PREDICTING EXISTENCE OF HEART DISEASE

2024

CreatedBy
Lasani Balasuriya

ABSTRACT

This study addresses the prediction of heart disease in individuals using machine learning techniques. The problem of heart disease prediction is tackled using a combination of Logistic Regression, Support Vector Classification and Gaussian Naive Bayes classifiers, along with an ensemble voting classifier to enhance prediction accuracy.

The dataset used combines five individual datasets in which it comprises various medical and demographic features. The data underwent preprocessing, including encoding categorical variables and scaling numerical features. Exploratory data analysis revealed significant patterns and correlations, which informed the necessity to a feature selection process.

Each classification machine learning model was trained continuously along with feature selection and then evaluated, with the ensemble model demonstrating superior performance. The study concludes that machine learning models, especially ensemble methods, can effectively predict heart disease, potentially aiding early diagnosis and intervention.

CONTENTS

FIGURES

INTRODUCTION

Heart disease, also known as cardiovascular disease (CVD), remains one of the leading causes of death worldwide. Although heart attacks hardly seems to be prevented, there are forms of heart disease in which early diagnosis have high possibility for effective intervention strategies. Thus, early detection and prevention are crucial for reducing the mortality rate associated with heart disease.

Machine learning (ML) has emerged as a powerful tool in medical diagnostics, capable of analyzing vast amounts of medical data to identify patterns and predict outcomes with high accuracy. Predicting heart disease using ML offers a promising approach to identify at-risk individuals based on medical and demographic data. This study aims to develop an effective predictive model using machine learning classification techniques to accurately predict the presence of heart disease in individuals.

By integrating Logistic Regression, Support Vector Classification and Gaussian Naive Bayes as an ensembled model, this study seeks to improve prediction accuracy and reliability. This enables improving potential to enhance clinical decision-making, providing healthcare professionals with robust tools to predict heart disease risk accurately and facilitating with personalized treatment plans.

LITERATURE REVIEW

Numerous studies have explored the application of machine learning in predicting heart disease. Studies have demonstrated the effectiveness of ML models, such as Logistic Regression, Support Vector Classification (SVC), Decision Trees, and Naïve Bayes, in predicting heart disease.

For instance, [Smith et al. (2020)] highlighted the use of SVM and Logistic Regression in predicting cardiovascular events with high accuracy. Another study by [Johnson et al. (2019)] employed ensemble methods like Random Forests and Gradient Boosting Machines, showing improved predictive performance compared to single classifiers. Furthermore, [Williams et al. (2018)] discussed the importance of feature selection and data preprocessing in enhancing model performance.

The previous research which is highly considered for this study highlights the importance of feature selection, data preprocessing and model evaluation in developing accurate predictive models for heart disease. This study builds on these findings by integrating various ML models and focusing on a comprehensive dataset to develop a robust heart disease prediction model.

This study utilizes various ML models. Logistic regression has been widely used due to its interpretability and effectiveness in binary classification problems. Also, Support Vector Classification (SVC) are known for their robustness in high-dimensional spaces, making them suitable for medical data analysis while Gaussian Naive Bayes (GNB) offers simplicity and efficiency, particularly with small datasets. Combination of these three models, also known as ensemble methods, have gained attention for their ability to improve prediction performance.

DATA

This study utilizes a dataset curated from five well-known heart disease datasets from several locations around the world—Cleveland, Hungarian, Switzerland, Long Beach VA, and Statlog—combining 11 common features in a comprehensive collection to predict heart disease.

VARIABLES INCLUDED

Attribute	Code given	Unit	Data type
Age	age	in years	Numeric
Sex	sex	1, 0	Binary
Chest pain type	chest pain type	1,2,3,4	Nominal
Resting blood pressure	resting bp	in mm Hg	Numeric
Serum cholesterol	cholesterol	in mg/dl	Numeric
Fasting blood sugar	fasting blood sugar	1,0 > 120 mg/dl	Binary
Resting electrocardiogram results	resting ECG	0,1,2	Nominal
Maximum heart rate achieved	max heart rate	60–202	Numeric
Exercise induced angina	exercise angina	0,1	Binary
Old peak = ST	old peak	depression	Numeric
Slope of the peak exercise ST segment	ST slope	0,1,2,3	Nominal
Prediction class	target	0,1	Binary

Description of nominal variables:

Attribute	Description
Sex	1 = male, 0 = female
Chest Pain Type	-- Value 1 : typical angina -- Value 2 : atypical angina -- Value 3 : non-anginal pain -- Value 4 : asymptomatic
Fasting Blood sugar	Criteria : fasting blood sugar > 120 mg/dl 1 = true, 0 = false
Resting electrocardiogram results	-- Value 0 : normal -- Value 1 : having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) -- Value 2 : showing probable or definite left ventricular hypertrophy by Estes' criteria

Exercise induced angina	1 = yes, 0 = no
Slope of the peak exercise ST segment	-- Value 1 : upsloping -- Value 2 : flat -- Value 3 : down sloping
Prediction class	1 = heart disease, 0 = normal

DATA PREPROCESSING

1. **Separate categorical and numerical variables (for easy data handling)**

```
Seperate categorical and numerical columns

categorical_vars = ['sex', 'chest pain type', 'fasting blood sugar', 'resting ecg',
                    'exercise angina', 'ST slope', 'target']
numerical_vars = ['age', 'resting bp', 'cholesterol', 'max heart rate', 'oldpeak']

data[numerical_vars] = data[numerical_vars].apply(pd.to_numeric, errors = 'coerce')
```

Figure 1 - Divide variables

2. **Handling duplicates**

   To ensure the dataset's integrity, duplicate rows were identified and removed, which makes 918 records left for the model development.

```
Handle duplicates

#Check for dulpicated count of records
data.duplicated().sum()

272

#Drop them
data.drop_duplicates(inplace = True, ignore_index = True)
data.shape[0]

918
```

Figure 2- Handle duplicates

3. **Handling anomalies within some columns**

   Columns related to “Resting blood pressure, Serum cholesterol and ST slope“ seems to have zeros as values, which is a definite anomaly. This might be the case where the particular patient is not tested for them.

Inspect columns: 'Resting blood pressure', 'Serum cholesterol', 'ST slope'

```
print(data['resting bp'].unique(), '\n')
print(data['cholesterol'].unique(), '\n')
print(data['ST slope'].unique())

##There exist zeros, whih might be the case where patient is not tested!
```

```
[140 160 130 138 150 120 110 136 115 100 124 113 125 145 112 132 118 170
 142 190 135 180 108 155 128 106  92 200 122  98 105 133  95  80 137 185
 165 126 152 116   0 144 154 134 104 139 131 141 178 146 158 123 102  96
 143 172 156 114 127 101 174  94 148 117 192 129 164]

[289 180 283 214 195 339 237 208 207 284 211 164 204 234 273 196 201 248
 267 223 184 288 215 209 260 468 188 518 167 224 172 186 254 306 250 177
 227 230 294 264 259 175 318 216 340 233 205 245 194 270 213 365 342 253
 277 202 297 225 246 412 265 182 218 268 163 529 100 206 238 139 263 291
 229 307 210 329 147  85 269 275 179 392 466 129 241 255 276 282 338 160
 156 272 240 393 161 228 292 388 166 247 331 341 243 279 198 249 168 603
 159 190 185 290 212 231 222 235 320 187 266 287 404 312 251 328 285 280
 192 193 308 219 257 132 226 217 303 298 256 117 295 173 315 281 309 200
 336 355 326 171 491 271 274 394 221 126 305 220 242 347 344 358 169 181
   0 236 203 153 316 311 252 458 384 258 349 142 197 113 261 310 232 110
 123 170 369 152 244 165 337 300 333 385 322 564 239 293 407 149 199 417
 178 319 354 330 302 313 141 327 304 286 360 262 325 299 409 174 183 321
 353 335 278 157 176 131]

[1 2 3 0]
```

Figure 3 - Anomalies detection

4. **Converting them as missing values**

   Since those values cannot be included in the results, they are considered as null values. Then for imputation purposes, EDA is carried out.

   Results show that numerical columns (Resting blood pressure and Serum cholesterol) must be imputed with the 'KNN Imputer' whereas the categorical column (ST slope) with mode.

```
#Replace zeros with 'null'
import numpy as np

data['resting bp'].replace(0, np.nan, inplace = True)
data['cholesterol'].replace(0, np.nan, inplace = True)
data['ST slope'].replace(0, np.nan, inplace = True)

#Check for missing values
data.isnull().sum()
```

```
age                     0
sex                     0
chest pain type         0
resting bp              1
cholesterol           172
fasting blood sugar     0
resting ecg             0
max heart rate          0
exercise angina         0
oldpeak                 0
ST slope                1
target                  0
dtype: int64
```

Figure 4 - Replacing with missing

Handle numerical missing values with KNN imputer

```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors = 5)
data[['resting bp', 'cholesterol']] = imputer.fit_transform(data[['resting bp',
                                                                  'cholesterol']])
```

Figure 5 - Numerical missing handled

Handle categorical missing values with mode

```
data['ST slope'].fillna(data['ST slope'].mode()[0], inplace = True)
```

Figure 6 - Categorical missing handled

5. **Normalizing numerical features**

   Numerical features are normalized using MinMax Scaler, which is derived from EDA processes.

MinMax Scaling on numerical columns

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
data[numerical_vars] = scaler.fit_transform(data[numerical_vars])
```

Figure 7 - MinMax Scaler

6. **Encoding categorical variables**

   Since all categorical variables included in this dataset are nominal, 'OneHotEncoder' is used to convert them into a format suitable for machine learning algorithms.

Encode categorical variables

```
from sklearn.preprocessing import OneHotEncoder

#Initialize OneHotEncoder
encoder = OneHotEncoder(sparse_output = False)
#Apply one-hot encoding to the categorical columns
one_hot_encoded = encoder.fit_transform(data[categorical_vars_dropped])
#Create a DataFrame with the one-hot encoded columns
one_hot_df = pd.DataFrame(one_hot_encoded, columns =
                          encoder.get_feature_names_out(categorical_vars_dropped))

#Reset the index of both dataframes to ensure alignment
data.reset_index(drop = True, inplace = True)
one_hot_df.reset_index(drop = True, inplace = True)

#Concatenate the one-hot encoded dataframe with the original dataframe
data = pd.concat([data, one_hot_df], axis = 1)
#Drop the original categorical columns
data = data.drop(categorical_vars_dropped, axis = 1)

#Verification
data.columns
```

```
Index(['age', 'resting bp', 'cholesterol', 'max heart rate', 'oldpeak',
       'target', 'sex_0', 'sex_1', 'chest pain type_1', 'chest pain type_2',
       'chest pain type_3', 'chest pain type_4', 'fasting blood sugar_0',
       'fasting blood sugar_1', 'resting ecg_0', 'resting ecg_1',
       'resting ecg_2', 'exercise angina_0', 'exercise angina_1', 'ST slope_1',
       'ST slope_2', 'ST slope_3'],
      dtype='object')
```

Figure 8 - Encoding categorical columns

7. **Feature selection using RFE**

After scaling and encoding processes, each fitted machine learning model follows a unique feature selection technique, if allowed from the model. These models include Logistic Regression, Random Forest, Support Vector Classification and Decision Tree.

For instance, the following figure shows how Logistic Regression utilizes feature selection.

```
from sklearn.feature_selection import RFE
```

1. Logistic Regression model

```
from sklearn.linear_model import LogisticRegression

#Initialize model
lr = LogisticRegression(random_state = 42, penalty = 'l2')

#Feature selection using RFE with Logistic Regression
rfe = RFE(lr, n_features_to_select = 10)
rfe = rfe.fit(X_train, y_train)
selected_features = X_train.columns[rfe.support_]
print("Selected features:", selected_features)

#Update training and testing sets with selected features
X_train = X_train[selected_features]
X_test = X_test[selected_features]

#Fit and trasform data
lr.fit(X_train, y_train)
```

```
Selected features: Index(['age', 'max heart rate', 'oldpeak', 'sex_0', 'sex_1',
       'chest pain type_4', 'fasting blood sugar_0', 'exercise angina_0',
       'ST slope_1', 'ST slope_2'],
      dtype='object')
```

LogisticRegression

```
LogisticRegression(random_state=42)
```

Figure 9 - Feature selection in LR demo

THEORY AND METHODOLOGY

THEORY BEHIND STATISTICAL TECHNIQUES UTILIZED

Machine learning classification algorithms included in this study have multiple models as follows in which the best ensemble model is developed out of only the ones with overall high accuracies.

Logistic Regression : Logistic regression is a statistical method for binary classification that models the probability of a binary outcome based on one or more predictor variables. It uses the logistic function to map predicted values to probabilities.

Random Forest : Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. It improves accuracy and controls overfitting by averaging the predictions from numerous trees, which reduces variance and increases robustness.

Support Vector Classification **(SVC)** : SVM is a supervised learning model that finds the optimal hyperplane to separate different classes in the feature space. It is effective in high-dimensional spaces and is robust against overfitting.

K-Nearest Neighbors (KNN) : KNN is a non-parametric, instance-based learning algorithm that classifies data points based on the majority class among its k-nearest neighbors in the feature space. It is simple, intuitive, and effective for smaller datasets, though it can be computationally expensive for large datasets.

Gaussian Naive Bayes (GNB) : GNB is a probabilistic classifier based on Bayes' theorem, assuming that features are independent given the class. It is computationally efficient and works well with small datasets.

Ensemble Voting Classifier : The voting classifier combines multiple models to improve prediction accuracy. In this study, a soft voting classifier is used, which averages the probabilities predicted by individual models.

Cross-validation : Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent dataset. It is used to prevent overfitting and ensure that the model performs well on unseen data by partitioning the data into training and testing sets multiple times.

Hyperparameter Tuning : Hyperparameter tuning involves selecting the optimal set of hyperparameters for a learning algorithm. Techniques like Grid Search or Random Search can be used to find the best parameters that improve model performance.

Performance Metrics : Different metrics such as accuracy, precision, recall, F1-score, confusion matrix and AUC of ROC are used to evaluate the performance of classification models. These metrics provide a detailed insight into the model's ability to classify correctly and handle imbalanced datasets.

METHODOLOGY STEPS

1. **Data collection :** The dataset, named 'hear_disease.csv', is sourced from a reliable repository from the Kaggle platform.
2. **Data preprocessing :** Includes encoding categorical variables, scaling numerical features, selecting relevant features using RFE and so on as mentioned in the above section.
3. **Exploratory data analysis (EDA) :** Aligned with preprocessing techniques, EDA is performed to understand the distribution of features, detect outliers, identify correlations and detect patterns and associations within variables.
4. **Model training :**
    - Dataset being split into training and testing sets to evaluate model performance.
    - Various machine learning classifiers being employed: Logistic Regression, Random Forest, Support Vector Classification (SVC), K-Nearest Neighbors (KNN), Decision Trees, and Naïve Bayes.
    - Hyper-parameter tuning performed using cross-validation to optimize model parameters and improve performance.
5. **Model evaluation :** Evaluating based on metrics such as accuracy, precision, recall, F1-score, confusion matrix and AUC of ROC.
6. **Ensemble model :** Checking for overall high accuracies and combining the best three trained models using a voting classifier.
7. **Prediction :** Using the trained models to predict the existence of heart disease for a new observation.

EXPLORATORY DATA ANALYSIS (EDA)

EDA is performed to gain insights into the dataset and understand the relationships between features. Main techniques include :

- **Descriptive statistics** : Summary statistics for numerical features are computed to understand the central tendency and dispersion.
- **Visualizations (Data distribution)** : Histograms and boxplots to visualize the distribution of numerical features.
- **Correlation analysis** : Heatmap to identify significant correlations (relationships) between features.
- **Feature analysis** : Analyzing the impact of each feature on the target variable.

DISTRIBUTION OF EXISTENCE IN HEART DISEASE (TARGET VARIABLE)

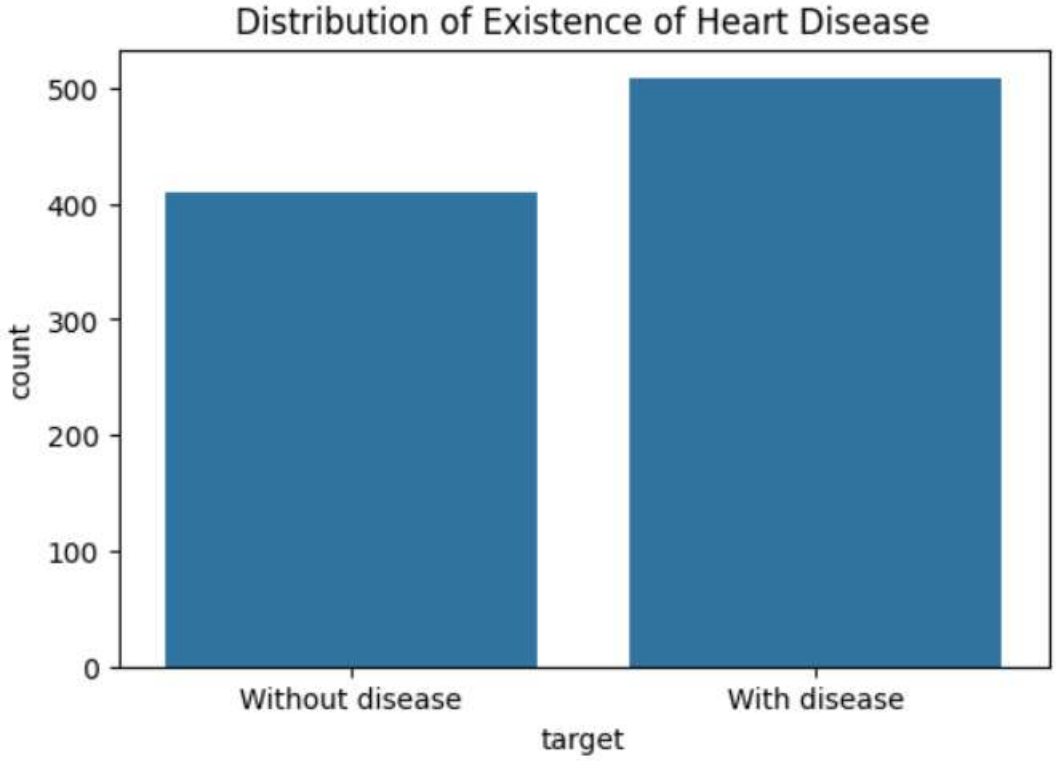


Figure 10 - Distribution of target

Distribution of the target variable shows that individuals taken into study have ones suffering from heart disease more than the ones who are not. However, heavy class imbalancement is not detected, which implies that it is not necessary to carry out techniques like over-sampling/ under-sampling/ SMOTE for the responsible variable.

CATEGORICAL VARIABLES' DISTRIBUTIONS

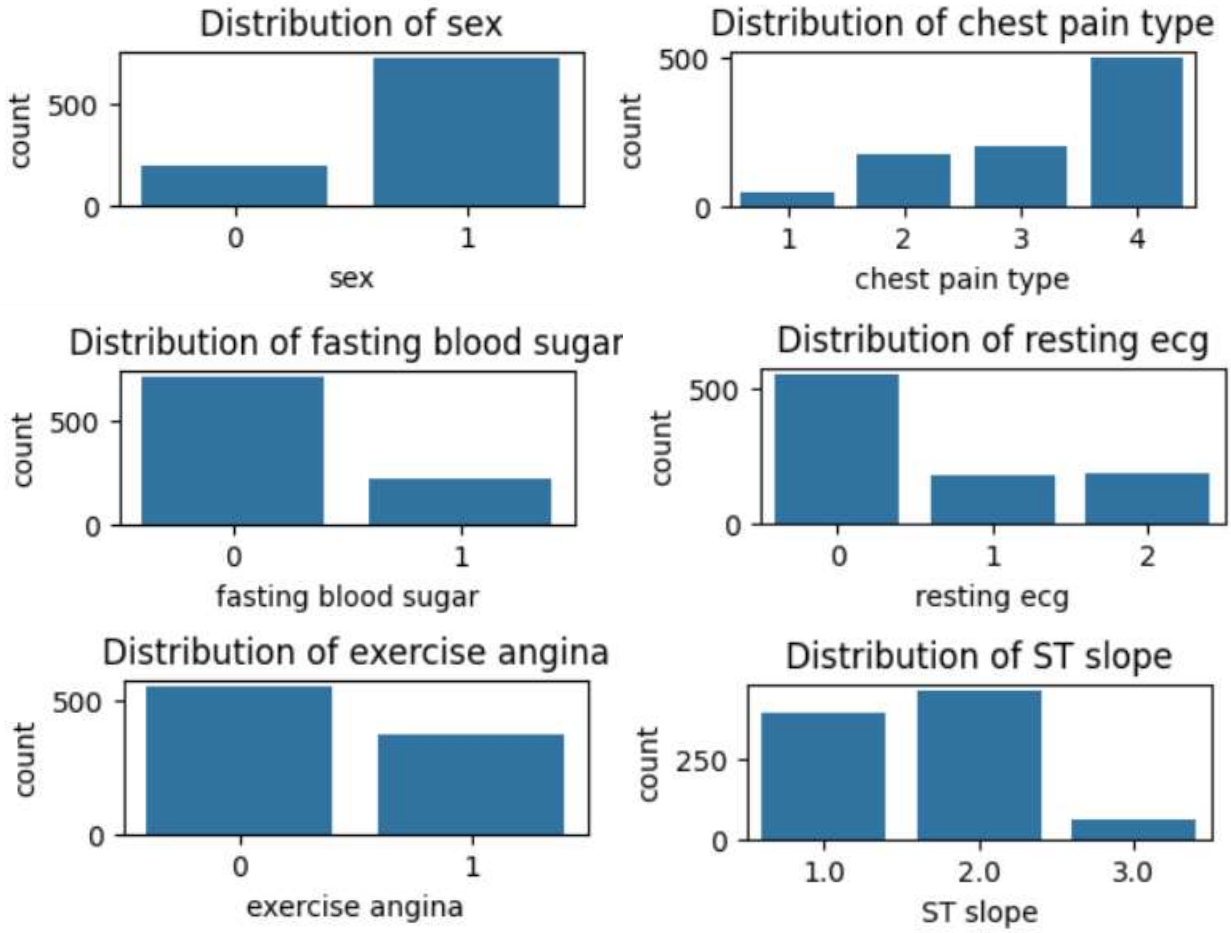


Figure 11 - Categorical column distributions

Considering these count plots of categorical variables in the study, it indicates that

- most individuals are males than females,
- suffered mostly from asymptotic pain than typical angina, atypical angina or non-anginal pain in the chest,
- have fasting blood sugar mostly less than 120 mg/dl,
- obtained normal resting ECG results rather than having ST-T wave abnormality or left ventricular hypertrophy,
- have negative exercise-induced angina more than positives, and
- have up-sloping or flat line of the peak exercise ST segment rather than down-sloping in.

AGE DISTRIBUTION BY SEX

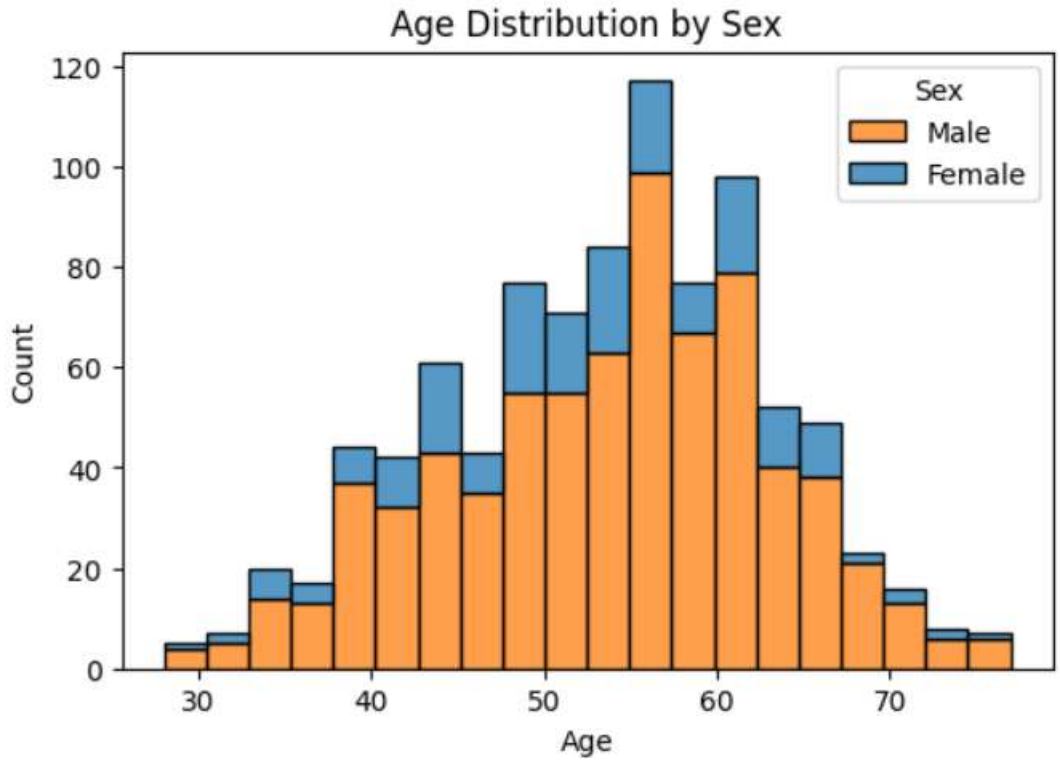
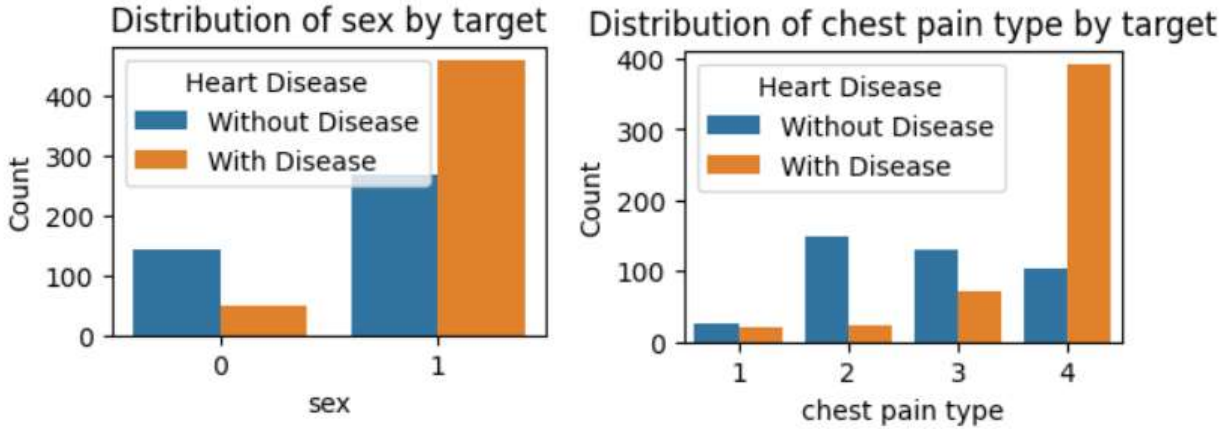


Figure 12 - Age by sex

The bar graph shows most of the individuals in the study either male or female are distributed around the age of 55 of years whereas ages of every individual ranges from 25 to 80. It is clear that the percentage of males in the study are higher than females as well. Nearly close fractions of women over men are distributed inside every age category, which indicates both males and females exist in each age category.

CATEGORICAL VARIABLE DISTRIBUTIONS BY EXISTENCE OF HEART DISEASE



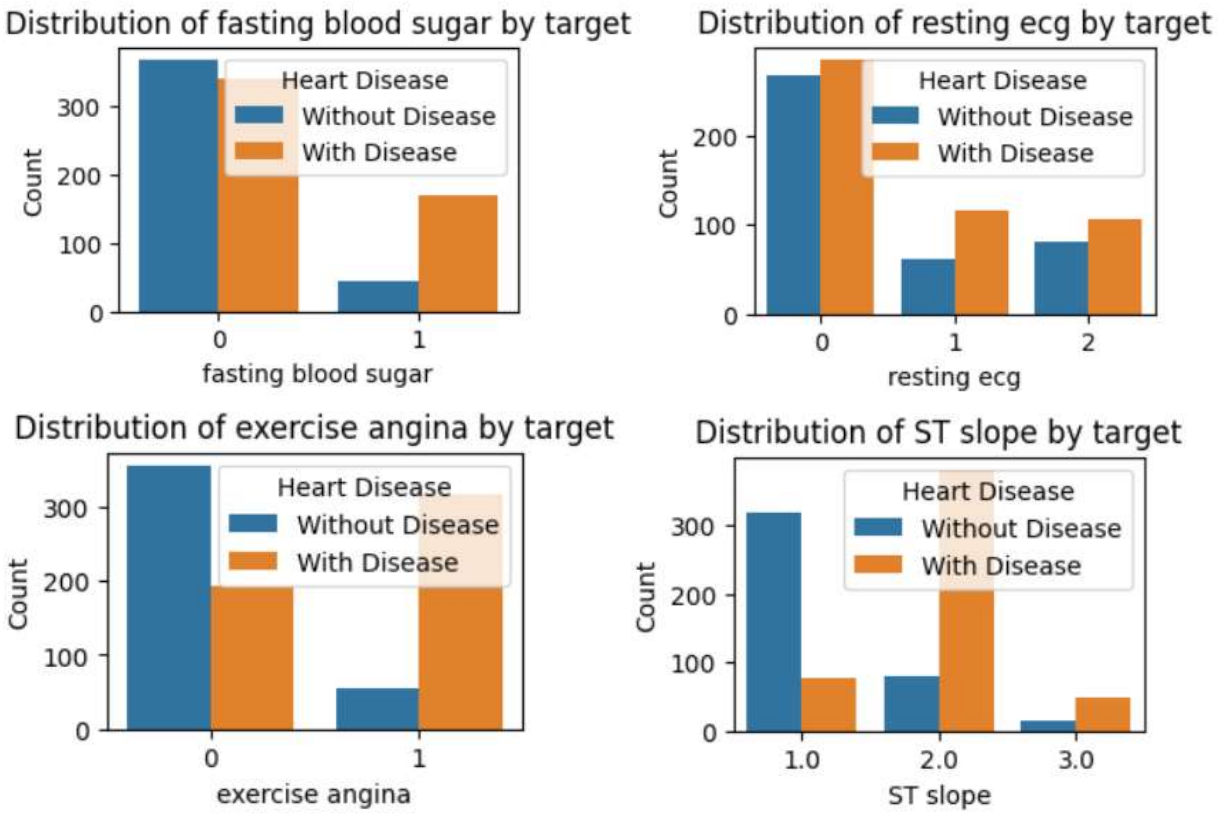


Figure 13 - Categorical distributions by target

Clear indications are,

- there is a high chance that males suffer from heart disease than females,
- individuals with asymptomatic typed chest pain might suffer heavily from heart disease more than others while individuals with atypical angina typed chest pain type would suffer less than others,
- individuals with fasting blood sugar greater than 120 mg/dl seems to have a high possibility to suffer from heart disease than others,
- individuals with any of the resting ECG results have roughly equal chances on being diagnosed with heart disease,
- individuals having positive exercise induced angina have high risk of heart disease than ones with negative results,
- individuals with no slope of the peak exercise ST segment is having high probability of risk while ones with upslope is having low risk of heart disease.

NUMERICAL VARIABLES' DISTRIBUTIONS

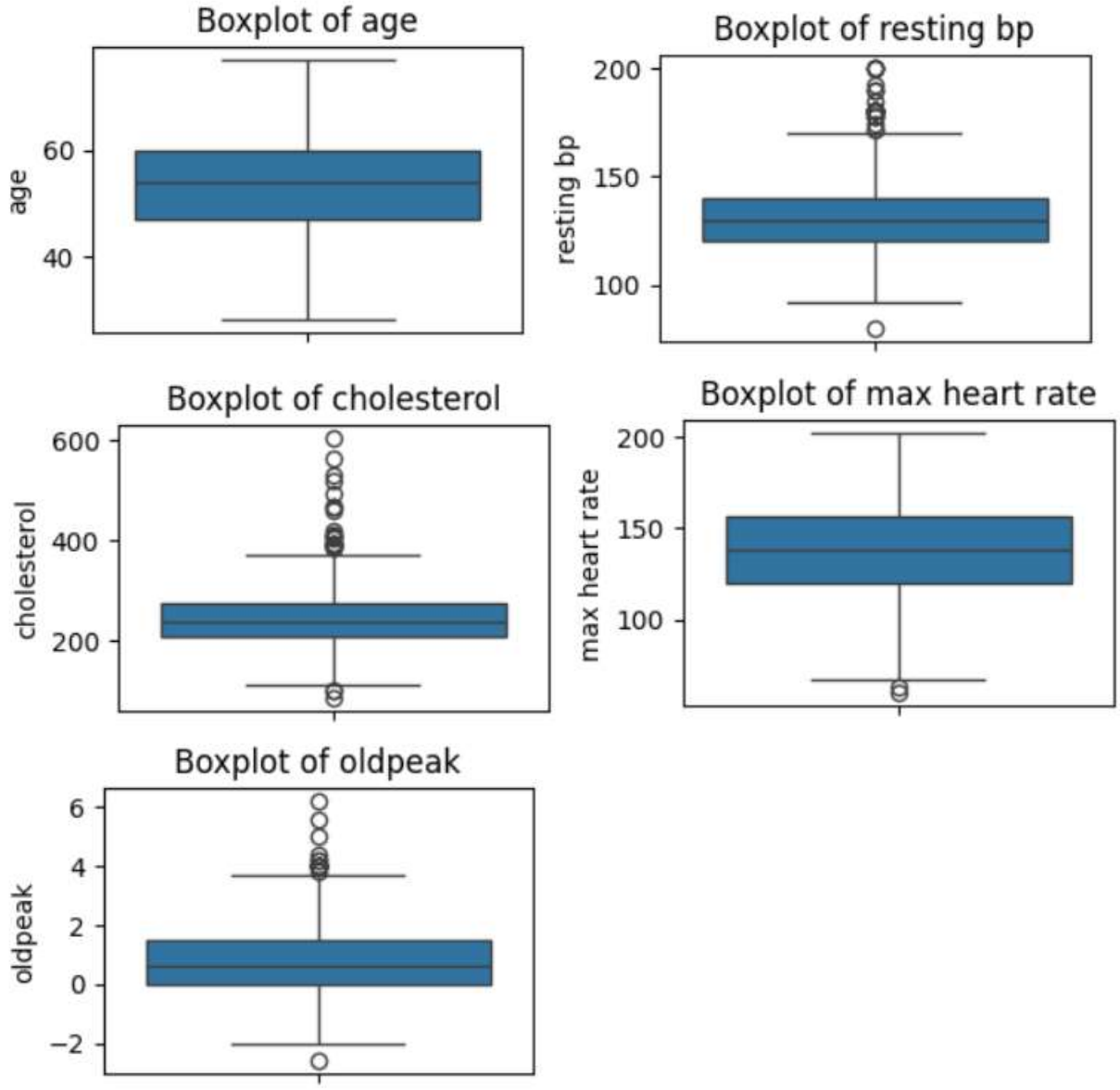


Figure 14 - Numerical column distributions

The boxplots indicate that,

- most of the individuals are in the age of 50-60 years,
- with resting blood pressure results mostly being scattered around the value of 140 mmHg.
- with cholesterol levels around 200 mg/dl but a limited number of individuals having levels of more than 400 mg/dl as well,
- with maximum heart rates mostly ranging around 120-150 per minute, which might increase up to 200 or drop until 60 too,
- with ST depression induced by exercise relative to rest being mostly between 0-2.

Also, since there exist a considerable number of outliers in the columns, ‘Resting blood pressure' and 'Serum cholesterol', it is better to impute their missing values with KNN imputer method.

DESCRIPTIVE STATISTICS OF NUMERICAL VARIABLES

	age	resting bp	cholesterol	max heart rate	oldpeak
count	918.000000	917.000000	746.000000	918.000000	918.000000
mean	53.510893	132.540894	244.635389	136.809368	0.887364
std	9.432617	17.999749	59.153524	25.460334	1.066570
min	28.000000	80.000000	85.000000	60.000000	-2.600000
25%	47.000000	120.000000	207.250000	120.000000	0.000000
50%	54.000000	130.000000	237.000000	138.000000	0.600000
75%	60.000000	140.000000	275.000000	156.000000	1.500000
max	77.000000	200.000000	603.000000	202.000000	6.200000

Figure 15 - Descriptive statistics

These summary statistics give insights on values mentioned above.

MAXIMUM HEART RATE BY EXERCISE INDUCED ANGINA

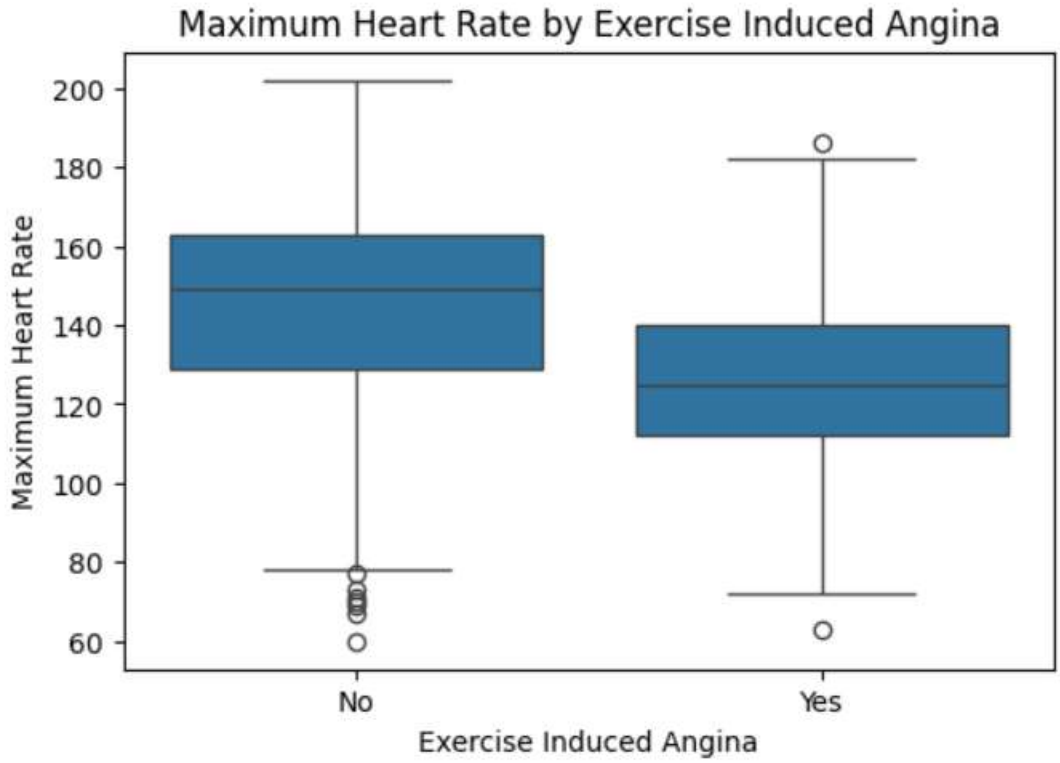


Figure 16 - Maximum heart rate by exercise induced angina

As per this result shows, maximum heart rate seems not to trigger that much when there is exercise-induced angina (pain in the chest when heart work harder) unlike when exercise-induced angina does not exist, in most of the individuals in which it might not be the case in the practical world.

DEPRESSION PEAK BY SLOPE OF THE PEAK EXERCISE ST SEGMENT

Figure 17 - Depression peak by slope of the peak exercise ST segment

In the violin plot, when considering about patients with the risk of disease, density curve for ST depression level for those with up-sloped trends seems to be highly scattered around -2 to 4 with zero depression levels in individuals being the peak value. For a flatted ST slope, the depression levels tend to have equal densities around -1 to 4 whereas for a downslope, it is around 0 to 6, which is greater than other kinds of ST slopes. This implies when ST slopes seem to be moving more towards downwards, the depression levels seems to be going up with the individuals who suffer from heart disease. However, the risk-free individuals seems to have very low densities when ST slopes are flatted or down sloped whereas there is a peak point of zero depression levels with up-sloped ST slopes. This does not seem to give better insights on individuals without heart disease.

PAIR PLOT OF NUMERICAL VARIABLES COLOURED BY HEART DISEASE EXISTENCE

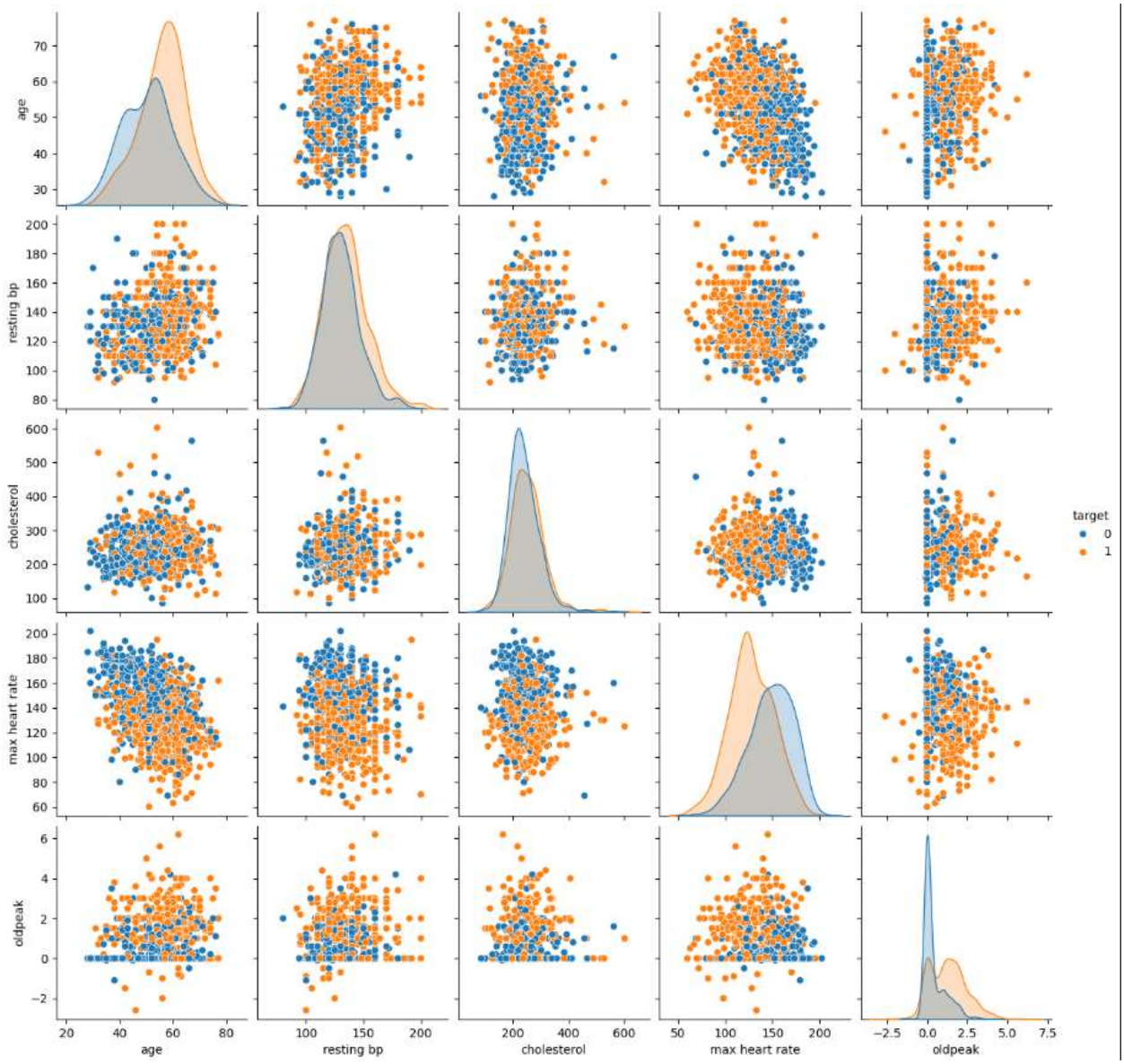


Figure 18 - Pair plots

Each numerical variable seems to have a slight to no association with another variable. These scatterplots do not give good insights as in associations between numerical variables, although ST depression induced by exercise relative to rest (old peak) seems to have more of higher values with other variables, compared to other plots.

The diagonal plots show each variable's distribution with the existence of the disease where the diseased seem to be growing in older individuals, also maximum heart rate seems to be lower in the diseased ones than risk-free individuals and there is enough possibility in higher depression levels on diseased ones as well.

Since there seems to be no unusual scenarios in the scatterplots, MinMaxScaler is used for scaling numerical features.

CORRELATION MATRIX ON NUMERICAL VARIABLES

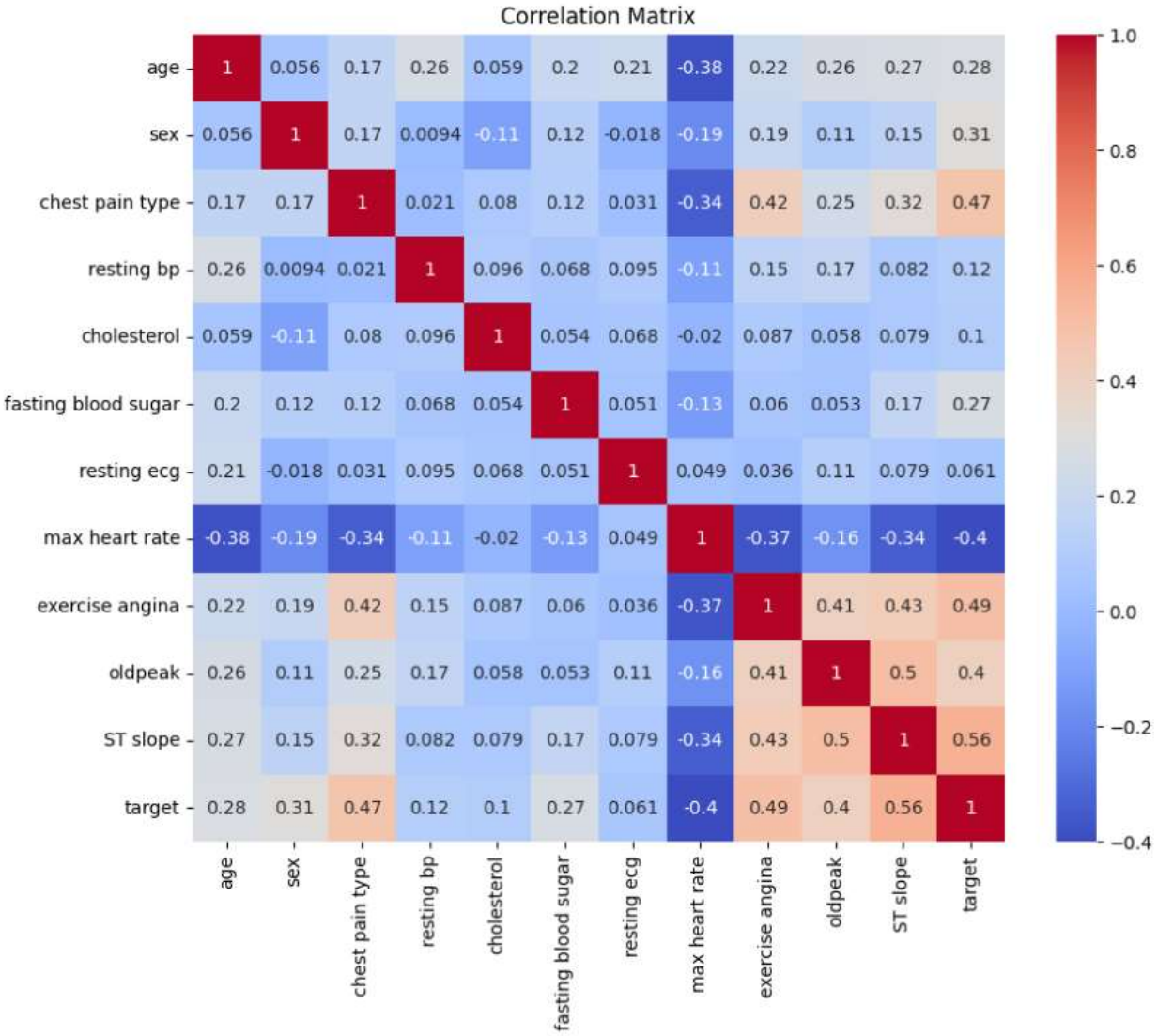


Figure 19 - Correlation matrix

It is clear that multicollinearity does not exist between any variable, (since all correlations are less than 0.6) in which removing columns is not needed to move on with the modeling process.

It is important to note that a considerable amount of multicollinearity exists between variables, 'ST slope' and 'target', which is satisfactory, then also between variables 'old peak' and 'ST slope' as well. They imply considerably low association between variables.

ADVANCED ANALYSIS

Advanced analysis involves the application of machine learning models to the preprocessed data. Key steps included:

- **Splitting data :** Splitting the dataset into training test sets for training and evaluation.
- **Model training :** Various classifiers are trained on the training data, and their performance was evaluated on the testing data.
- **Feature selection :** Using RFE for feature selection while training some models.
- **Hyper-parameter tuning with CV :** Grid Search cross validation technique is used to find the best optimized hyper-parameters for each model.
- **Prediction and evaluation on test data :** Model is used to predict on test data and evaluate by accuracy score, confusion matrix and classification report.
- **Ensemble model :** According to evaluation results, combining models with the top three performances in prediction using a soft voting classifier.
- **Model evaluation :** Assessing the models using accuracy, confusion matrix, and classification report.
- **Prediction :** A new observation is predicted from the new ensemble model.

The performance of individual models and the ensemble model is compared to identify the best predictive model for heart disease detection.

SPLITTING THE DATA

```
from sklearn.model_selection import train_test_split

X = data.drop('target', axis = 1)
y = data['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
                                                    random_state = 42)
```

Figure 20 - Splitting data

Splitting data into 80% on training and 20% on testing, keeping train data for training.

MODEL TRAINING WITH FEATURE SELECTION

Some models such as LR, RF, SVM and Decision Tree enables feature selection for the encoded variables set while other models (KNN and GNB) do not. Each model is optimized as such the input of the selected features improves performance of the particular model. The package 'Recursive Feature Elimination (RFE)' aids in feature selection process.

```
from sklearn.feature_selection import RFE
```

Figure 21 - Feature selection with RFE package

1. **Logistic Regression model :** 10 optimum features

```
Selected features: Index(['age', 'max heart rate', 'oldpeak', 'sex_0', 'sex_1',
       'chest pain type_4', 'fasting blood sugar_0', 'exercise angina_0',
       'ST slope_1', 'ST slope_2'],
      dtype='object')
LogisticRegression(random_state=42)
```

Figure 22 - LR training

2. **Random Forest Classifier model :** 15 optimum features

```
Selected features: Index(['age', 'max heart rate', 'oldpeak', 'sex_0', 'sex_1',
       'chest pain type_4', 'fasting blood sugar_0', 'exercise angina_0',
       'ST slope_1', 'ST slope_2'],
      dtype='object')
RandomForestClassifier(random_state=42)
```

Figure 23 - RF training

3. **Support Vector Machine (SVM) model :** 10 optimum features

```
Selected features: Index(['age', 'max heart rate', 'oldpeak', 'sex_0', 'sex_1',
       'chest pain type_4', 'fasting blood sugar_0', 'exercise angina_0',
       'ST slope_1', 'ST slope_2'],
      dtype='object')
SVC(kernel='linear', probability=True, random_state=42)
```

Figure 24 - SVM training

4. **K Nearest Neighbors (KNN) Classifier Model :** Feature selection is inaccessible

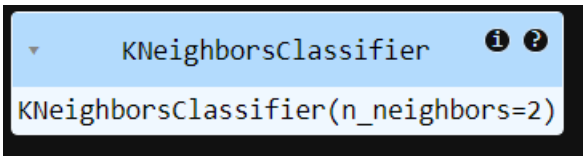


Figure 25 - KNN training

5. **Gaussian Naive Bayes Model :** Feature selection is inaccessible

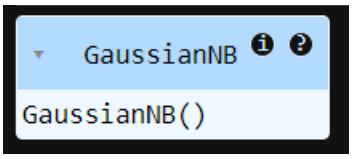


Figure 26 - GNB training

6. **Decision Tree Model :** 10 optimum features

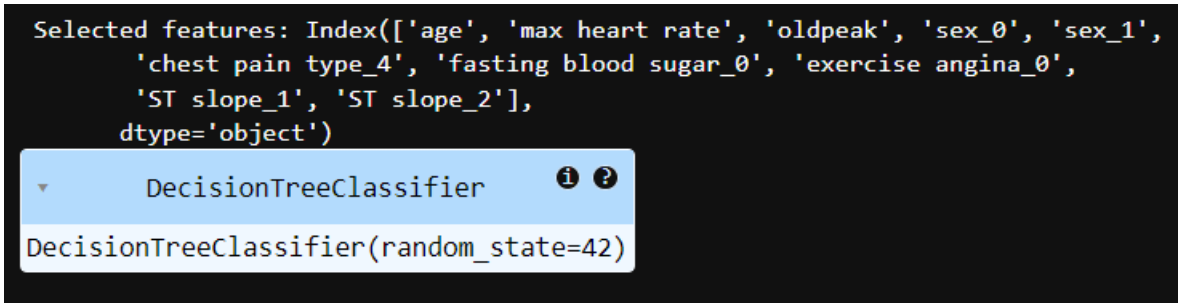


Figure 27 - Decision Tree training

HYPER-PARAMETER TUNING WITH CV

Each model having a set of hyper-parameters so that by using cross validation, they are tuned to get the best estimators.

1. **Logistic Regression model :**

```
LR best parameters:  {'C': 0.9, 'max_iter': 1000, 'solver': 'newton-cg'}
```

Figure 28 - LR best parameters

2. **Random Forest Classifier model :**

```
RF best parameters:  {'criterion': 'gini', 'max_depth': None, 'max_features': 'sqrt', 'min_
samples_leaf': 2, 'min_samples_split': 14, 'n_estimators': 51}
```

Figure 29 - RF best parameters

3. **Support Vector Machine (SVM) model :**

```
SVM best parameters:  {'C': 10, 'kernel': 'poly'}
```

Figure 30 - SVM best parameters

4. **K Nearest Neighbors (KNN) Classifier Model :**

```
KNN best parameters:  {'algorithm': 'auto', 'leaf_size': 1, 'n_neighbors': 17, 'p': 1, 'wei
ghts': 'uniform'}
```

Figure 31 - KNN best parameters

5. **Gaussian Naive Bayes Model :**

```
GNB best parameters:  {'var_smoothing': 1e-323}
```

Figure 32 - GNB best parameters

6. **Decision Tree Model :**

```
Decision Tree best parameters:  {'class_weight': 'balanced', 'criterion': 'gini', 'max_dept
h': 4, 'max_features': 'sqrt', 'min_samples_leaf': 3, 'min_samples_split': 6}
```

Figure 33 - Decision Tree best parameters

PREDICTION AND EVALUATION ON TEST DATA

With the best parameters, the test set is predicted for 'y', then the performances are evaluated with the actual 'target' values in the data.

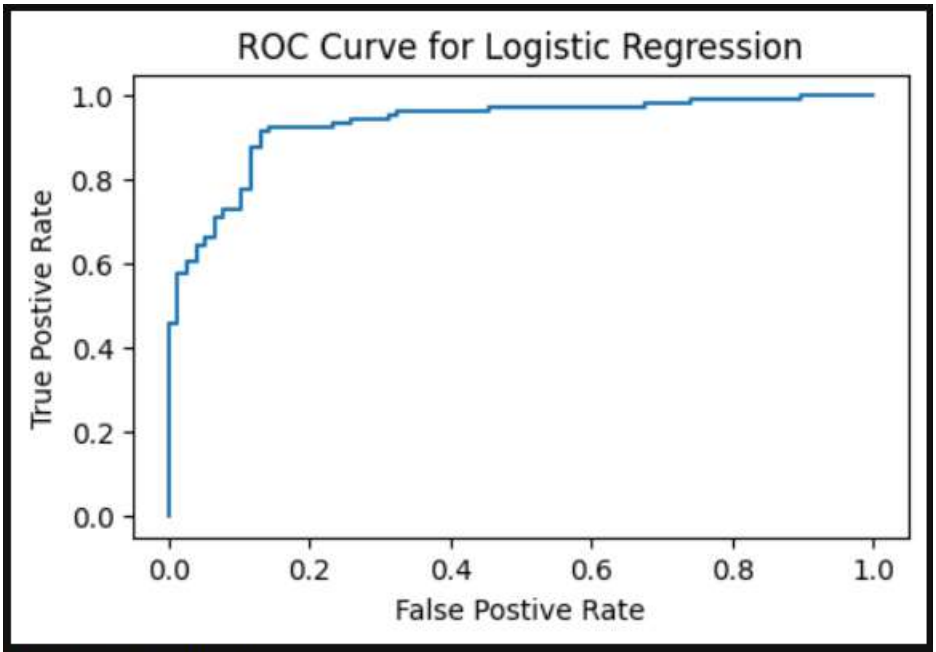
Also, in order to check for model evaluation further, (to check if the model is of good fit for prediction purposes) by taking the prediction probabilities and their false positive rates and true positive rates with the comparison of actual 'target' values, both the ROC curve (Receiver Operating Characteristic curve) and its AUC (Area Under Curve) metrics are obtained.

1. **Logistic Regression model :**

```
Logistic Regression Model:
Accuracy: 86.95652173913044
Confusion Matrix:
 [[68  9]
 [15 92]]
Classification Report:              precision    recall  f1-score   support

           0       0.82      0.88      0.85        77
           1       0.91      0.86      0.88       107

    accuracy                           0.87       184
   macro avg       0.87      0.87      0.87       184
weighted avg       0.87      0.87      0.87       184
```




```
93.13023425172958
```

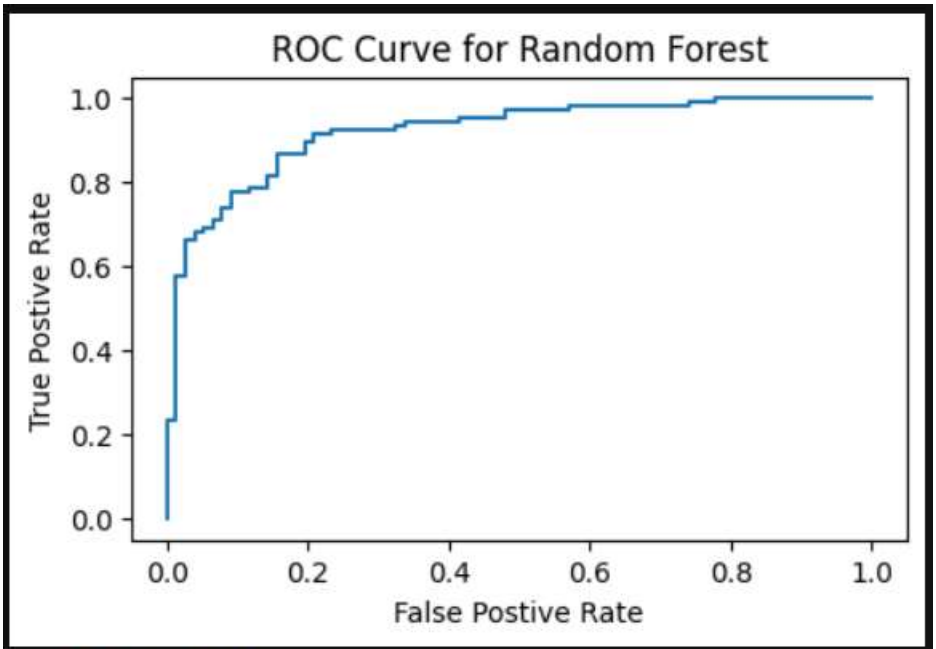
Figure 34 - LR evaluation

2. Random Forest Classifier model :

```
Random Forest Classifier model:
Accuracy: 83.69565217391305
Confusion Matrix:
 [[65 12]
 [18 89]]
Classification Report:              precision    recall  f1-score   support

           0       0.78      0.84      0.81        77
           1       0.88      0.83      0.86       107

    accuracy                           0.84       184
   macro avg       0.83      0.84      0.83       184
weighted avg       0.84      0.84      0.84       184
```




```
92.13496783590242
```

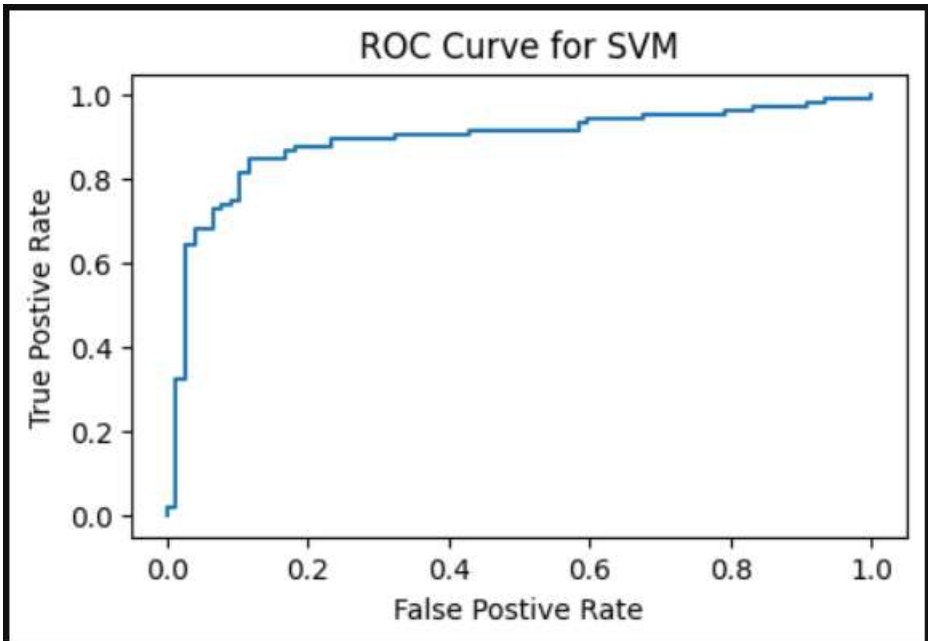
Figure 35 - RF evaluation

3. Support Vector Machine (SVM) model :

```
Support Vector Machine model:
Accuracy: 85.86956521739131
Confusion Matrix:
 [[68  9]
 [17 90]]
Classification Report:              precision    recall  f1-score   support

           0       0.80      0.88      0.84        77
           1       0.91      0.84      0.87       107

    accuracy                           0.86       184
   macro avg       0.85      0.86      0.86       184
weighted avg       0.86      0.86      0.86       184
```




```
88.96710765869645
```

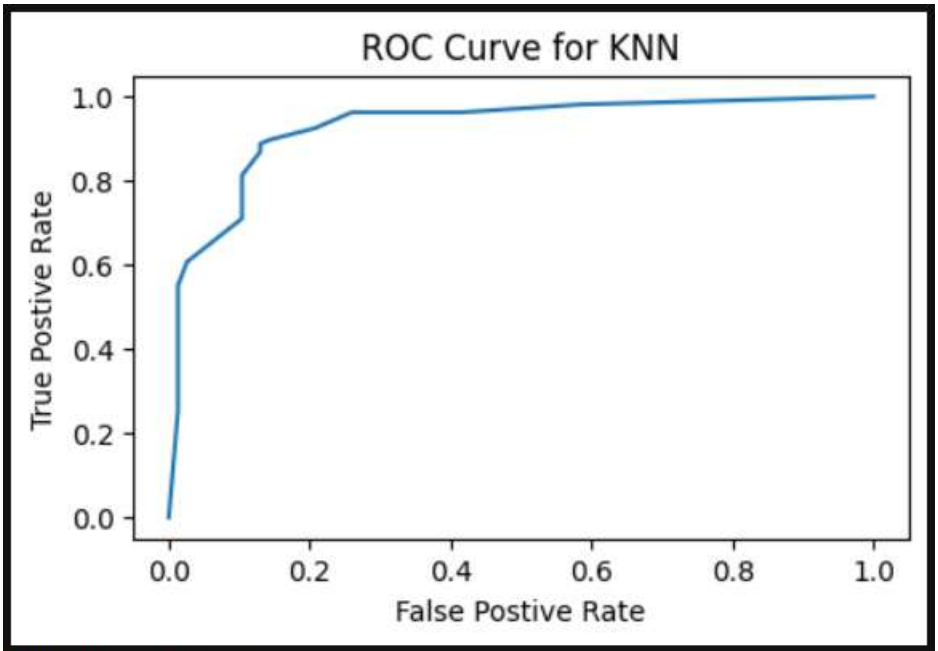
Figure 36 - SVM evaluation

4. K Nearest Neighbors (KNN) Classifier Model :

```
K Nearest Neighbours Model:
Accuracy: 77.17391304347827
Confusion Matrix:
 [[74  3]
 [39 68]]
Classification Report:              precision    recall  f1-score   support

           0       0.65      0.96      0.78        77
           1       0.96      0.64      0.76       107

    accuracy                           0.77       184
   macro avg       0.81      0.80      0.77       184
weighted avg       0.83      0.77      0.77       184
```




```
92.79645588056802
```

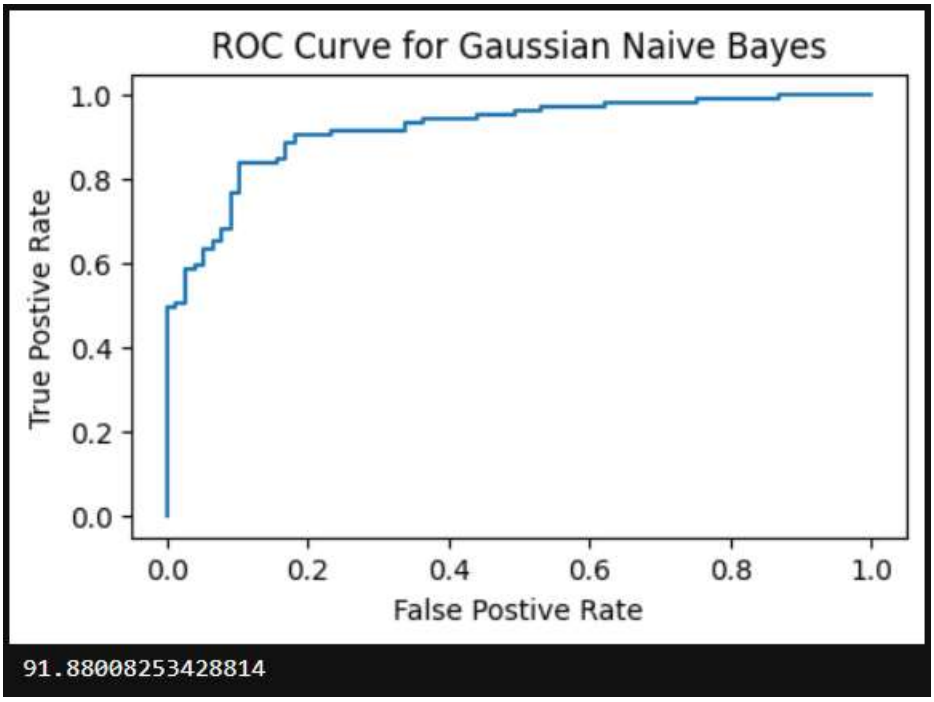
Figure 37 - KNN evaluation

5. **Gaussian Naive Bayes Model :**

```
Gaussian Naive Bayes Model:
Accuracy: 85.32608695652173
Confusion Matrix:
 [[67 10]
 [17 90]]
Classification Report:              precision    recall  f1-score   support

           0       0.80      0.87      0.83        77
           1       0.90      0.84      0.87       107

    accuracy                           0.85       184
   macro avg       0.85      0.86      0.85       184
weighted avg       0.86      0.85      0.85       184
```




```
91.88008253428814
```

Figure 38 - GNB evaluation

6. Decision Tree Model :

```
Decision Tree Model:
Accuracy: 76.63043478260869
Confusion Matrix:
 [[66 11]
 [32 75]]
Classification Report:              precision    recall  f1-score   support

           0       0.67      0.86      0.75        77
           1       0.87      0.70      0.78       107

    accuracy                           0.77       184
   macro avg       0.77      0.78      0.77       184
weighted avg       0.79      0.77      0.77       184
```

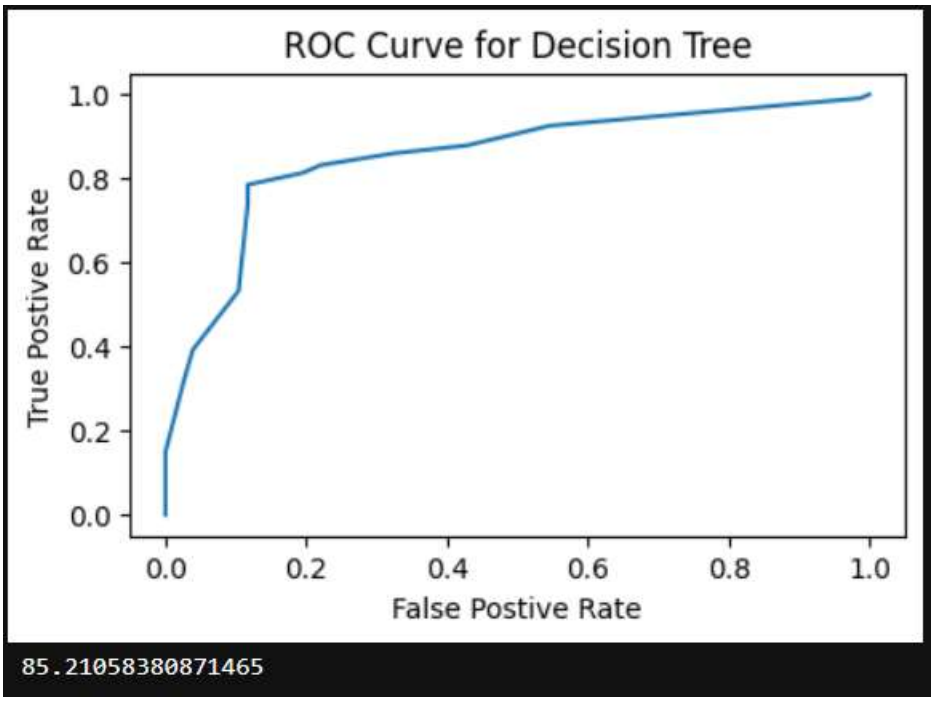


Figure 39 - Decision Tree evaluation

Conclusion from the predictions :

According to the highest accuracy and other evaluation metrics like precision, recall, F1-score, ROC curve and AUC value, the most suitable (3 best) models would be LR, SVM and Gaussian Naive Bayes. (with around 85% of accuracy and AUC metrics being more than 88% which is of better fit for prediction)

Finally, these 3 models are combined for improved performance.

ENSEMBLE MODEL WITH EVALUATION

Combining predictions of LR, SVM and GNB models is performed using an ensemble method, called 'VotingClassifier'.

```
from sklearn.ensemble import VotingClassifier
```

Figure 40 - Voting classifier import

It is clear that combining models stabilizes predictions, reducing overfitting or other anomalies when a model itself is used for prediction of a new observation.

```
Voting Classifier Model:
Accuracy: 85.32608695652173
Confusion Matrix:
 [[67 10]
 [17 90]]
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.87      0.83        77
           1       0.90      0.84      0.87       107

    accuracy                           0.85       184
   macro avg       0.85      0.86      0.85       184
weighted avg       0.86      0.85      0.85       184
```

Figure 41 - Ensemble model evaluation

NEW OBSERVATION PREDICTION

Using the ensemble model, a new synthetic observation is predicted.

```
#Create new observation
feature_names = ['age', 'sex', 'chest pain type', 'resting bp', 'cholesterol',
                 'fasting blood sugar', 'resting ecg', 'max heart rate',
                 'exercise angina', 'oldpeak', 'ST slope']
new_obs = np.array([[55, 1, 2, 187, 215, 1, 2, 125, 0, 0.4, 1]])

#Convert new observation to DataFrame
new_obs_data = pd.DataFrame(new_obs, columns = feature_names)
```

Figure 42 - New observation array

After a new observation array is inserted, it must go through MinMax scaling and OneHotEncoder method just before the prediction processes in order to ensure consistency of results.

```
#Perform MinMax Scaling as before
new_obs_data[numerical_vars] = scaler.transform(new_obs_data[numerical_vars])

#Encode categorical variables as before
new_obs_encoded = pd.get_dummies(new_obs_data,
                                 columns = ['sex', 'chest pain type',
                                            'fasting blood sugar', 'resting ecg',
                                            'exercise angina', 'ST slope'])

#Align the new observation with the training data columns
new_obs_encoded = new_obs_encoded.reindex(columns = X_train.columns, fill_value = 0)

#Make predicting using LR, SVM and GNB models seperately
pred_lr = best_lr.predict(new_obs_encoded)
print("LR Prediction: ", pred_lr[0])
pred_svm = best_svm.predict(new_obs_encoded)
print("SVM Prediction: ", pred_svm[0])
pred_gnb = best_gnb.predict(new_obs_encoded)
print("GNB Prediction: ", pred_gnb[0])

#Predicting using ensemble model
predict_voting = voting_clf.predict(new_obs_encoded)
print("Ensemble Model Prediction ", predict_voting[0])
```

```
LR Prediction:  1
SVM Prediction:  1
GNB Prediction:  1
Ensemble Model Prediction  1
```

Figure 43 - New observation prediction

Both the individual models itself and the ensemble model predicts that the new individual with given factors and diagnosis is a patient with heart disease.

GENERAL DISCUSSION AND CONCLUSION

In order to find the best model for an accurate prediction, this study used a variety of machine learning techniques to predict heart disease. The analysis initiated with comprehensive data preprocessing, which included scaling numerical characteristics, encoding categorical variables, and handling missing values. Important patterns and connections were found through exploratory data analysis, which aided in directing the feature selection procedure. Recursive Feature Elimination (RFE) was used to identify optimal features for models Logistic Regression, Random Forest, Support Vector Machine and Decision Tree. To enhance model performance, cross-validation was used for hyper-parameter adjustment. After evaluating these models using the test data, it was revealed that LR, SVM and GNB had the best accuracy (around 85%) as well as the rest of evaluation metrics being the best. An ensemble method called VotingClassifier was used to combine the predictions of LR, SVM, and GNB in order to further improve performance. This approach decreased overfitting while simultaneously stabilizing predictions. The ensemble model demonstrated improved performance than the individual models in heart disease prediction, offering a reliable instrument for early detection and treatment of heart disease. Ultimately, the model's consistency and dependability in practical applications were confirmed by predicting on a new synthetic observation.

Conclusion:

- ✓ The dataset was suitably prepped for machine learning modelling using the data preprocessing procedures.
- ✓ Significant patterns were found and the feature selection process was guided by exploratory data analysis.
- ✓ Logistic Regression, Support Vector Machine, and Gaussian Naive Bayes models were the top performers in terms of accuracy and other metrics.
- ✓ Cross-validation and hyper-parameter adjustment significantly enhanced the models' performance.
- ✓ The best predictive performance and least amount of overfitting were achieved by the ensemble approach, VotingClassifier, which combined the models, LR, SVM, and GNB.
- ✓ The ensemble model was validated and found reliable for predicting heart disease of individuals in new observations.

REFERENCES

Dataset selected : https://www.kaggle.com/datasets/mexwell/heart-disease-dataset/code?datasetId=4755824&sortBy=voteCount

For methodology selection : https://www.researchgate.net/profile/V-V-Ramalingam/publication/325116774_Heart_disease_prediction_using_machine_learning_techniques_A_survey/links/5d48560a299bf1995b68266f/Heart-disease-prediction-using-machine-learning-techniques-A-survey.pdf