İZMİR BAKIRÇAY ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜH. A.B.D.

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING- A CASE STUDY

YÜKSEK LİSANS TEZİ

Burcu AKÇA

Tez Danışmanı: Doç. Dr. Orhan ER

Ağustos 2022

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING-A CASE STUDY

Yüksek Lisans Tezi

Burcu AKÇA İzmir 2022

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING-A CASE STUDY

Burcu AKÇA

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Orhan ER

İzmir
İzmir Bakırçay Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ağustos 2022

Bu tez çalışması BAP Komisyonu tarafından kabul edilen........................... no.lu proje kapsamında desteklenmiştir (Tez Çalışması Bilimsel Araştırma Projeleri Komisyonu tarafından desteklenmişse).

JÜRİ VE ENSTİTÜ ONAYI

İzmir Bakırçay Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim dalında öğrenim görmekte olan Burcu AKÇA'nın "FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING-A CASE STUDY" başlıklı tezi 08/08/2022 tarihinde aşağıdaki jüri tarafından değerlendirilerek "İzmir Bakırçay Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Jüri Üyeleri	**Unvanı Adı Soyadı**	**İmza**
Üye (1. Tez Danışmanı)	Doç. Dr. Orhan ER	
Üye (2. Tez Danışmanı)		
Üye	Doç. Dr. Deniz KILINÇ	
Üye	Dr. Öğr. Üyesi Ahmet Sertol KÖKSAL	
Üye		

Prof. Dr. Serkan ÇINARLI
Lisansüstü Eğitim Enstitüsü Müdürü

FINAL APPROVAL FOR THESIS

This thesis titled “FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING - A CASE STUDY” has been prepared and submitted by Burcu AKÇA in partial fulfillment of the requirements in “İzmir Bakırçay University Directive on Graduate Education and Examination” for the Degree of Master of Science Department has been examined and approved on 08/08/2022.

Committee Members	**Title, Name, and Surname**	**Signature**
Member (1. Thesis Supervisor)	Assoc. Prof. Dr. Orhan ER	
Member (2. Thesis Supervisor)		
Member	Assoc. Prof. Dr. Deniz KILINÇ	
Member	Asst. Prof. Dr. Ahmet Sertol KÖKSAL	
Member		

Prof. Dr. Serkan ÇINARLI

Director of Graduate Education Institute

ÖZET

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING-A CASE STUDY

Burcu AKÇA

Bilgisayar Mühendisliği Anabilim Dalı

İzmir Bakırçay Üniversitesi, Lisansüstü Eğitim Enstitüsü, Ağus.2022

Danışman: Doç. Dr. Orhan ER

Günümüzde sağlık alanında yapılan yapay zekâ çalışmalarının en önemli girdisi sağlık verisidir. Sağlık verisinin alan bilgisi uzmanları ve hekimler tarafından toplanması ve makine öğrenme algoritmalarında eğitilmesi oldukça zahmetli bir iş olup bu verilerin doğru algoritma ve parametreler ile işlenmesi, çalışmaların başarısını ortaya koymaktadır. Bu nedenlerden ötürü sağlık verisini işlemek isteyen akademisyenlere yol gösterici olması arzusu ile bir biyomedikal veri seti üzerinde özellik mühendisliği pilot çalışması amaçlandı. Bu amaç doğrultusunda uluslararası bir veri tabanından kalp yetmezliği ile ilgili örnek bir veri seti kullanıldı. Bu tezin amacına uygun olarak belirlenen veriler üzerinde yapay zekâ yöntemleri ve parametre optimizasyonu için farklı modeller kurularak deneysel çalışmalar yapıldı.

Yapılan bu çalışmada veri seti üzerinde tahmine dayalı öğrenme modelleri kullanılarak hangi yapay zekâ algoritmalarının hangi parametre setleri ile en doğru sonuca ulaşıldığı raporlandı. Sonuçlar incelendiğinde özellik mühendisliğinin veri seti üzerindeki olumlu-olumsuz performans değişimlerini kıyaslayarak karar destek sistemi oluşturmak isteyen akademisyenlere önerilerde bulunuldu. Gelecek çalışmalara zemin olacağı düşünülen bu çalışmanın farklı alanlardaki sağlık verileri için de örnek alınabileceği öngörülmektedir.

Anahtar Sözcükler: Biyomedikal Veri İşleme; Özellik Mühendisliği; Kalp Yetmezliği.

SUMMARY

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING-A CASE STUDY

Burcu AKÇA

Department of Computer Engineering

Izmir Bakircay University, Graduate Education Institute, August 2022

Supervisor: Assoc. Prof. Dr. Orhan ER

Today, the most important input of artificial intelligence studies in the field of health is medical data. The collection of medical data by field specialists and physicians and training the machine learning algorithms is a very laborious task and processing these data with the right algorithms and parameters determines the success of the study. For these reasons, a dataset on heart failure from an international database was used as a model study by feature engineering on a biomedical dataset, with the desire to guide academics who want to process health data. For this thesis, experimental studies were carried out for parameter optimization with artificial intelligence methods.

In this study, which artificial intelligence algorithm performs best is specified, by using predictive learning models on the data set. When the results were examined, suggestions were made to the academicians who wanted to create a decision support system by comparing the positive-negative performance changes on the feature engineering dataset. This study is believed to form a basis for future studies, which also may set an example for health data in different fields.

Keywords: Biomedical Data Processing; Feature Engineering; Heart Failure.

ABSTRACT

FEATURE ENGINEERING IN BIOMEDICAL DATA PROCESSING - A CASE STUDY

Burcu AKÇA

Department of Computer Engineering

Izmir Bakircay University, Graduate Education Institute, August 2022

Supervisor: Assoc. Prof. Dr. Orhan ER

Cardiovascular diseases cause approximately 17.9 million deaths each year and 32% of deaths worldwide. 85% of these deaths were due to heart attacks and strokes (URL 1, 2022). These diseases usually occur in the form of myocardial infarction and heart failure.

The data set consisting of a total of 299 samples shows which method gives higher accuracy with many methods such as Artificial Neural Networks, Fine Gaussian SVM, Fine KNN, Weighted KNN, Subspace KNN, Boosted Trees, and Bagged Trees.

As a result, it is seen that there are algorithms that can predict the diagnosis of heart failure with full accuracy (100%) according to the data obtained. This study shows that accurately predicting whether a heart failure patient will survive with which artificial intelligence algorithm will provide high accuracy.

Keywords: Biomedical Data Processing; Feature Engineering; Heart Failure.

08/08/2022

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın İzmir Bakırçay Üniversitesi tarafından kullanılan "Word Count'' bilimsel intihal tespit programıyla tarandığını ve hiçbir şekilde "intihal içermediğini" beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

(İmza)

....

Burcu AKÇA

* *Bu belgenin ciltlenmiş tezin "Abstract"tan sonraki sayfasında ıslak imzanız ile (fotokopi olmayacak) yer alması gerekmektedir.*

08/08/2022

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved by the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis, and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "Word Count" scientific plagiarism detection program used by Izmir Bakircay University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

....

Burcu AKÇA

(Name and SURNAME of the Student)

** This document has to be included with your original signature (no photocopies) on the page following the "Abstract" page of the bound copy of the thesis.*

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, supervisor Assoc. Prof. Dr. Orhan ER, for his excellent guidance, care, and patience throughout my study. I could not complete this work without his help, and professional guidance. I am also thankful for his great kindness and support.

I am grateful to my parents who have never failed to support me.

TABLE OF CONTENTS

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

HF	: Heart failure
ANN	: Artificial neural network
KNN	: K-Nearest Neighbour
SVM	: Support Vector Machine
DT	: Decision Trees
NB	: Naive Bayes
RF	: Rheumatoid Factor
CPK	: Creatinine Phosphokinase
ACC	: Accuracy

1. INTRODUCTION

The most important input of artificial intelligence studies in today's health field is health data. Collecting health data by field experts and physicians and training the machine learning algorithms is a very laborious task and processing these data with the right algorithms and parameters reveals the success of the study. For these reasons, this study was planned with the desire to guide academicians who want to process health data. It is aimed to determine which artificial intelligence algorithm gives the most accurate result by using predictive learning models on the data set consisting of 299 heart failure patient samples by using artificial intelligence algorithms with MATLAB. The estimation method, in which the highest accuracy is determined with artificial intelligence algorithms on this data set, is realized for the first time.

The continuity of human life depends on the smooth functioning of all organs. In addition, the heart, which is the most important vital organ in the human body after the brain, is responsible for pumping and distributing the blood that carries the oxygen and nutrients the body needs.

Cardiovascular diseases defined as heart or vascular diseases include various medical conditions such as coronary heart disease, cerebrovascular diseases, stroke, heart failure, hypertensive, and rheumatic heart diseases. 32% of deaths worldwide are due to cardiovascular diseases. People with cardiovascular disease or who are at high cardiovascular risk due to the presence of one or more risk factors such as hypertension, diabetes, and hyperlipidemia need an accurate, effective, continuous treatment and monitoring program (Erdas, 2020, p. 6).

Heart failure (HF), one of the cardiovascular diseases, is a clinical syndrome characterized by deterioration in body functions because of the decrease in the ability of the heart to pump blood or fill with blood. HF often occurs with an increase in left ventricular filling pressure, and risk factors such as hypertension, diabetes, obesity, high cholesterol level, stress, and smoking accelerate the progression to heart failure.

Chronic heart failure is a complex clinical condition that affects the quality of life of patients worldwide. The patient and death rate has been increasing over the years. Heart failure causes an increase in infections, increased treatment costs, prolonged hospital stay, and ultimately a decrease in the quality of life of heart failure patients and causes an excessive burden on family and society (Costa, 2020, p. 831). This causes an economic burden. There are some errors in the examination due to symptoms resembling other diseases, so when it comes to heart disease, these small mistakes can cost a life in the future (Ishaq et al. 2021, p. 2).

Predicting heart failure has become a priority for physicians. However, predicting HF-related events in clinical practice has generally failed to achieve high accuracy to date (Pfeffer & Braunwald, 2016). For this reason, electronic records can be considered a useful source of information in revealing hidden and implicit correlations and relationships between patient data for clinical practice (Chicco, 2020, p. 16).

Due to increasing medical data over time, healthcare professionals need to leverage machine learning algorithms to analyze data and assist in accurate and precise diagnoses. (Ishaq et. al, 2021, p. 4). Machine learning applied to medical records can be an effective tool both to predict the survival of each patient with heart failure symptoms and to identify the most important clinical risk factors to cause it. (Martinez-Amezcua et al., 2020, p. 10). Due to the successful prediction and classification results, shown in ready-made data sets, machine learning algorithms are frequently used in academic studies in recent years. (Chaturvedi et al., 2016 p. 28).

Some studies on the use of machine learning methods on heart failure in the literature can be summarized as follows: Patients with HF have a high mortality rate; thus, clinicians require reliable prognostic information to make wise decisions about how to use palliative care, medicine, devices, and grafts. (Wilstrup & Cave, 2021). Smith et al. aimed to develop a prognostic risk model that could distinguish vitally high and low-risk patients among HF patients and investigate the effect of EF and left ventricular wall thickness on the associated risk estimation (Nauta et al., 2018). They worked on developing and externally validating risk models to predict hospitalizations due to HF (Gianluigi & Lund, 2017, p. 7). Long-term survival after hospitalization for acute heart failure differences in the prognosis of acutely decompensated chronic and new-onset acute HF (Tan et al., 2010, p. 217)

A study conducted at the Faisalabad Institute of Cardiology and Faisalabad Allied Hospital, Pakistan, used time-dependent Cox regression and Kaplan Meier survival plots, a traditional biostatistics model that uses much fewer covariates than the Seattle Heart Failure Model, to estimate the mortality rate of 299 heart failure patients admitted to the hospital. (Ahmad et. al., 201, p. 7). The datasets, along with the analysis explanations and results of this study, were made publicly available online, making them freely accessible to the scientific community (Akgül et al., 2013, p. 425), (Voors et al., 2017, p. 627). In another study, they analyzed the same dataset to detail two different gender-based mortality prediction models. (Zahid et al.,2019). Although these studies, presented promising results, the problem was solved using standard bio-statistical methods (Oladimeji & Oladimeji, 2020, p. 90). Such methods were insufficient for large-scale data sets (Ishaq et. al. 2021, p. 2). Most researchers conducted their studies on HF patients using the linear mixed method. However, Seid et al. suggested that this linear mixed impact method applications and separate Weibull or semiparametric (Cox) proportional hazard model analysis for such data are not appropriate when associating the changeable patient health status (Moyehodie et al, 2021).

Gürfidan et al. utilized 67% of the 299 information as preparing and 33% as test information and got the most noteworthy exactness esteem as 83% with the Back Vector Machine among diverse calculations. Rahayu connected Artificial Neural Network (ANN), Decision Trees (DT), K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and Romaroid Faktör (RF) calculations for once more the same information set with the Destroyed and resample strategy. As a result, he accomplished the leading precision of 94.31% with the RF calculation utilizing the resampling procedure. (Gürfidan R, Ersoy M., 2021, p. 13).

This study utilized the dataset of the medical records of 299 HF patients admitted to the Faisalabad Institute of Cardiology and Allied Hospital in Faisalabad (Punjab, Pakistan) from April-December 2015 and shared anonymously in the international database UCI-Irvine Machine Learning Repository.

In the dataset, 105 were female and 194 were male, and the age of the patients ranged from 40 to 95. All 299 patients had left ventricular systolic dysfunction and were classified as New York Heart Association (NYHA) class III or IV as HF stage (Nunez et.al. 2017, p. 430).

This dataset (Wilstrup & Cave, 2021) includes the realization status information of 12 features and targets, namely mortality, that can be used to predict HF-related deaths. The mentioned 13 attributes and their definitions are as follows: Year, Anemia, Creatinine, Diabetes, Ejection fraction, High blood pressure, Platelets, Serum creatinine, Serum sodium, Gender, Smoking, Time, and Death event.

In this thesis, after introducing the definition of the problem, in the material method section, database definition, feature engineering and artificial intelligence methods are explained in the second chapter. Later, many models have been tried for experimental studies and are given in section 3. Finally, the 4th chapter of the thesis was completed with the interpretation of the experimental results and suggestions.

2. MATERIAL and METHODS

2.1. Data Description

Various factors affect the quality of life of heart failure patients. These are modifiable factors and non-modifiable factors. Among the modifiable factors, lifestyle habits such as diabetes mellitus, systolic blood pressure, cardiomyopathy, antihypertensive drug use, hyperlipidemia, lipid-lowering drug use, hormone replacement therapy, smoking status, physical inactivity, and alcohol consumption have been identified as critical factors. Gender, age, and heredity were defined as non-modifiable factors (Katz et al., 2017, p. 275). These uncontrollable factors cause complications such as pneumonia, pulmonary embolism, stroke, organ failure, sudden death, and disability (Costa, 2020, p. 831).

In this study, (Ahmad et. al., 2017, p. 7) utilized the dataset of the medical records of 299 HF patients admitted to the Faisalabad Institute of Cardiology and Allied Hospital in Faisalabad (Punjab, Pakistan) from April-December 2015 and shared anonymously in the international database UCI-Irvine Machine Learning Repository.

Of the patients in the dataset, 105 were female and 194 were male, and the age of the patients ranged from 40 to 95. All 299 patients had left ventricular systolic dysfunction and were classified as New York Heart Association (NYHA) class III or IV as HF stage (Nunez et.al. 2017, p. 430). This dataset (Wilstrup & Cave, 2021) includes the realization status information of 12 features and targets, namely mortality, that can be used to predict HF-related deaths. The mentioned 13 attributes and their definitions are as follows:

1. Year; patient age in years,
2. Anemia; decrease in red blood cells or hemoglobin,
3. Creatinine; CPK enzyme levels in the blood (mcg/L),
4. Diabetes, whether the patient has diabetes,
5. Ejection fraction; percentage of blood leaving the heart with each contraction
6. High blood pressure, whether the patient has hypertension
7. Platelets; platelets in the blood (kilo platelets/mL)
8. Serum creatinine; serum creatinine level in the blood (mg/dL)

9. Serum sodium; serum sodium level in the blood (mEq/L)

10. Gender: female or male

11. Smoking; whether the patient smokes or not

12. Time; patient's follow-up period in days

13. Death event; the patient's death status during the follow-up period

The characteristics of age, serum creatinine, left ventricular dysfunction and pulmonary hypertension in the data set are numerical. In binary categorical (binary) features, “0” indicates that the risk factor is not present, and “1” indicates that the risk factor is present. Six of the features, including anemia, hypertension, diabetes, gender, smoking, and death event, were converted into binary to make the dataset used for the classification task. The registrar assumed that patients with hematocrit levels of less than 36% had anemia (Chen et. al. 2020, p. 1445).

Creatinine phosphokinase (CPK) alludes to the level of the CPK chemical within the blood. When muscle tissue is harmed, CPK is discharged into the blood. Therefore, high CPK levels within the patient's blood may show HF or harm (Vistarini et. al. 2014, p. 238). phosphokinase (CPK) refers to the level of the CPK enzyme in the blood. When muscle tissue is damaged, CPK is released into the blood. Therefore, high CPK levels in the patient's blood may indicate HF or injury (Salim et. al. 2020, p.139)

EF indicates how much blood the left ventricle pumps with each contraction. Serum creatinine is an organic waste formed by muscle metabolism (Erdas. C, 2020, p. 6). Sodium is a mineral that serves the proper functioning of muscles and nerves. The serum sodium test is a routine blood examination that shows whether the patient has normal levels of sodium in their blood.

The abnormally low sodium level in the blood may be due to HF. The death or survival status used as a target in our classification study indicates that the patient died or survived before the end of a mean follow-up period of 130 days, ranging from 4 to 285 days (Chen et. al. 2020, s. 1445). The patient's survival (mortality =0) and death (mortality =1) were expressed in binary. Regarding the data set imbalance, the data set has an imbalance of approximately 2:1, since the number of patients who survived was 203 (67.89%) and the number of patients who died was 96 (32.11%)

Among these clinical features, the first twelve features are accepted as independent variables. A part of the data set is shown in Table 2.1.

Table 2.1. Data Set Sampling

Age	Anemia	Creatinine phosphokinase	Diabetes	Ejection fraction	High Blood pressure	Platelets	Serum creatinine	Serum sodium	Sex	Smoking	Time	Death event
75	0	582	0	20	1	265000	1,9	130	1	0	4	1
55	0	7861	0	38	0	263358	1,1	136	1	0	6	1
65	0	146	0	20	0	162000	1,3	129	1	1	7	1
50	1	111	0	20	0	210000	1,9	137	1	0	7	1
65	1	160	1	20	0	327000	2,7	116	0	0	8	1
90	1	47	0	40	1	204000	2,1	132	1	1	8	1
75	1	246	0	15	0	127000	1,2	137	1	0	10	1
60	1	315	1	60	0	454000	1,1	131	1	1	10	1
65	0	157	0	65	0	263358	1,5	138	0	0	10	1
80	1	123	0	35	1	388000	9,4	133	1	1	10	1
75	1	81	0	38	1	368000	4	131	1	1	10	1
62	0	231	0	25	1	253000	0,9	140	1	1	10	1
45	1	981	0	30	0	136000	1,1	137	1	0	11	1
50	1	168	0	38	1	276000	1,1	137	1	0	11	1
49	1	80	0	30	1	427000	1	138	0	0	12	2
82	1	379	0	50	0	47000	1,3	136	1	0	13	1
87	1	149	0	38	0	262000	0,9	140	1	0	14	1
45	0	582	0	14	0	166000	0,8	127	1	0	14	1
70	1	125	0	25	1	237000	1	140	0	0	15	1
48	1	582	1	55	0	87000	1,9	121	0	0	15	1
65	1	52	0	25	1	276000	1,3	137	0	0	16	2
65	1	128	1	30	1	297000	1,6	136	0	0	20	1
68	1	220	0	35	1	289000	0,9	140	1	1	20	1
53	0	63	1	60	0	368000	0,8	135	1	0	22	2
75	0	582	1	30	1	263358	1,83	134	0	0	23	1
80	0	148	1	38	0	149000	1,9	144	1	1	23	1
95	1	112	0	40	1	196000	1	138	0	0	24	1
70	0	122	1	45	1	284000	1,3	136	1	1	26	1
58	1	60	0	38	0	153000	5,8	134	1	0	26	1
82	0	70	1	30	0	200000	1,2	132	1	1	26	1
94	0	582	1	38	1	263358	1,83	134	1	0	27	1
85	0	23	0	45	0	360000	3	132	1	0	28	1
50	1	249	1	35	1	319000	1	128	0	0	28	1
50	1	159	1	30	0	302000	1,2	138	0	0	29	2
65	0	94	1	50	1	188000	1	140	1	0	29	1
69	0	582	1	35	0	228000	3,5	134	1	0	30	1
90	1	60	1	50	0	226000	1	134	1	0	30	1
82	1	855	1	50	1	321000	1	145	0	0	30	1

2.2. Feature Engineering

Machine learning calculations learn from input information related to the area of the subject. It is exceptionally vital to nourish the calculation with the proper information for the arrangement of an indicated issue. Indeed, if the information is fundamental and collected carefully, significant features ought to be included within the framework in an indicated organize and affectability. To urge this noteworthy information format:

1.Data selection
2.Data pre-processing
3.Data transformation

In the scope of machine learning, a feature is a measurable variable that is used to explain some part of individual data objects. For example, sepal length and petal length are some of the features that are used to describe species of iris flower in the Iris Data Set (Dua & Graff, 2017).

To design effective machine learning models, comprehensive and independent features that explain the underlying information on the target variable should be presented. Feature engineering is the process of transforming, pre-processing, and selecting features on the collected data sets. Even with the recent developments in the data analytics and machine learning area, most of the designed algorithms are not fully capable of understanding the reasoning behind the target variables only being applied to a collected data set. Machine learning experts are needed for generating features to extract useful information for machine learning models to work.

Extracting meaningful features requires extensive domain knowledge. The process of feature engineering is not a simple line but rather a cycle of learning that goes back and forth between the feature engineering stage and model development stage. There are three broadly utilized' 'data preprocessing ‘‘steps within the writing: cleaning, organizing, and examining. These exchanges are carried out, separately, as takes after: to begin with, the superfluous and lost information are cleaned from the information. At that point, a designing handle known as numerical values and normalization is performed on the cleaning information.

Machine learning procedures cannot work on substance data. Inputs given to the organization are changed over into numerical values. Sexual introduction can be given as a case for this alter (digitized as male = 1, female = 2). Thus, the normalization gets ready is carried out to expect the highlights associated with the inputs of the organization from having a one-sided effect on the organization. After ensuring that all inputs have a homogeneous effect on the organization, looking at trade is carried out. Two common techniques are utilized here: customary data division and k-fold cross validation.

Exchanges are connected. Information change steps are moreover called include designing. By with scaling exchange at the point of classifying the highlights inside themselves and deciding their characteristics by selecting a particular subset from the whole existing information set, the ‘‘Data Selection’’ transaction is exhausted. common, there's a want to choose the whole information set that exists with the logic of’ ’much better ‘‘. This may not be genuine. It is essential to know which information influences the issue that should be unraveled. Information choice for vital information can be done on suspicions and within the way of affirming the presumption afterward. In this think, 70% of information was arbitrarily chosen for preparation and 30% of the information for test set traditionally.

The final step is the’ ’data conversion’’ exchange which straightforwardly influences the issue region of the calculation utilized. Numerous information change exchanges may be required in numerous considers. There are three common information change strategies: scaling, breaking down, and combining. When the writing is inspected, it is seen that the exchanges of combining, breaking down, or scaling is done together within the preprocessing step of numerous problems.

It is the exchange of exploring the impacts of highlights on the arrangement after starting. The vital component investigation (PCA) strategy is the essence of the foremost broadly utilized methods here. PCA strategy is additionally a measurement decrease and includes an extraction strategy that gives extensive results beneath the presumption that the information incorporates a regular conveyance. The combining exchange is based on the method of combining the passages including comparable characteristics amid the highlight. The algorithms such as profound learning perform these information transformation exchanges as a closed box inside themselves, but this exchange needs to be done by a master in machine learning calculations.

In this think, we utilized the essential operations in include building. No extra preparation was done after a test diminishment preparation made by the database proprietor. In expansion, we centered as it were on making strides in the execution of highlight building on this dataset. The data operations were performed on the whole information set. A while later, the information set was partitioned with distinctive procedures for preparing and testing. In this way, conceivable factual blunders are anticipated.

2.2.1. Data selection

Data Selection is done by selecting a specific subset from the entire existing dataset. There is usually a desire to select the entire existing dataset with the philosophy of much better. This may not be true. It is necessary to know which data influences the problem that needs to be solved. Data selection can be made on assumptions about the data that is important and in a way that can confirm the assumption afterward.

2.2.2. Data pre-processing

There are 3 commonly used "data pre-processing" steps: cleaning, formatting, and sampling. These operations are performed as follows: first, unnecessary and missing data is cleaned on the data. Then, numerical values and formatting, known as normalization, are performed on the cleaned data. Machine learning methods cannot work on text data. The inputs to the network are converted to numeric values. Then, the normalization process is performed to prevent the properties applied to the inputs of the network from having a biased effect on the network. After ensuring that all inputs have a homogeneous effect on the network, sampling is done. Two methods commonly used here are traditional data segmentation and k-fold cross validation.

2.2.3 Data transforming

Data transformation is a method that directly affects the problem area of the algorithm used. There are three general methods of data transformation: scaling, attribute parsing, and aggregation. It is often seen that merging, parsing, or scaling jobs for many problems are done together in the preprocessing step.

2.3. Artificial Intelligence Algorithms

Artificial intelligence algorithms refer to software possibilities that bring real-world assets to the digital platform. AI algorithms are handled in three different categories: supervised learning, unsupervised learning, and reinforcement learning.

Computers and computer-based systems that have the ability to learn and evaluate with artificial intelligence algorithms; It is used to process the data and increase the performance of the results.

2.3.1. Support Vector Machine (SVM)

Whereas the bolster vector machine was initially utilized to partition the two classes, it has been created over time and has been effectively utilized in relapse, classification, and exception discovery issues with nonlinear frameworks. It may be a directed parametric machine learning calculation based on measurable learning theory. To partition the two classes within the SVM calculation, a parallel line/hyperplane is drawn between the information that produces up the classes. The structure utilized to partitioned classes is spoken to as a line in two-dimensional space, and as a plane in three-dimensional space. The information closest to the hyperplane is called bolster vectors. The edge between the bolster vectors of inverse classes is maximized, in this way making it stronger.

2.3.2. Artificial Neural Networks (ANN)

Artificial neural systems (ANNs), one of the foremost common machine learning strategies, are frameworks shaped by the combination of straightforward data preparing units called neurons. ANNs are very competent of learning nonlinear connections between factors and recognizing high-order connections. The control of ANNs to demonstrate complex connections comprises not in complex scientific models, but the intuitively get together of expansive numbers of basic neurons. ANNs are models that can be fully applied to supervised, unsupervised, and reinforcement learning algorithms.

2.3.3. k-Nearest Neighbor Algorithm (KNN)

K-NN is known as one of the best and most seasoned non-parametric directed classification approaches among machine learning calculations within the writing. By characterizing a extraordinary number k within the add up to information set, the mean/mode classes of the closest neighbors are gotten, and the modern protest is doled out to the course closest to its neighbors. The separations of the unused question to its neighbors can be calculated with capacities such as Euclid. It contains a vigorous structure against preparing information given the k-number is expansive sufficient. When the information set and k measure increment, the handling time increment significantly, and in this approach, all these remove calculations must be kept in memory. Hence, the choice of k esteem is greatly vital.

2.3.4. Decision Tree Classifier Algorithm

The choice tree calculation falls beneath the category of administered learning. They are utilized to illuminate both relapse and classification issues. The choice tree employments tree representation to illuminate the issue where each leaf hub compares to a course name and the qualities are spoken to at the inward hub of the tree. Measurements are one of the prescient modeling approaches utilized in information mining and machine learning.

Tree models in which the target variable can take a discrete set of values are called classification trees; in these tree structures, clears out speak to lesson names and branches speak to combinations of properties that provide rise to these course names. Choice trees where the target variable can take ceaseless values are called relapse trees. In uncertainty investigation, a choice tree can be utilized to speak to choices outwardly and clearly.

2.3.5. Accuracy

The proportion of predictions that a classification model is correct. Accuracy in multiclass classification is defined as:

Accuracy = Correct guesses / Total number of samples

In binary classification, accuracy has the following definition:

Accuracy = True positives + True negatives / Total number of samples

3. EXPERIMENTAL STUDIES

In this part, we evaluate the artificial neural network results based on classification accuracy. For this purpose, a dataset containing 13 features (12 Inputs - 1 Output) and 299 samples were used. The traditional validation approach is used to evaluate the performance of the proposed algorithms.

Heart failure diagnostic data was tested with many different machine learning techniques to demonstrate the success of the study. For this purpose, Logistic Regression, Naive Bayes, Linear SVM, Cubic SVM, Fine Gauss SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN, Weighted KNN, Subspace KNN, Boosted Trees, Bagged Trees, RUS Boosted Trees methods have been tested and the classification performance results obtained with different classifiers using all features are shown in Table 3.1.

3.1. Experimental Studies with Dataset Unnormalized

In this section, our data set is not normalized. Different models have been tested on raw data. The performance results of the models are given below:

Table 3.1.1. Machine learning techniques for comparison-no validation

Machine Learning Algorithms		**No Validation (ACC-%)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	93%	84%
2	Tree (Medium Tree)	92%	72%
3	Tree (Coarse Tree)	86%	69%
4	Linear Discriminant	85%	67%
5	Quadratic Discriminant	79%	67%
6	Logistic Regression	85%	67%
7	Naive Bayes (Kernel)	76%	67%
8	Naive Bayes (Gaussian)	84%	68%
9	SVM (Linear Support Vector Machine)	84%	67%
10	SVM (Quadratic)	90%	66%
11	SVM (Cubic)	**98%**	32%
12	SVM (Fine Gaussian)	**100%**	67%
13	SVM (Medium Gaussian)	90%	67%
14	SVM (Coarse Gaussian)	78%	67%
15	KNN (Fine KNN)	**100%**	**100%**
16	KNN (Medium KNN)	78%	69%
17	KNN (Coarse KNN)	69%	67%
18	KNN (Cosine KNN)	77%	32%
19	KNN (Cubic KNN)	79%	69%
20	KNN (Weighted KNN)	**100%**	**100%**
21	Ensemble (Boosted Trees)	**100%**	73%
22	Ensemble (Bagged Trees)	**99%**	98%
23	Ensemble (Subspace Discriminant)	80%	67%
24	Ensemble (Subspace KNN)	**100%**	**100%**
25	Ensemble (RUSBoosted Trees)	**97%**	73%
	Best Results	**100%**	**100%**

The dataset is divided by the no validation method and the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.1.1. According to this table, SVM (Cubic), SVM (Fine Gaussian), KNN (Fine KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Bagged Trees), Ensemble (Subspace KNN), Ensemble (RUSboosted Trees) models have the highest accuracy rate.

Quadratic Discriminant, Naive Bayes (Kernel), SVM (Coarse Gaussian), KNN (Fine KNN), KNN (Medium KNN), KNN (Coarse KNN), KNN (Cosine KNN), KNN (Cubic KNN) models have the lowest accuracy rate.

Table 3.1.2. Machine learning techniques for comparison-holdout validation-20-80%

Machine Learning Algorithms		**Holdout Validation-20/80% (ACC-%)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	78%	61%
2	Tree (Medium Tree)	78%	55%
3	Tree (Coarse Tree)	79%	**67%**
4	Linear Discriminant	**84%**	**67%**
5	Quadratic Discriminant	79%	**67%**
6	Logistic Regression	**83%**	**67%**
7	Naive Bayes (Kernel)	78%	**67%**
8	Naive Bayes (Gaussian)	**81%**	**67%**
9	SVM (Linear Support Vector Machine)	79%	**67%**
10	SVM (Quadratic)	71%	66%
11	SVM (Cubic)	61%	28%
12	SVM (Fine Gaussian)	67%	**67%**
13	SVM (Medium Gaussian)	78%	**67%**
14	SVM (Coarse Gaussian)	79%	**67%**
15	KNN (Fine KNN)	57%	49%
16	KNN (Medium KNN)	69%	57%
17	KNN (Coarse KNN)	67%	**67%**
18	KNN (Cosine KNN)	76%	32%
19	KNN (Cubic KNN)	71%	57%
20	KNN (Weighted KNN)	72%	52%
21	Ensemble (Boosted Trees)	79%	61%
22	Ensemble (Bagged Trees)	**81%**	52%
23	Ensemble (Subspace Discriminant)	**83%**	**67%**
24	Ensemble (Subspace KNN)	61%	49%
25	Ensemble (RUSBoosted Trees)	72%	47%
	Best Results	**84%**	**67%**

The dataset is divided by the holdout validation (20-80%) method and the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.1.2. According to this table, Linear Discriminant, Logistic Regression, Naive Bayes (Gaussian), Ensemble (Bagged Trees), and Ensemble (Subspace Discriminant) models have the highest accuracy rate. Quadratic Discriminant SVM (Cubic), KNN (Fine KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

Table 3.1.3. Machine learning techniques for comparison-holdout validation-15-85%

Machine Learning Algorithms		**Holdout Validation-15/85% (ACC-%)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	70%	52%
2	Tree (Medium Tree)	70%	59%
3	Tree (Coarse Tree)	72%	**70%**
4	Linear Discriminant	**81%**	68%
5	Quadratic Discriminant	70%	68%
6	Logistic Regression	79%	68%
7	Naïve Bayes (Kernel)	70%	68%
8	Naïve Bayes (Gaussian)	72%	68%
9	SVM (Linear Support Vector Machine)	**81%**	68%
10	SVM (Quadratic)	68%	36%
11	SVM (Cubic)	77%	65%
12	SVM (Fine Gaussian)	68%	68%
13	SVM (Medium Gaussian)	77%	68%
14	SVM (Coarse Gaussian)	72%	68%
15	KNN (Fine KNN)	59%	50%
16	KNN (Medium KNN)	70%	**70%**
17	KNN (Coarse KNN)	70%	68%
18	KNN (Cosine KNN)	72%	31%
19	KNN (Cubic KNN)	72%	**70%**
20	KNN (Weighted KNN)	72%	54%
21	Ensemble (Boosted Trees)	77%	65%
22	Ensemble (Bagged Trees)	**81%**	47%
23	Ensemble (Subspace Discriminant)	77%	68%
24	Ensemble (Subspace KNN)	65%	50%
25	Ensemble (RUSBoosted Trees)	70%	50%
	Best Results	**81%**	**70%**

The dataset is divided by the holdout validation (15-85%) method and the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.1.3. According to this table, Linear Discriminant, SVM (Linear Support Vector Machine), and Ensemble (Bagged Trees) models have the highest accuracy rate.

Quadratic Discriminant, SVM (Quadratic), SVM (Fine Gaussian), KNN (Fine KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

Table 3.1.4. Machine learning techniques for comparison-k-fold-cross-validation (k=3)

Machine Learning Algorithms		**k-fold-cross-validation (k=3) (ACC-%)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	77%	62%
2	Tree (Medium Tree)	77%	62%
3	Tree (Coarse Tree)	78%	64%
4	Linear Discriminant	**83%**	**67%**
5	Quadratic Discriminant	76%	66%
6	Logistic Regression	82%	**67%**
7	Naïve Bayes (Kernel)	75%	66%
8	Naïve Bayes (Gaussian)	77%	66%
9	SVM (Linear Support Vector Machine)	**82%**	**67%**
10	SVM (Quadratic)	75%	63%
11	SVM (Cubic)	73%	64%
12	SVM (Fine Gaussian)	67%	**67%**
13	SVM (Medium Gaussian)	**80%**	**67%**
14	SVM (Coarse Gaussian)	74%	**67%**
15	KNN (Fine KNN)	69%	57%
16	KNN (Medium KNN)	76%	61%
17	KNN (Coarse KNN)	67%	**67%**
18	KNN (Cosine KNN)	76%	32%
19	KNN (Cubic KNN)	75%	61%
20	KNN (Weighted KNN)	75%	60%
21	Ensemble (Boosted Trees)	75%	62%
22	Ensemble (Bagged Trees)	**81%**	57%
23	Ensemble (Subspace Discriminant)	78%	**67%**
24	Ensemble (Subspace KNN)	63%	57%
25	Ensemble (RUSBoosted Trees)	75%	54%
	Best Results	**83%**	**67%**

The dataset is divided by the **k**-fold-cross-validation (k=3) method, and the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.1.4. According to this table, Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), SVM (Medium Gaussian), and Ensemble (Bagged Trees) models have the highest accuracy rate.

Quadratic Discriminant SVM (Fine Gaussian), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

Table 3.1.5. Machine learning techniques for comparison-k-fold-cross-validation (k=5)

Machine Learning Algorithms		**k-fold-cross-validation (k=5) (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	78%	56%
2	Tree (Medium Tree)	78%	61%
3	Tree (Coarse Tree)	**81%**	**67%**
4	Linear Discriminant	**82%**	**67%**
5	Quadratic Discriminant	74%	**67%**
6	Logistic Regression	**82%**	**67%**
7	Naïve Bayes (Kernel)	76%	**67%**
8	Naïve Bayes (Gaussian)	76%	**67%**
9	SVM (Linear Support Vector Machine)	**80%**	**67%**
10	SVM (Quadratic)	74%	62%
11	SVM (Cubic)	72%	51%
12	SVM (Fine Gaussian)	67%	**67%**
13	SVM (Medium Gaussian)	78%	**67%**
14	SVM (Coarse Gaussian)	73%	**67%**
15	KNN (Fine KNN)	65%	52%
16	KNN (Medium KNN)	75%	61%
17	KNN (Coarse KNN)	68%	**67%**
18	KNN (Cosine KNN)	74%	32%
19	KNN (Cubic KNN)	75%	61%
20	KNN (Weighted KNN)	73%	56%
21	Ensemble (Boosted Trees)	77%	62%
22	Ensemble (Bagged Trees)	**81%**	51%
23	Ensemble (Subspace Discriminant)	78%	**67%**
24	Ensemble (Subspace KNN)	64%	52%
25	Ensemble (RUSBoosted Trees)	79%	46%
	Best Results	**82%**	**67%**

The dataset is divided by the k-fold-cross-validation (k=5) method, and the model performance performances obtained in PCA Enable and Disable states are given in Table 3.1.5. According to this table, Tree (Coarse Tree), Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), and SVM (Linear Support Vector Machine) models have the highest accuracy rate.

Quadratic Discriminant, SVM (Fine Gaussian), KNN (Fine KNN), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

Table 3.1.6. Machine learning techniques for comparison-k-fold-cross-validation (k=10)

Machine Learning Algorithms		**k-fold-cross-validation (k=10) (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	77%	57%
2	Tree (Medium Tree)	78%	59%
3	Tree (Coarse Tree)	**81%**	64%
4	Linear Discriminant	**81%**	**67%**
5	Quadratic Discriminant	76%	66%
6	Logistic Regression	**81%**	**67%**
7	Naïve Bayes (Kernel)	76%	66%
8	Naïve Bayes (Gaussian)	78%	66%
9	SVM (Linear Support Vector Machine)	**81%**	**67%**
10	SVM (Quadratic)	74%	58%
11	SVM (Cubic)	72%	48%
12	SVM (Fine Gaussian)	67%	**67%**
13	SVM (Medium Gaussian)	79%	**67%**
14	SVM (Coarse Gaussian)	74%	**67%**
15	KNN (Fine KNN)	64%	55%
16	KNN (Medium KNN)	72%	56%
17	KNN (Coarse KNN)	69%	**67%**
18	KNN (Cosine KNN)	72%	32%
19	KNN (Cubic KNN)	73%	56%
20	KNN (Weighted KNN)	72%	55%
21	Ensemble (Boosted Trees)	**80%**	63%
22	Ensemble (Bagged Trees)	**84%**	55%
23	Ensemble (Subspace Discriminant)	78%	**67%**
24	Ensemble (Subspace KNN)	60%	55%
25	Ensemble (RUSBoosted Trees)	**80%**	48%
	Best Results	**84%**	**67%**

The dataset is divided by the k-fold-cross-validation (k=10) method and the model performance performances obtained in PCA Enable and Disable states are given in Table 3.1.6. According to this table, Tree (Coarse Tree), Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), Ensemble (Boosted Trees), Ensemble (Bagged Trees), and Ensemble (RUSBoosted Trees) models have the highest accuracy rate.

SVM (Fine Gaussian), KNN (Fine KNN), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

3.2. Experimental Studies with Dataset Normalized

The dataset is divided by the no validation method and the normalization of the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.1.

Table 3.2.1. Machine learning techniques for comparison-no validation-normalization-I

Normalized Dataset **Machine Learning Algorithms**		**No Validation** **(ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	80**%**	80**%**
2	Tree (Medium Tree)	70**%**	60**%**
3	Tree (Coarse Tree)	60**%**	50**%**
4	Linear Discriminant	50**%**	50**%**
5	Quadratic Discriminant	30**%**	50**%**
6	Logistic Regression	50**%**	50**%**
7	Naïve Bayes (Kernel)	20**%**	50**%**
8	Naïve Bayes (Gaussian)	50**%**	50**%**
9	SVM (Linear Support Vector Machine)	50**%**	50**%**
10	SVM (Quadratic)	70**%**	52**%**
11	SVM (Cubic)	90**%**	50**%**
12	SVM (Fine Gaussian)	**100%**	50**%**
13	SVM (Medium Gaussian)	70**%**	50**%**
14	SVM (Coarse Gaussian)	30**%**	52**%**
15	KNN (Fine KNN)	100**%**	**100%**
16	KNN (Medium KNN)	30**%**	50**%**
17	KNN (Coarse KNN)	50**%**	50**%**
18	KNN (Cosine KNN)	30**%**	30**%**
19	KNN (Cubic KNN)	30**%**	50**%**
20	KNN (Weighted KNN)	**100%**	**100%**
21	Ensemble (Boosted Trees)	**100%**	60**%**
22	Ensemble (Bagged Trees)	**100%**	**100%**
23	Ensemble (Subspace Discriminant)	42**%**	50**%**
24	Ensemble (Subspace KNN)	**100%**	100%
25	Ensemble (RUSBoosted Trees)	90**%**	60**%**
	Best Results	**100%**	**100%**

Table 3.2.2. Machine learning techniques for comparison-no validation-normalization-II

Normalized **Machine Learning Algorithms**		**No Validation** **(ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	28**%**	28**%**
2	Tree (Medium Tree)	28**%**	28**%**
3	Tree (Coarse Tree)	20**%**	20**%**
4	Linear Discriminant	Failed*	Failed*
5	Quadratic Discriminant	Failed*	Failed*
6	Logistic Regression	Failed*	Failed*
7	Naïve Bayes (Kernel)	**88%**	**88%**
8	Naïve Bayes (Gaussian)	Failed*	Failed*
9	SVM (Linear Support Vector Machine)	40**%**	60**%**
10	SVM (Quadratic)	**88%**	**88%**
11	SVM (Cubic)	**88%**	**88%**
12	SVM (Fine Gaussian)	**88%**	**88%**
13	SVM (Medium Gaussian)	**88%**	**88%**
14	SVM (Coarse Gaussian)	40**%**	60**%**
15	KNN (Fine KNN)	**88%**	**88%**
16	KNN (Medium KNN)	8**%**	12**%**
17	KNN (Coarse KNN)	4**%**	4**%**
18	KNN (Cosine KNN)	8**%**	12**%**
19	KNN (Cubic KNN)	8**%**	12**%**
20	KNN (Weighted KNN)	**88%**	**88%**
21	Ensemble (Boosted Trees)	**88%**	**88%**
22	Ensemble (Bagged Trees)	**88%**	**88%**
23	Ensemble (Subspace Discriminant)	4**%**	4**%**
24	Ensemble (Subspace KNN)	52**%**	**88%**
25	Ensemble (RUSBoosted Trees)	**84%**	**84%**
	Best Results	**88%**	**88%**

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset with normalized is divided by the no validation method and the model performances obtained in PCA Enable and Disable states are given in Table 3.2.2.

According to this table, SVM (Quadratic), SVM (Cubic), SVM (Fine Gaussian), SVM (Medium Gaussian), KNN (Fine KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Bagged Trees), Ensemble (RUSBoosted Trees) models have the highest accuracy rate.

KNN (Medium KNN), KNN (Coarse KNN), KNN (Cosine KNN), KNN (Cubic KNN) Ensemble (Subspace Discriminant) models have the lowest accuracy rate.

Table 3.2.3. Machine learning techniques for comparison-holdout validation-20-80%-normalization-I

Normalized Dataset **Machine Learning Algorithms**		**Holdout Validation 20/80%** **(ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	**80%**	**80%**
2	Tree (Medium Tree)	**80%**	70**%**
3	Tree (Coarse Tree)	**80%**	**100%**
4	Linear Discriminant	**10%**	**100%**
5	Quadratic Discriminant	**80%**	**100%**
6	Logistic Regression	**10%**	**100%**
7	Naïve Bayes (Kernel)	**80%**	**100%**
8	Naïve Bayes (Gaussian)	**90%**	**100%**
9	SVM (Linear Support Vector Machine)	**80%**	**100%**
10	SVM (Quadratic)	50**%**	**100%**
11	SVM (Cubic)	10**%**	0**%**
12	SVM (Fine Gaussian)	40**%**	**100%**
13	SVM (Medium Gaussian)	**80%**	**100%**
14	SVM (Coarse Gaussian)	**80%**	**100%**
15	KNN (Fine KNN)	0**%**	50**%**
16	KNN (Medium KNN)	40**%**	70**%**
17	KNN (Coarse KNN)	40**%**	**100%**
18	KNN (Cosine KNN)	70**%**	10**%**
19	KNN (Cubic KNN)	50**%**	70**%**
20	KNN (Weighted KNN)	60**%**	60**%**
21	Ensemble (Boosted Trees)	**80%**	**80%**
22	Ensemble (Bagged Trees)	**90%**	60**%**
23	Ensemble (Subspace Discriminant)	**100%**	**100%**
24	Ensemble (Subspace KNN)	10**%**	50**%**
25	Ensemble (RUSBoosted Trees)	60**%**	50**%**
	Best Results	**100%**	**100%**

The dataset is divided by the no validation method and the normalization of the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.3.

Table 3.2.4. Machine learning techniques for comparison-holdout validation-20-80%-normalization-II

Normalized Dataset **Machine Learning Algorithms**		**Holdout Validation 20/80%** **(ACC - %)**	
		PCA **Disable**	**PCA** **Enable**
1	Tree (Fine Tree)	0**%**	0**%**
2	Tree (Medium Tree)	0**%**	0**%**
3	Tree (Coarse Tree)	0**%**	0**%**
4	Linear Discriminant	Failed *	0**%**
5	Quadratic Discriminant	Failed*	0**%**
6	Logistic Regression	Failed*	0**%**
7	Naïve Bayes (Kernel)	0**%**	0**%**
8	Naïve Bayes (Gaussian)	Failed*	0**%**
9	SVM (Linear Support Vector Machine)	**83**	**83%**
10	SVM (Quadratic)	0**%**	**83%**
11	SVM (Cubic)	0**%**	**83%**
12	SVM (Fine Gaussian)	**83%**	**83%**
13	SVM (Medium Gaussian)	**83%**	**83%**
14	SVM (Coarse Gaussian)	**83%**	**83%**
15	KNN (Fine KNN)	0**%**	0**%**
16	KNN (Medium KNN)	0**%**	0**%**
17	KNN (Coarse KNN)	0**%**	0**%**
18	KNN (Cosine KNN)	0**%**	0**%**
19	KNN (Cubic KNN)	0**%**	0**%**
20	KNN (Weighted KNN)	0**%**	0**%**
21	Ensemble (Boosted Trees)	0**%**	0**%**
22	Ensemble (Bagged Trees)	0**%**	0**%**
23	Ensemble (Subspace Discriminant)	0**%**	0**%**
24	Ensemble (Subspace KNN)	0**%**	0**%**
25	Ensemble (RUSBoosted Trees)	0**%**	0**%**
	Best Results	**83%**	**83%**

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset is divided by the normalization-holdout validation (80-20%) method and the model performance performances obtained in PCA Enable and Disable states are given in Table 3.2.4.

According to this table, SVM (Linear Support Vector Machine), SVM (Fine Gaussian), SVM (Medium Gaussian), and SVM (Coarse Gaussian) models have the highest accuracy rate.

Tree (Fine Tree), Tree (Medium Tree), Tree (Coarse Tree), Naïve Bayes (Kernel), SVM (Quadratic), SVM (Cubic), KNN (Fine KNN), KNN (Medium KNN), KNN (Coarse KNN), KNN (Cosine KNN), KNN (Cubic KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Subspace Discriminant), Ensemble (Subspace KNN), Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

Table 3.2.5. Machine learning techniques for comparison-holdout validation-15-85%-normalization-I

Normalized Dataset Machine Learning Algorithms		**Holdout Validation 15/85% (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	50%	50%
2	Tree (Medium Tree)	50%	70%
3	Tree (Coarse Tree)	60%	**100%**
4	Linear Discriminant	**100%**	**90%**
5	Quadratic Discriminant	50%	**90%**
6	Logistic Regression	**90%**	**90%**
7	Naïve Bayes (Kernel)	50%	**90%**
8	Naïve Bayes (Gaussian)	60%	**90%**
9	SVM (Linear Support Vector Machine)	**100%**	**90%**
10	SVM (Quadratic)	40%	10%
11	SVM (Cubic)	**80%**	**90%**
12	SVM (Fine Gaussian)	40%	**90%**
13	SVM (Medium Gaussian)	**80%**	**90%**
14	SVM (Coarse Gaussian)	60%	**90%**
15	KNN (Fine KNN)	50%	50%
16	KNN (Medium KNN)	50%	100%
17	KNN (Coarse KNN)	50%	**90%**
18	KNN (Cosine KNN)	60%	60%
19	KNN (Cubic KNN)	60%	**100%**
20	KNN (Weighted KNN)	60%	60%
21	Ensemble (Boosted Trees)	**80%**	**90%**
22	Ensemble (Bagged Trees)	**100%**	40%
23	Ensemble (Subspace Discriminant)	**80%**	**90%**
24	Ensemble (Subspace KNN)	30%	50%
25	Ensemble (RUSBoosted Trees)	50%	50%
	Best Results	**100%**	**100%**

The dataset is divided by the holdout validation-15-85% method and the normalization of the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.5.

Table 3.2.6. Machine learning techniques for comparison-holdout validation-15-85%-normalization-II

Normalized Dataset **Machine Learning Algorithms**		**Holdout Validation** **15/85% (ACC - %)**	
		PCA **Disable**	**PCA** **Enable**
1	Tree (Fine Tree)	**50%**	41%
2	Tree (Medium Tree)	**50%**	41%
3	Tree (Coarse Tree)	**50%**	41%
4	Linear Discriminant	Failed*	Failed*
5	Quadratic Discriminant	Failed*	Failed*
6	Logistic Regression	Failed*	Failed*
7	Naïve Bayes (Kernel)	**50%**	41%
8	Naïve Bayes (Gaussian)	Failed*	Failed*
9	SVM (Linear Support Vector Machine)	**50%**	41%
10	SVM (Quadratic)	41%	25%
11	SVM (Cubic)	**50%**	25%
12	SVM (Fine Gaussian)	16%	0%
13	SVM (Medium Gaussian)	25%	41%
14	SVM (Coarse Gaussian)	25%	41%
15	KNN (Fine KNN)	41%	16%
16	KNN (Medium KNN)	41%	41%
17	KNN (Coarse KNN)	41%	41%
18	KNN (Cosine KNN)	41%	41%
19	KNN (Cubic KNN)	41%	41%
20	KNN (Weighted KNN)	41%	25%
21	Ensemble (Boosted Trees)	41%	25%
22	Ensemble (Bagged Trees)	41%	25%
23	Ensemble (Subspace Discriminant)	**50%**	**50%**
24	Ensemble (Subspace KNN)	41%	16%
25	Ensemble (RUSBoosted Trees)	16%	8%
	Best Results	**50%**	**50%**

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset is divided by the normalization holdout validation (15-85%) method and the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.6. According to this table, Tree (Fine Tree), Tree (Medium Tree), Tree (Coarse Tree), SVM (Linear Support Vector Machine), SVM (Cubic), and Ensemble (Subspace Discriminant) models have the highest accuracy rate.

SVM (Fine Gaussian), SVM (Medium Gaussian), SVM (Coarse Gaussian) Ensemble (RUSBoosted), Trees) models have the lowest accuracy rate.

Table 3.2.7. Machine learning techniques for comparison- k-fold-cross-validation (k=3)-normalization-I

Normalized Dataset **Machine Learning Algorithms**		**k-fold-cross-validation (k=3)** **(ACC- %)**	
		PCA **Disable**	**PCA** **Enable**
1	Tree (Fine Tree)	70**%**	**90%**
2	Tree (Medium Tree)	70**%**	**90%**
3	Tree (Coarse Tree)	80**%**	**90%**
4	Linear Discriminant	**100%**	**100%**
5	Quadratic Discriminant	70**%**	**100%**
6	Logistic Regression	**100%**	**100%**
7	Naïve Bayes (Kernel)	60**%**	**100%**
8	Naïve Bayes (Gaussian)	70**%**	**100%**
9	SVM (Linear Support Vector Machine)	**100%**	**100%**
10	SVM (Quadratic)	60**%**	**90%**
11	SVM (Cubic)	50**%**	**90%**
12	SVM (Fine Gaussian)	20**%**	**100%**
13	SVM (Medium Gaussian)	**90%**	**100%**
14	SVM (Coarse Gaussian)	60**%**	**100%**
15	KNN (Fine KNN)	30**%**	70**%**
16	KNN (Medium KNN)	70**%**	70**%**
17	KNN (Coarse KNN)	20**%**	**100%**
18	KNN (Cosine KNN)	70**%**	70**%**
19	KNN (Cubic KNN)	60**%**	**80%**
20	KNN (Weighted KNN)	60**%**	**80%**
21	Ensemble (Boosted Trees)	60**%**	**90%**
22	Ensemble (Bagged Trees)	**90%**	70**%**
23	Ensemble (Subspace Discriminant)	**80%**	**100%**
24	Ensemble (Subspace KNN)	70**%**	70**%**
25	Ensemble (RUSBoosted Trees)	60**%**	60**%**
	Best Results	**100%**	**100%**

The dataset is divided by the k-fold-cross-validation (k=3) method and the normalization of the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.7.

Table 3.2.8. Machine learning techniques for comparison k-fold-cross-validation (k=3)-normalization-II

Normalized Dataset **Machine Learning Algorithms**		**k-fold-cross-validation (k=3)** **(ACC - %)**	
		PCA **Disable**	**PCA** **Enable**
1	Tree (Fine Tree)	28%	**36%**
2	Tree (Medium Tree)	28%	**36%**
3	Tree (Coarse Tree)	28%	**36%**
4	Linear Discriminant	Failed*	Failed*
5	Quadratic Discriminant	Failed *	Failed*
6	Logistic Regression	Failed*	Failed*
7	Naïve Bayes (Kernel)	28%	24%
8	Naïve Bayes (Gaussian)	Failed*	Failed*
9	SVM (Linear Support Vector Machine)	24%	28%
10	SVM (Quadratic)	24%	28%
11	SVM (Cubic)	20%	28%
12	SVM (Fine Gaussian)	20%	24%
13	SVM (Medium Gaussian)	**32%**	24%
14	SVM (Coarse Gaussian)	**32%**	32%
15	KNN (Fine KNN)	20%	20%
16	KNN (Medium KNN)	16%	16%
17	KNN (Coarse KNN)	**32%**	32%
18	KNN (Cosine KNN)	12%	16%
19	KNN (Cubic KNN)	16%	16%
20	KNN (Weighted KNN)	20%	28%
21	Ensemble (Boosted Trees)	16%	20%
22	Ensemble (Bagged Trees)	20%	20%
23	Ensemble (Subspace Discriminant)	28%	28%
24	Ensemble (Subspace KNN)	20%	20%
25	Ensemble (RUSBoosted Trees)	20%	20%
	Best Results	**32%**	**36%**

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset is divided by the normalization k-fold-cross-validation (k=3) method and the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.8. According to this table, SVM (Medium Gaussian), SVM (Coarse Gaussian), and KNN (Coarse KNN) models have the highest accuracy rate.

KNN (Medium KNN), KNN (Cosine KNN), KNN (Cubic KNN) Ensemble (Boosted Trees) models have the lowest accuracy rate.

Table 3.2.9. Machine learning techniques for comparison- k-fold-cross-validation (k=5)-normalization-I

Normalized Dataset Machine Learning Algorithms		**k-fold-cross-validation (k=5) (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	**80%**	70%
2	Tree (Medium Tree)	**80%**	**80%**
3	Tree (Coarse Tree)	**90%**	**100%**
4	Linear Discriminant	**100%**	**100%**
5	Quadratic Discriminant	60%	**100%**
6	Logistic Regression	**100%**	**100%**
7	Naïve Bayes (Kernel)	70%	**100%**
8	Naïve Bayes (Gaussian)	70%	**100%**
9	SVM (Linear Support Vector Machine)	**90%**	**100%**
10	SVM (Quadratic)	60%	**90%**
11	SVM (Cubic)	40%	50%
12	SVM (Fine Gaussian)	20%	**100%**
13	SVM (Medium Gaussian)	**80%**	**100%**
14	SVM (Coarse Gaussian)	50%	**100%**
15	KNN (Fine KNN)	50%	60%
16	KNN (Medium KNN)	60%	**80%**
17	KNN (Coarse KNN)	80%	**100%**
18	KNN (Cosine KNN)	60%	50%
19	KNN (Cubic KNN)	60%	80%
20	KNN (Weighted KNN)	50%	70%
21	Ensemble (Boosted Trees)	70%	**90%**
22	Ensemble (Bagged Trees)	**90%**	50%
23	Ensemble (Subspace Discriminant)	**80%**	**100%**
24	Ensemble (Subspace KNN)	50%	60%
25	Ensemble (RUSBoosted Trees)	**80%**	40%
	Best Results	**100%**	**100%**

The dataset is divided by the k-fold-cross-validation (k=5) method and the normalization of the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.9.

Table 3.2.10. Machine learning techniques for comparison- k-fold-cross-validation (k=5)-normalization-II

Normalized Dataset **Machine Learning Algorithms**		**k-fold-cross-validation (k=5) (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	52%	52%
2	Tree (Medium Tree)	52%	52%
3	Tree (Coarse Tree)	52%	52%
4	Linear Discriminant	Failed*	Failed*
5	Quadratic Discriminant52	Failed*	Failed*
6	Logistic Regression	Failed*	Failed*
7	Naïve Bayes (Kernel)	48%	48%
8	Naïve Bayes (Gaussian)	Failed*	Failed*
9	SVM (Linear Support Vector Machine)	48%	48%
10	SVM (Quadratic)	48%	52%
11	SVM (Cubic)	52%	48%
12	SVM (Fine Gaussian)	48%	48%
13	SVM (Medium Gaussian)	48%	48%
14	SVM (Coarse Gaussian)	48%	48%
15	KNN (Fine KNN)	48%	40%
16	KNN (Medium KNN)	48%	48%
17	KNN (Coarse KNN)	48%	48%
18	KNN (Cosine KNN)	40%	48%
19	KNN (Cubic KNN)	48%	48%
20	KNN (Weighted KNN)	48%	48%
21	Ensemble (Boosted Trees)	44%	44%
22	Ensemble (Bagged Trees)	48%	44%
23	Ensemble (Subspace Discriminant)	48%	48%
24	Ensemble (Subspace KNN)	48%	40%
25	Ensemble (RUSBoosted Trees)	20%	12%
	Best Results	52%	52%

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset is divided by the normalization k-fold-cross-validation (k=5) method, and the model performance performances obtained in PCA Enable and Disable states are given in Table 3.2.10. According to this table, Tree (Coarse Tree), Tree (Medium Tree), Tree (Coarse Tree), and SVM (Cubic) models have the highest accuracy rate.

Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

Table 3.2.11. Machine learning techniques for comparison- k-fold-cross-validation (k=10)-normalization-I

Normalized Dataset **Machine Learning Algorithms**		**k-fold-cross-validation (k=10) (ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	70%	70%
2	Tree (Medium Tree)	**80**%	**80**%
3	Tree (Coarse Tree)	**90**%	**90**%
4	Linear Discriminant	**90**%	**100**%
5	Quadratic Discriminant	70%	**100**%
6	Logistic Regression	**90**%	**100**%
7	Naïve Bayes (Kernel)	70%	**100**%
8	Naïve Bayes (Gaussian)	**80**%	**100**%
9	SVM (Linear Support Vector Machine)	**90**%	**100**%
10	SVM (Quadratic)	60%	70%
11	SVM (Cubic)	50%	50%
12	SVM (Fine Gaussian)	30%	**100**%
13	SVM (Medium Gaussian)	**80**%	**100**%
14	SVM (Coarse Gaussian)	60%	**100**%
15	KNN (Fine KNN)	20%	70%
16	KNN (Medium KNN)	50%	70%
17	KNN (Coarse KNN)	40%	**100**%
18	KNN (Cosine KNN)	50%	0%
19	KNN (Cubic KNN)	50%	70%
20	KNN (Weighted KNN)	50%	70%
21	Ensemble (Boosted Trees)	**80**%	**90**%
22	Ensemble (Bagged Trees)	**100**%	70%
23	Ensemble (Subspace Discriminant)	**80**%	**100**%
24	Ensemble (Subspace KNN)	0%	70%
25	Ensemble (RUSBoosted Trees)	**80**%	50%
	Best Results	**100**%	**100**%

The dataset is divided by the k-fold-cross-validation (k=10) method and the normalization of the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.2.11.

Table 3.2.12. Machine learning techniques for comparison-cross validation-10-normalization-II

Normalized Dataset **Machine Learning Algorithms**		**k-fold-cross-validation (k=10)** **(ACC - %)**	
		PCA Disable	**PCA Enable**
1	Tree (Fine Tree)	24%	20%
2	Tree (Medium Tree)	24%	20%
3	Tree (Coarse Tree)	24%	20%
4	Linear Discriminant	Failed*	Failed*
5	Quadratic Discriminant	Failed*	Failed*
6	Logistic Regression	Failed*	Failed*
7	Naïve Bayes (Kernel)	36%	**40**%
8	Naïve Bayes (Gaussian)	Failed*	Failed*
9	SVM (Linear Support Vector Machine)	28%	32%
10	SVM (Quadratic)	**40**%	**40**%
11	SVM (Cubic)	32%	28%
12	SVM (Fine Gaussian)	28%	32%
13	SVM (Medium Gaussian)	32%	36%
14	SVM (Coarse Gaussian)	32%	32%
15	KNN (Fine KNN)	20%	24%
16	KNN (Medium KNN)	24%	16%
17	KNN (Coarse KNN)	32%	32%
18	KNN (Cosine KNN)	4%	4%
19	KNN (Cubic KNN)	24%	16%
20	KNN (Weighted KNN)	36%	**40**%
21	Ensemble (Boosted Trees)	32%	36%
22	Ensemble (Bagged Trees)	32%	36%
23	Ensemble (Subspace Discriminant)	28%	32%
24	Ensemble (Subspace KNN)	20%	24%
25	Ensemble (RUSBoosted Trees)	12%	8%
	Best Results	**40**%	**40**%

* Failed: Untested due to inappropriate inputs for the algorithm.

The dataset is divided by the normalization k-fold-cross-validation (k=10) method, and the model performance performances obtained in PCA Enable and Disable states are given in Table 3.2.12. According to this table, SVM (Quadratic), Naive Bayes (Kernel), and KNN (Weighted KNN) models have the highest accuracy rate.

KNN (Cosine KNN), and Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

3.3. Experimental Studies with Dataset Parameter Optimization

We performed parameter optimization for the worst results we obtained in previous experiments in this section.

Table. 3.3.1. Machine learning techniques for comparison-no validation-advanced

			Box Constraint Level	Kernel Scale Mode	Multiclass Method	%	Kernel Scale Mode	Manuel Kernel scale	Multiclass Method	%
Parameter Optimization and No Validation	**Linear SVM**	**Linear**	1	Auto	One-vs-one	40	Manuel	1	One-vs-all	16
		Gaussian	1	Auto	One-vs-one	88	Manuel	1	One-vs-all	80
		Quadratic	1	Auto	One-vs-one	88	Manuel	1	One-vs-all	76
		Cubic	1	Auto	One-vs-one	88	Manuel	1	One-vs-all	80
	Linear SVM	Linear	2	Auto	One-vs-one	88	Manuel	2	One-vs-all	20
		Gaussian	2	Auto	One-vs-one	88	Manuel	2	One-vs-all	64
		Quadratic	2	Auto	One-vs-one	88	Manuel	2	One-vs-all	64
		Cubic	2	Auto	One-vs-one	88	Manuel	2	One-vs-all	72

In the No validation method, the values in Table 3.3.1 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2.

Similarly, when the kernel scale mode is manual in the no validation method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.3.1 when multiclass method one-vs-all is taken.

Table 3.3.2. Machine learning techniques for comparison-holdout validation-20-80%-advanced

Parameter Optimization and **Holdout Validation 20-80%**	**Linear SVM**		**Box Constraint Level**	**Kernel Scale Mode**	**Multiclass Method**	**%**	**Kernel Scale Mode**	**Manuel Kernel scale**	**Multiclass Method**	**%**
		Linear	1	Auto	One-vs-one	8.3	Manuel	1	One-vs-all	8.3
		Gaussian	1	Auto	One-vs-one	8.3	Manuel	1	One-vs-all	16.7
		Quadratic	1	Auto	One-vs-one	0	Manuel	1	One-vs-all	0
		Cubic	1	Auto	One-vs-one	0	Manuel	1	One-vs-all	0
	Linear SVM	Linear	2	Auto	One-vs-one	0	Manuel	2	One-vs-all	8.3
		Gaussian	2	Auto	One-vs-one	0	Manuel	2	One-vs-all	8.3
		Quadratic	2	Auto	One-vs-one	0	Manuel	2	One-vs-all	8.3
		Cubic	2	Auto	One-vs-one	0	Manuel	2	One-vs-all	0

In the holdout validation (20-80%) method, the values in Table 3.3.2 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2.

Similarly, when the kernel scale mode is manual in the holdout validation (20-80%) method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.3.2 when multiclass method one-vs-all is taken.

Table. 3.3.3. Machine learning techniques for comparison- k-fold-cross-validation (k=3)-advanced

			Box Constraint Level	Kernel Scale Mode	Multiclass Method	%	Kernel Scale Mode	Manuel Kernel scale	Multiclas s Method	%
Parameter Optimization **k-fold-cross-validation (k=3)**	**Linear SVM**	Linear	1	Auto	One-vs-one	32	Manuel	1	One-vs-all	12
		Gaussian	1	Auto	One-vs-one	24	Manuel	1	One-vs-all	24
		Quadratic	1	Auto	One-vs-one	24	Manuel	1	One-vs-all	32
		Cubic	1	Auto	One-vs-one	28	Manuel	1	One-vs-all	36
	Linear SVM	Linear	2	Auto	One-vs-one	12	Manuel	2	One-vs-all	28
		Gaussian	2	Auto	One-vs-one	12	Manuel	2	One-vs-all	24
		Quadratic	2	Auto	One-vs-one	12	Manuel	2	One-vs-all	16
		Cubic	2	Auto	One-vs-one	12	Manuel	2	One-vs-all	28

In the k-fold-cross-validation (k=3) method, the values in Table 3.3.3 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2.

Similarly, when the kernel scale mode is manual in the k-fold-cross-validation (k=3) method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.3.3 when the multiclass method one-vs-all is taken.

4. RESULTS AND CONCLUSION

In this thesis, the accuracy of artificial intelligence algorithms is tested by using predictive learning models on the data set consisting of 299 heart failure patient samples. The main finding is as follows:

The no validation performance obtained in PCA Enable and PCA Disable states are given in Table 3.1. According to this table, SVM (Cubic), SVM (Fine Gaussian), KNN (Fine KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Bagged Trees), Ensemble (Subspace KNN), Ensemble (RUSboosted Trees) models have the highest accuracy rate.

Quadratic Discriminant, Naive Bayes (Kernel), SVM (Coarse Gaussian), KNN (Fine KNN), KNN (Medium KNN), KNN (Coarse KNN), KNN (Cosine KNN), KNN (Cubic KNN) models have the lowest accuracy rate.

The holdout validation (20-80%) method and the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.2. According to this table, Linear Discriminant, Logistic Regression, Naive Bayes (Gaussian), Ensemble (Bagged Trees), and Ensemble (Subspace Discriminant) models give the highest accuracy results. Quadratic Discriminant SVM (Cubic), KNN (Fine KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

The holdout validation (15-85%) method and the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.3. According to this table, Linear Discriminant, SVM (Linear Support Vector Machine), and Ensemble (Bagged Trees) models have the highest accuracy rate. Quadratic Discriminant, SVM (Quadratic), SVM (Fine Gaussian), KNN (Fine KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

By the k-fold-cross-validation (k = 3) method, the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.4. According to this table, Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), SVM (Medium

Gaussian), and Ensemble (Bagged Trees) models have the highest accuracy rate. Quadratic Discriminant SVM (Fine Gaussian), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

By the k-fold-cross-validation (k = 5) method, the model performances obtained in PCA Enable and Disable states are given in Table 3.5. According to this table, Tree (Coarse Tree), Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), and SVM (Linear Support Vector Machine) models have the highest accuracy rate. Quadratic Discriminant, SVM (Fine Gaussian), KNN (Fine KNN), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

By the k-fold-cross-validation (k =10) method and the model performance performances obtained in PCA Enable and PCA Disable states are given in Table 3.6. According to this table, Tree (Coarse Tree), Linear Discriminant, Logistic Regression, SVM (Linear Support Vector Machine), Ensemble (Boosted Trees), Ensemble (Bagged Trees), and Ensemble (RUSBoosted Trees) models have the highest accuracy rate. SVM (Fine Gaussian), KNN (Fine KNN), KNN (Coarse KNN), and Ensemble (Subspace KNN) models have the lowest accuracy rate.

By the data normalization & no validation method and the model performances obtained in PCA Enable and Disable states are given in Table 3.7. According to this table, SVM (Quadratic), SVM (Cubic), SVM (Fine Gaussian), SVM (Medium Gaussian), KNN (Fine KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Bagged Trees), Ensemble (RUSBoosted Trees) models have the highest accuracy rate. KNN (Medium KNN), KNN (Coarse KNN), KNN (Cosine KNN), KNN (Cubic KNN) Ensemble (Subspace Discriminant) models have the lowest accuracy rate.

The data normalization-holdout validation (20-80%) method and the model performances obtained in PCA Enable and Disable states are given in Table 3.8. According to this table, SVM (Linear Support Vector Machine), SVM (Fine Gaussian), Trees), Ensemble (Subspace Discriminant), Ensemble (Subspace KNN), and Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

By the data normalization & holdout validation (15-85%) method and the model performances obtained in PCA Enable and PCA Disable states are given in Table 3.9. According to this table, Tree (Fine Tree), Tree (Medium Tree), Tree (Coarse Tree), SVM (Linear Support Vector Machine), SVM (Cubic), and Ensemble (Subspace Discriminant) models have the highest accuracy rate. SVM (Fine Gaussian), SVM (Medium Gaussian), SVM (Coarse Gaussian) Ensemble (RUSBoosted), Trees) models have the lowest accuracy rate.

By the data normalization & k-fold-cross-validation (k = 3) method and the model performances obtained in PCA Enable and PCA, disable states are given in Table 3.10. According to this table, SVM (Medium Gaussian), SVM (Coarse Gaussian), and KNN (Coarse KNN) models have the highest accuracy rate. KNN (Medium KNN), KNN (Cosine KNN), KNN (Cubic KNN) Ensemble (Boosted Trees) models have the lowest accuracy rate.

The dataset with normalization is divided by the k-fold-cross-validation (k = 5) method and the model performances obtained in PCA Enable and Disable states are given in Table 3.11. According to this table, Tree (Coarse Tree), Tree (Medium Tree), Tree (Coarse Tree), and SVM (Cubic) models have the highest accuracy rate. Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

The dataset with normalization is divided by the k-fold-cross-validation (k = 10) method and the model performances obtained in PCA Enable and Disable states are given in Table 3.12. According to this table, SVM (Quadratic), Naive Bayes (Kernel), and KNN (Weighted KNN) models have the highest accuracy rate. KNN (Cosine KNN), and Ensemble (RUSBoosted Trees) models have the lowest accuracy rate.

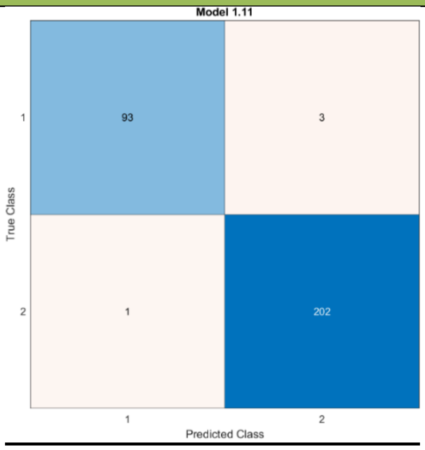
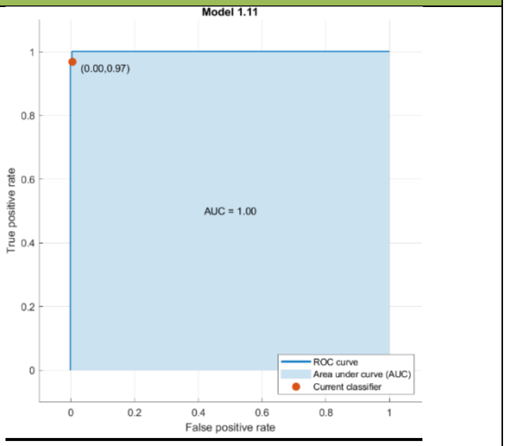
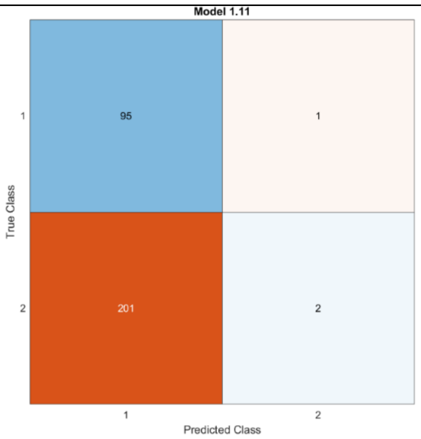
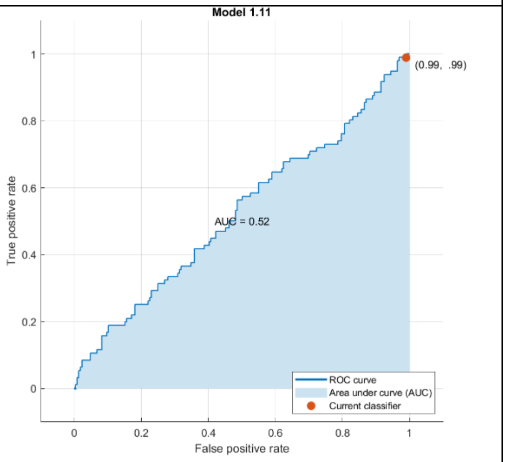
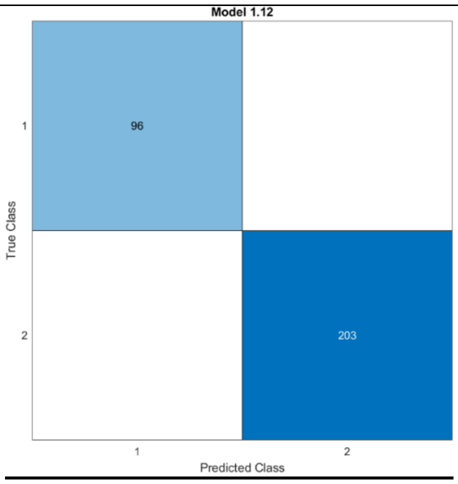
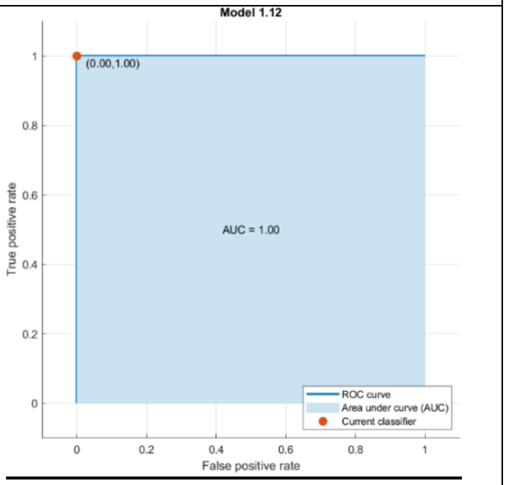
In the no validation method, the values in Table 3.13 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2. Similarly, when the kernel scale mode is manual in the no validation method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.13 when multiclass method one-vs-all is taken.

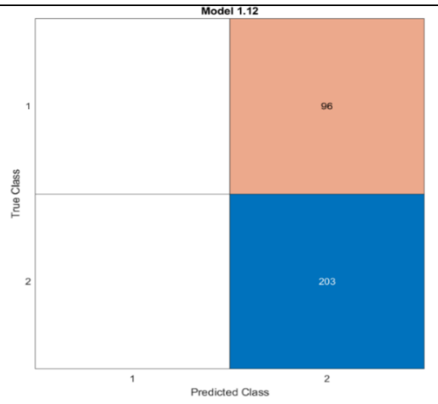
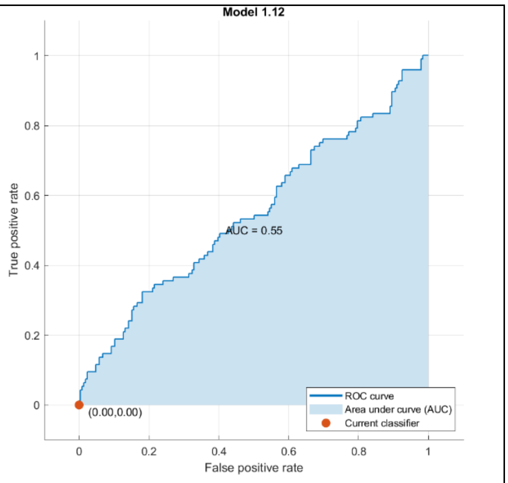
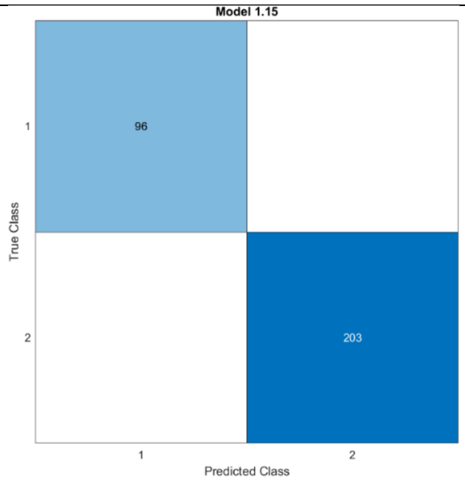
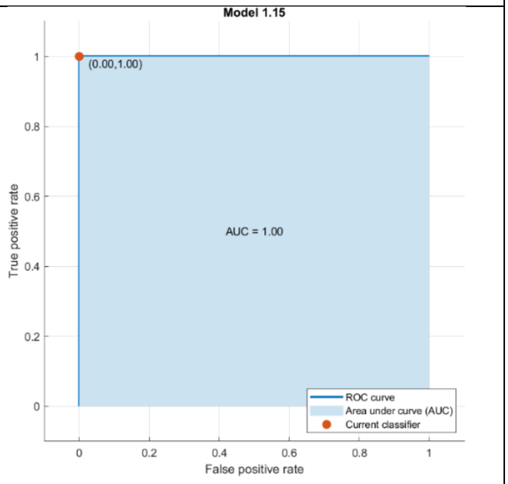
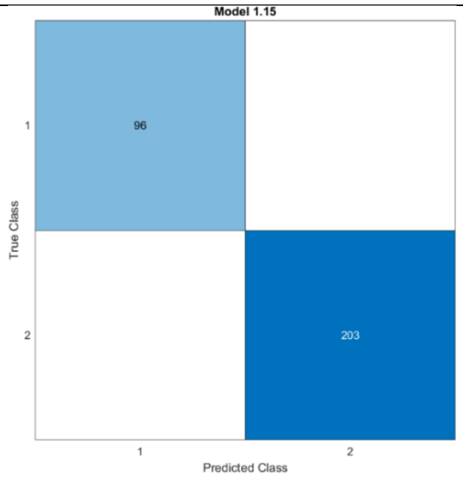
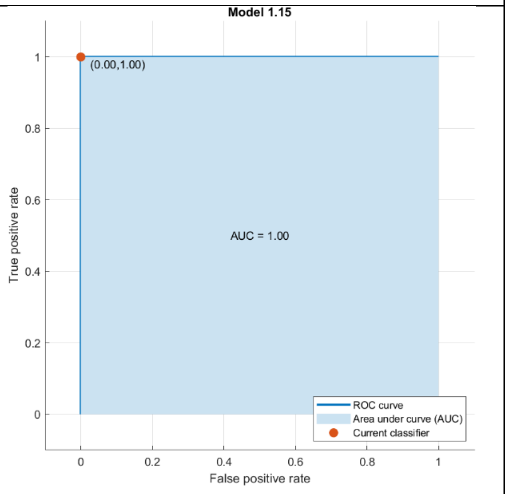
In the holdout validation (20-80%) method, the values in Table 3.14 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2. Similarly, when the kernel scale mode is manual in the holdout validation (20-80%)

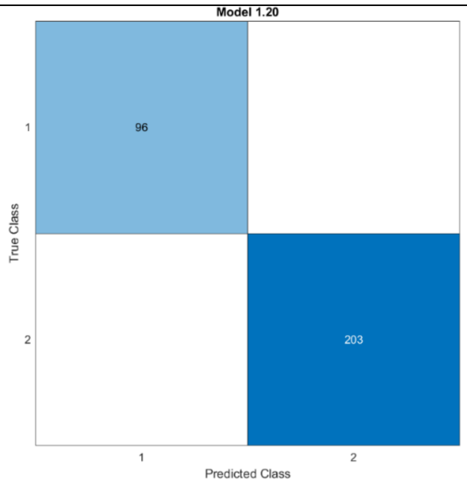
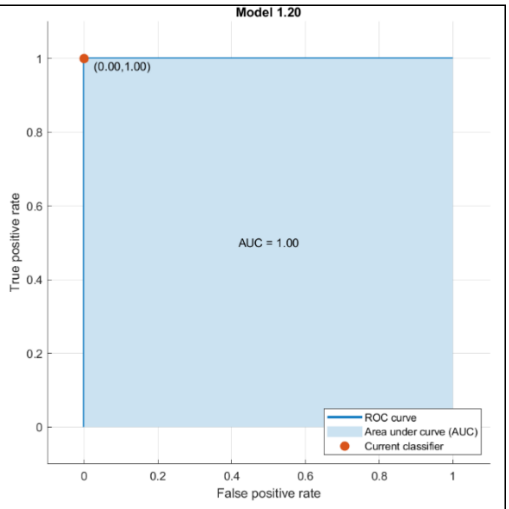
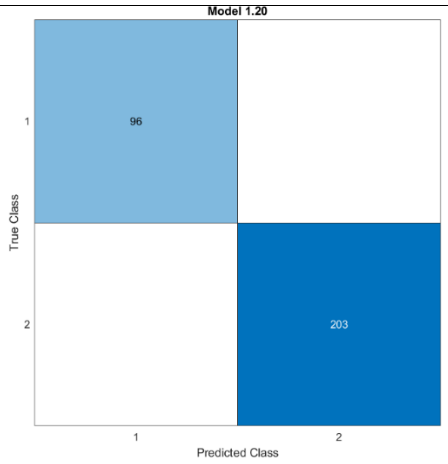
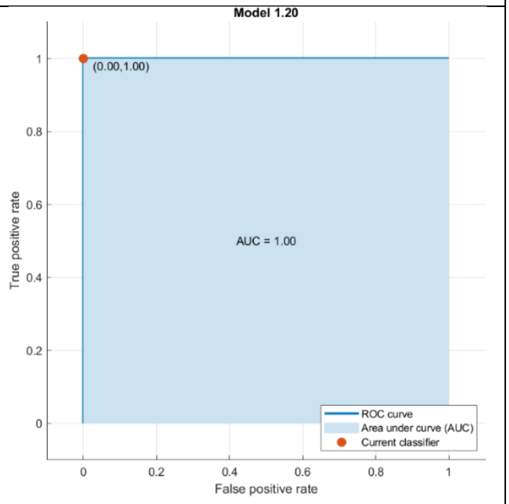
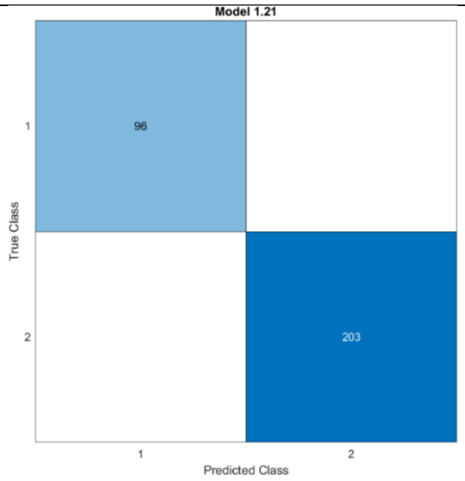
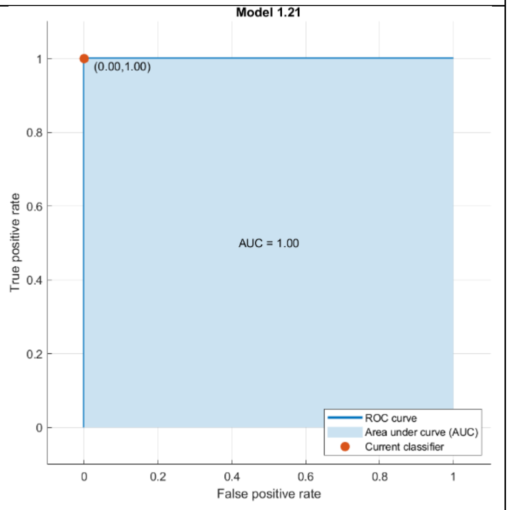
method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.14 when multiclass method one-vs-all is taken.

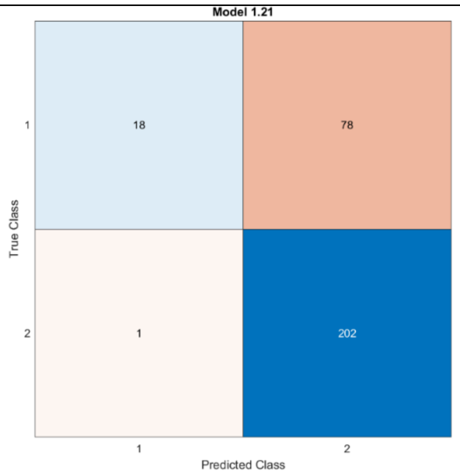
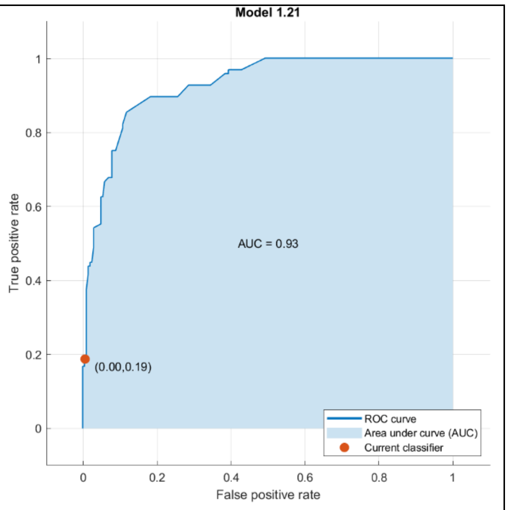
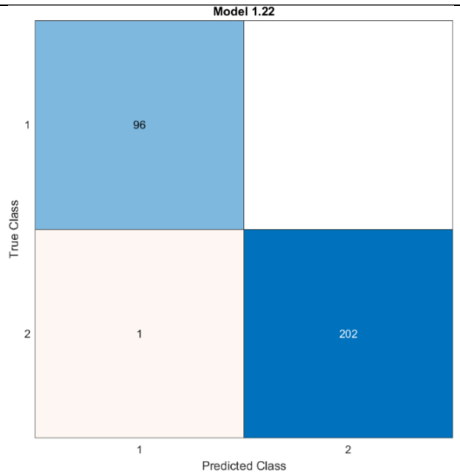
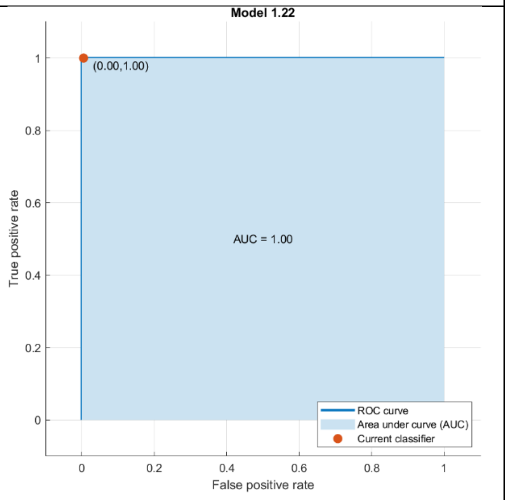
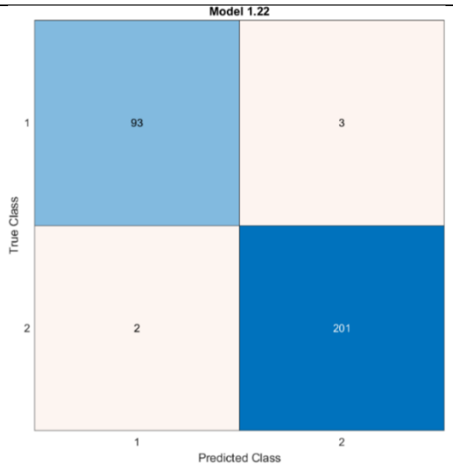
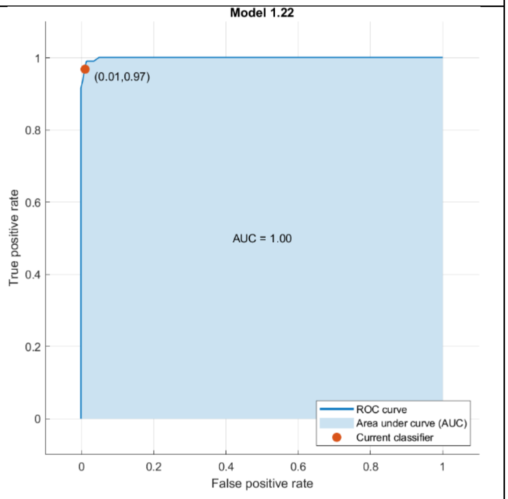
In the k-fold-cross-validation (k = 3) method, the values in Table 3.15 are obtained when linear SVM is run in linear, gaussian, quadratic, and cubic by taking box constraint levels 1 and 2. Similarly, when the kernel scale mode is manual in the k-fold-cross-validation (k = 3) method, and the manual kernel scale is 1 and 2, the percentage values are given in Table 3.15 when the multiclass method one-vs-all is taken.

Table 4.1. Confusion matrix and ROC of highest and lowest accuracy performance of all models (No validation)

MODEL NO	CONFUSION MATRIX	ROC
Highest Accuracy Results		
1.11 **PCA Disable**		
1.11 **PCA Enable**		
1.12 **PCA Disable**		

1.12 **PCA** **Enable**		
1.15 **PCA** **Disable**		
1.15 **PCA** **Enable**		

1.20 **PCA** **Disable**		
1.20 **PCA** **Enable**		
1.21 **PCA** **Disable**		

1.21 **PCA** **Enable**		
1.22 **PCA** **Disable**		
1.22 **PCA** **Enable**		

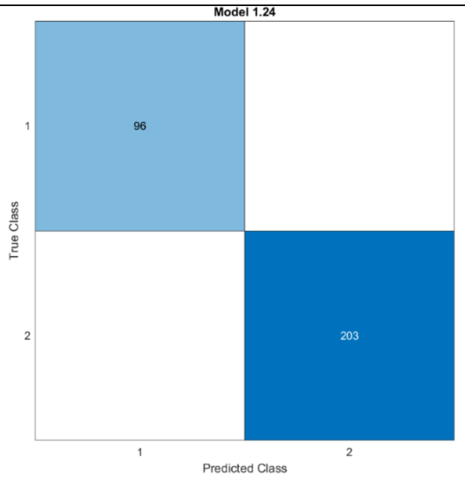
<table>
<tr>
<td>

1.24

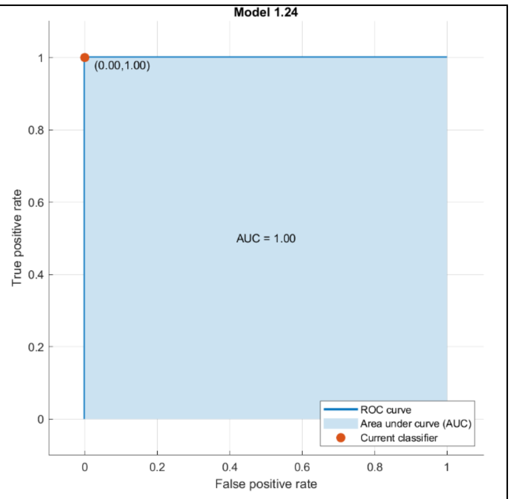
PCA

Disable

</td>
<td>



</td>
<td>



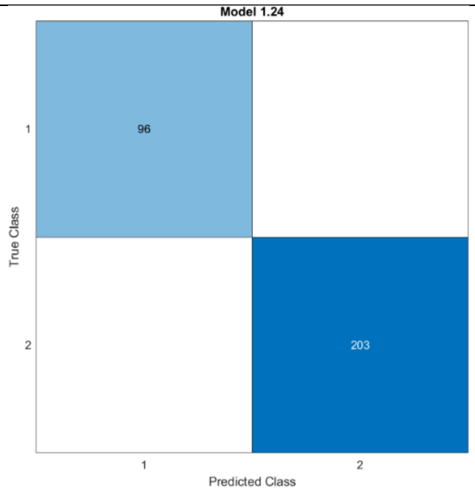
</td>
</tr>
<tr>
<td>

1.24

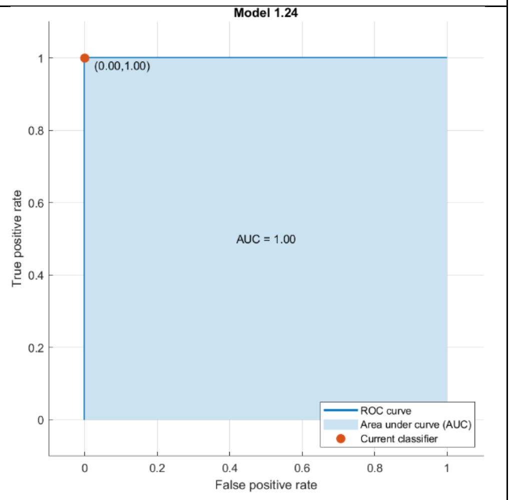
PCA

Enable

</td>
<td>



</td>
<td>



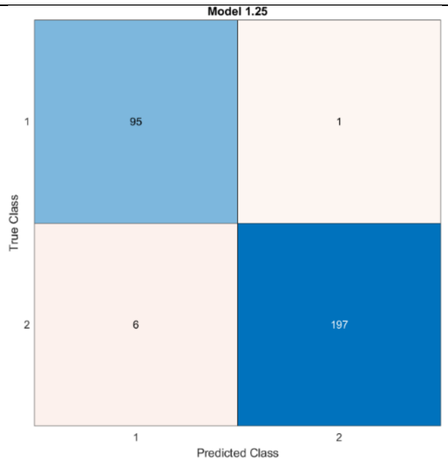
</td>
</tr>
<tr>
<td>

1.25

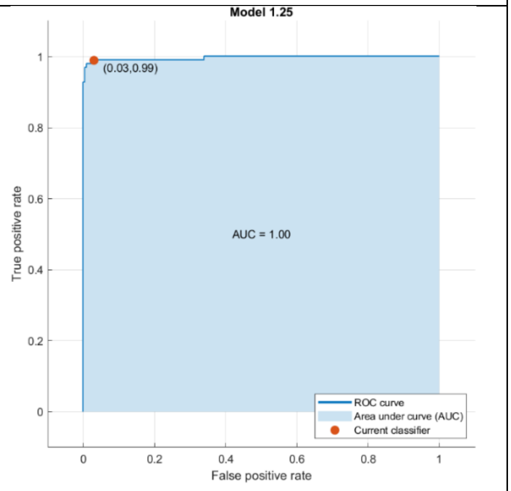
PCA

Disable

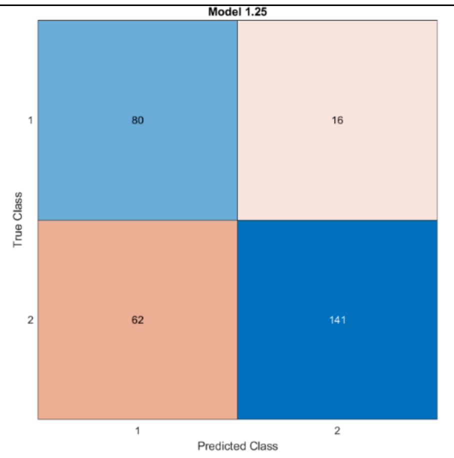
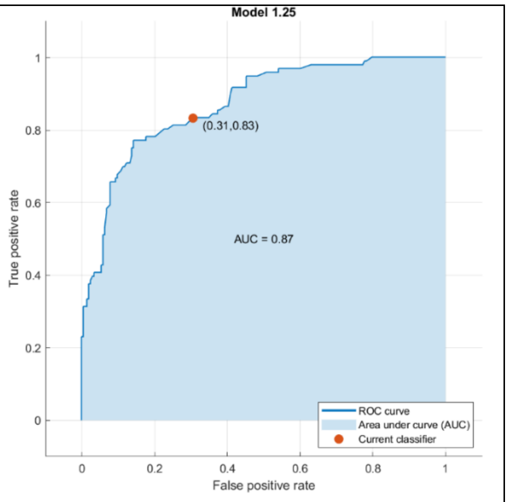
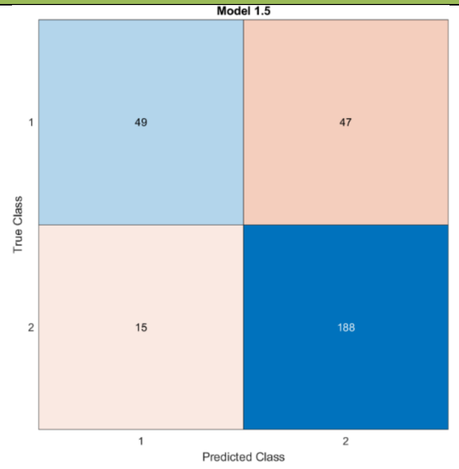
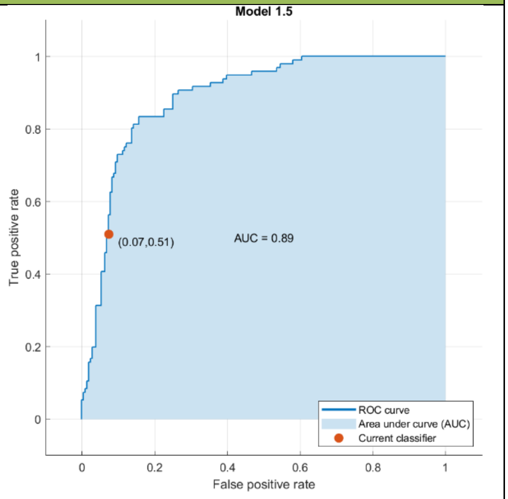
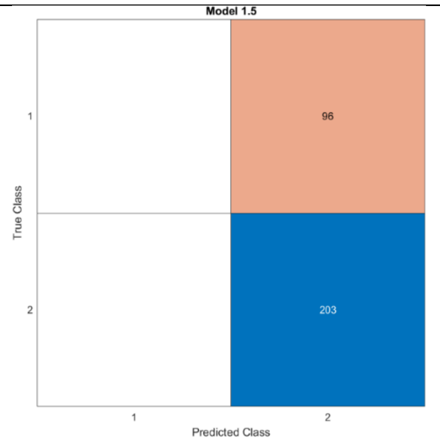
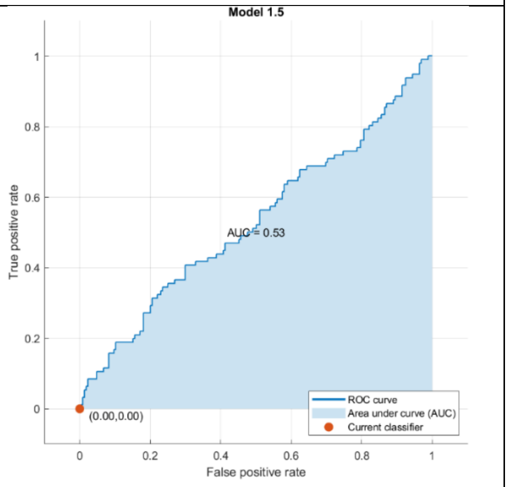
</td>
<td>



</td>
<td>



</td>
</tr>
</table>

1.25 **PCA** **Enable**		
Lowest Accuracy Results		
1.5 **PCA** **Disable**		
1.5 **PCA** **Enable**		

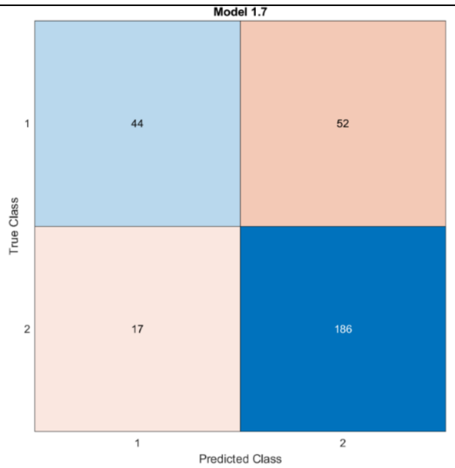
<table>
<tr>
<td>

1.7

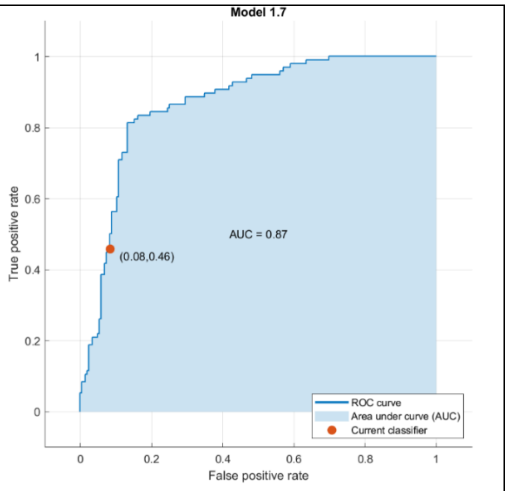
PCA

Disable

</td>
<td>



</td>
<td>



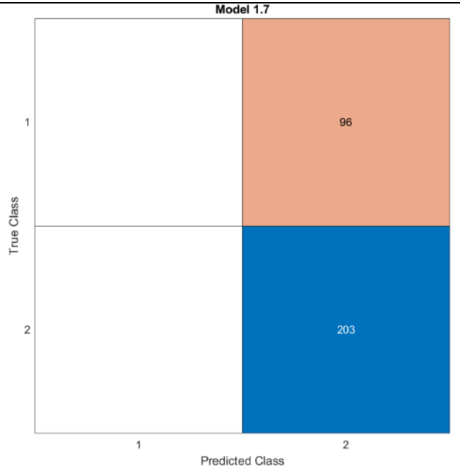
</td>
</tr>
<tr>
<td>

1.7

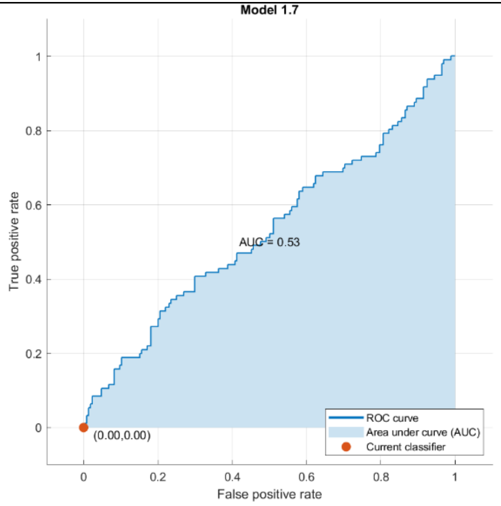
PCA

Enable

</td>
<td>



</td>
<td>



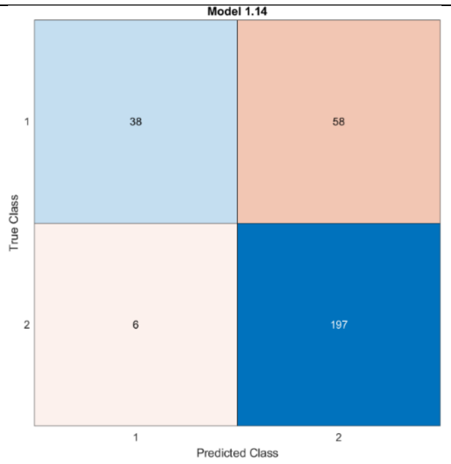
</td>
</tr>
<tr>
<td>

1.14

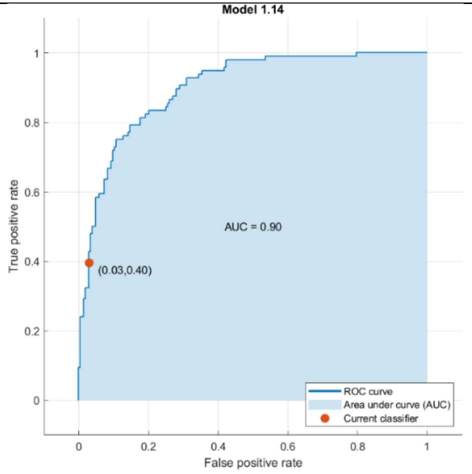
PCA

Disable

</td>
<td>

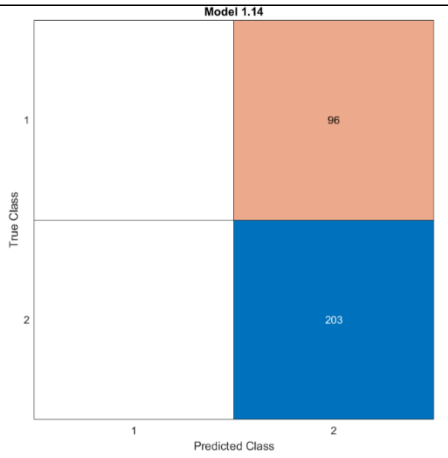
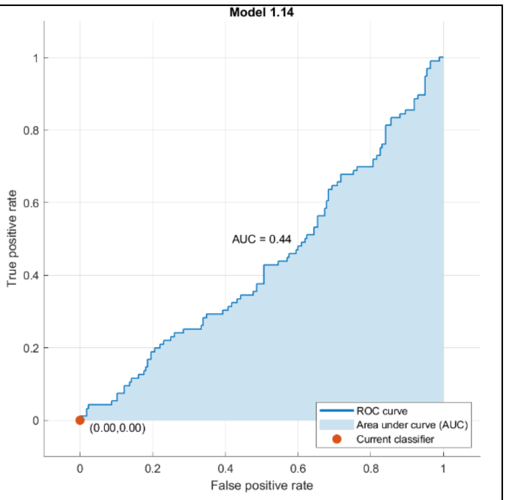


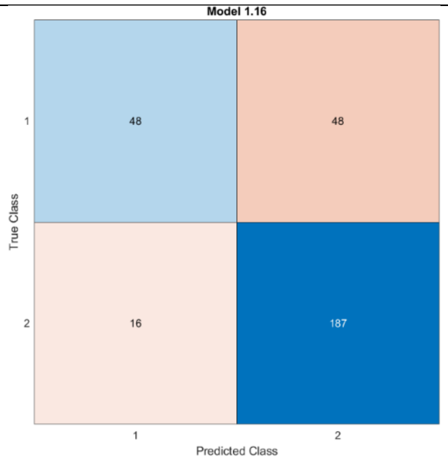
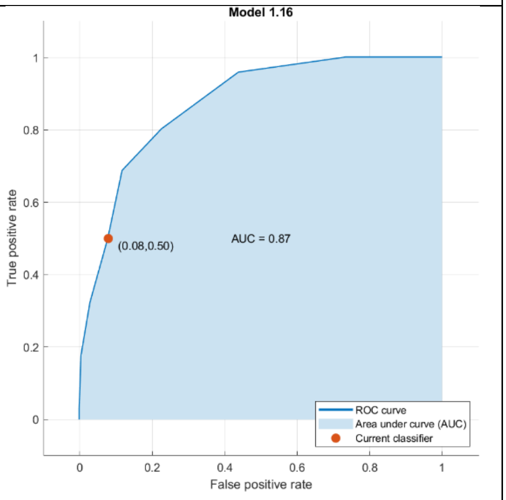
</td>
<td>

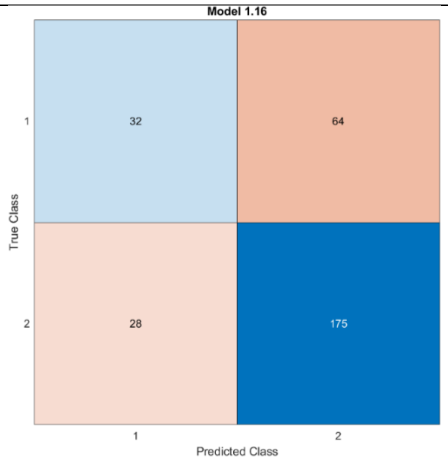
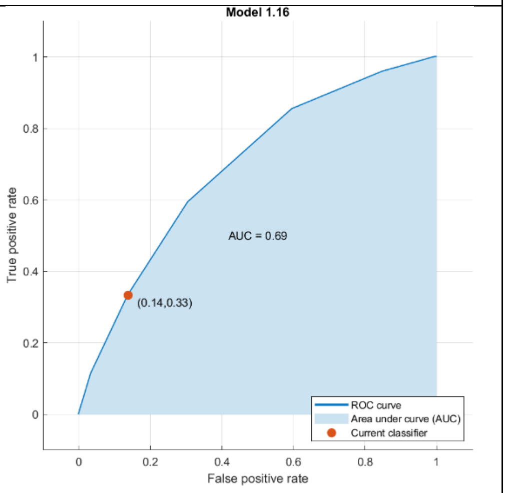


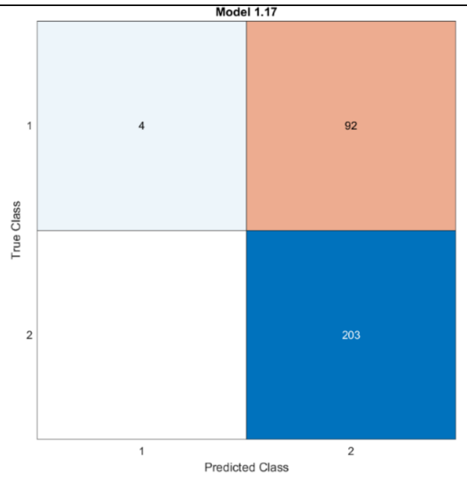
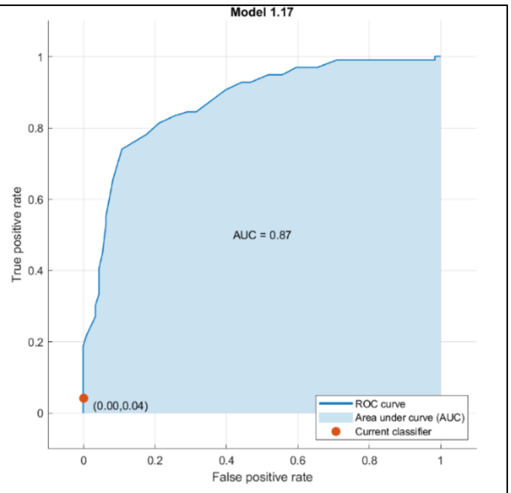
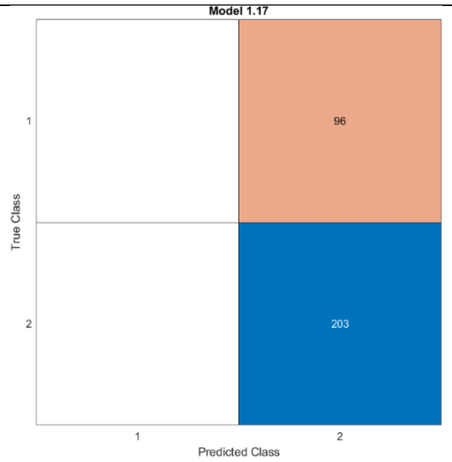
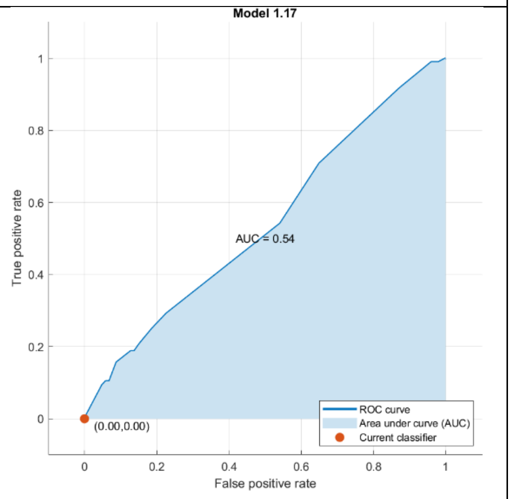
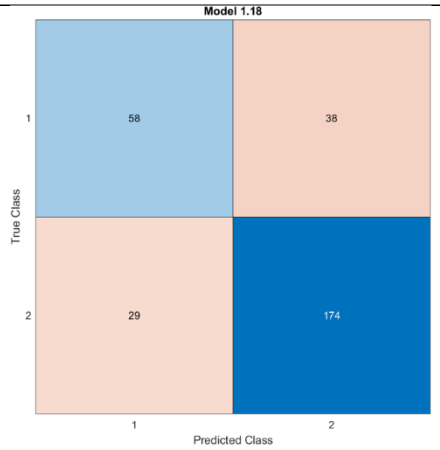
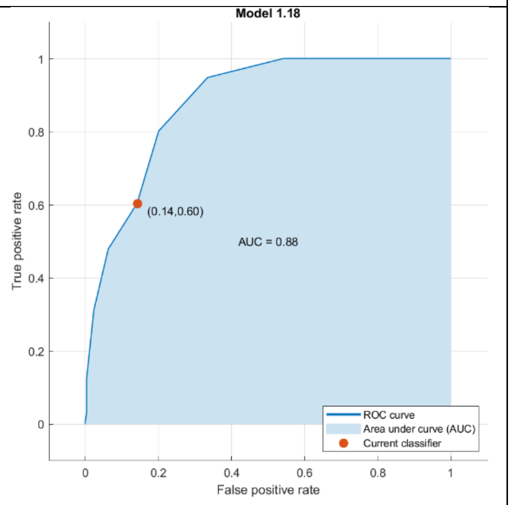
</td>
</tr>
</table>

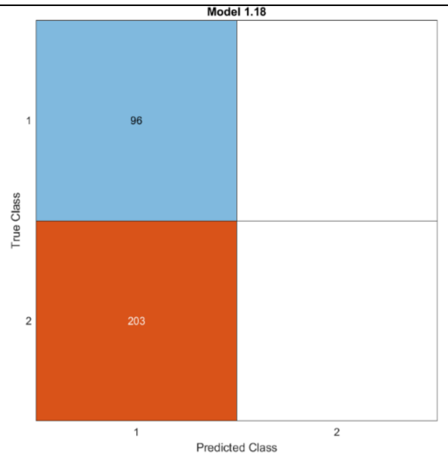
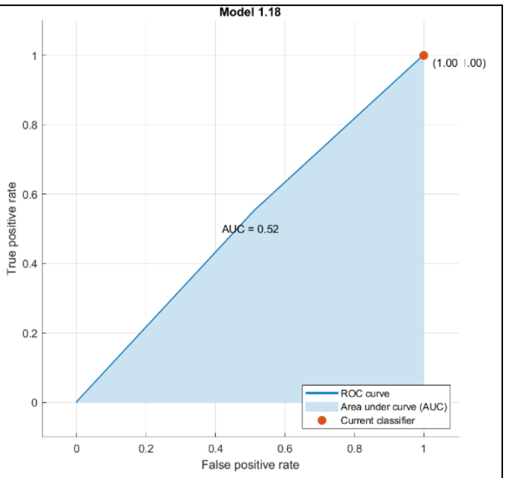
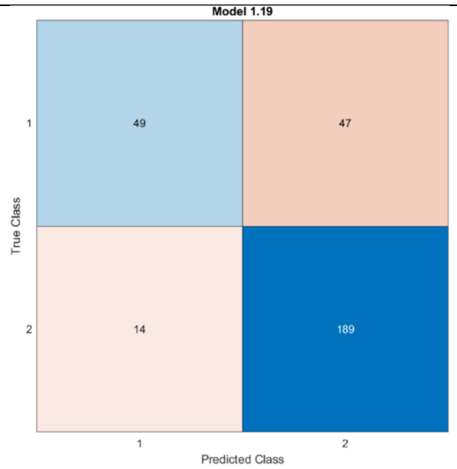
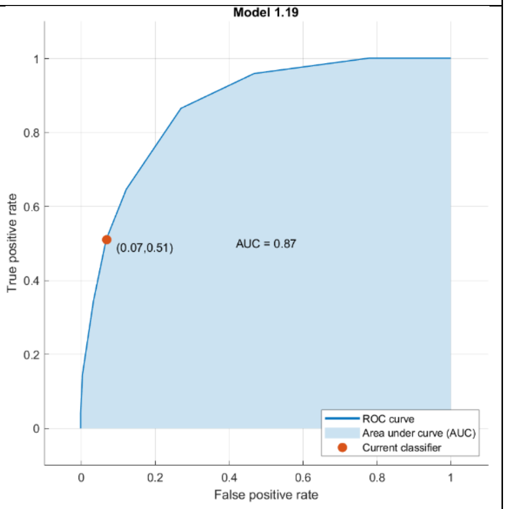
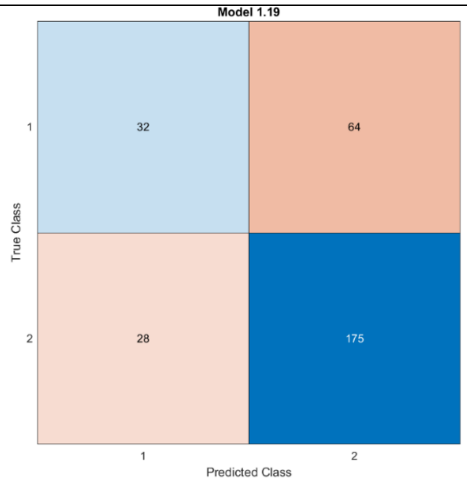
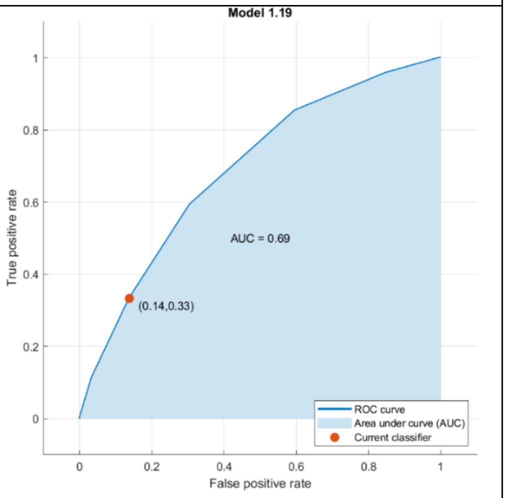
<table>
<tr>
<td><b><u>1.14</u></b>

<b><u>PCA</u></b>
<b><u>Enable</u></b></td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td><b><u>1.16</u></b>

<b><u>PCA</u></b>
<b><u>Disable</u></b></td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td><b><u>1.16</u></b>

<b><u>PCA</u></b>
<b><u>Enable</u></b></td>
<td>
</td>
<td>
</td>
</tr>
</table>

1.17 **PCA** **Disable**		
1.17 **PCA** **Enable**		
1.18 **PCA** **Disable**		

1.18 **PCA** **Enable**		
1.19 **PCA** **Disable**		
1.19 **PCA** **Enable**		

When all these results are examined; in the case of the PCA value of the no validation method is passive, SVM (Cubic), SVM (Fine Gaussian), KNN (Fine KNN), KNN (Weighted KNN), Ensemble (Boosted Trees), Ensemble (Bagged Trees), Ensemble (Subspace KNN)), Ensemble (RUSboosted Trees) models give the highest accuracy, and when PCA is active, KNN (Fine KNN) KNN (Weighted KNN) Ensemble (Subspace KNN) models give the most accurate results. By evaluating these results, confusion matrix and roc graft Table 4.1. has also been given.

For this thesis, in which a dataset related to heart failure was used by feature engineering on the biomedical dataset, experimental studies were carried out for parameter optimization of the data with artificial intelligence methods. In this study, which artificial intelligence algorithm gave the most accurate results by using predictive learning models on the data set is demonstrated. When the results were examined, suggestions were made to the academicians who wanted to create a decision support system by comparing the positive-negative performance changes on the feature engineering dataset. It can be suggested that this study can a basis for future studies, it will also be an example for health data in different fields.

REFERENCES

Ahmad T, Munir A, Bhatti SH, Aftab M, Raza MA (2017) Survival analysis of heart failure patients: a case study. PloS one 12(7):181001

Akgul A et al (2013) Koroner arter baypas greftleme sonrası erken mortalitenin belirlenmesinde standart, lojistik Euroscore ve Euroscore II’nin kars¸ılas¸tırılması. Anadolu Kardiyol Dergisi 13:425–431

Boyd CR, Tolson MA, Copes WS (1987) Evaluating trauma care: The TRISS method. J Trauma 27(4):370–378.

Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees (statistics/probability series). CRC Press, Hoboken.

Breiman L (2001) Random forests. Mach Learn 45(1):532

Chaturvedi V, Parakh N, Seth S, Bhargava B, Ramakrishnan S, Roy A, Anand K, et al (2016) Heart failure in India: The INDUS (India Ukieri study) study. J Pract Cardiovasc Sci 2:28–35

Chicco D, Giuseppe J (2020) Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Med Inform Decis Mak 20(1):16

Costa LL et al (2020) Quality of life of chronic heart failure patients. Open Journal of Nursing 10(9):831–857

C. Wilstrup, C. Cave (2021) Combining symbolic regression with the Cox proportional hazards model improves the prediction of heart failure deaths.

D. H. Katz, R. C. Deo, F. G. Aguilar, S. Selvaraj, E. E. Martinez, L. Beussink-Nelson, K-Y. A. Kim, J. Peng, M. R. Irvin, H. Tiwari, D. C. Rao, D. K. Arnett, S. J. Shah, (2017), Phenomapping for the identification of hypertensive patients with the myocardial substrate for heart failure with preserved ejection fraction, J Cardiovasc Translate Res. 10 (3) 275-284.

Dong, G., & Liu, H. (2018). Feature engineering for machine learning and data *analytics.* Boca Raton, FL: CRC Press.

Dua, D., & Graff, C. (2017). UCI machine learning repository. Retrieved from http://archive.ics.uci.edu/ml

Erdas¸C¸ & Olcer D (2020) A machine learning-based approach to detect survival of heart failure patients. TIPTEKNO. 2020:1–4

E. T. Mesquita, D. C. Grion, M. C. Kubrusly, B. B. F. F. Silva, É. A. R. Santos, (2018), Phenotype mapping of heart failure with preserved ejection fraction, Int J Cardiovasc Sci. 31 (6) 652-661

Freund Y, Schapire R, Abe N (1999) A short introduction to boosting. J Jpn Soc Artif Intell 14:771–780

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 2001:1189–1232.

Gardner WA (1984) Learning characteristics of stochastic-gradient-descent algorithms: a general study, analysis, and critique. Signal Process 6(2):113–133.

Gianluigi S, Lund LH (2017) Global public health burden of heart failure. Card Fail Rev 3(1):7

Gürfidan R, Ersoy M (2021) Classification of death related to heart failure by machine learning algorithms. Adv Artif Intell Res 1(1):13–18.

Ishaq A, Sadiq S, Umer M, Ullah S, Mirjalili S, Rupapara V, Nappi M (2021) Improving the prediction of heart failure patients' survival using smote and effective data mining techniques. IEEE Access 9:39707–39716

Karakuş M., Er O. (2022), A comparative study on prediction of survival event of a heart failure patients using machine learning algorithms", Neural Computing and Applications.

Lassus JP et al (2013) Long-term survival after hospitalization for acute heart failure-differences in the prognosis of acutely decompensated chronic and new-onset acute heart failure. Int J Cardiol 168(1):458–462

Lee HC, Yoon HK, Nam K, Cho YJ, Kim TK, Kim WH, Bahk JH (2018) Derivation and validation of machine learning approaches to predict acute kidney injury after cardiac surgery. J Clin Med 7(10):322

L. He, S. Chen, Y. Liang, M. Hou, J. Chen, (2020), Infilling the missing values of groundwater level using time and space series: a case of Nantong City, east coast of China, Earth Science Informatics 13 (4) 1445-1459.

Martinez-Amezcua P, Haque W, Khera R, Kanaya AM, Sattar N, Lam CS, et al (2020) The upcoming epidemic of heart failure in South Asia. Circle Heart Fail 13(10):7218

M. W. Akhter, D. Aronson, F. Bitar, S. Khan, H. Singh, R. P. Singh, A. J. Burger, U. Elkayam, (2004), Effect of elevated admission serum creatinine and its worsening on outcome in hospitalized patients with decompensated heart failure, Am J Cardiol. 94(7) 957-960.

Moyehodie YA, Yesuf KM, Sied AA, Masresha BM (2021) Determinants of pulse rate change and time-to default from treatment among congestive heart failure patients in a file-hiwot referral hospital, Bahir dar, Ethiopia; comparison of separate and joint models. Res Square

Nauta JF, Jin X, Hummel YM, Voors AA (2018) Markers of left ventricular systolic dysfunction when left ventricular ejection fraction is normal. Eur J Heart Fail 20:1636–1638

N. Vistarini, A. Deschamps, R. Cartier, (2014) Preoperative creatinine clearance affects long-term survival after off-pump coronary artery bypass surgery, Can J Cardiol. 30(10) S238-S239.

Nunez J, Garcia S, Nunez E, Bonanad C, Bodı´ V, M Miñana G, Santas E, Escribano D, Bayes GA, Pascual FD, Chorro FJ, Sanchis J (2017) Early serum creatinine changes and outcomes in patients admitted for acute heart failure: the cardio-renal syndrome revisited. Eur Heart J Acute Cardiovasc Care 6(5):430–440

Oladimeji OO, Oladimeji O (2020) Predicting survival of heart failure patients using classification algorithms. JITCE J Inf Technol Comput Eng. 4(02):90–94

Pfeffer MA, Braunwald E (2016) Treatment of heart failure with preserved ejection fraction reflections on its treatment with an aldosterone antagonist. J Am Med Assoc JAMA Cardiol 1(1):7–8,

Pe´rez A, Larranaga P, Inza I (2006) Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes. Int J Approx Reasoning 43(1):1–25.

Rahayu S, Purnama JJ, Pohan AB, Nugraha FS, Nurdiani S, Hadianti S (2020) Prediction of survival of heart failure patients using random forest. Jurnal Pilar Nusa Mandiri 16(2):255–260.

Sharaff A, Gupta H (2019) Extra-tree classier with metaheuristics approach for email classification. In: Proc. Adv. Comput. Commun. Comput. Sci. Springer, Singapore, pp 189–197

Schölkopf B, Burges C, Vapnik V (1996) Incorporating invariances in support vector learning machines. In: Proc. Int. Conf. Artif. Neural Netw. Springer, Berlin, pp 47–52.

S. V. Salim, A. Alvaro, J. B. Emelia, S. B. Marcio, W. C. Clifton, P. C. April, et al., heart disease and stroke statistics| (2020) update: A report from the American heart association, Circulation. 141 (9) (2020) 139-596.

Seid, A., (2014), Joint modeling of longitudinal CD4 cell counts and time-to-default from HAART treatment: a comparison of separate and joint models. Electronic Journal of Applied Statistical Analysis, 7(2): p. 292-314.

Tan LB, Williams SG, Tan DK, Cohen-Solal A (2010) So many definitions of heart failure: are they all universally valid A critical appraisal, Expert review of cardiovascular therapy 8(2):217–228

Voors AA et al (2017) Development and validation of multivariable models to predict mortality and hospitalization in patients with heart failure. Eur J Heart Fail 19(5):627–634

Wajner A et al (2017) Causes and predictors of in-hospital mortality in patients admitted with or for heart failure at a tertiary hospital in Brazil. Arq Bras Cardiol 109(4):321–330

Zahid F., Ramzan S, Faisal S, Hussain I (2019) Gender-based survival prediction models for heart failure patients: a case study in Pakistan. Plos One 14(2):0210602

URL1 (2022). World Health Organization https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases (Date of Access: 05.07.2022)

CURRICULUM VITAE

After completing her primary and secondary education in Ankara, she completed his undergraduate studies at Kırıkkale University Industrial Engineering and Anadolu University Business Administration. She completed her undergraduate education at ‘TUSAŞ-Turkish Aerospace Industries Inc. NDI Quality Control Bench Capacity Determination 'and has completed her work with the third part of she. She completed his master's degree in Ankara Yıldırım Beyazıt University Management and Organization and Ankara Yıldırım Beyazıt University Management Information Systems Departments with the studies ‘Car Sales Automation Information System Case Study’ and ‘Analysis of the Benefits of Information Systems Used in Businesses: Yiğit Akü’. After her undergraduate education, she worked as an industrial engineer in a corporate company in Ankara. She later started her own business. She is currently working as an expert at Bakırçay TEKMER.