EARLY HEART DISEASE DIAGNOSIS AND PREDICTION USING MACHINE LEARNING ALGORTIHMS

by

Umut Ekin Kaya

DATS 599 Project Report

Yeditepe University

Data Science MSc. Program

2023

ACKNOWLEDGEMENTS

First of all I would like to thank my advisor Dr. Name Onur Demir for his guidance and support throughout my project.

ABSTRACT

EARLY HEART DISEASE DIAGNOSIS AND PREDICTION USING MACHINE LEARNING ALGORTIHMS

Heart related diseases are the most deadly of all types of diseases. Every year millions of people are dying because of cardiovascular diseases and early detection of heart diseases play vital a role for preventing these deaths. In this paper focus on early detection of patients which may have potentially heart related diseases which will lead to heart failure. It is a supervised learning project and classification problem. To get best results nine machine learning algorithms have been deployed and various data preprocessing methods have been used. Each algorithm is compared with each other in different scenarios. Four different metrics are taken into consideration while evaluating performance of machine learning algorithms. As a result from these nine algorithms best performance was delivered by random forest classifier with an accuracy of %88, Precision of %90, F1 score of 0.905 and MCCC of 0.76. This algorithm determines whether patience is under the risk of heart failure or not.

Key words: Heart failure prediction, machine learning algorithms, cardiovascular diseases

TABLE OF CONTENTS

1. 1

1.1. Importance of Topic 3

1.2. Project Aims3

2. **Hata! Yer işareti tanımlanmamış.**

2.1 3

2.2 3

2.3 3

2.4 3

2.5 3

2.6 3

2.7 3

2.8 3

2.9 3

2.10 3

3. 9

3.1. **Hata! Yer işareti tanımlanmamış.**

3.2. 13

3.3. 14

4. 20

4.1. 23

4.2. **Hata! Yer işareti tanımlanmamış.**

5. 23

6. 23

6.1. 27

6.2. **Hata! Yer işareti tanımlanmamış.**

7. 23

Bibliography 29

LIST OF FIGURES

Figure 1.1 Heart Disease Map 1

Figure 3.1.1 Age Distribution Graph 9

Figure 3.1.2 Sex Distribution Graph 9

Figure 3.1.3 Chest Pain Type Distribution Graph 10

Figure 3.1.4 Resting Blood Pressure Distribution Graph 10

Figure 3.1.5 Cholesterol Distribution Graph 10

Figure 3.1.6 Fasting Blood Sugar Distribution Graph 11

Figure 3.1.7 Resting Electrocardiogram Results Distribution Graph 11

Figure 3.1.8 Maximum Heart Rate Results Distribution Graph 12

Figure 3.1.9 Oldpeak Distribution Graph 12

Figure 3.1.10 Exercise Induced Angina Distribution Graph 12

Figure 3.1.11 Slope of the peak exercise Distribution Graph 13

Figure 3.2.1 Untouched data correlation table 13

Figure 3.2.2 Untouched data correlation matrix 14

Figure 4.1.1 Confusion Matrix Example 16

Figure 4.2.1 One of the Tree Example in Random Forest 18

Figure 4.2.2 Feature Importances for Random Forest 19

Figure 5.1.1 Pseudocode of Grid Search Algorithm 20

Figure 5.1.2 Pseudocode of LigthGbm Grid Search 21

Figure 5.1.3 Project Summary Workflow Chart 22

LIST OF TABLES

Table 2.1 Results of models analysis for first related work 3

Table 2.2 Results of models analysis for second related work 3

Table 2.3 Results of models analysis for third related work 4

Table 2.4 Results of models analysis for fourth related work 5

Table 2.5 Results of models analysis for fifth related work 5

Table 2.6 Results of models analysis for sixth related work 6

Table 2.7 Results of models analysis for seventh related work 7

Table 2.8 Results of mortality predictions by models 7

Table 2.9 Results of models analysis for ninth related work 7

Table 2.10 Results of models analysis for tenth related work 8

Table 6.1.1 Comparative Analysis (%66 train data) 23

Table 6.1.2 Confusion Matrix (%66 train data) 23

Table 6.1.3 Comparative Analysis (%20 train data) 24

Table 6.1.4 Confusion Matrix (%20 train data) 24

LIST OF SYMBOLS / ABBREVIATIONS

GLMN	Generalized Linear Model Net
LR	Logistic Regression
CART	Classification and Regression Tree
RF	Random Forest
AB	Adaboost
SVM	Support Vector Machine
NN	Neural Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

1. INTRODUCTION

1.1 Importance of Topic

For years, heart-related diseases have been one of the deadliest diseases in the world. Even though disease itself isn't new, research history of diseases goes back to 20th century. According to the British Heart Foundation there are 550 million people living with heart diseases and it seems like these numbers will increase due to modern unhealthy lifestyle and overpopulation. In 2019 the global death count was 19 million and every 1 of 3 deaths was caused by heart related disease (34 per cent). It means that every day 50,000 people was die due to heart diseases. The bad part of this situation is that numbers or ratios were not that high in the last decades. In 1990 the ratio was 27 per cent.

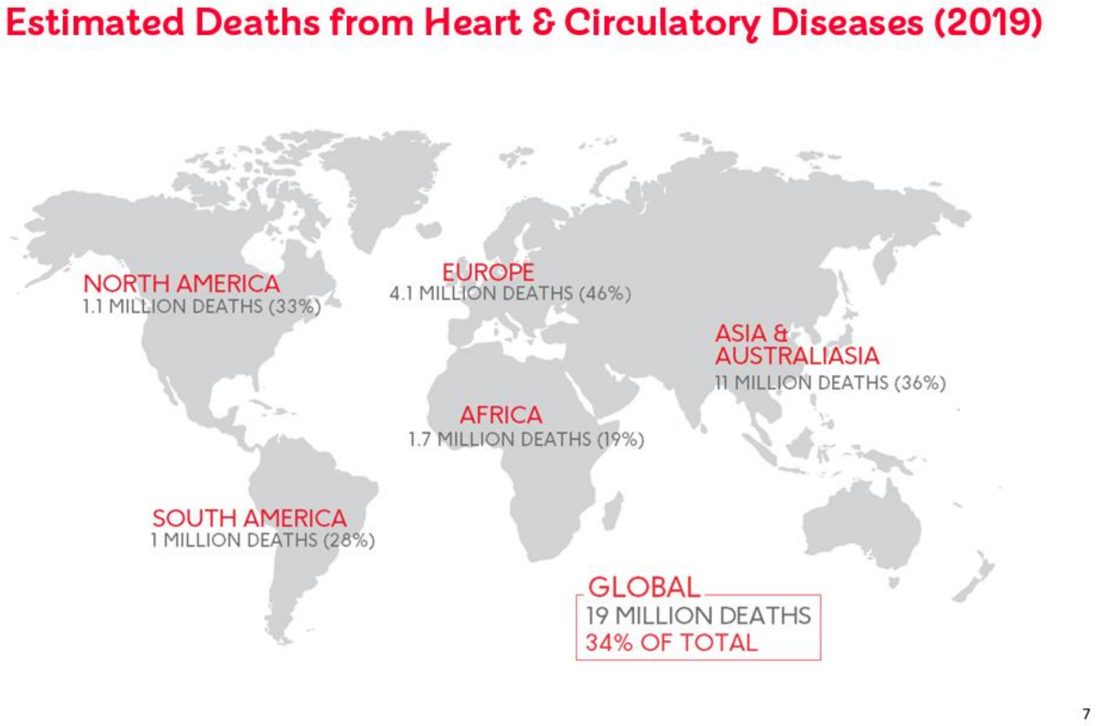


Figure 1.1 Heart Disease Map

Most popular heart diseases are Coronary Heart Disease (Ischaemic Heart Disease; CHD), Stroke (Cerebrovascular Disease; CBVD), Heart Failure, Congenital Heart Diseases.

Four out of five cardiovascular disease deaths are due to heart strokes and attacks. In 2019 there were 100 million stroke survivors and 65 million people had heart failure.

Early detection plays a vital role for these diseases. There are some symptoms and features that may be related with heart related diseases. When we can create links between them or find clear relation It is possible to detect them earlier than before they become fatal.

1.2 Project Aims

The main purpose of the project is constructing a predictive classification model by using various machine learning algorithms and choosing the best suited model which will determine whether patience has potential to have heart related disease or not. The reason behind the choosing machine learning based approach is to find clear patterns between features which consists of both categorical and numerical features.

Also, before building any machine learning model, a crucial objective of this project is to find and implement best data preprocessing methods which will affect robustness of results directly. Statistical and encoding strategies are both applied during the preprocessing stage. I wanted to demonstrate and discuss the impact and importance of these preprocessing methods for this classification problem.

2. RELATED WORK

In this section there will be 10 academic articles which have the same or similar problem with my project. Which are;

2.1 Same Research with My Project

This research is used exactly same dataset with my project and their results are;

Models	Accuracy (%)	Specificity (%)	Sensitivity (%)
Logistic Regression	83.3	82.3	86.3
K neighbors	84.8	77.7	85.0
SVM	83.2	78.7	78.2
Random forest	80.3	78.7	78.2
Decision tree	82.3	78.9	78.5
DL	94.2	83.1	82.3

Table 2.1 Results of models analysis for first related work

2.2 Prediction of Heart Attack in Stroke Patients

Main purpose of this research was predicting potential patients who may have a heart attack. The dataset consists of 46520 patients and 38606 of them are adult. My dataset only contains adults. Also the raw data has 35 features and 2 of them eliminated during future selection. Their results are;

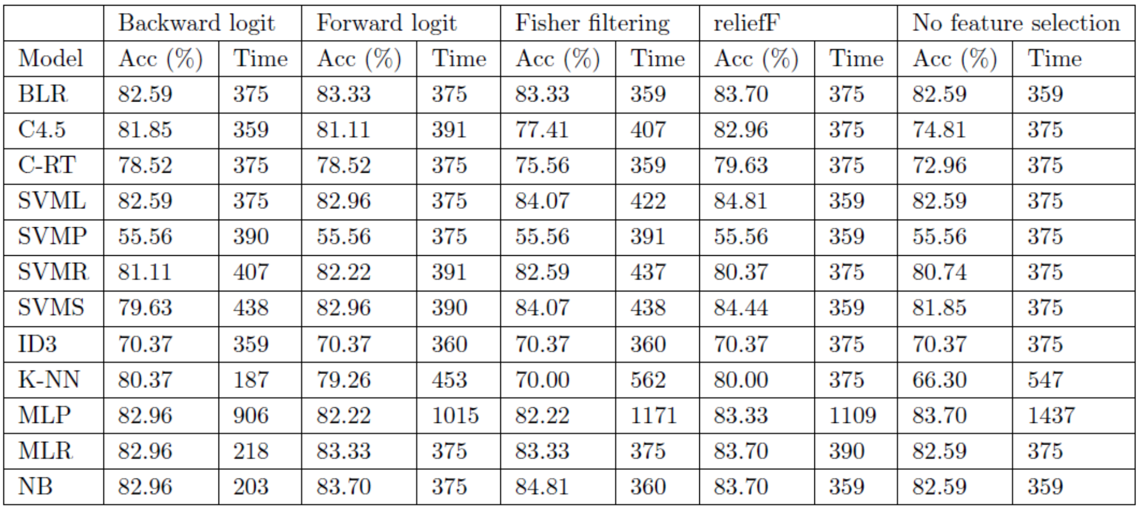
Model	Precission	Accuracy	F1 Score
Random Forest	70.05	70.29	66.13
Adaboost	65.37	66.75	61.56
KNN	58.67	62.13	60.28
DT	63.59	64.55	58.37
SVM	59.11	58.98	48.32

Table 2.2 Results of models analysis for second related work

The reason their model’s performances are lower than my project is that their data consists of too many patients unlike my dataset.

2.3 Heart Attack Prediction by Feature Selection

Main goal of this research is using and testing different feature selection methods in order to a create clear relation of features which should certainly lead to heart attack. The dataset which they used consists of 270 patients and 76 unique features. After using various feature selection models they have chosen 13 unique features and built their machine learning algorithms with these features. In addition, 9 out of the 13 features they used are the same as in the dataset of my project. The performance of models are;



	Backward logit		Forward logit		Fisher filtering		reliefF		No feature selection	
Model	Acc (%)	Time	Acc (%)	Time	Acc (%)	Time	Acc (%)	Time	Acc (%)	Time
BLR	82.59	375	83.33	375	83.33	359	83.70	375	82.59	359
C4.5	81.85	359	81.11	391	77.41	407	82.96	375	74.81	375
C-RT	78.52	375	78.52	375	75.56	359	79.63	375	72.96	375
SVML	82.59	375	82.96	375	84.07	422	84.81	359	82.59	375
SVMP	55.56	390	55.56	375	55.56	391	55.56	359	55.56	375
SVMR	81.11	407	82.22	391	82.59	437	80.37	375	80.74	375
SVMS	79.63	438	82.96	390	84.07	438	84.44	359	81.85	375
ID3	70.37	359	70.37	360	70.37	360	70.37	375	70.37	375
K-NN	80.37	187	79.26	453	70.00	562	80.00	375	66.30	547
MLP	82.96	906	82.22	1015	82.22	1171	83.33	1109	83.70	1437
MLR	82.96	218	83.33	375	83.33	375	83.70	390	82.59	375
NB	82.96	203	83.70	375	84.81	360	83.70	359	82.59	359

Table 2.3 Results of models analysis for third related work

2.4 Prediction of Coronary Artery Disease

Purpose of this project is predicting patients which have the potential to have any coronary artery disease. Their dataset consists of 303 patients of which 216 of them have disease. 3 machine learning models are used for this research and their results are;

Model	**Accuracy (%)**
Logistic Regression	89.61
Support Vector Machine	91.37
Artificial Neural Network	93.35

Table 2.4 Results of models analysis for fourth related work

2.5 Survival Analysis After Heart Attack

Purpose of this project using EHR (Electronic Health Record) of patients directly in order to find clear relation between selected features and build a more optimized model which performs better than the current model. From 119.649 patient records only 5044 of them were chosen as the main data source. Training dataset has 1560 patients and the test dataset has 3484 patients. Each patient's condition was monitored for a period of 5 years. If they survive 5 years. And machine learning algorithms have tried to predict the future of patients who have had heart stroke and EX means the patient couldn’t survive.

	1 Year		**2 Year**		**5 Year**	
Models	**BL**	**EX**	**BL**	**EX**	**BL**	**EX**
Decision Tree	0.60	0.66	0.50	0.50	0.50	0.50
Random Forest	0.62	0.80	0.65	0.72	0.62	0.72
Ada Boost	0.59	0.74	0.66	0.71	0.61	0.68
SVM	0.56	0.46	0.61	0.52	0.55	0.38
Logistic Regression	0.68	0.81	0.7	0.74	0.61	0.73

Table 2.5 Results of models analysis for fifth related work

Since, Logistic Regression has clearly better results, researchers have chosen the Logistic Regression algorithm as their final model.

2.6 Improved Heart Failure Prediction

Main purpose of this study is to take results from previous studies which use the same data and improve all of the machine learning algorithms that have been used in previous research. For this study 1013 numbers of patient's records have been used in order to predict the potential heart failure risks of patients which have some symptoms. Also 2 new unused machine learning algorithms were employed. These machine learning algorithms are Random forest and Naïve Bayes.

Models	**This Study**	**[UCI, Rapid Miner, 2019 [7]**	**[UCI, Matlab, 2017] [9]**	**[UCI, Weka, 2017] [9]**
Decision Tree	93.19%	82.22%	60.9%	67.7%
Logistic Regression	87.36%	82.56%	65.3%	67.3%
Random Forest	89.14%	84.17%	-	-
Naïve Bayes	87.27%	84.24%	-	-
SVM	92.30%	84.85%	67%	63.9%

Table 2.6 Results of models analysis for sixth related work

2.7 Survival Prediction by Using Serum Creatinine and Ejection Fraction

Main purpose of this research is predicting survival chances of patients after heart failure by using the patient's body features, symptoms and clinical laboratory test values. In this study 299 unique patients who had heart failure. For accurate predictions several classifiers have been employed.

Models	**MCC**	**F1 Score**	**Accuracy**	**TP rate**	**TN rate**
Random Forest	0.384	0.547	0.740	0.491	0.864
Decision Tree	0.376	0.554	0.737	0.532	0.831
Gradient Boosting	0.367	0.527	0.738	0.477	0.860
Linear Regression	0.332	0.475	0.730	0.394	0.892
Neural Network	0.262	0.483	0.680	0.428	0.815
Naïve Bayes	0.224	0.364	0.696	0.279	0.898
SVM radial	0.159	0.182	0.690	0.122	0.967

SVM linear	0.107	0.115	0.684	0.072	0.981
KNN	-0.025	0.148	0.624	0.121	0.866

Table 2.7 Results of models analysis for seventh related work

2.8 Modern Machine Learning Algorithms and Logistic Regression

The goal of this study is comparing machine learning algorithms with logistic regression and predicting outcomes of patients which had heart failure. There were 2 sources for this study, first one is claim based records from patients and second one is electronic medical records of patients. Dataset was consists of 9502 patients who are around 78 years old.

Models	**Accuracy (claims only)**	**Accuracy(Claims + EMR)**
Logistic Regression	0.158	0.152
LASSO	0.157	0.152
CART	0.165	0.161
Random Forest	0.156	0.150
GBM	0.156	0.148

Table 2.8 Results of mortality predictions by models

2.9 Hospitalization in Heart Failure Patients

The purpose of this study is to compare 8 different machine learning algorithms and find the model with best performance and predicting hospitalization risks for the patients who had heart failure.

Models	**Sensivity**	**Accuracy**
GLMN	77.8	81.2
LR	54.7	58.9
CART	44.3	63.5
RF	54.9	72.6
AB	57.3	67.1
SVM	57.3	69.9
NN	61.6	68.2

Table 2.9 Results of models analysis for ninth related work

2.10 Death or Readmission prediction after Heart Failure

Aim of this study is to compare various machine learning algorithms in order to find the best model which will predict readmission rate of patients. Predicting readmission rates will reduce health costs significantly.

Model	**Accuracy**
Logistic Regression	62.56
Random Forest	76.39
Decision Tree	66.97
SVM	71.8
Neural Network	64.93

Table 2.10 Results of models analysis for tenth related work

3. ANALYSIS AND DESIGN

3.1 Data Explanation

I have used a dataset from kaggle which is about Heart Failure. This data set is a combination of 5 different dataset which contains total of 918 observations from 5 different countries. There is 11 feature in the dataset which are;

Age: It is a numerical feature that indicates the age of a patient.

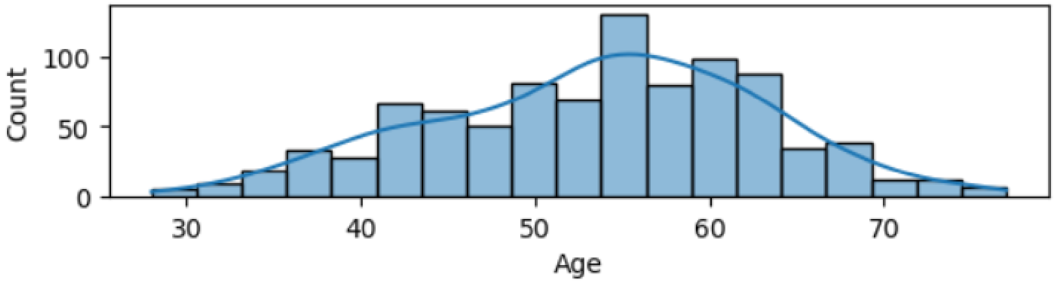


Figure 3.1.1 Age Distribution Graph

Sex: It is a categorical feature and M represents Males, F represents Females.

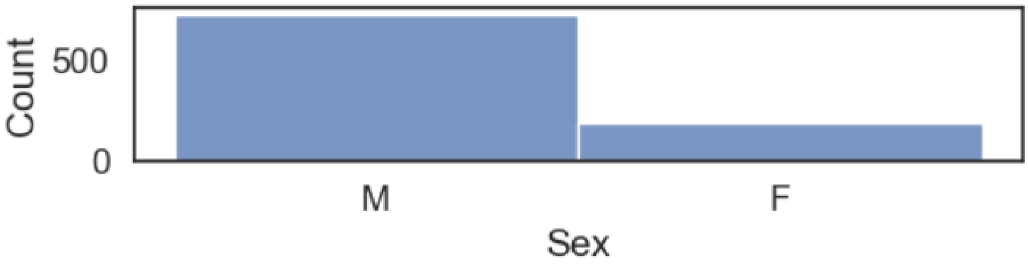


Figure 3.1.2 Sex Distribution Graph

Chest Pain Type: It is a categorical feature and there are 4 types of chest pain which are; TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain and ASY: Asymptomatic.

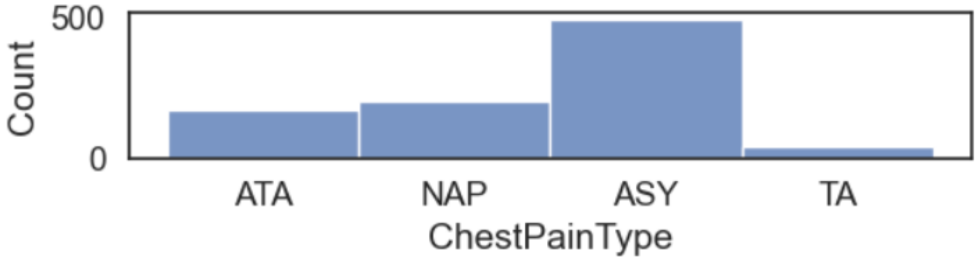


Figure 3.1.3 Chest Pain Type Distribution Graph

Resting Blood Pressure: It is a numerical feature and less than 120 mmHg is considered as healthy and if the value is higher than 180, the patient is diagnosed with hypertension.

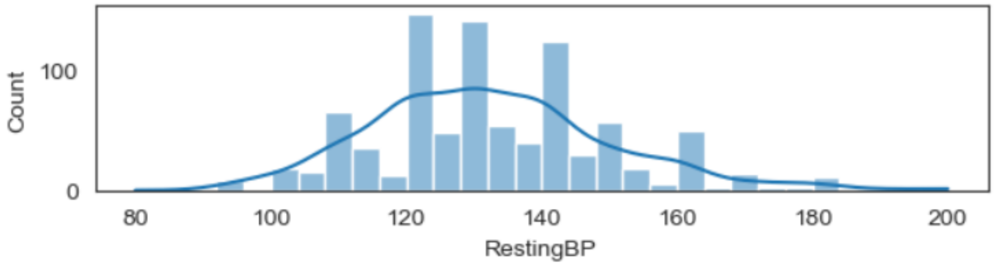


Figure 3.1.4 Resting Blood Pressure Distribution Graph

Cholesterol: It is a numerical feature and having less than 240 mg/dL is considered as healthy.

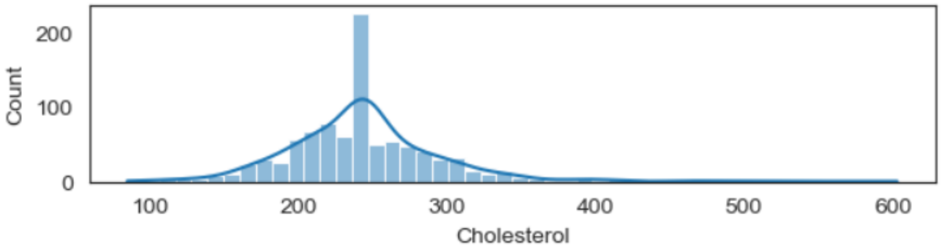


Figure 3.1.5 Cholesterol Distribution Graph

Fasting Blood Sugar: Even though fasting blood sugar has numerical value in this dataset we will use it as a categorical feature. 1 means having higher than 120 mg/dL and 0 means having lower than 120 mg/dL fasting blood sugar value.

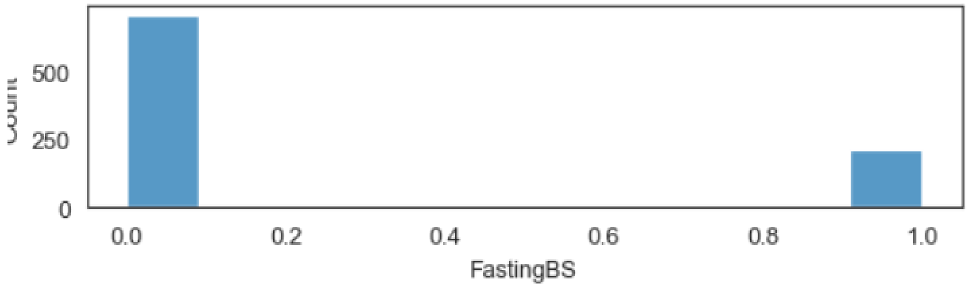


Figure 3.1.6 Fasting Blood Sugar Distribution Graph

Resting Electrocardiogram Results: Resting electrocardiography is a test which detects abnormalities and finds evidence for heart disease. In the dataset ECG is a categorical feature and Normal indicates normal results, ST indicates having S-T wave abnormality and LVH indicates probable or definite left ventricular hypertrophy by Estes' criteria.

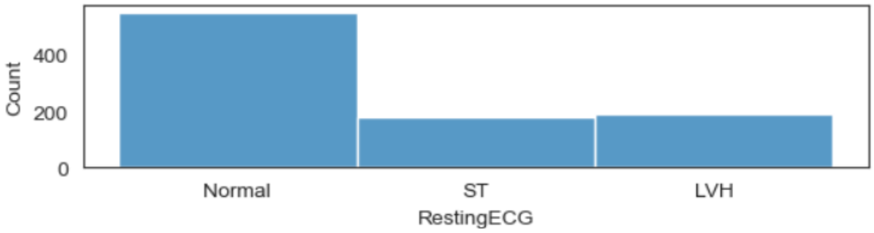


Figure 3.1.7 Resting Electrocardiogram Results Distribution Graph

Maximum Heart Rate: It is a numerical feature and represents the heart beat amount for 1 minute. Between 60 and 100 is considered normal however since it depends on age these values can be different for different ages.

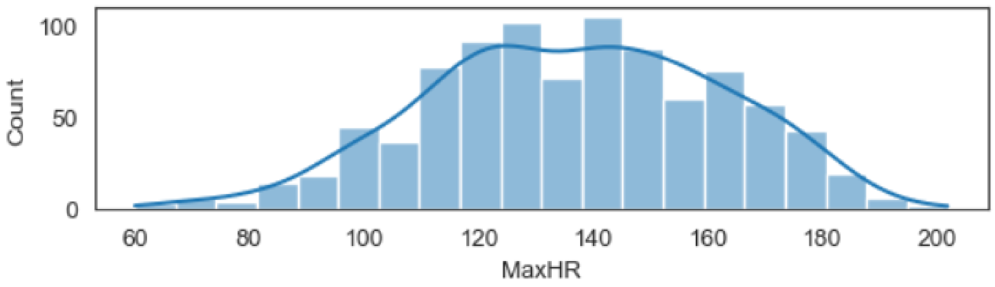


Figure 3.1.8 Maximum Heart Rate Results Distribution Graph

Oldpeak: Numeric value of a ST depression induced by exercise relative to rest

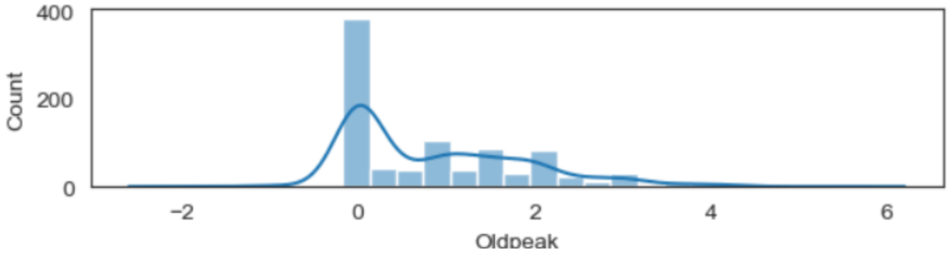


Figure 3.1.9 Oldpeak Distribution Graph

Exercise Induced Angina: It represents pain in the chest that comes on with exercise, stress, or other things that make the heart work harder. In the dataset It is categorical value and Y means yes N means no.

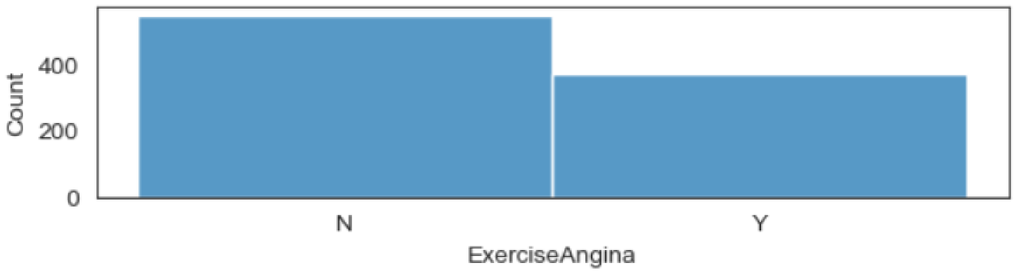


Figure 3.1.10 Exercise Induced Angina Distribution Graph

Slope of the Peak Exercise: Slope of the peak exercise ST segment. Up means upsloping, Flat means flat, Down means downsloping.

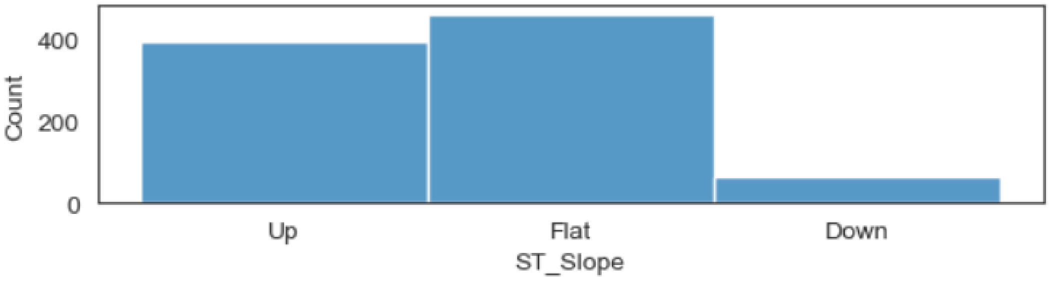
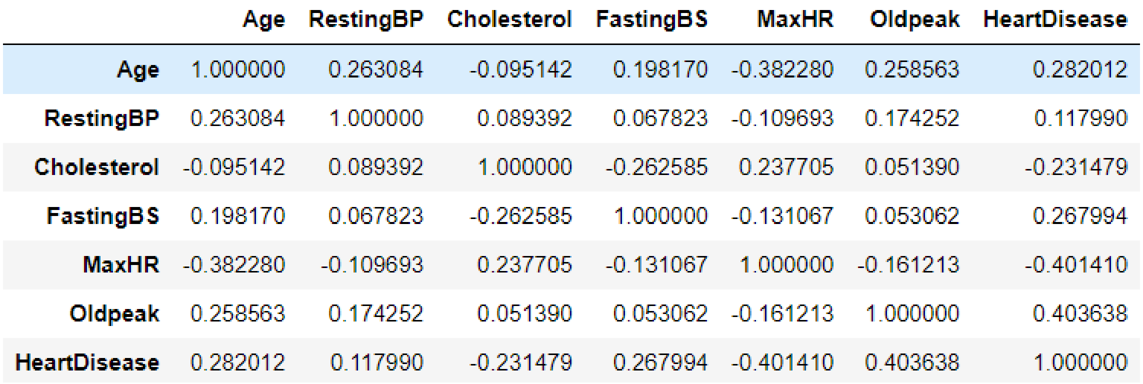


Figure 3.1.11 Slope of the peak exercise Distribution Graph

Since it is a classification problem finally, It has the binary Heart Disease feature.

3.2 Data Analysis

Before starting the preprocess I have checked the correlation between each value. Their correlation ratio was between 0.4 and -0.4 which is an acceptable range. Dataset does not have high correlation relationships such as 0.9 etc. However the cholesterol was negatively correlated with heart disease and unless It is not HDL (good cholesterol) there should be a problem in the data set. Since it was not clear the which type of cholesterol is given therefor I have checked cholesterol distribution and I have realized there are zero values which cannot be true. I have changed zeros with mean and the correlation between having heart disease became 0.09. When we consider good cholesterol and bad cholesterol cancel each other, having no strong correlation is safer than assuming the dataset only has good cholesterol.



	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1.000000	0.263084	-0.095142	0.198170	-0.382280	0.258563	0.282012
RestingBP	0.263084	1.000000	0.089392	0.067823	-0.109693	0.174252	0.117990
Cholesterol	-0.095142	0.089392	1.000000	-0.262585	0.237705	0.051390	-0.231479
FastingBS	0.198170	0.067823	-0.262585	1.000000	-0.131067	0.053062	0.267994
MaxHR	-0.382280	-0.109693	0.237705	-0.131067	1.000000	-0.161213	-0.401410
Oldpeak	0.258563	0.174252	0.051390	0.053062	-0.161213	1.000000	0.403638
HeartDisease	0.282012	0.117990	-0.231479	0.267994	-0.401410	0.403638	1.000000

Figure 3.2.1 Untouched data correlation table

When the numerical features are analyzed except maximum heart rate, all of the features have positive relation with having heart disease and it is quite reasonable. However when we analyze categorical features 3 out of 4 types of chest pain have negative correlation also having normal ECG and Upward ST Slope have negative correlation with heart disease.

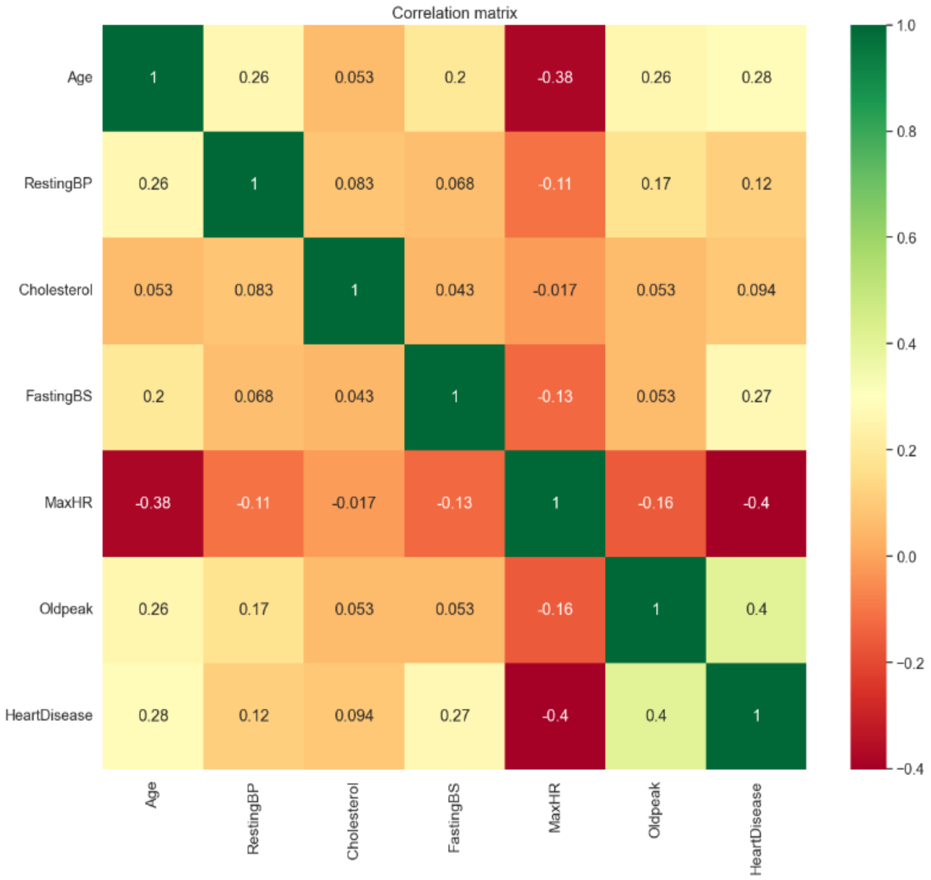


Figure 3.2.2 Untouched data correlation matrix

Analyzing correlation between 11 features and having heart disease gave a clean idea about the problem and better have better knowledge about features.

3.3 Preprocessing

For find outliers of each feature IQR (Interquartile Range) test has been performed. In order to perform IQR firstly %25 (Q1) and %75 (Q3) quantile calculated and their differences assigned as IQR after that lower bound (Q1-1.5*IQR) and upper bound (Q3 + 1.5*IQR) calculated and the values which do not belong to this range excluded from data however this reduced performances of all models so removing outliers didn’t help the improve performance of models. In addition this method, z score test also applied and it didn’t work as well.

As an alternative approach I have analyzed their distribution graphs and rather than eliminating outliers I have performed some cleaning operations however the original dataset was performing better so only change which affects instance count was the eliminating 0 heart rate patient, in addition to this filled every 0 cholesterol value with mean of cholesterol.

As the next step in order to increase performance of models I have performed 2 type scalings which are standard scaler and min-max scaler. Even though it didn’t increase performance of the best model drastically, it still provided slight improvement, however improved some of the models performance so much so my numerical features have placed between 0 and 1 due to min max scaling. The formula of scaler is;

$$X'=(X - X\_min) / (X\_max\_Xmin)$$

Final step of preprocessing was encoding categorical values into binary values. In order to work without problem Since it has no specific formula I have used one hot encoder for this process.

After encoding the feature selection process has been employed however for the best result all features included to test and train datasets.

Also I haven’t performed regularization in the preprocess step since some of the algorithms have both l1 and l2 regularization as a parameter and the rest of the algorithms are not benefit or they can’t work with either l1 or l2 regularization.

4. Model

In this section, I will explain what the performance metrics that I used for evaluating my models and how the evaluation was done by using these metrics. Then I will mention the machine learning models which I have used in the project and then I will explain the working principles of the Random Forest Classifier algorithm, which I have selected as the main algorithm for my the project.

4.1 Performance Evaluation Methods

4.1.1 Confusion Matrix

Confusion matrix is a performance evaluation for classification models. Matrix have 4 parameters which are TP, FP, FN, TN. TP refers to “True Positive” which means model classified correctly instances which are positive. TN refers to “True Negative” which means model classified correctly instances which are negative. FP refers to “False Positive” which means even though instances are negative still the model classified them as positive and lastly FN refers to “False Negative” which means even though instances are positive, the model classifies them as negative. In addition I will provide examples directly from my project.

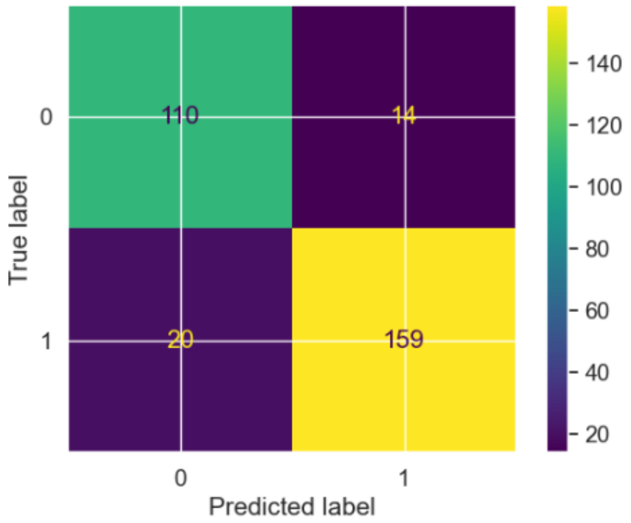


Figure 4.1.1 Confusion Matrix Example

4.1.2 Accuracy

Accuracy is a performance metric that measures the overall ability of the model to make correct predictions. It means that all of the parameters which are True Positive, False Positive, True Negative and False Negative are important for this metric.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)

4.1.3 Precision

Precision is a performance metric which evaluates the only performance of models on true instances. It means that only True Positives and False Positives are important for this method.

Precision = True Positive / (True Positive + False Positive)

4.1.4 F1 Score

F1 score is a harmonic mean of precision and recall metrics. Recall is a performance metric which will evaluate the performance of models on only positive instances. Formula of recall is; Recall = TP / (TP + FN). So the formula of F1 Score is;

F1= 2 * (Precision * Recall) / (Precision + Recall)

4.1.5 Matthews Correlation Coefficient (MCCC)

Matthew Correlation Coefficient is a metric which measures the overall performance a of model.

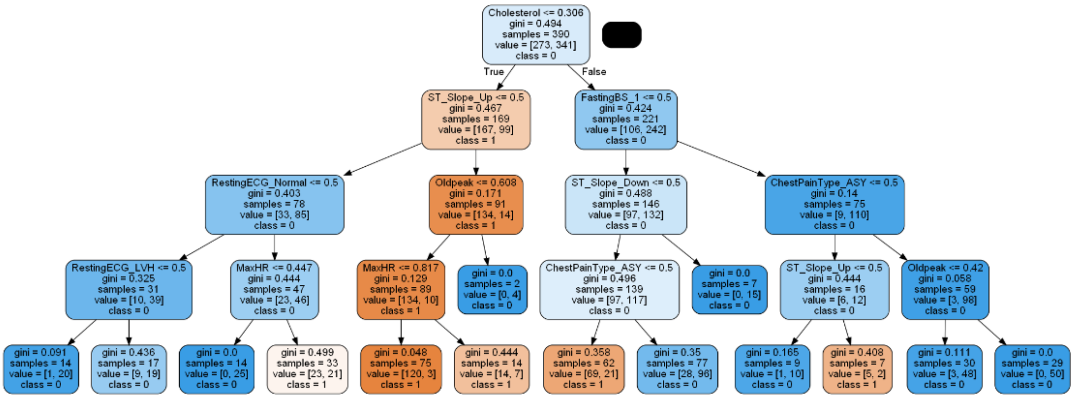
MCC = (TP*TN – FP*FN) / SQRT ((TP+FT)+(TP+FN)+(TN+FP)+(TN+FN))

4.2 Model

Since my project is a classification problem all models that I choose are classifier. 9 different machine learning algorithms were used. These algorithms are;

LightGBM Classifier, XGBoost Classifier, Support Vector Machine, K-Nearest Neighbor, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Neural Network Classifier and Naïve Bayes and after all of the preprocess and optimization processes Random Forest classifier has been chosen.

Basic working principle of random forest classifier is firstly the algorithm creates individual trees which use random subsamples of training data. Then these trees are splitting based on the feature that delivers the best chosen criteria which can be gini, gain or impurity etc. In my model gini was chosen as a separator metric. And since my n_estimators 100 random forest had created 100 trees with 4 leaves and split them according to best gini score on chosen feature.



Cholesterol <= 0.306 gini = 0.494 samples = 390 value = [273, 341] class = 0
True
False
ST_Slope_Up <= 0.5 gini = 0.467 samples = 169 value = [167, 99] class = 1
FastingBS_1 <= 0.5 gini = 0.424 samples = 221 value = [106, 242] class = 0
RestingECG_Normal <= 0.5 gini = 0.403 samples = 78 value = [33, 85] class = 0
Oldpeak <= 0.608 gini = 0.171 samples = 91 value = [134, 14] class = 1
ST_Slope_Down <= 0.5 gini = 0.488 samples = 146 value = [97, 132] class = 0
ChestPainType_ASY <= 0.5 gini = 0.14 samples = 75 value = [9, 110] class = 0
RestingECG_LVH <= 0.5 gini = 0.325 samples = 31 value = [10, 39] class = 0
MaxHR <= 0.447 gini = 0.444 samples = 47 value = [23, 46] class = 0
MaxHR <= 0.817 gini = 0.129 samples = 89 value = [134, 10] class = 1
gini = 0.0 samples = 2 value = [0, 4] class = 0
ChestPainType_ASY <= 0.5 gini = 0.496 samples = 139 value = [97, 117] class = 0
gini = 0.0 samples = 7 value = [0, 15] class = 0
ST_Slope_Up <= 0.5 gini = 0.444 samples = 16 value = [6, 12] class = 0
Oldpeak <= 0.42 gini = 0.058 samples = 59 value = [3, 98] class = 0
gini = 0.091 samples = 14 value = [1, 20] class = 0
gini = 0.436 samples = 17 value = [9, 19] class = 0
gini = 0.0 samples = 14 value = [0, 25] class = 0
gini = 0.499 samples = 33 value = [23, 21] class = 1
gini = 0.048 samples = 75 value = [120, 3] class = 1
gini = 0.444 samples = 14 value = [14, 7] class = 1
gini = 0.358 samples = 62 value = [69, 21] class = 1
gini = 0.35 samples = 77 value = [28, 96] class = 0
gini = 0.165 samples = 9 value = [1, 10] class = 0
gini = 0.408 samples = 7 value = [5, 2] class = 1
gini = 0.111 samples = 30 value = [3, 48] class = 0
gini = 0.0 samples = 29 value = [0, 50] class = 0

Figure 4.2.1 One of the Tree Example in Random Forest

Once all trees are built each tree votes for their class label and most voted class label are chosen as predicted class label for these trees. Finally all the predictions of individual decision trees aggregated and they have start to voting for the final class label of the whole algorithm. Most voted class label will be chosen as the algorithm's final class label.

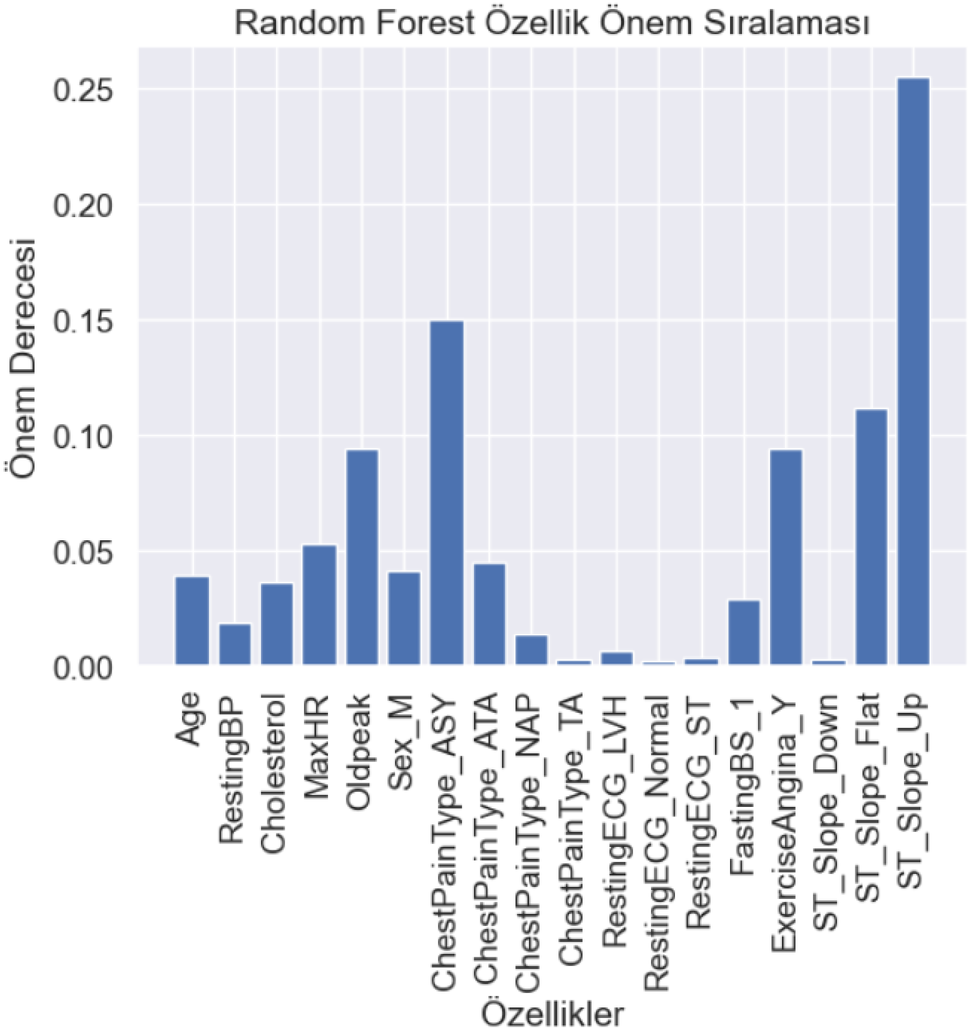


Figure 4.2.2 Feature Importances for Random Forest

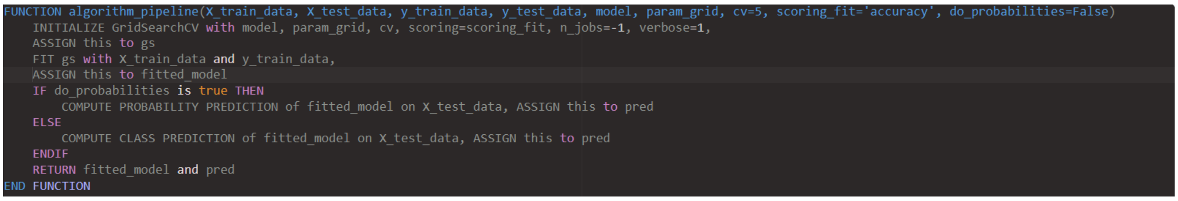
5. IMPLEMENTATION

The first step of the project was finding an appropriate dataset. Before making any progress I have evaluated and changed four different dataset. The reason behind this is the dataset that I have found must be aligned well with my project goals and shouldn’t have major issues for any machine learning algorithm. Also the quality of the dataset is one of the most important things for my project.

After evaluating the dataset the next step was preprocessing. Firstly, missing values have been filled with the average of the feature. Scaling and encoding algorithms were applied to the dataset.

Next stage was building various machine learning models. Firstly without focusing any hyperparameter only algorithms were built. After building algorithms, the most important part was the grid searching process.

In order to create a proper grid search algorithm on python firstly I have defined a new algorithm pipeline function. As a parameter it has both train and test data, cross validation which is 5 for my project, a metric that evaluates the performance of parameters which is accuracy and GridSearchCv function which is part of scikit-learning library. Since GridSearchCv function has Its unique parameters I have to define them as well. Pseudocode that define algorithm_pipeline is;

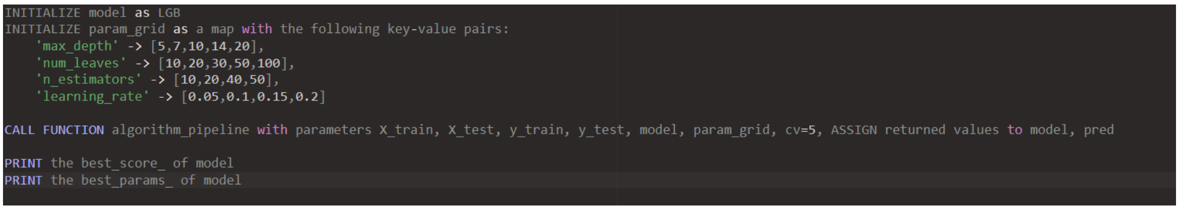


```
FUNCTION algorithm_pipeline(X_train_data, X_test_data, y_train_data, y_test_data, model, param_grid, cv=5, scoring_fit='accuracy', do_probabilities=False)
    INITIALIZE GridSearchCV with model, param_grid, cv, scoring=scoring_fit, n_jobs=-1, verbose=1,
    ASSIGN this to gs
    FIT gs with X_train_data and y_train_data,
    ASSIGN this to fitted_model
    IF do_probabilities is true THEN
        COMPUTE PROBABILITY PREDICTION of fitted_model on X_test_data, ASSIGN this to pred
    ELSE
        COMPUTE CLASS PREDICTION of fitted_model on X_test_data, ASSIGN this to pred
    ENDIF
    RETURN fitted_model and pred
END FUNCTION
```

Figure 5.1.1 Pseudocode of Grid Search Algorithm

After creating a grid search function I have started customized it for every different machine learning algorithm since every model has their unique hyperparameters. In order to save time my method was to create general intervals for major and important hyperparameters and then use them in order to find optimal values. After finding more optimal results, I have created more specific intervals which will be close to previous best parameter values and I have repeated this process until I find the best values for important hyperparameters. With this method not only have I found better values for hyperparameters but I have saved lots of time since some of the hyperparameters have static values then I have eliminated them from grid searching.

As e example I have used max_depth, num_leaves, n_estimators, random_state, learning_rate, reg_alpha and reg_lamda which are regularization functions, subsample, tree_learner, colsample_bytree hypermaters for LightGbm algorithm. However my final grid search didn’t have all these hyperparameters. Which was like;



```
INITIALIZE model as LGB
INITIALIZE param_grid as a map with the following key-value pairs:
    'max_depth' -> [5,7,10,14,20],
    'num_leaves' -> [10,20,30,50,100],
    'n_estimators' -> [10,20,40,50],
    'learning_rate' -> [0.05,0.1,0.15,0.2]

CALL FUNCTION algorithm_pipeline with parameters X_train, X_test, y_train, y_test, model, param_grid, cv=5, ASSIGN returned values to model, pred

PRINT the best_score_ of model
PRINT the best_params_ of model
```

Figure 5.1.2 Pseudocode of LigthGbm Grid Search

Final stage in my project was evaluating results of each model according to selected scoring metrics and chose most suitable machine learning for this problem.

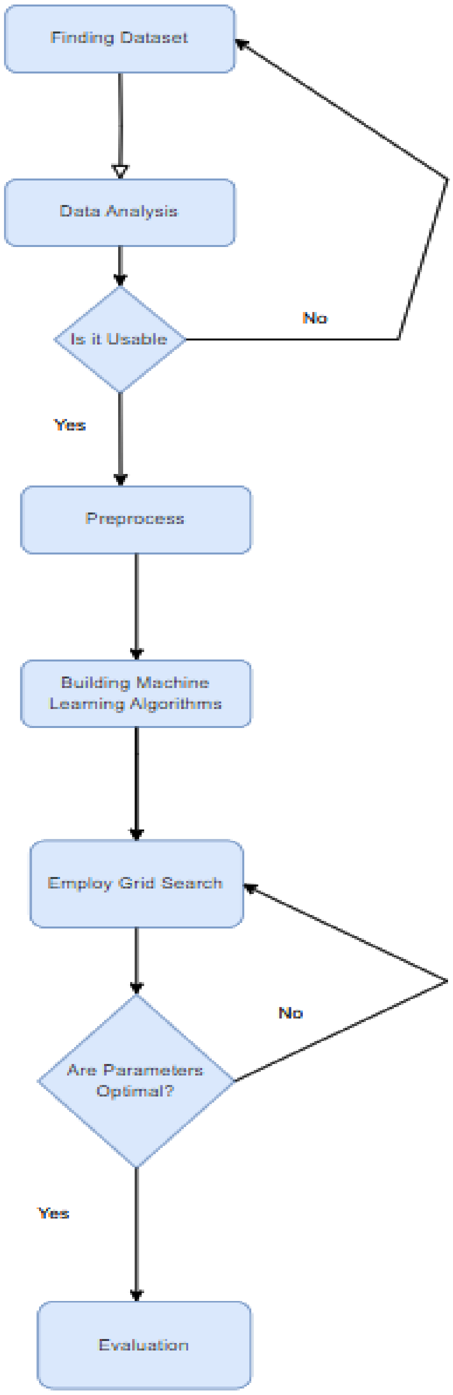


Figure 5.1.3 Project Summary Workflow Chart

6. RESULTS

6.1 Performance of Models

Results are produced from a heart failure dataset which has information of 917 patients. Mean Age of patients was 58 which is relatively young for this project.

In first trial every model at first trained with %66 of the dataset and %33 of the dataset used for the test which means 614 patient’s records have been used as training data and 303 patient’s records have been used as test dataset.

Model	Accuracy	Precision	F1 Score	MCCC
K-Nearest Neighbors	0.864	0.887	0.885	0.720
Neural Network	0.867	0.897	0.887	0.728
Naïve Bayes	0.844	0.897	0.863	0.686
Logistic Regression	0.858	0.909	0.875	0.713
Random Forest	0.887	0.900	**0.905**	0.767
Decision Tree	0.854	0.894	0.874	0.703
XGBoost	0.864	0.892	0.884	0.721
SVM	0.864	0.901	0.883	0.723
LightGBM	0.887	0.919	0.903	0.770

Table 6.1.1 Comparative Analysis (%66 train data)

Model	TN	FP	FN	TP
K-Nearest Neighbors	104	20	21	158
Neural Network	106	18	22	157
Naïve Bayes	107	17	30	149
Logistic Regression	109	15	28	151
Random Forest	106	18	16	**163**
Decision Tree	106	18	26	153
XGBoost	105	19	22	157
SVM	107	17	24	155
LightGBM	110	14	20	159

Table 6.1.2 Confusion Matrix (%66 train data)

In the second trial models had been fed with %20 of initial dataset and %80 of dataset has used as test dataset. It means that model had learned 183 patient's records and tried predict whether 734 patient had heart failure or not.

Model	**Accuracy**	**Precision**	**F1 Score**	**MCCC**
K-Nearest Neighbors	0.828	0.832	0.848	0.6515
Neural Network	0.841	0.875	0.853	0.6830
Naïve Bayes	0.833	0.874	0.845	0.637
Logistic Regression	0.818	0.860	0.831	0.713
Random Forest	0.856	0.868	**0.871**	0.710
Decision Tree	0.811	0.874	0.819	0.630
XGBoost	0.843	0.868	0.856	0.684
SVM	0.817	0.840	0.834	0.631
LightGBM	0.852	0.869	0.866	0.702

Table 6.1.3 Comparative Analysis (%20 train data)

Model	**TN**	**FP**	**FN**	**TP**
K-Nearest Neighbors	256	71	55	352
Neural Network	279	48	68	339
Naïve Bayes	279	48	74	333
Logistic Regression	274	53	80	327
Random Forest	273	54	51	**356**
Decision Tree	282	45	93	314
XGBoost	275	52	63	344
SVM	263	64	70	337
LightGBM	275	53	55	352

Table 6.1.4 Confusion Matrix (%20 train data)

Since one of the main goals of this project is to try various machine learning algorithms and find the best one for early detection of heart diseases, every metric is important. However, for choosing the best model, I will use F1 Score. The reason behind choosing F1 score is It takes all of the predictions into consideration. It means that True Negatives, True Positives, False Negatives and False Positives will have an impact on my model. Also since the main goal of the project is predicting the people that may have heart diseases True positives are more valuable and Random Forest algorithms have better scores on both F1 and TP. However LightGBM also has quite similar results with Random Forest.

Random forest has %88 accuracy and 0.905 F1 score. It means that the model can predict correctly every 88 patience from 100 patience. Also random forest was better than every algorithm in order to find people who have heart failure potential. However if our focus is finding healthy people then LightGBM can be a better option.

Since I didn’t have a large dataset, I hadn’t any problem with time. However, if our dataset becomes larger, especially in the case of introducing new parameters, then random forest may not be the best algorithm for the project. In terms of time usage for small datasets using random forest algorithms is a good idea. However It doesn’t mean that random forest will always perform better in small datasets because due to nature of algorithm, in the data there must be diversity.

If the sole purpose of the study is only getting high accuracy scores then using random forest may be a good idea since due to working principle It always tries to reduce overfitting.

Also If the computer has limited memory probably using neither random forest nor boosting algorithms in more complex datasets will not be a good choice. Since these algorithms are heavy memory users.

Also if the purpose of the project is not only making correct predictions but also understanding the work principle or monitoring each stage of the algorithm then random forest is definitely not a good option. In the case of selecting large n_estimators it is impossible to follow steps since there may be hundreds of decision trees.

6.2 Evaluation and Discussion

During training session models trained with two different portions of the main dataset. First one used %66 of data as a train set and second one %20 of data as train set. The reason behind this experiment was measuring the model's reaction in case of lack of train data. Because of the structure of the dataset I hadn’t observed huge drops model’s performances however every model delivered worse results when trained with %20 of dataset. Another reason for this experiment is trying to understand if there is an overfit case. Since my dataset isn’t too large I had a suspicion that using %66 of the dataset as a train can lead to overfit. However the results are quite acceptable when we use less train data for each model and more convenient. Every model has higher than %80 accuracy and f1 scores were higher than 0.8.

Another experiment was about measuring the hyperparameters of models. With and without grid searching I have been using lots of different hyperparameters with different values and I clearly observed that both XGBoost and LightGBM, Random Forest, KNN and Neural Network algorithms are really sensitive to hyperparameters. For example changing n_estimator changes the results of tree based algorithms instantly.

Another thing that I want to learn is can my problem be solved by an unsupervised learning algorithm. Expect KNN every single algorithm that I have used was supervised learning algorithms and that means they heavily need the results in order to create connection between features. So that’s why I have included the KNN algorithm which is a supervised learning algorithm and it’s performance clearly proved that my classification problem can be solved by using an unsupervised learning algorithm.

In order to understand the impacts of outliers I have excluded them from my main dataset and feed my models with new dataset however results have become much worse. One reason is models are benefitting from outliers, another reason may be models are good at dealing with useless or bad impacting outliers however keeps beneficial records.

Also even though it was clear from correlation between features and having heart disease still I wanted to reduce the number of features. The reason behind this experiment was observing the impacts of excluding the feature which has relatively low correlation with target value on performances of models. However best performance metrics came from the natural data not the one which has reduced features.

I have used python for everything in my project. Which means all of the experiments, models, data analysis and preprocess was done in python. Since It has provided easy access and a user friendly interface I have chosen Jupyter Notebook which is a computational and documentation environment, to use Python in my project.

7. CONCLUSION & FUTURE WORK

With this project I want to show that people have suffering from certain diseases and early detection is so important. Using machine learning algorithms it is possible to analyze certain symptoms and create connections between each other which will lead to the conclusion of potential disease. The reason I have chosen heart related diseases is that for over decades they have been the most deadly disease. Every year millions of people are dying because of cardiovascular diseases. This project gave me a chance to apply things that I have learned from lessons by doing some experiments and comparisons and these experiments gave me more insight and improve myself in this field.

My first experiment was about choosing the right ratio for splitting my dataset and also observing the impacts of this process. Every single model had worse performance when I dropped my train dataset ratio from %66-20. My chosen model, which is a random forest, had %88.7 accuracy and 0.905 F1 score when I trained my model with %66 of initial data however when I reduce It to %20 then accuracy dropped to %85.6 and F1 score dropped to 0.871.

Another thing that I wanted to figure out was how models react to changes in their hyperparameters. Are they hyperparameter sensitive or they are overfitting because of hyperparameters. Even though not all models are sensitive to their respective parameters still there are vital parameters which can change the results completely. For example for the lightgbm algorithm If I chose n_estimators 1 then model's accuracy was %70 however when I increase it 2 then it accuracy becomes %83 however If I have chosen 3 then accuracy will drop then I make it 4 then accuracy increase to %86 etc. The result of these experiments is that most of the algorithms that I have chosen are hyperparameter sensitive and It is important to choose good values for these parameters.

The learning type of algorithm is not that important for my project as long as the model can be classified. For example xgboost is a supervised learning algorithm and KNN is an unsupervised learning algorithm however they both deliver acceptable and close results.

With %66 train data KNN and XGBoost had the same accuracy and with %20 train data KNN had better accuracy which was %82.8 however still not the best one.

When I have exclude outliers from dataset every single model's performances dropped. For example with all data Random Forest had 0.905 F1 score and Decision Tree had 0.874 however when I exclude outliers from my dataset their F1 scores have become 0.876 and 0.824. So even though in theory getting rid of outliers is a good idea still It is not valid for every dataset.

Final experiment was about features. I have checked and tried to find unnecessary or bad features for models however using all of the features delivered the best results. 15 features had %87 accuracy for lightgbm and 18 was %88.7. (Feature selection has performed after encoding process which is why I had 18 feature in total not 11)

As a result my final algorithm recommendation for this project is Random Forest Classifier algorithm. The values that I got from performance metrics are; %88.7 accuracy, 0.9 precision, 0.905 F1 score, 0.767 MCCC and from 303 patients, the model has classified the status of 269 patients correctly.

In the future there are some improvements for this project. Firstly new features or the symptoms must be added to this project. In case some of the deep learning algorithms can be used for gathering more features for this datasets. This dataset is created by medical records of patients; however with the help of NLP (natural language processing) algorithms it is possible to generate more data or features directly from doctor's records and their hand writings. Also not only NLPs but with the help of image processing algorithms (CNNs) the new data can be generated from directly x-ray images. Since the dataset is not complex, I have used most common machine learning algorithms, however if the dataset becomes larger, more complex deep learning algorithms which can have more layers can be used. In fact, a dataset should have hundreds or even thousands of features for more accurate and decisive decisions. I believe that using machine learning algorithms in healthcare has a bright future. In the near future we can see the machine learning models being used by doctors as a decision support system or some of the diseases will become non-threatening for us.

Bibliography

[1] Bharti, R., Khamparia, A., Shabaz, M., Dhiman, G., Pande, S., & Singh, P. (2021). Prediction of heart disease using a combination of machine learning and deep learning. Computational intelligence and neuroscience, 2021.

[2] M. Wang, X. Yao and Y. Chen, "An Imbalanced-Data Processing Algorithm for the Prediction of Heart Attack in Stroke Patients," in IEEE Access, vol. 9, pp. 25394-25404, 2021, doi: 10.1109/ACCESS.2021.3057693.

[3] TAKCI, HİDAYET (2018) "Improvement of heart attack prediction by the feature selection methods," Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 26: No. 1, Article 1. https://doi.org/10.3906/elk-1611-235

[4] Learning, M. (2017). Heart disease diagnosis and prediction using machine learning and data mining techniques: a review. Adv. Comput. Sci. Technol, 10(7), 2137-2159.

[5] Panahiazar M, Taslimitehrani V, Pereira N, Pathak J. Using EHRs and Machine Learning for Heart Failure Survival Analysis. Stud Health Technol Inform. 2015;216:40-4. PMID: 26262006; PMCID: PMC4905764.

[6] Alotaibi, F. S. (2019). Implementation of machine learning model to predict heart failure disease. International Journal of Advanced Computer Science and Applications, 10(6).

[7] Chicco, D., Jurman, G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Med Inform Decis Mak 20, 16 (2020). https://doi.org/10.1186/s12911-020-1023-5

[8] Desai RJ, Wang SV, Vaduganathan M, Evers T, Schneeweiss S. Comparison of Machine Learning Methods With Traditional Models for Use of Administrative Claims With Electronic Medical Records to Predict Heart Failure Outcomes. JAMA Netw Open. 2020;3(1):e1918962. doi:10.1001/jamanetworkopen.2019.18962

[9] Lorenzoni G, Sabato SS, Lanera C, Bottigliengo D, Minto C, Ocagli H, De Paolis P, Gregori D, Iliceto S, Pisanò F. Comparison of Machine Learning Techniques for Prediction

of Hospitalization in Heart Failure Patients. Journal of Clinical Medicine. 2019; 8(9):1298. https://doi.org/10.3390/jcm8091298

[10] Awan, S. E., Bennamoun, M., Sohel, F., Sanfilippo, F. M., and Dwivedi, G. (2019) Machine learning-based prediction of heart failure readmission or death: implications of choosing the right model and the right metrics, ESC Heart Failure, 6: 428– 435. https://doi.org/10.1002/ehf2.12419.