoji na predpostavki, da imata spremenljivki normalno razporeditev. Želimo dobiti korelacijo med dvema spremenljivkama $\{x_1, x_2, \dots, x_n\}$ in $\{y_1, y_2, \dots, y_n\}$. Pearsonov korelacijski faktor r je izračunan za vsako značilko po formuli:

$$\mathrm{r} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^{n}(x_i - \bar{x})^2][\sum_{i=1}^{n}(y_i - \bar{y})^2]}} \tag{3.3}$$

kjer je $\bar{x}$ povprečna vrednost značilke, x_i vrednost i-tega objekta. V primeru, da je vrednost Pearsonovega korelacijskega koeficienta $r = 1$, imata značilka in pripadnost razredu največjo pozitivno korelacijo. Če je vrednost $r = -1$, imata značilka in pripadnost razredu največjo negativno korelacijo. V primeru, da je $r = 0$, med značilko in pripadnostjo razredu ni korelacije.

3.2.2 Khendallov korelacijski faktor

Khendalov korelacijski koeficient predstavlja stopnjo skladnosti (več inverzij pomeni manjši koeficient) med dvema stolpcema po velikosti razvrščenih in rangiranih pozicij spremenljivk. Gre za neparametrični test, ki ne stoji na nobeni predpostavki o statističnem raztrosu spremenljivk. Želimo dobiti korelacijo med spremenljivkama $\{x_1, x_2, \dots, x_n\}$ in $\{y_1, y_2, \dots, y_n\}$Določen par spremenljivk (x_j, y_j) je skladen z (x_i, y_i), kjer $i \neq j$. To velja, kadar se rangirane pozicije ujemajo. Khendallov korelacijski faktor τ_b je za vsako značilko izračunan po formuli:

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{3.4}$$

$$\mathrm{n}_0 = \mathrm{n}(\mathrm{n} - 1)/2 \tag{3.5}$$

$$\mathrm{n}_1 = \sum_i t_i(t_i - 1)/2 \tag{3.6}$$

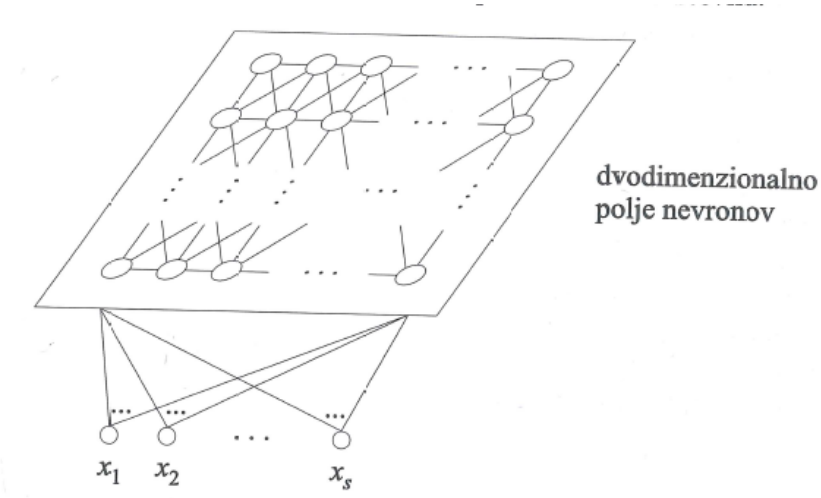
$$\mathrm{n}_2 = \sum_j u_j(u_j - 1)/2 \tag{3.7}$$

kjer je n_c število skladnih parov, n_d število neskladnih parov, t_i je število izenačenih rangiranih pozicij za prvo spremenljivko in u_j je število izenačenih rangiranih pozicij za drugo spremenljivko. V primeru, da je vrednost Khendallovega korelacijskega koeficienta $\tau_b = 1$, imata značilka in pripadnost razredu največjo pozitivno korelacijo. Če je vrednost $\tau_b = -1$, imata značilka in pripadnost razredu največjo negativno korelacijo. V primeru, da je $\tau_b = 0$, med značilko in pripadnostjo razredu ni korelacije.

3.2.3 Samoorganizacijske gruče

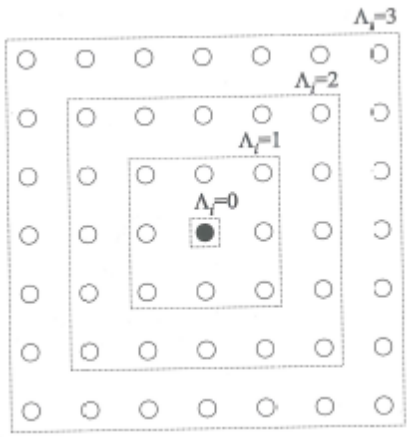
Samoorganizacijske gruče (SOM) je leta 1982 razvil Kohonen, ko je skušal posneti bistvo računskih preslikav v možganih. Samoorganizacijske gruče ali Kohonenove nevronske mreže, kot je že v imenu nakazano, temeljijo na principu samoorganizacije, ki jo dosežemo z uporabo nenadzorovanega učenja. Ideja samoorganizacije je v avtomatski klasifikaciji razredov vzorcev, ki so predstavljeni v n-dimenzionalnem prostoru značilk. Algoritem SOM samodejno zaznava gruče v prostoru brez vnaprej znane klasifikacije vzorcev. Zaznava gruč poteka tako, da se utežni vektorji nevronov skozi proces nenadzorovanega učenja prilagodijo posameznim gručam vzorcev in tako postanejo neke vrste reprezentativni vzorci posameznih razredov [2].

Dvodimenzionalni Kohonenov model nevronske mreže predstavlja slika 3.2. V splošnem gre za nevronsko mrežo, kjer so nevroni med seboj povezani v 1D, 2D ali 3D mrežo. Vhodni podatki imajo obliko vektorja dimenzij $(x_1, x_2, \dots, x_s)$.



Slika 3.2: Dvodimenzionalni Kohonenov model [2]

Ker so nevroni v Kohonenovi mreži lokalno povezani, govorimo o soseščini nevrona. Oblike sosešči̇n nevrona so prikazane na sliki 3.3, kjer vidimo sosešči̇no Λ_i , ki je dvodimenzionalna in ima obliko kvadrata.



Slika 3.3: Kvadratna sosešči̇na dvodimenzionalne Kohonenove mreže [2]

Spodaj je prikazan pseudo-algoritem delovanja SOM. Značilnost delovanja SOM je ta, da je klasifikacija, oz. preslikava vhodnih vzorcev v razrede, topološko urejena. To pomeni, da se sosednji nevroni v mreži odzivajo na sosednje razrede vzorcev (tj. razrede, ki so si podobni in v prostoru značilk blizu). V 2. vrstici se izvrši naključna izbira začetnih vrednosti uteži $\boldsymbol{w_j}(0)$. Zaželeno je, da so te vrednosti majhne. V 6. vrstici izberemo zmagoviti nevron, tako da

uporabimo kriterij najmanjše razdalje. V 7. vrstici prilagodimo utežne vektorje sosednjih nevronov v skladu z enačbo:

$$\boldsymbol{w_j}(\mathrm{n}+1) = \begin{cases} \boldsymbol{w_j}(n) + \eta(n)[\boldsymbol{x}(n) - \boldsymbol{w_j}(n)], & j \in \Lambda_{i(x)}(n) \\ \boldsymbol{w_j}(n), & sicer \end{cases} \tag{3.8}$$

kjer je n trenutna iteracija (»epoch«), $\eta(n)$ hitrost učenja in $\Lambda_{i(x)}(n)$ funkcija sosednosti zmagovitega nevrona. Tako $\eta(n)$ kot $\Lambda_{i(x)}(n)$ se dinamično spreminjata tekom učenja, in sicer z namenom doseganja čim boljših rezultatov.

Alogritem Samoorganizacijske gruče SOM

Vhodi: $\boldsymbol{x}$ – vhodni vzorci

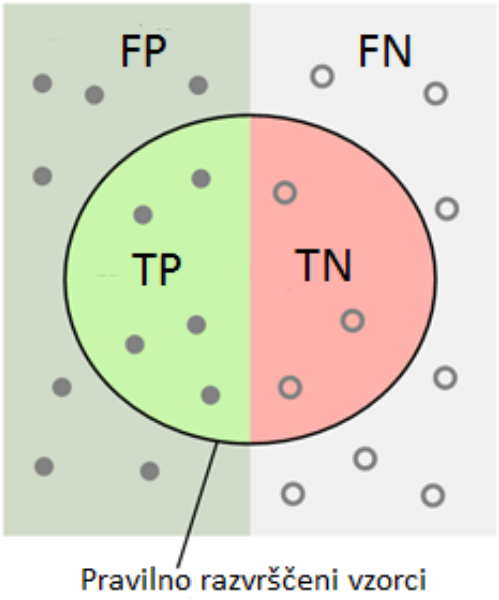
Izhodi: ustrezni izhodni nevron

1: **function**: KOHONENOVA MREŽA
2: inicializacija začetnih uteži $\boldsymbol{w}$
3: **while** razlika med novo in staro mrežno utežjo ni dovolj majhna **do**
4: **foreach** učni vzorec $\boldsymbol{x}$ => naključni vrstni red jemanja vzorcev
5: izračunaj razdalje med vektorjem $\boldsymbol{x}$ in vektorji $\boldsymbol{w}$ posameznih nevronov v soseščini
6: izmed nevronov izberemo tistega, ki ima najmanjšo razdaljo med vektorjema $\boldsymbol{x}$ in $\boldsymbol{w}$
7: prilagodi uteži v vsej mreži oz. v predpisani soseščini nevronov Λ
8: zmanjšaj soseščino nevronov Λ in hitrost učenja η
9: **end foreach**
10: **end while**
11: **end function**

Slika 3.4: Pseudo-algoritem delovanja SOM [2]

3.3 Metrike evalvacije razvrščanja

Podatke se v primeru binarnih vrednosti izhodnih spremenljivk najbolje pokaže s pomočjo množice (slika 3.5) ali pa s pomočjo kontingenčne matrike (preglednica 3.1). V slednji so v zgornjem levem kotu pravilno klasificirani pozitivni rezultati (TP), v desnem zgornjem pa napačno klasificirani pozitivni rezultati (FP), v spodnjem levem kotu so napačno klasificirani negativni rezultati (FN) in v spodnjem desnem kotu pravilno klasificirani negativni rezultati (TN).



Slika 3.5: Prikaz binarnih podatkov kot množice [11]

Preglednica 3.1: Kontingenčna matrika

	Označeno kot P	Označeno kot N
Napovedano kot P	TP	FP
Napovedano kot N	FN	TN

Natančnost se v primeru vrednosti izhodnih spremenljivk meri kot pogostost primernega odziva oz. predikcije, s strani razvrščevalnega algoritma. Je razmerje med številom ustreznih in številom vseh predikcij (število vseh objektov):

$$\text{natančnost} = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.2}$$

Občutljivost in **specifikativnost** sta statistični metriki za pravilno razvrščene pozitivne in negativne primere in sta enaki [11]:

$$\text{občutljivost} = \frac{TP}{TP + FN} \tag{3.3}$$

$$\text{specifikativnost} = \frac{TN}{TN + FP} \tag{3.4}$$

MCC je mera, ki kaže kvaliteto binarne klasifikacije. Gre za Matthewsov korelacijski koeficient. Ta pokaže povezavo med predvidenimi in opazovanimi binarnimi klasifikacijami. Njegove vrednosti se gibljejo med +1 in -1. MCC da vrednost +1, kadar razvrščevalni algoritem poda popolno predikcijo, vrednost -1, kadar je napoved povsem napačna in vrednost 0, kadar gre le za naključno ugibanje KA [12].

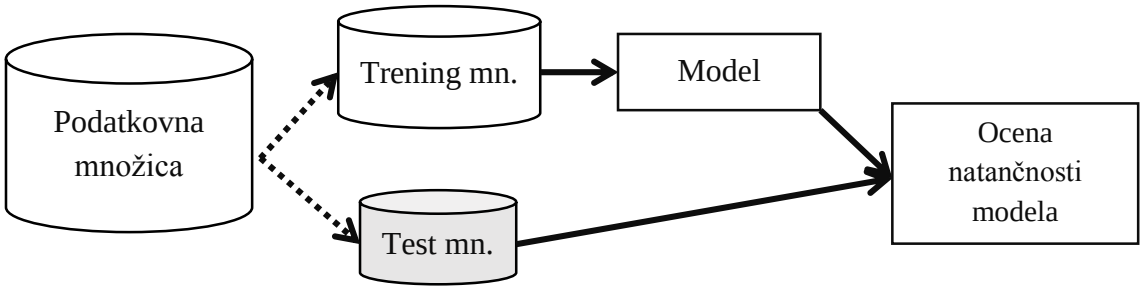
$$\text{MCC} = \frac{TP\ x\ TN + FP\ x\ FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.5}$$

3.4 Tehnike validacije

Za začetek se moramo vprašati, kateri so tisti podatki, ki so najprimernejši za merjenje učinkovitosti razvrščanja. Učinkovitost lahko ocenimo na podlagi podatkov, s pomočjo katerih smo oblikovali model. Kar je deloma prav, vendar pa se je treba zavedati, da bi bila učinkovitost takega modela napovedana preveč optimistično. Kajti razvrščevalni algoritem je v postopku učenja modela, že prilagodil uteži glede na trening populacijo. Tak model bi torej bil pristranski (»biased«). Primernejša metoda je merjenje glede na podatke, katerih model še ni videl. Podatkovna množica je tako razdeljena na dva dela. Iz prvega se ustvari model, kar lahko poimenujemo učna množica ali trening primeri. Na drugem delu pa se oceni točnost prvega, kar imenujemo testna množica ali testni primeri. V nadaljevanju so prikazane različne tehnike razčlenitve končne množice podatkov in testiranje učinkovitosti poljubnega razvrščevalnega algoritma.

3.4.1 Zadržanje

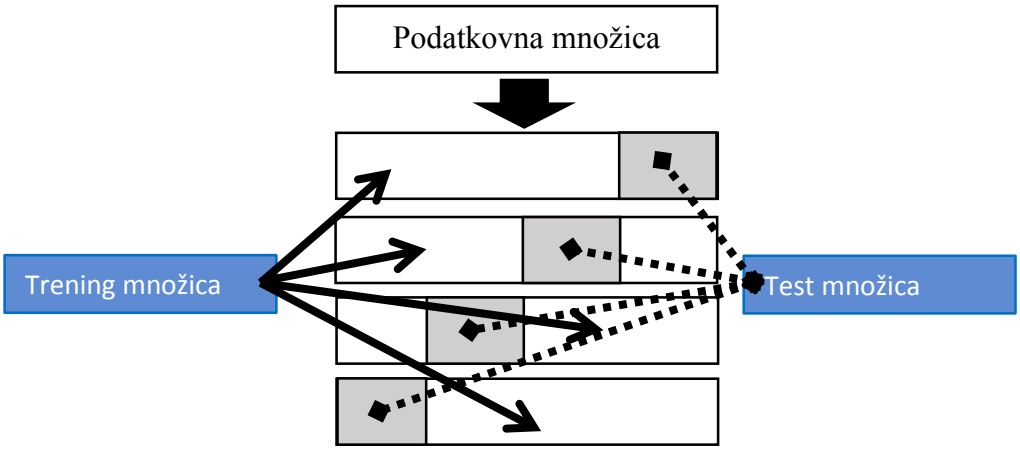
Zadržanje (»hold-out«) je najbolj preprosta tehnika za validacijo delovanja razvrščevalnega algoritma. Zadržanje izvedemo tako, da podatkovno množico razdelimo na trening (običajno od 60 % do 90 % učne množice) in testne primere (običajno od 10 % do 40 %). Postopek delitve prikazuje slika 3.1. Prednost metode je v njeni preprostosti in primernosti za validiranje uspešnosti razvrščanja na velikih podatkovnih množicah. Slabost pa je predvsem v tem, da način razdelitve podatkovne množice vpliva na rezultate razvrščanja. Tako lahko s srečnim naključjem dobimo »na videz« zelo dobre rezultate, ki pa niso pravilni.



Slika 3.6: Validacija poljubnega KA z zadržanjem

3.4.2 Navzkrižna validacija

Pri testu učinkovitosti KA , ki ga izvajamo na majhni populaciji, se običajno uporablja navzkrižna validacija (»cross validation«). Podatkovno množico razdelimo na določeno število manjših vzorcev. Najpogostejša je k-kratna navzkrižna validacija. Slednja poteka tako, da podatkovno množico razdelimo na k enako velikih delov (najpogostejša vrednost števila k je 10). Potem uporabimo $k-1$ vzorcev za učenje ter izgradnjo modela, učinkovitost razvrščanja pa testiramo z izvzetim k-tim vzorcem (test množica v prvi iteraciji). Postopek ponovimo k krat, tako dobimo povprečje uspešnosti izbranega razvrščevalnega modela. Poudarimo, da je za testno množico vsak k-ti del uporabljen natanko enkrat. Prednost te metode je v tem, da za validacijo razvrščevalnega modela uporabimo celotno podatkovno množico.

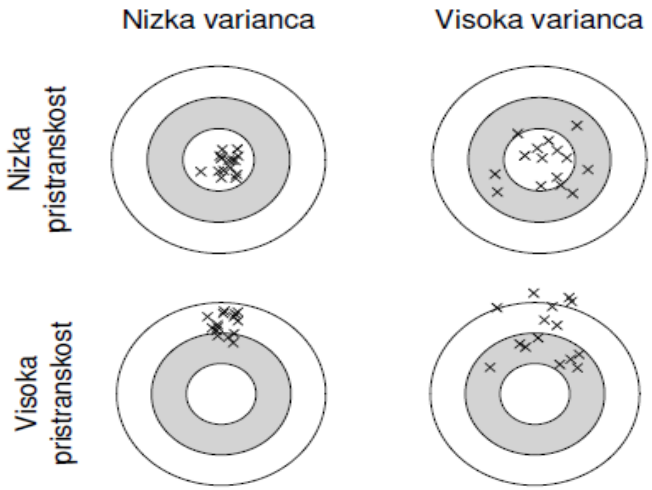


Slika 3.7: Navzkrižna validacija (k=4) poljubnega KA

Pogosto se uporablja tudi **stratificirana navzkrižna validacija**, kjer so elementi posameznega k vzorca izbrani tako, da med njimi velja določeno razmerje kar se tiče pripadnosti. Poseben primer je tudi **navzkrižna validacija izpusti-enega LOSO**, kjer je k enak številu elementov učne množice. LOSO se uporablja, kadar imamo opravka z zelo majhno učno množico. V našem primeru bomo za ugotavljanje učinkovitosti razvitega sistema uporabili LOSO validacijsko shemo.

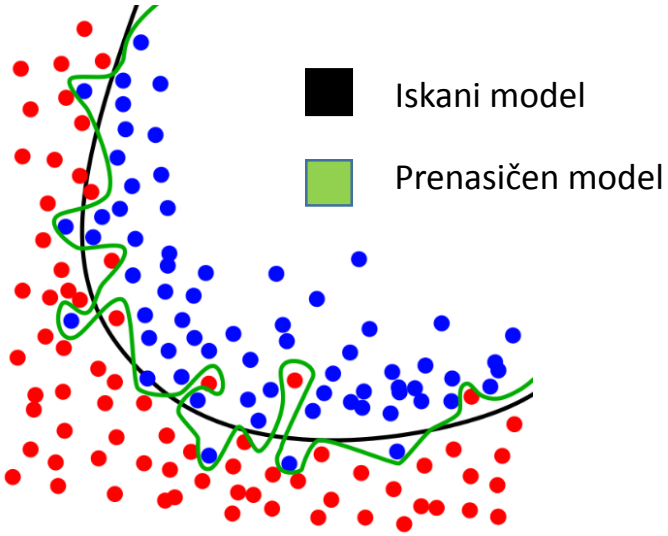
3.5 Prenasičenje

Pri prenasičenju (»over-fitting«) razvrščevalni algoritem vrne model, ki se prekomerno prilagodi trening množici. Tak model daje zelo dobre rezultate, če natančnost preverjamo na trening množici, na podlagi katere je bil tudi zasnovan. Ob tem pa navadno ne daje dovolj dobrih (optimalnih) rezultatov na testni množici. Pravimo, da ima takšen model slabo sposobnost generalizacije. Bistvo učenja je sposobnost generalizacije, kjer gre za to, da tudi na področjih, kjer nismo videli nobenega primera, čim bolj pravilno določimo razred primera, na osnovi obdelanih primerov. Generalizacija je sposobnost posploševanja, zajemanja splošnih bistvenih značilnosti podatkov. Nasprotje od prenasičenja je nenasičenje, kjer je model preveč splošen in posledično ne zazna pomembnih zakonitosti znotraj učne množice. Nenasičeni modeli imajo navadno visoko pristranskost.



Slika 3.8: Varianca in pristranskost [10]

Razvrščevalni algoritmi, ki oblikujejo nelinearne odločitvene krivulje, so še posebej nagnjeni k prenasičenju. Prenasičeni modeli imajo veliko stopnjo variance. To pomeni, da se prekomerno prilagodijo majhnemu naključnemu šumu , ki je vsebovan v podatkih. Slika 3.9 prikazuje potek odločitvene krivulje za prenasičen (zelena barva) in ustrezni model (črna barva).



Slika 3.9: Primerjava med prenasičenim in iskanim modelom [11]

4 PREGLED NEKATERIH RAZVRŠČEVALNIH ALGORITMOV

Preden se lotimo reševanja problema, je treba izbrati primeren algoritem, ki bo uspel ustrezno rešiti naš problem. Omenimo, da ne obstaja en najboljši razvrščevalni algoritem, ki deluje najboljše na vseh vrstah problemov. Na določenih vrstah podatkov se določeni algoritmi odrežejo slabše kot drugi. Zato je pomembno, da izberemo pravi algoritem, ki bo v skladu z našimi potrebami dosegal ustrezne rezultate. V nadaljevanju bodo predstavljeni nekateri razvrščevalni algoritmi, s katerimi se srečujemo najpogosteje.

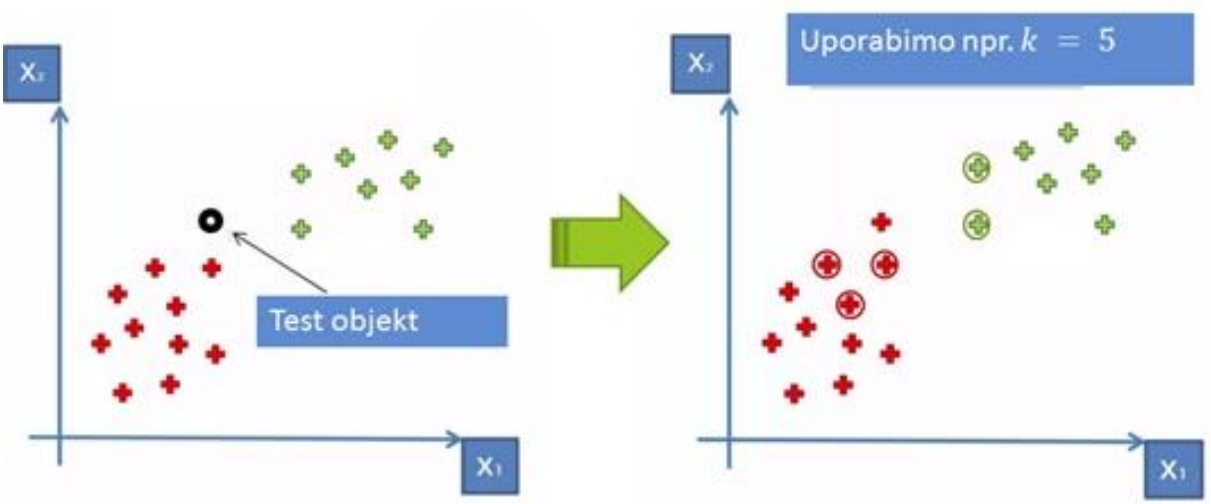
Obstaja veliko dejavnikov, ki vplivajo na primernost uporabljenega razvrščevalnega algoritma. Spodaj je navedenih nekaj izmed teh [13]:

- **Natančnost**: podaja delež pravilnih razvrstitev izbranega algoritma. Treba se je zavedati, da so določene napake potencialno resnejše in nadzirati stopnjo napak za nekatere ključne razrede.
- **Hitrost razvrščanja**: v določenih primerih hitrost razvrščanja (podajanja rezultatov) izbranega algoritma predstavlja pomembno vlogo. Tako je lahko razvrščevalni algoritem z 90 % natančnostjo razvrščanj, v določenih okoljih (npr. sortiranje pošte) primernejši od 100- krat počasnejšega s 95 % natančnostjo.
- **Razumljivost**: v primeru, da človek izvaja postopek razvrščanja, mora imeti majhno število jasnih pravil. V nasprotnem primeru lahko pride do napak. Pomemben dejavnik je tudi človeško zaupanje v razvrščevalni sistem. V zgodovini so se že zgodile nesreče (npr. Chernobyl in Three-Mile Island), ko je avtomatska naprava zaznala potrebo po zaustavitvi sistema, a operater preprosto ni verjel v resnost priporočila.
- **Čas učenja** oz. **izgradnje modela**: je še posebej pomemben v hitro spreminjajočih se okoljih, kjer je potrebna hitra prilagoditev okolju in uspešna razvrstitev. Treba je vedeti, da včasih »hitro« pomeni že majhno število objektov, ki pa lahko zadostujejo za izgradnjo kvalitetnega modela.

4.1 K-najbližjih sosedov

K-najbližjih sosedov (k-NN) je neparametrična metoda, ki se uporablja za razvrščanje vzorcev. Vhodi so sestavljeni iz trening primerov v prostoru značilk. Izhod je v primeru razvrščanja vzorcev napoved pripadnosti določeni skupini oz. neka diskretna vrednost. Pripadnost posameznega vzorca ali objekta (element vzorca) se na podlagi k-NN določi s postopkom večinskega glasovanja (objekt pripada istemu razredu kot večina njegovih sosedov). Lastnosti posameznega objekta si lahko predstavljamo kot vektor v prostoru značilk.

V primeru $k = 1$, pripadnost objekta določa njegov najbližji sosed oz. najbližji vektor (skupaj tako pripadata isti skupini). Spodnja slika prikazuje delovanje k-NN algoritma. Populacija je sestavljena iz rdečih in zelenih križcev v 2D prostoru značilk. Križci predstavljajo trening populacijo, barva pa pripadnost posameznega objekta. Črn krožec predstavlja novo meritev oz. objekt, katere pripadnost želimo napovedati.



Slika 4.1: Delovanje k-NN razvrščevalnega algoritma [14]

Prednosti k-NN so hitro učenje oz. »leno učenje« (algoritem si mora le zapomniti vse trening primere, skorajda ne uporablja matematičnih operacij), enostavna razlaga metode in interpretacija rezultatov. Slabost pa je v tem, da je glavnina procesiranja prisotna pri klasifikaciji novega primera. Posledično je postopek klasifikacije precej dolgotrajnejši,

kot pri drugih metodah. V primeru, da želimo napovedati razred objekta oz. izhode testnih primerov, moramo preračunati vse razdalje v prostoru značilk. V nasprotju z ostalimi razvrščevalnimi algoritmi v smislu uteži, nam metoda *k*-NN, ne nudi »pravega modela«. Učinkovitejša je v nizko-dimenzionalnem prostoru značilk.

Že sama beseda »najbližjih« nakazuje, da je za zagon testiranja k-NN algoritma potreben izračun razdalje. Razdaljo od testnega objekta $\mathbf{x}_j$ do trening objekta $\mathbf{x}_q$ običajno merimo z Minkowski-jevo razdaljo L^p, ki je podana kot [15]:

$$L^p(\mathbf{x}_j, \mathbf{x}_q) = \left(\sum_i |x_{j,i} - x_{q,i}|^p \right)^{1/p} \tag{4.1}$$

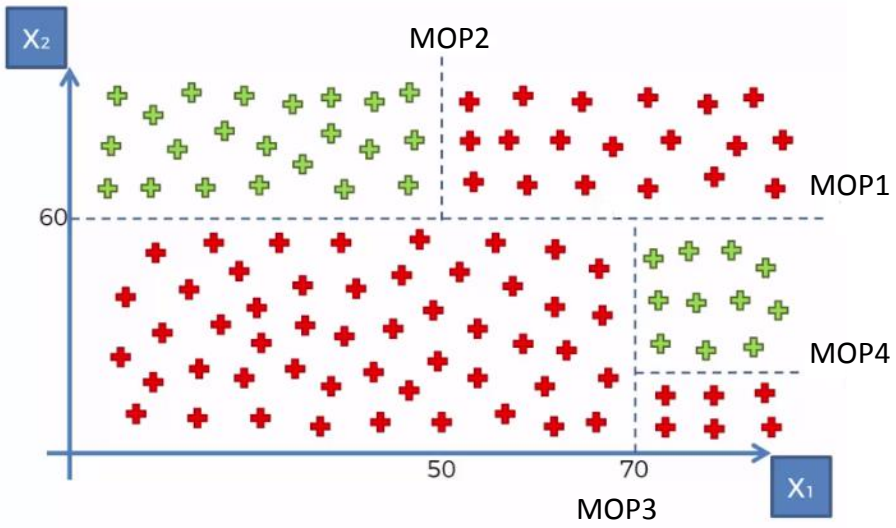
V primeru $p = 2$ imamo opravka z Evklidsko razdaljo, ki se po navadi uporablja kadar imamo opravka s prostorom značilk, kjer značilke metrično popisujejo podobne lastnosti. Primer Evklidske razdalje so višina, širina in dolžina tekočega traku. V primeru, da je $p = 1$ pa predstavljajo podobne lastnosti. Manhattanova razdalja se uporablja v primeru, da imamo opravka z značilkami, ki metrično popisujejo nepodobne lastnosti kot so starost, višina in spol pacienta. Na razdaljo pomembno vpliva red velikosti značilk, zato je potrebna normalizacija.

4.2 Odločevalna drevesa

Odločevalna drevesa (OD) so ena izmed najbolj preprostih, v veliko primerih pa tudi ena izmed najbolj učinkovitih, različic razvrščevalnih algoritmov. OD predstavlja funkcijo, ki na podlagi vektorja vhodnih vrednosti poda izhod. Izhod je v primeru razvrščanja vzorcev enak pripadnosti določeni skupini. OD se pogosto uporabljajo kot strategija za prepoznavo odločitev, s katerimi bomo najlažje dosegli ciljno stanje/dejanje (npr. v ekonomiji, pri zasnovi računalniških igric ipd.). OD so pred dvema desetletjema veljala za eno izmed najbolj popularnih metod strojnega učenja. Vendar se je njihova uporaba, zaradi slabe generalizacije in pojava naprednejših izpeljank algoritma (npr. odločevalni gozdovi), močno zmanjšala. [17].

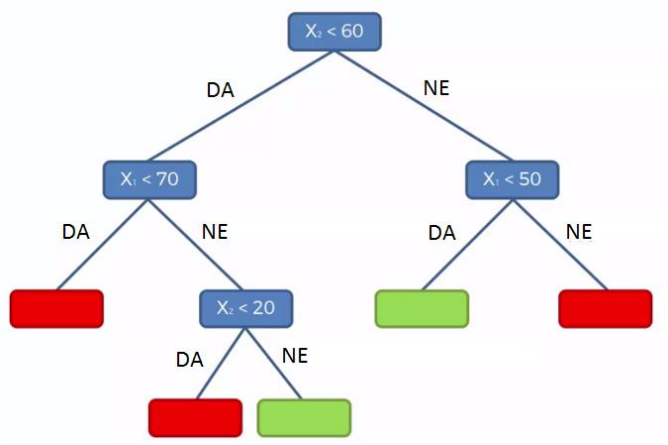
Na spodnji sliki je prikazana populacija, ki je sestavljena iz rdečih in zelenih križcev. Algoritem OD jo loči, in sicer s pomočjo mejnih odločitvenih premic (MOP) v 2D oz. s pomočjo mejnih odločitvenih ravnin v večdimenzionalnem prostoru zančilk. Algoritem med procesom učenja izvaja delitve populacije s pomočjo MOP v zaporedju MOP1, MOP2, MOP3, … , MOP5. Zaporedje in lokacija premic je določena na podlagi informacijske entropije (enačba 4.2). Entropija je merilo negotovosti naključne spremenljivke. Določa količino informacije, ki jo dobimo, če izvedemo poskus. V enačbi predstavlja S trenutno izbiro podatkov, X nabor razredov, x razred in $p(x)$ delež elementov v podatkovni zbirki X, ki spadajo v razred x [16]. Želimo izbrati tako delitev, da je entropija čim nižja. V primeru, da je entropija $H(S) = 0$ velja, da se vsi vzorci uvrščajo v isti razred.

$$H(S) = \sum_{x \in X} p(x) log_2((p(x)) \tag{4.2}$$



Slika 4.2: Prikaz delovanja OD algoritma [14]

Slika 3.2 prikazuje shemo odločilnega drevesa, kjer končne vrednosti (v našem primeru zelene in rdeče križce) imenujemo terminalni listi, modre križce pa odločevalna vozlišča. V primeru, da želimo določiti pripadnost nekega objekta, se le sprehodimo po odločevalnem drevesu od zgoraj navzdol (sledeč ustreznim odločitvam, ki jih sprejmemo na podlagi lastnosti objekta), dokler ne dosežemo terminalnega lista, ki nam poda pripadnost testiranega objekta.



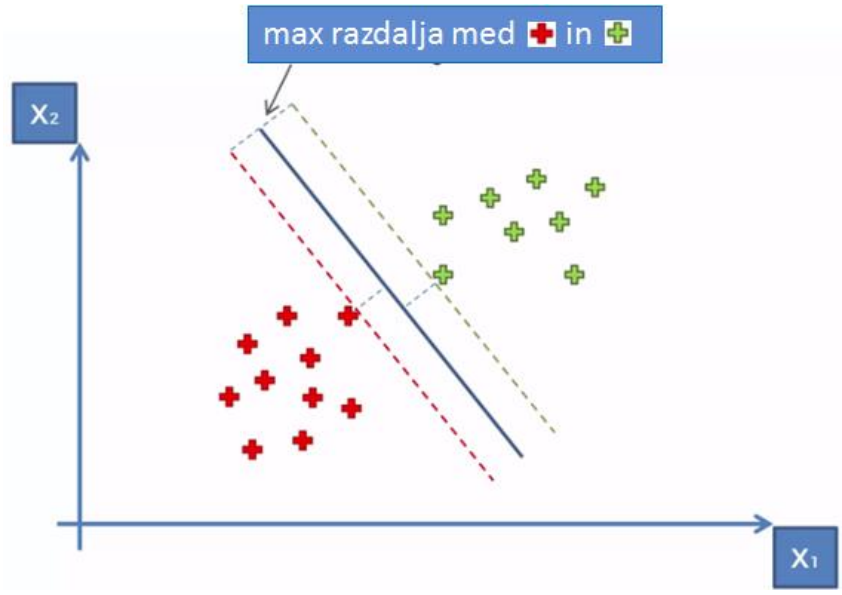
Slika 4.3: Shematski prikaz OD [14]

Odločevalna drevesa so primerna le v primeru, ko je število elementov v populaciji veliko večje, kot je število značilk, s katerimi je populacija popisana. Priporočeno je vsaj 2^n elementov oz. trening primerov, kjer je n enak številu značilk oz. številu dimenzij v prostoru značilk. Odločevalna drevesa načeloma odlično funkcionirajo z največ desetimi dimenzijami in tisočerimi trening primeri [15].

4.3 Metoda podpornih vektorjev

Čeprav je bila metoda razvita že v šestdesetih, je na svoji prepoznavnosti pridobila šele v devetdesetih letih prejšnjega stoletja. Primerna je za vsakogar, saj ne zahteva predhodnega znanja o strojnem učenju. V najpreprostejši obliki nam pove (brez uporabe kernel trika), ali so naši podatki linearno ločljivi ali ne. Metoda podpornih vektorjev (SVM) ima sposobnost prikazovanja kompleksnih funkcij. A je kljub temu odporna na prekomerno prilagajanje modela (»over-fitting«) ter posledico tega prilagajanja, tj. slabo generalizacijo. Namen je optimalna ločitev prostora na dva dela. Gre za deterministično metodo, kar pomeni, da je rešitev, ki predstavlja optimalno ločitev prostora le ena. Velja za neparametrično metodo, ker obdrži trening primere in potencialno shrani vse podatke, na podlagi katerih je oblikovan model. Po drugi strani pa v praksi SVM algoritem dejansko uporabi le majhen odstotek trening primerov oz. le tiste na podlagi katerih zasnuje podporne vektorje.

Na spodnji sliki je prikazana populacija sestavljena iz rdečih in zelenih križcev. Prostor SVM algoritem loči s pomočjo odločitvene premice v 2D, odločitvene ravnine v 3D oz. s pomočjo odločitvenih hiper-ravnin v večdimenzionalnem prostoru zančilk. Iz slike je razvidno, da premica, ki ločuje populacijo rdečih in zelenih križcev, stoji na podpornih vektorjih najbližjih objektov nasprotne pripadnosti v prostoru značilk.



Slika 4.4: Delovanje SVM algoritma [14]

Hiper-ravnino v splošnem opišemo z enačbo 4.2. Kjer je $\mathbf{w}^T$ vektor uteži, ki podaja smer hiper ravnine v določeni koordinatni smeri, $\mathbf{x}_j$ je vektor objekta v prostoru značilk in b je zamik.

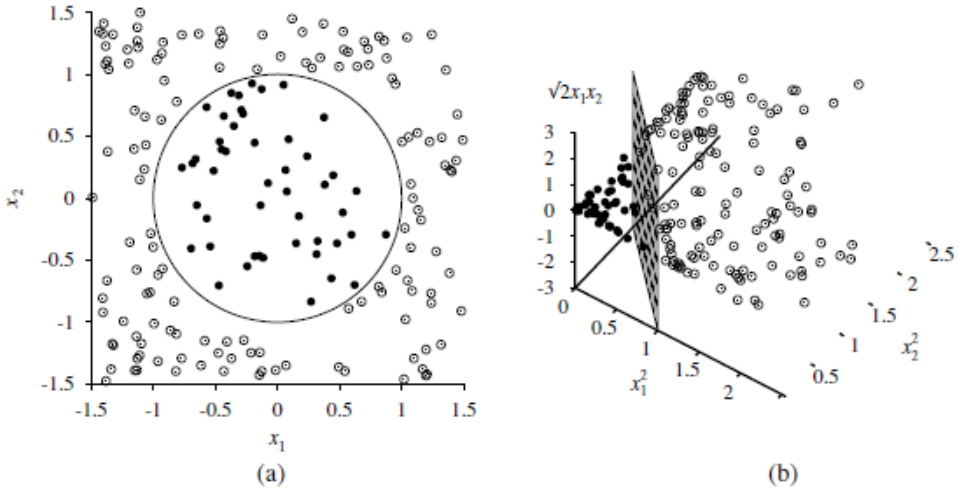
Pripadnost objekta določenemu razredu je definirana z enačbama 4.3 in 4.4. V našem primeru (slika 4.4) velja enačba 4.3, če je testni objekt postavljen nad premico. V primeru, ko je testni objekt postavljen pod premico pa velja enačba 4.4. Pripadnost objekta je določena v obliki rdečega križca. Uteži in zamik se iterativno prilagajajo tekom treniranja SVM modela.

$$\mathbf{w}^T * \mathbf{x}_j + b = 0 \tag{4.2}$$

$$\boldsymbol{w}^T * \boldsymbol{x}_j + b > 0 \quad => \text{objekt} = \text{zeleni} + \tag{4.3}$$

$$\boldsymbol{w}^T * \boldsymbol{x}_j + b < 0 \quad => \text{objekt} = \text{rdeči} + \tag{4.4}$$

SVM-ji ustvarijo mejno ločitveno hiper-ravnino. Vseeno pa imajo sposobnost, da lahko z uporabo kernel trika, ločijo podatke v višjih dimenzijah. Velikokrat podatki v originalnem prostoru niso ločljivi, so pa ločljivi v višje dimenzionalnem prostoru značilk (slika 4.5). Na sliki vidimo, da krožna (nelinearna) odločitvena krivulja (a) postane linearna v tri-dimenzionalnem prostoru. Visoko dimenzionalni linearni separator je v bistvu nelinearen v prvotnem prostoru značilk. To pomeni, da lahko uporabimo veliko več bolj ali manj zapletenih kernelovih funkcij za razvrščanje vzorcev[15]. Nekatere pogosteje uporabljene kernelove funkcije so gaussova, linearna, polinomska, eksponenčna in sigmoidna [18].



Slika 4.5: (a) dvo-dimenzionalna trening populacija z pozitivnimi primeri prikazanimi kot črni krožci in negativnimi kot beli krožci. Resnična odločitvena krivulja je enaka $x_1^2 + x_2^2 \leq 1$. (b) enaka populacija je preslikana v tri-dimenzionalen prostor značilk $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$[15]

4.4 Naivni Bayes

Naivni Bayesov klasifikator predstavlja pogojno neodvisnost vrednosti različnih značilk pri danem razredu. Naivni Bayesov KA lahko uporabljamo tako za zvezne, kot tudi za diskretne značilke. Zvezne značilke je potrebno najprej diskretizirati. Za diskretizacijo zveznih značilk se najbolje odnese mehka diskretizacija, kjer en primer ne pripada samo enemu intervalu (mehke meje). Kadar obstajajo močne odvisnosti med značilkami, Naivni Bayesov KA odpove. Naloga slednjega KA je, da s pomočjo učne množice podatkov aproksimira apriorne verjetnosti razredov $P(r_k), k = 1, \dots, n_0$ in pogojne verjetnosti razredov $r_k, k = 1, \dots, n_0$ pri dani vrednosti v_i značilke $A_i,\ i = 1, \dots, a$: $P(r_k|v_i)$. Za ocenjevanje apriornih verjetnosti se uporablja Laplaceov zakon zaporednosti:

$$P(r_k) = \frac{N_k + 1}{N + n_0} \tag{4.5}$$

kjer je N_k število trening primerov iz razreda r_k in N število vseh trening primerov. Za ocenjevanje pogojnih verjetnosti se uporablja m-ocena:

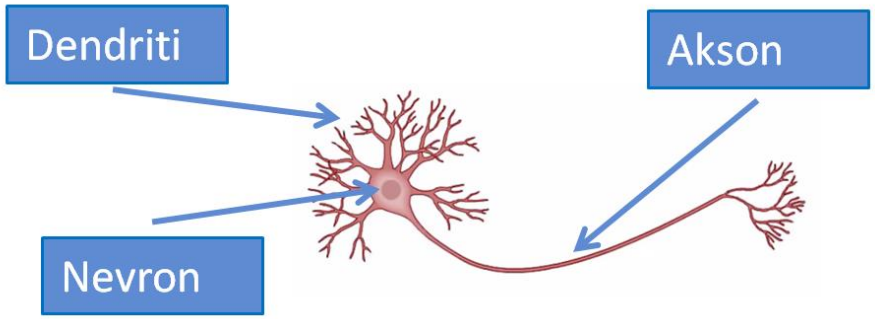
$$P(r_k|v_i) = \frac{N_{k,i} + mP(r_k)}{N_i + m} \tag{4.6}$$

kjer je $N_{k,i}$ število trening primerov iz razreda r_k in z vrednostjo i-te značilke v_i ter N_i število vseh trening primerov z vrednostjo i-te značilke v_i. Osrednja formula Naivnega Bayesovega KA je podana kot [1]:

$$P(r_k|V) = P(r_k) \prod_{i=1}^{a} \frac{P(r_k|v_i)}{P(r_k)} \tag{4.5}$$

4.5 Nevronske mreže

Nevroni so v biološkem svetu gradniki živčevja. Njihova osrednja funkcija je prevajanje impulzov. Številni nevroni se po izgledu močno razlikujejo, kar je posledica visoke specializiranosti nevronov. Nevron sestavljajo jedro, dendriti in nevrit. Dendritov je ponavadi več, so bolj razvejani in preko njih nevron prejme signal. Signal se nato obdela v jedru nevrona in se pošlje naprej po nevritu do ciljnih nevronov ali efektornih celic. Z učenjem oz. interakcijo z okoljem se določene povezave med nevroni oz. skupki nevronov krepijo. Moč obdelave velike količine kompleksnih podatkov bioloških nevronskih mrež izhaja iz velike količine medsebojno povezanih nevronov ter visoko stopnjo vzporednega delovanja. Tega pa niti z najnovejšimi super računalniki nismo zmožni poustvariti.



Slika 4.6: Prikaz biološkega nevrona

Umetne nevronske mreže posnemajo delovanje bioloških nevronskih mrež. Razvite so bile v petdesetih letih prejšnjega stoletja, vendar se do dobe računalnikov niso prav veliko uporabljale. Nevrone povezujejo sinaptične povezave oziroma sinapse, vsaka povezava ima svojo utež. Uteži s pozitivnimi vrednostmi so vzbujajoče, z negativnimi pa zavirajoče. Osnovni element za gradnjo umetne nevronske mreže je umetni nevron. Umetni nevron (slika 4.7) je kot informacijsko procesna enota, sestavljen iz množice sinaps s sinaptičnimi utežmi, sumacije oz. seštevalnika in aktivacijske funkcije. Izhod seštevalnika, ki sešteje produkte vhodnih uteži z vhodnimi signali, imenujemo aktivacija. Ta je vhod v aktivacijsko funkcijo, ki služi omejevanju amplitude izhoda nevrona. Iz matematičnega vidika je nevron element, ki zadosti funkciji:

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + w_0\right) \tag{4.7}$$

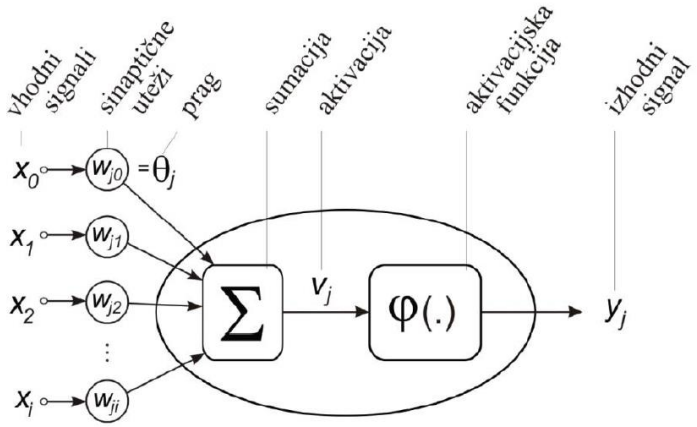
kjer y predstavlja izhod iz nevrona, $\varphi()$ aktivacijsko funkcijo, w_i uteži, x_i vhodne vrednosti nevrona ter w_0 zamik nevrona. Nevron nato sešteje vrednosti zmnožka posamezne vhodne

vrednosti in pripadajoče uteži. Rezultat se uporabi kot argument aktivacijske funkcije. Tako se posledično ustvarijo izhodne vrednosti nevrona. Zamik nevrona je člen, ki nam omogoča učinkovitejši popis realnih problemov, ki so po navadi nelinearne narave. Grafično nam zamik omogoča premik aktivacijske funkcije nevrona po x-osi.

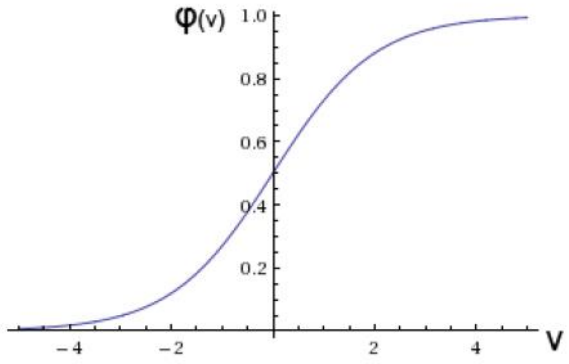
Najpogosteje uporabljena aktivacijska funkcija je sigmoidna in je predstavljena v naslednji enačbi:

$$\varphi(\nu) = \frac{1}{1 + e^{-a\nu}} \tag{4.8}$$

kjer je ν aktivacija in a parameter naklona. Graf funkcije je prikazan na sliki 4.8.



Slika 4.7: Sestavni deli umetnega nevrona



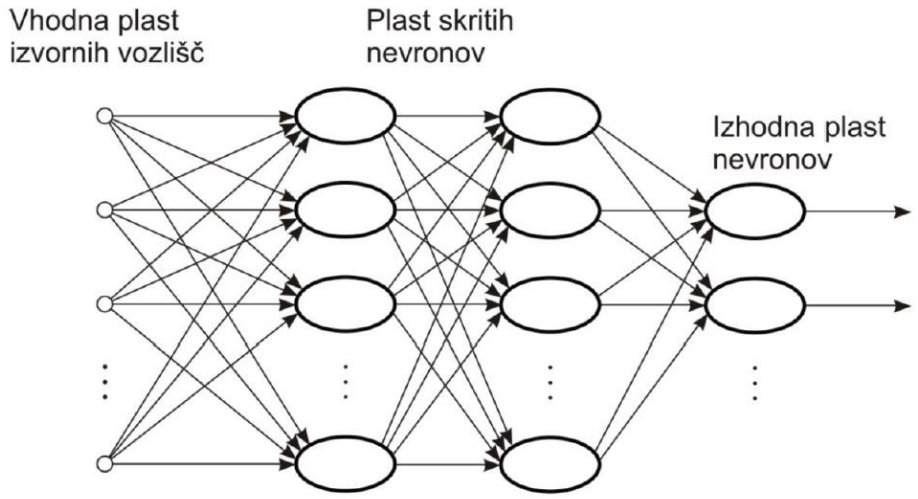
Slika 4.8: Prikaz sigmoidne funkcije

4.5.1 Topologija nevronske mreže

Topologija nevronske mreže nam pove, koliko nevronov bo v mreži in kako bodo med seboj povezani. Poznane so številne topologije, najbolj pogosto pa se v aplikacijah uporablja usmerjena nevronska mreža (»feed-forward neural network«). Uporabili smo jo tudi v našem primeru, za napovedovanje Parkinsonove bolezni, Usmerjene nevronske mreže imajo naslednje omejitve:

- Povezave med nevroni v istem nivoju niso dovoljene,
- Niso dovoljene povezave nazaj na prejšnje nivoje in
- Ni dovoljeno preskakovanje enega ali več nivojev pri povezavi naprej.

Poznamo tri osnovne plasti nevronskih mrež, in sicer vhodno plast, plast skritih nevronov in izhodno plast nevronov (slika 2.3). Skritih plasti nevronov je lahko poljubno število. Od njihove izbire je v veliki meri odvisna učinkovitost nevronske mreže. Pri tem se moramo zavedati, da večje število skritih plasti nevronov ne pomeni tudi nujno boljše rešitve [19]. Spodnja slika prikazuje usmerjeno nevronsko mrežo globile $L = 3$ z dvema skritima plastema.



Slika 4.9: Usmerjena nevronska mreža

4.5.2 Algoritem vzvratnega širjenja napake

Večplastne usmerjene nevronske mreže učimo s popularnim algoritmom vzvratnega razširjanja napake (»error back-propagation algorithm«). V literaturi se izraz pogosto pojavlja v svoji skrajšani obliki, tj. vzvratno razširjanje (»back-prop«). V osnovi sestavljata proces vzvratnega razširjanja dva prehoda skozi različne plasti, in sicer prehod naprej in prehod nazaj. Delovanje vzvratnega razširjanja napake lahko ponazorimo z pseudo-algoritmom, ki je prikazan na sliki 4.11. V 2. vrstici utežem in pragom mreže predpišemo dovolj majhni, ponavadi enakomerno porazdeljeni naključni števili. Mreži v vsakem ciklu for (4. vrstica) predstavimo en učni objekt [$\boldsymbol{x}^{(p)}$, $\boldsymbol{d}^{(p)}$], kar je v skladu z vzorčnim načinom učenja (nasprotje je paketni način učenja).

Alogritem Učni algoritem usmerjene večplastne nevronske mreže

Vhodi: [$\boldsymbol{x}^{(p)}$, $\boldsymbol{d}^{(p)}$], $p = 1, 2, \dots, N$ – učni primeri

Izhod: **w** – utežni vektor

```
1:  function: VZVRATNO-RAZŠIRJANJE (x^(p), d^(p), w)
2:    Postavi uteži na začetne vrednosti
3:    while napaka ni dovolj majhna do
4:      foreach učni vzorec [x^(p), d^(p)], p = 1, 2, ..., N
5:          Treniraj mrežo na tekočem vzorcu prehod naprej, prehod nazaj
6:      end foreach
7:    end while
8:  end function
```

Slika 4.10: Pseudo-algoritem delovanja vzvratnega razširjanja[2]

V 5. vrstici najprej opravimo **prehod naprej**, kjer izračunamo aktivacijske in funkcijske signale. V tem prehodu v mrežo vstopi vstopni vektor, ki se širi plast za plastjo. Tekom tega postopka se uteži ne prilagajajo. Vstopni vektor potuje od začetne plasti, preko skritih, do izhodne plasti. Na koncu mreža proizvede množico izhodov, ki predstavljajo odgovor mreže. Mrežni aktivacijski nivo za nevron j v plasti l je podan z enačbo:

$$\mathrm{v}_j^{(l)}(n) = \sum_{i=0}^{s_j^{(l)}} \mathrm{w}_j^{(l)}(n)\mathrm{y}_i^{(l-1)}(n) \tag{4.9}$$

kjer je $s_j^{(l)}$ število vhodov v nevron j v plasti l, $\mathrm{y}_i^{(l-1)}(n)$ je funkcijski signal navrona i v prejšni plasti $l-1$ v n-ti iteraciji in $\mathrm{w}_j^{(l)}(n)$ je utež nevrona j v plasti l, ki je povezan z nevronom i v plasti $l-1$. V primeru, da nevroni uporabljajo sigmoidalno aktivacijsko funkcijo enačbo (4.9) vstavimo spodnjo enačbo:

$$\mathrm{y}_j^{(l)}(n) = \frac{1}{1 + e^{-\mathrm{v}_j^{(l)}(n)}} \tag{4.10}$$

V primeru, da je $l = 0$, ko imamo opravka z vhodno plastjo, kjer ni nevronov in velja $\mathrm{y}_j^{(0)}(n)$ je enak ustreznemu vhodu $x_j^{(p)}$. Na koncu izračunamo signal napake pri j-tem izhodu:

$$e_j = \mathrm{d}_j^{(p)} - o_j(n) \tag{4.11}$$

Kjer je $\mathrm{d}_j^{(p)}$ j-ti element željenega odzivnega vektorja $\boldsymbol{d}^{(p)}$, z $o_j(n) = \mathrm{y}_j^{(L)}(n)$ označimo izhodni signal nevrona v plasti L [2].

Nato izvedemo **prehod nazaj** in vzvratna izračunavanja. V tem prehodu se uteži spremenijo glede na učenje s popravljanjem napake. Dejanski odziv mreže odštejemo do željenega, da proizvedemo signal napake, ki se širi nazaj skozi mrežo. Tako se s prilagoditvijo uteži, odziv mreže približa želenemu. Izračunamo lokalne gradiente ϭ v smeri nazaj in plast za plastjo. Gradient za nevron j v izhodni plasti L je:

$$\mathrm{ϭ}_j^{(L)}(n) = o_j(n)\left[1 - o_j(n)\right]\left[\mathrm{d}_j^{(p)} - o_j(n)\right] \tag{4.12}$$

Gradient za nevron j v skriti plasti l je:

$$l\mathrm{ϭ}_j^{(l)}(n) = y_j^{(l)}(n)\left[1 - y_j^{(l)}(n)\right]\sum_k \mathrm{ϭ}_k^{(l+1)}(n)\,\mathrm{w}_{kj}^{(l+1)}(n) \tag{4.13}$$

Kjer vsota teče po vseh nevronih v $(l+1)$-ti plasti, ki so povezani z j-tim nevronom v plasti l. Prilagoditev uteži poteka po pravilu delta:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta\mathrm{ϭ}_j^{(l)}(n)\mathrm{y}_i^{(l-1)}(n) \tag{4.14}$$

kjer je η hitrost učenja. Hitrost učenja vpliva na stabilnost učenja. V primeru da je premajhna, so računski časi razvrščanja visoki. Če pa je vrednost nastavljena previsoko, se lahko zgodi, da algoritem ne bo konvergiral k stabilni rešitvi. Pogosto se v enačbi (4.14) pojavlja tudi člen z pozabljanjem α, ki pomaga preprečiti, da bi učni algoritem obstal v lokalnem minimumu [2].

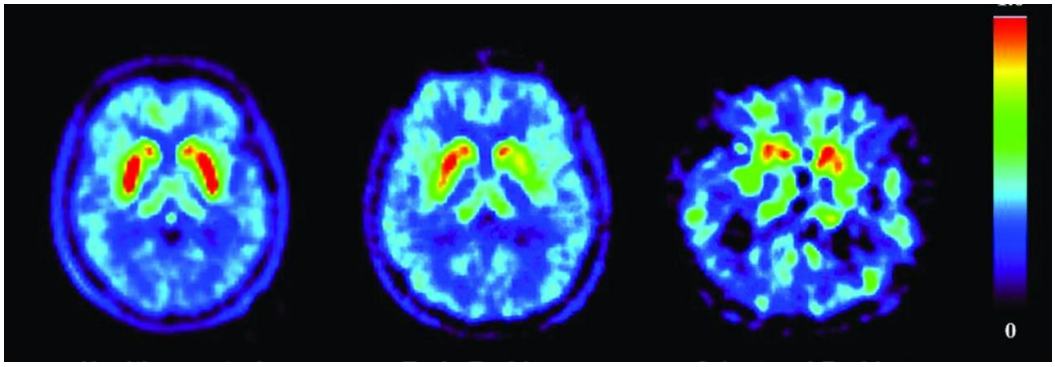
5 PRIMER UPORABE UMETNE INTELIGENCE ZA RAZVRŠČANJE VZORCEV

5.1 Parkinsonova bolezen in vhodni podatki

Parkinsonova bolezen (PB) je dobila ime po Jamesu Parkinsonu, ki je leta 1817, v delu »*An Essay on the Shaking palsy*«, prvi opisal tresenje dlani v mirovanju ter upočasnjeno gibanje. Oboje je zaznal pri svojih pacientih [14]. Bolezen se običajno pojavi pri starejših ljudeh in povzroči motnje v govoru in motoričnih sposobnostih (pisanje, ravnotežje itd.). Gre za 2. najpogostejšo bolezen tretjega življenjskega obdobja, takoj za Alzheimerjevo boleznijo. PB prizadene na milijone ljudi, po nekaterih ocenah je na svetu več kot 10 milijonov primerov. V prihodnosti pa lahko pričakujemo, da bo s staranjem prebivalstva v večini razvitejših držav število obolelih za PB vse večje. PB prizadene predvsem telesno gibanje bolnika in ga sčasoma onesposobi za samostojno življenje. Ob zgodnji diagnozi in implementaciji dopaminske terapije je napredovanje bolezni mogoče upočasniti. Vendar dopaminska terapija sčasoma postane neučinkovita. Vzroka in zdravila za PB do danes še ne poznamo v celoti.

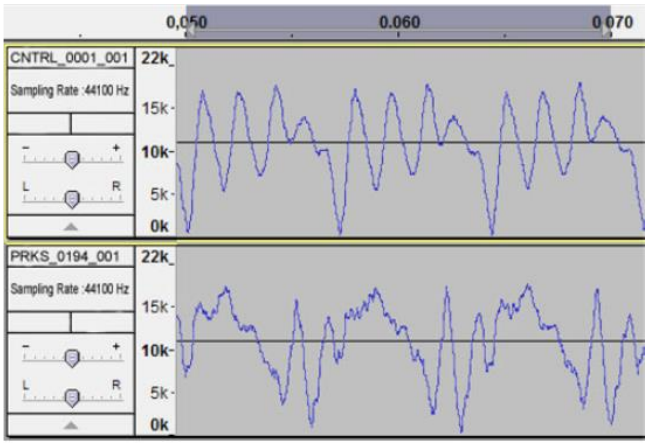
PB je kronična, napredujoča in nevrodegenerativna bolezen. Povzroči postopni razpad centralnega živčnega sistema, kar ima neposredne posledice na motorične sposobnosti bolnika. Večina nevronov, ki proizvajajo dopamin, tvori v možganih črno substanco imenovano »*substantia nigra*«. Slednja začne bledeti, kar povzroči otežen prehod živčnega signala po hrbtenjači [20]. Posledice se kažejo kot zmanjšano in upočasnjeno gibanje pacienta ter nezmožnost naključnega odziva oz. refleksnega giba. Prepoznavni znaki bolezni so zmanjšana izraznost obraza, motnje v REM fazi spanja, povečana rigidnost mišic in tresoče dlani v fazi gibanja bolnika. Ljudje s PB lahko tudi izgubijo čut za vonj [21, 22]. Umske sposobnosti bolnikom (vsaj v zgodnji fazi) ne pešajo. Čeprav so videti otopeli, delno zaradi borne mimike obraza, so v resnici mentalno bistri.

Metode diagnosticiranja PB potekajo z invazivnimi metodami, kar zakomplicira življenje pacientov. Slika 5.1 prikazuje scintigrafijsko sliko možganskega delovanja treh pacientov. Pacient na desni ima PB. Pri njemu vidimo, da je razpršenost delovanja centrov v možganih, drugačna kot pri zdravem pacientu.



Slika 5.1: Prikaz scintigrafijske slike zdravega pacienta (na levi), pacienta z PB v zgodnji fazi (na sredini) in pacienta z PB v pozni fazi [23]

Ljudje, ki obolevajo za PB, trpijo za različnimi motnjami govora kot so: disfonija (okvarjena uporaba glasu oz. hripavost), hipofonija (zmanjšan obseg glasu), monotonija (zmanjšan intonančni obseg) in dizartrija (težave artikulacije z zvoki ali zlogi). Da bi se izognili uporabi invazivnih metod, smo na podlagi raziskave [24] opravili meritve govora/glasilk. Spodnja slika prikazuje graf amplitude glasu zdrave in obolele osebe.



Slika 5.2: Prikaz govora PB bolnika in zdravega posameznika [24]

5.2 Vhodni podatki

Kot že rečeno, so bile na podlagi raziskave [24] narejene meritve govora/glasilk. Podatki opravljenih meritev so dostopni na [25] University of Irvine (UIC) machine learning. V sklopu raziskave je bilo testiranih 20 zdravih posameznikov (10 moških, 10 žensk) in 20 pacientov z PB (14 moških, 6 žensk). Starost zdravih posameznikov se je gibala med 43 in 77 leti (povprečje 64.86, standardna deviacija 8.97), starost pacientov z PB pa med 45 in 83 (povprečje 62.55, standardna deviacija 10.79). Meritve so opravljene z mikrofonom tipa Trust MC-1500 z frekvenčnim razponom od 50 Hz do 13 kHz. Mikrofon je bil nastavljen na 96 kHz, 30 dB in pozicioniran v razdalji 10 cm od ust posameznika. V poteku raziskave [24] so udeleženci najprej opravili zdravniški pregled. Zdravniki so jih prosili naj preberejo vnaprej pripravljeno besedilo, ki je vključevalo glasovne vzorce. V tem kontekstu je udeleženec prebral ali izgovoril 26 vzorcev glasu, ki so vsebovali številke od 1 do 10, 4 ritmične stavke, 9 besed in samoglasnike »a«, »o« in »u«. Na podlagi 26-ih vzorcev zvoka 40-ih udeležencev, so s pomočjo programa Praat [26], podane časovno-frekvenčne značilke. Te so prikazane v spodnji preglednici.

Preglednica 5.1: Prikaz časovno-frekvenčnih značilk pridobljenih na osnovi testiranj 40-ih posameznikov [24]

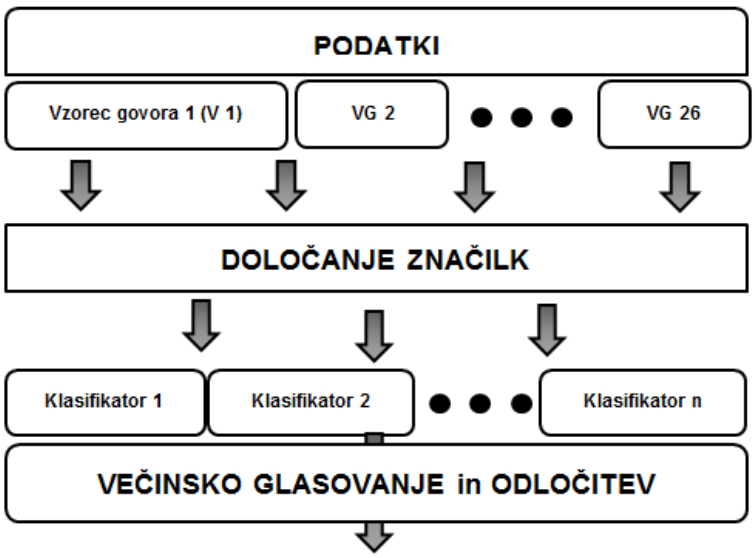
Zap. št. zničilke	Značilka	Povprečje	Stand. dev.	Skupina
1	**Tresenje (lokalno)**	2,67952	1,76505	Frekvenčni parametri
2	**Tresenje (lokalno, absolutno)**	0,00017	0,00011	
3	**Tresenje (rap)**	1,24705	0,97946	
4	**Tresenje (ppq5)**	1,34832	1,13874	
5	**Tresenje (ddp)**	3,74116	2,93844	
6	**Število impulzov**	12,91839	5,45220	Impulzni parametri
7	**Število period**	1,19489	0,42007	
8	**Povprečna perioda**	5,69960	3,01518	
9	**Standardna deviacija period**	7,98355	4,84089	
10	**Šimer (lokalno)**	12,21535	6,01626	Amplitudni parametri
11	**Šimer (lokalno, dB)**	17,09844	9,04554	
12	**Šimer (apq3)**	0,84601	0,08571	
13	**Šimer (apq5)**	0,23138	0,15128	
14	**Šimer (apq11)**	9,99954	4,29130	
15	**Šimer (dda)**	163,3683	56,02168	
16	**Delež lokalno neizvajanih okvirov**	168,7276	55,96991	Voicing parametri
17	**Število glasovnih prekinitev**	27,54763	36,67262	
18	**Stopnja glasovnih prekinitev**	134,5381	47,05806	
19	**Mediana intonance**	234,8760	121,5412	Intonančni parametri
20	**Povprečna intonanca**	109,7442	150,0277	
21	**Standardna deviacija**	105,9692	149,4171	
22	**Najmanjša intonanca**	0,00655	0,00188	
23	**Maksimalna intonanca**	0,00084	0,00072	
24	**Avtokorelacija**	27,68286	20,97529	Harmoničnostni parametri
25	**Hrup-proti-harmoničnosti**	1,13462	1,16148	
26	**Harmoničnost-proti-hrupu**	12,37001	15,16192	

5.3 Predstavitev algoritma

Spodnja slika prikazuje shematski prikaz delovanja algoritma, ki na podlagi časovno-frekvenčnih značilk, pridobljenih na osnovi 26-ih zvočnih vzorcev 40-ih posameznikov, prepozna ali objekt ima parkinsonovo bolezen ali ne. Gre seveda za problem razvrščanja, v želji po zaznavi vzorcev zvočnih indikatorjev bolezni. Podatkovna množica je ločena na trening in testno množico. Ta razločitev je v skladu z izpusti-enega LOSO navzkrižno validacijo, kjer je k enak številu posameznikov.

V najpreprostejši obliki, kar pomeni brez redukcije značilk, je vsak vzorec glasu, obdelan z razvrščevalnim algoritmom (klasifikator). Ta na podlagi vzorca glasu določi ali posameznik ima Parkinsonovo bolezen ali ne. Končna odločitev o prisotnosti PB je podana s pomočjo večinskega glasovanja, kjer se upoštevajo odločitve razvrščevalnih algoritmov, ki so obdelali posamezne vzorce glasu. V primeru izenačenja je (npr. brez redukcije značilk imamo 26 klasifikatorjev in lahko pride do izenačenja) večinsko glasovanje nagnjeno proti ocenitvi posameznika kot bolnika z PB. Namreč, bolje je dodatno preiskati osebo, ki nima PB, kot pa ne ukrepati v primeru, da bolnik ima PB.

Na sliki 5.3 je prikazan način delovanja algoritma za zaznavo PB. Uporabljena procedura določanja oz. redukcije značilk lahko pripelje do tega, da posamezen vzorec glasu ostane značilk in je tako izločen iz algoritma (n je lahko ≤ 26).



Slika 5.3: Shematski prikaz delovanja algoritma za zaznavo PB

5.4 Določitev značilk

Določitev značilk ter redukcija podatkov sta izpeljani na podlagi Samo-organizacijskih gruč, Pearsonovega in Khendallovega korelacijskega koeficienta. Želja je prepoznati pomembne značilke (v primeru korelacij) oz. videti, če transformacija v nižje dimenzijski prostor značilk (v primeru samoorganizacijskih gruč) izboljša uspešnost razvrščanja vzorcev. V primeru Prearsonovega korelacijskega koeficienta, smo obdržali vse značilke, za katere velja $r > |0|$, $r > |0{,}25|$, $r > |0{,}30|$, $r > |0{,}35|$ in $r > |0{,}40|$. Vse značilke, ki ne zadoščajo temu pogoju so izločene. Iz preglednice 5.2 je razvidno, da vzorec zvoka »kratki stavek 1«, za r vse razen $r > |0|$, ostane brez značilk in je tako izločen iz algoritma za zaznavo PB.

Preglednica 5.2: Izbrane časovno-frekvenčne značilke z uporabo različnih Pearsonovih korelacijskih koeficientov

Vzorec zvoka	**Uporabljene značilke (r>|0|)**	**Uporabljene značilke (r>|0.25|)**	**Uporabljene značilke (r>|0.30|)**	**Uporabljene značilke (r>|0.35|)**	**Uporabljene značilke (r>40)**
"a"	Vsi	24	Nobena	Nobena	Nobena
"o"	Vsi	19,24	24,19	Nobena	Nobena
"u"	Vsi	13,21	Nobena	Nobena	Nobena
Številka 1	Vsi	1,2,3,4,5,24	1,2,3,4,5,24	1,2,4	1,4
Številka 2	Vsi	1,2,8,9,10,11	2,8,9,10,11	10	Nobena
Številka 3	Vsi	12,13,14,17,19,23,25,26	17,19,23,25,26	17,19,23,25,26	17,25
Številka 4	Vsi	1,2,3,4,5,10,20,21	1,2,3,4,5,10	1,2,3,4,5	1,2,3,4,5
Številka 5	Vsi	24	24	24	Nobena
Številka 6	Vsi	10,23,26	Nobena	Nobena	Nobena
Številka 7	Vsi	17,19,24,26	Nobena	Nobena	Nobena
Številka 8	Vsi	9,10	9	Nobena	Nobena
Številka 9	Vsi	26	26	Nobena	Nobena
Številka 10	Vsi	1,2,3,5,8,9,11,23	Nobena	Nobena	Nobena
Kratki stavek 1	Vsi	Nobena	Nobena	Nobena	Nobena
Kratki stavek 2	Vsi	3,4,5,24,25,26	25,26	25	25
Kratki stavek 3	Vsi	3,4,5,10,25,26	4,10,25,26	10,26	26
Kratki stavek 4	Vsi	1,2,3,4,5,10,24,25,26	1,2,3,4,5,10,26	1,2,3,4,5,10,26	3,4,5,10
Beseda 1	Vsi	1,2,4,7	1,2	Nobena	Nobena
Beseda 2	Vsi	10	Nobena	Nobena	Nobena
Beseda 3	Vsi	17,19,23,25	17,19,23,25	17,19	17,19
Beseda 4	Vsi	3,5	Nobena	Nobena	Nobena
Beseda 5	Vsi	26	26	Nobena	Nobena
Beseda 6	Vsi	2,10	Nobena	Nobena	Nobena
Beseda 7	Vsi	17	Nobena	Nobena	Nobena
Beseda 8	Vsi	1,2,3,4,5,10,17,19,23,24,25	1,2,3,5,17,19,23,25	4,17,19	17,19
Beseda 9	Vsi	2,24	24	Nobena	Nobena
Število klasifikatorjev	26	25	16	10	8

Analogno zgornjemu primeru je določitev značilk narejena s pomočjo Khendallovega korelacijskega koeficienta τ_b, kjer smo obdržali vse značilke posameznega zvočnega vzorca, ki zadoščajo pogoju $\tau_b > |0|$, $\tau_b > |0{,}20|$, $\tau_b > |0{,}25|$, $\tau_b > |0{,}30|$ in $\tau_b > |0{,}35|$.

Preglednica 5.3: Izbrane časovno-frekvenčne značilke z uporabo različnih Khendallovih korelacijskih koeficientov

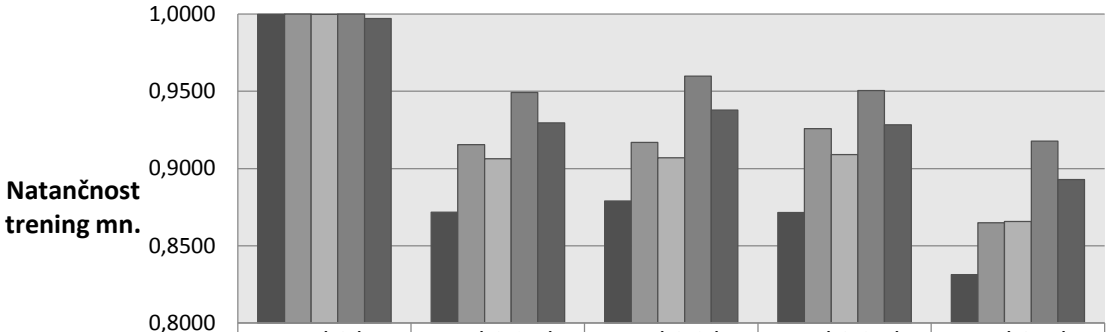
Vzorec zvoka	**Uporabljene značilke (τ_b>|0|)**	**Uporabljene značilke (τ_b>|0,2|)**	**Uporabljene značilke (τ_b>|0,25|)**	**Uporabljene značilke (τ_b>|0,3|)**	**Uporabljene značilke (τ_b>|0,35|)**
"a"	Vsi	6, 7, 9, 10, 14	10	Nobena	Nobena
"o"	Vsi	17, 24	24	24	24
"u"	Vsi	24	24	Nobena	Nobena
Številka 1	Vsi	1, 2, 3, 4, 5, 6, 7, 9 ,10, 24	1, 2, 3, 4, 5, 6, 24	1, 2, 4, 24	Nobena
Številka 2	Vsi	1, 2, 3, 4, 5, 6, 8, 9, 10, 11	1, 8, 9, 10, 11	9	Nobena
Številka 3	Vsi	12, 13, 14, 17, 19, 23, 24, 25, 26	12, 13, 17, 19, 23, 25, 26	17, 23, 25, 26	17,25,26
Številka 4	Vsi	1, 2, 3, 4, 5, 10, 20, 21	1, 2, 3, 4, 5, 10,	1, 2, 3, 4, 5,	1,2,3,4,5
Številka 5	Vsi	24	24	24	Nobena
Številka 6	Vsi	10, 24, 26	10, 26	Nobena	Nobena
Številka 7	Vsi	1, 3, 4, 5, 8, 11, 24	4, 5	4	Nobena
Številka 8	Vsi	9	9	9	Nobena
Številka 9	Vsi	2, 3, 4, 5, 21, 26	4, 26	4	Nobena
Številka 10	Vsi	1, 3, 5, 20, 23	23	Nobena	Nobena
Kratki stavek 1	Vsi	25, 26	Nobena	Nobena	Nobena
Kratki stavek 2	Vsi	3, 4, 5, 8, 10, 11, 17, 25, 26	24, 25, 26	25	25
Kratki stavek 3	Vsi	1, 2, 3, 4, 5, 10, 17, 24, 25, 26	10, 26	26	Nobena
Kratki stavek 4	Vsi	1, 2, 3, 4, 5, 10	1, 2, 3, 4, 5, 10	1, 3, 4, 5, 10, 25, 26	3,5
Beseda 1	Vsi	1, 2, 3, 4, 5, 7	1, 2, 4, 7	1, 4	Nobena
Beseda 2	Vsi	Nobena	Nobena	Nobena	Nobena
Beseda 3	Vsi	17, 19, 23, 25	17, 19, 25	17, 25	17
Beseda 4	Vsi	3, 5	Nobena	Nobena	Nobena
Beseda 5	Vsi	17 , 19, 26	Nobena	Nobena	Nobena
Beseda 6	Vsi	10, 17	10	Nobena	Nobena
Beseda 7	Vsi	3, 5, 23	Nobena	Nobena	Nobena
Beseda 8	Vsi	1, 2, 3, 4, 5, 10, 14, 17, 19, 23, 25	2, 17, 19, 25	17, 19	17
Beseda 9	Vsi	2, 3 4, 5, 24	24	Nobena	Nobena
Število klasifikatorjev	26	25	21	15	7

V primeru SOM pa je bila uporabljena 2D mreža nevronov z konfiguracijami SOM2x2, SOM3x3, SOM4x4 in SOM5x5. Topologija SOM5x5 pomeni, da smo prostor značilk posameznega zvočnega vzorca, ki na začetku znaša 26 dimenzij (26 značilk), preslikali v prostor s 25-imi dimenzijami. Za določitev oz. transformacijo značilk z uporabo SOM, smo uporabili vseh 26 klasifikatorjev v algoritmu za zaznavo PB.

5.5 Rezultati in diskusija rezultatov

Opravili smo testiranja petih različnih topologij usmerjenih nevronskih mrež (ANN), dveh z eno skrito plastjo s 5-imi in 10-imi nevroni (ANN5 in ANN10), dveh z dvema skritima plastema nevronov (ANN5-5 in ANN10-10) in ene s tremi skritimi plastmi nevronov (ANN5-10-5). ANN5-10-5 predstavlja usmerjeno nevronsko mrežo s 5-imi nevroni v prvi skriti plasti, 10-imi nevroni v drugi skriti plasti in 5-imi nevroni v tretji skriti plasti. Pri treniranju mreže smo uporabili 500 ponovitev (»epoch«). Po zaključku učenja smo naučeno nevronsko mrežo, v skladu z LOSO validacijsko shemo, ocenili z ustreznimi metrikami in postopek ponovili 30-krat. Ponovitve so bile nujne, ker je nevronska mreža podajala vsakič drugačne rezultate. Tako smo dobili povprečni odziv določene mreže in razpršenost odziva. Za vsako topologijo nevronske mreže smo spreminjali tudi Pearsonov korelacijski koeficient, Khendallov korelacijski koeficient in število nevronov SOM. Z uporabo redukcije značilk na podlagi Pearsonovega korelacijskega koeficienta, smo dobili rezultate za različne topologije usmerjenih nevronskih mrež. To je prikazano v nadaljevanju.

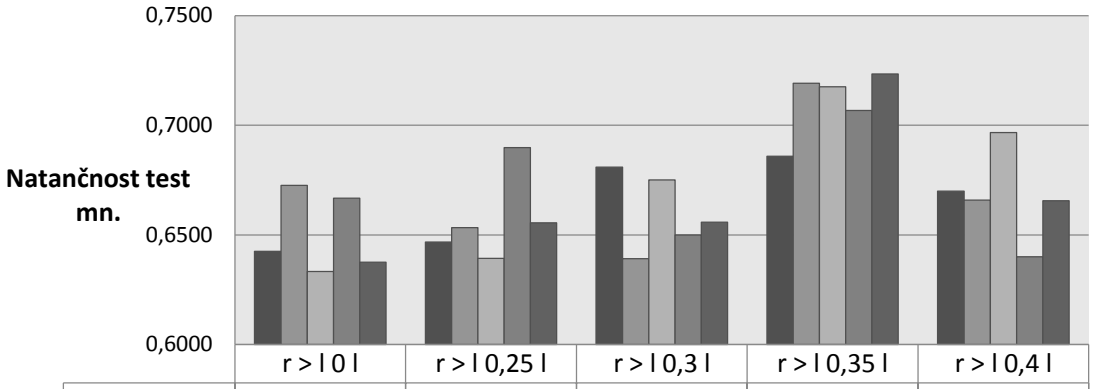
Slika 5.4 predstavlja natančnost razvrščanja (enačba (3.2)) trening množice, oz. podatkov s katerimi smo učili mrežo. Vidimo, da se ANN, brez redukcije značilk ($r > |0|$) prekomerno prilagaja. Z uporabo določitve značilk na podlagi Pearsonovega korelacijskega faktorja (pri uporabi značilk, ki ustrezajo pogoju $r > |0{,}25|$, $r > |0{,}30|$, $r > |0{,}35|$ in $r > |0{,}40|$), pa se stopnja prekomernega prilagajanja zmanjša. Topologija nevronske mreže vpliva na natančnost trening množice in tako lahko opazimo, da večje število skritih plasti pomeni večjo natančnost trening množice.



	r > | 0 |	r > | 0,25 |	r > | 0,3 |	r > | 0,35 |	r > | 0,4 |
ANN 5	0,9997	0,8718	0,8791	0,8715	0,8315
ANN 10	1,0000	0,9154	0,9169	0,9258	0,8650
ANN 5-5	0,9998	0,9064	0,9069	0,9090	0,8658
ANN 10-10	1,0000	0,9492	0,9599	0,9504	0,9178
ANN 5-10-5	0,9970	0,9296	0,9377	0,9282	0,8928

Slika 5.4: Natančnost trening množice različnih topologij ANN, z uporabo Pearsonovega korelacijskega koeficienta

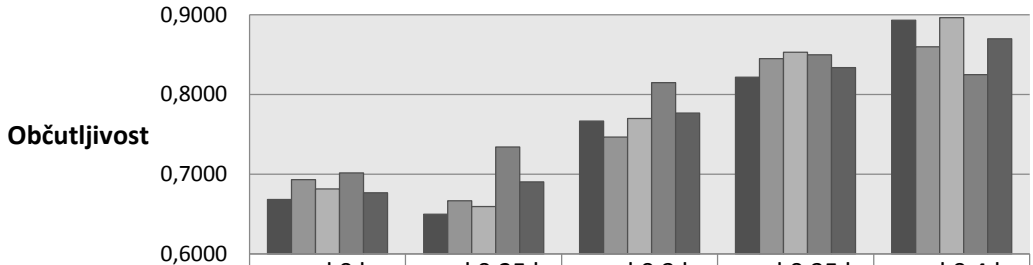
Slika 5.5 predstavlja natančnost (enačba (3.2)) testne množice, kadar imamo že naučeno mrežo in ji predstavimo podatke, katerih še ni videla. ANN brez redukcije značilk ($r > |0|$) dosega nižjo stopnjo natančnosti. Največja natančnost je dosežena z uporabo Pearsonovega korelacijskega koeficienta $r > |0{,}35|$, pravzaprav to velja za vse uporabljene topologije ANN. Najvišja testna natančnost pa je bila dosežena z topologijo ANN5-10-5 in sicer 72.34%.



	r > | 0 |	r > | 0,25 |	r > | 0,3 |	r > | 0,35 |	r > | 0,4 |
ANN 5	0,6425	0,6467	0,6808	0,6858	0,6700
ANN 10	0,6725	0,6533	0,6392	0,7192	0,6658
ANN 5-5	0,6333	0,6393	0,6750	0,7175	0,6967
ANN 10-10	0,6667	0,6898	0,6500	0,7067	0,6400
ANN 5-10-5	0,6375	0,6555	0,6558	0,7234	0,6655

Slika 5.5: Natančnost test množice različnih topologij ANN, z uporabo Pearsonovega korelacijskega koeficienta

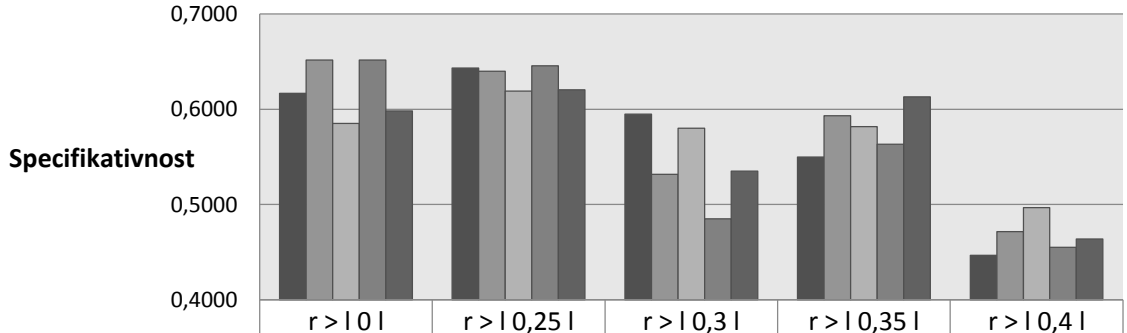
Slika 5.6 prikazuje graf občutljivosti sistema (enačba (3.3)). Občutljivost je mera za pravilno razvrščene pozitivne primere. Pozitivni primeri so v našem primeru bolniki z PB. Razvidno je, da se z redukcijo spremenljivk, na podlagi Pearsonovega korelacijskega koeficienta, poveča občutljivost algoritma za zaznavo PB. Topologija ANN skorajda nima vpliva na občutljivost.



	r > l 0 l	r > l 0,25 l	r > l 0,3 l	r > l 0,35 l	r > l 0,4 l
ANN 5	0,6683	0,6500	0,7667	0,8217	0,8933
ANN 10	0,6933	0,6667	0,7467	0,8450	0,8600
ANN 5-5	0,6817	0,6595	0,7700	0,8533	0,8967
ANN 10-10	0,7017	0,7341	0,8150	0,8500	0,8250
ANN 5-10-5	0,6767	0,6906	0,7767	0,8339	0,8699

Slika 5.6: Občutljivost različnih topologij ANN, z uporabo Pearsonovega korelacijskega koeficienta

Slika 5.7 prikazuje graf specifikativnosti (enačba (3.4)) v odvisnosti od topologije ANN in izbranega korelacijskega koeficienta. Specifikativnost je mera za pravilno razvrščene negativne primere (zdrave paciente). Razvidno je, da se z uporabo redukcijo, na podlagi Pearsonovega korelacijskega koeficienta, zmanjša specifikativnost algoritma za zaznavo PB. Topologija ANN nima bistvenega vpliva na specifikativnost.



	r > l 0 l	r > l 0,25 l	r > l 0,3 l	r > l 0,35 l	r > l 0,4 l
ANN 5	0,6167	0,6433	0,5950	0,5500	0,4467
ANN 10	0,6517	0,6400	0,5317	0,5933	0,4717
ANN 5-5	0,5850	0,6190	0,5800	0,5817	0,4967
ANN 10-10	0,6517	0,6455	0,4850	0,5633	0,4550
ANN 5-10-5	0,5983	0,6203	0,5350	0,6129	0,4638

Slika 5.7: Specifikativnost različnih topologij ANN z uporabo Pearsonovega korelacijskega koeficienta

Preglednica 5.3 nudi primerjavo dobljenih rezultatov z rezultati drugih, neodvisnih avtorjev. Vsi podatki so ocenjeni s pomočjo LOSO navzkrižne validacije. Preglednica prikazuje tudi vse najboljše rezultate dosežene na podlagi Pearsonovega in Khendallovega korelacijskega faktorja ter SOM-a. Najbolje se je odrezala redukcija značilk na podlagi Khendallovega korelacijskega koeficienta z 77.83 % natančnostjo pri $\tau_b > |0{,}3|$, z eno plastjo petih skritih nevronov ANN5. Najmanjša natančnost je bila dosežena s pomočjo SOM transformacije značilk, kjer smo pri uporabi 2D SOM$4x4$ mreže nevronov dosegli 56.44 % natančnost, z dvema plastema desetih skritih nevronov ANN10-10. V primeru, da izvajamo razvrščanje s surovimi podatki (brez redukcije značilk), so nevronske mreže dosegle najboljši rezultat natančnosti 67,25 %. V primeru določitve značilk, ki so jo uporabili drugi avtorji, se pod imenom A-MCFS skriva Pearsonov korelacijski faktor $r > |0{,}314|$. V našem primeru, smo s to metodo dosegli drugi najboljši rezultat, s 86.47 % natančnostjo (slednja natančnost je bila dosežena z različnimi topologijami posameznih ANN mrež, z uporabo različnih trening algoritmov in različnih aktivacijskih funkcij). Še uspešnešnejša je bila metoda podpornih vektorjev, ki je z uporabo kernelove funkcije RBF dosegla natančnost 87.5 %. Vidimo, da so naši rezultati povsem primerljivi z drugimi, in da bi z dodatno optimizacijo ANN nastavitev in primerno določitvijo značilk, potencialno lahko presegli natančnost metode SVM (RBF kernel).

Preglednica 5.4: Primerjava različnih rezultatov različnih razvrščevalnih algoritmov

Metoda razvrščanja	Določitev značilk	Natančnost test (%)	Občutljivost (%)	Specifikativnost (%)	MCC
k-NN (k=1)	/ [23]	53,37	49,62	57,12	0,0007
	A-MCFS [27]	70,00	80,00	60,00	0,4082
k-NN (k=3)	/ [23]	54,04	53,27	54,81	0,0008
	A-MCFS [27]	67,50	75,00	60,00	0,3540
k-NN (k=5)	/ [23]	54,42	53,65	55,19	0,0009
	A-MCFS [27]	72,50	70,00	75,00	0,4506
k-NN (k=7)	/ [23]	53,94	54,04	53,85	0,0008
	A-MCFS [27]	77,50	80,00	75,00	0,5507
Naïve Bayes	A-MCFS [27]	80,00	80,00	80,00	0,6000
SVM (linear kernel)	/ [27]	52,50	52,50	52,50	0,0006
	A-MCFS [27]	85,00	85,00	85,00	0,6000
SVM (RBF kernel)	/ [27]	55,00	60,00	50,00	0,1005
	A-MCFS [27]	87,50	90,00	85,00	0,7509
ANN 10	/	$67{,}25 \pm 4{,}52$	$69{,}33 \pm 6{,}66$	$65{,}17 \pm 5{,}65$	$0{,}3467 \pm 0{,}090$
ANN 5-10-5	Pearson	$72{,}34 \pm 4{,}54$	$83{,}39 \pm 7{,}14$	$61{,}29 \pm 5{,}86$	$0{,}4610 \pm 0{,}096$
ANN 5	Khendall	$77{,}83 \pm 4{,}44$	$77{,}50 \pm 6{,}12$	$78{,}17 \pm 4{,}64$	$0{,}5578 \pm 0{,}089$
ANN 10-10	SOM	$56{,}44 \pm 6{,}13$	$57{,}50 \pm 8{,}39$	$55{,}38 \pm 5{,}82$	$0{,}1293 \pm 0{,}123$
ANN (optimiz.)	A-MCFS	$86{,}47 \pm 3{,}27$	$88{,}91 \pm 4{,}79$	$84{,}02 \pm 5{,}10$	$0{,}7321 \pm 0{,}064$

6 SKLEP

Razvoj novih metod in učinkovitejših razvrščevalnih tehnik je ključnega pomena za uspešno obdelavo in izkoriščanje potenciala, ki ga predstavljajo zbirke podatkov. Odkrivanje znanja, ki je vsebovano v podatkih, ima velik potencial pri olajševanju človekovega vsakdanjika, izboljševanju varnosti, kvalitetnejših izdelkih, predvidevanju in še bi lahko naštevali. V magistrskem delu smo se osredotočili na proces razvrščanja kompleksnih vzorcev. Opisali smo področje umetne inteligence, strojnega učenja, sam proces in ocenitev razvrščanja ter različne algoritme za razvrščanje in določitev značilk.

V eksperimentalnem delu je bila izvedena implementacija opisanih metod. Lotili smo se problema določanja Parkinsonove bolezni s pomočjo zvoka. V ta namen je bila izdelana serija algoritmov, ki so sposobni uspešno zaznati prisotnost Parkinsonove bolezni. Doseženi rezultati so bili skladni s pričakovanji. Ugotavljamo, da izbira razvrščevalnega algoritma vpliva na uspešnost razvrščanja. Uporabljene so bile različne topologije mrež, ki so se odrezale različno dobro. To dokazuje, da sama topologija nevronske mreže vpliva na rezultate razvrščanja, ter da večje število nevronov in skritih plasti ne zagotavlja višje stopnje uspešnosti razvrščanja. S pomočjo določitve značilk smo izboljšali stopnjo uspešnosti razvrščanja. Določitve značilk na podlagi korelacijskih koeficientov nakazujejo tudi na različno stopnjo informacije o prisotnosti Parkinsonove bolezni, ki jo nosijo različni vzorci zvoka. S pomočjo postopka določitve značilk se zmanjša stopnja prekomernega prilagajanja razvrščevalnega algoritma, saj razvščevalni algoritem tekom učenja ne prilagaja modela manj pomembnim značilkam. Poleg tega, nam določitev značilk pospeši sam postopek razvrščanja.

Dobljeni rezultati nas vzpodbujajo k nadaljnjim raziskavam na tem področju. V nadaljevanju bi bilo smiselno opraviti meritve zvoka na večji populaciji ljudi. Z večjo populacijo bi narastla tudi natančnost algoritma za zaznavo Parkinsonove bolezni. V želji, določiti univerzalni vzorec zvoka, ki nosi največ informacij o prisotnosti Parkinsonove bolezni, bi raziskavo lahko razširili tudi na tujejezično prebivalstvo.

7 VIRI

[1] I. Kononenko, R. M. Šikonja, Inteligentni sistemi. Ljubljana: Založba FE in FERI, 2010.

[2] N. Guid, D. Strnad, Umetna inteligenca. Maribor: Fakulteta ze elektrotehniko, računalništvo in informatiko. 2007.

[3] School of informatics, University of Edinburgh [Splet], Dostopno: http://www.inf.ed.ac.uk/teaching/courses/ai2/module4/small_slides/small-agents.pdf [Datum dostopa: 28.7.2017].

[4] Artificial Intelligence, Agents and enviroments [Splet], Dostopno: https://dvikan.no/ntnu-studentserver/kompendier/artificial-intelligence-agents-and-environments.pdf [Datum dostopa: 28.7.2017].

[5] AITopics: An official publication of AAAI [Splet], Dostopno: https://aitopics.org/misc/brief-history [Datum dostopa: 1.8.2017].

[6] D. Poole, A. Mackworth, Artificial Intelligence: foundations of computational agents. New York: Cambridge University Press, 2010.

[7] Forbes, Applications of Artificial Intelligence In Use Today [Splet], Dosegljivo: https://www.forbes.com/sites/robertadams/2017/01/10/10-powerful-examples-of-artificial-intelligence-in-use-today/2/#10fd6cac3c8b [Datum dostopa: 3.8.2017].

[8] The Royal Society, Machine learning [Splet], Dosegljivo: https://royalsociety.org/~/media/policy/projects/machine-learning/publications/machine-learning-report.pdf [Datum dostopa: 27.7.2017].

[9] J. Han, M. Kamber in J. Rei, Data mining: Concepts and techniques, 3. izdaja. Amsterdam: Elsevier 2011.

[10] S. Karakatič (2017). »Metoda alokacije za klasifikacijo neuravnoteženih podatkov« (doktorska disertacija). Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko.

[11] Wikipedija, Precision and recall, Overfitting, Parkinson's disease [Splet], Dosegljivo: https://en.wikipedia.org/wiki/Precision_and_recall [Datum dostopa: 6.6.2017].

[12] Dato, How to evaluate machine learning models [Splet], Dosegljivo: http://blog.dato.com/how-to-evaluate-machine-learning-models-part-1-orientation [Datum dostopa: 20.5.2017].

[13] Machine Learning, Neural and Statistical Classification [Splet], Dostopno: https://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf [Datum dostopa: 1.8.2017].

[14] Udemy, Machine Learning A-Z™: Hands-ON Python & R Data Science [Splet], Dosegljivo: https://www.udemy.com/machinelearning/learn/v4/overview. [Datum dostopa: 3.8.2017].

[15] J. Stuart, P. Norvig, Artificial Intelligence: A Modern Approach. 3. izdaja. New York: Prentice Hall, 2010.

[16] An Introduction to Data Mining [Splet], Dosegljivo: http://www.saedsayad.com/decision_tree.htm [Datum dostopa: 2.8.2017].

[17] A. Criminisi, J. Shotton, Decision forests for computer vision and medical image analysis in computer vision and pattern recognition. London: Springer-Verlag, 2013.

[18] MachineLearningKernels [Splet], Dosegljivo: http://mlkernels.readthedocs.io/en/latest/kernels.html [Datum dostopa: 4.8.2017].

[19] S. Klančnik, J. Balič, F. Čuš, »Intelligent prediction of miling strategy using neural networks«. Control Cybern, let. 39, št. 1, str. 9-22, november 2009.

[20] A. Samii, J. G. Nutt, B. R. Ransom, »Parkinson's disease«. Lancet, št. 636, str. 1783–1793, 2004.

[21] J. Jankovic, »Parkinson's disease: Clinical features and diagnosis«. Journal Neurol. Neurosurgery Psychiatry, let. 79, št.4, str. 468-376, 2007.

[22] S. Skodda, »Aspects of speech rate and regularity in Parkinson's disease«. Journal of the Neurological Sciences, št. 15, str. 231-6, 2011.

[23] American Journal of Neuroradiology [Splet], Dosegljivo: http://www.ajnr.org/content/36/2/229 [Datum dostopa: 4.8.2017].

[24] B. Erdogdu Sakar, M. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, O. Kursun, »Parkinsons Speech Dataset with multiple Types of Sund Recordings Data Set«. UIC Machine Respository, 2014.

[25] UCI Machine Respository, Parkinson Speech Dataset with Multiple Types of Sound Recordings Data Set [Splet], Dosegljivo: https://archive.ics.uci.edu/ml/datasets/Parkinson+Speech+Dataset+with++Multiple+Types+of+Sound+Recordings [Datum dostopa: 8.8.2017]

[26] Praat, Doing phonetics by computer [Splet]. Dosegljivo: http://www.praat.org/ . [Datum dostopa 5.12.2016]

[27] M. Behroozi, A. Sami, "A multiple-classifier framework for Parkinson's disease detection based on various vocal tests". International Journal of Telemedicine and Applications, št. 2016, april 2016.