HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS

A course-project Report

Submitted by

SOURABH GHAGHRE

in partial fulfillment of the requirements for the award of the degree of

Master of Technology

in

Mechanical Engineering

(Industrial Engineering & Management)

Department of Mechanical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

NIT CAMPUS PO, CALICUT

KERALA, INDIA 673601

May 2023

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: NITC **Signature:**

Date: **Name:**

Roll No.:

ABSTRACT

This abstract presents a study on predicting heart diseases using machine learning and gradient boosting methods. The study utilizes three popular gradient boosting algorithms, namely LightGBM, XGBoost, and CatBoost, to train predictive models. The results indicate that LightGBM outperforms the other two algorithms in terms of predictive accuracy. The study highlights the potential of using advanced machine learning techniques for early detection of heart diseases, which can lead to timely intervention and improved health outcomes.

Heart disease, also known as cardiovascular disease, is a leading cause of death worldwide. It refers to a range of conditions that affect the heart and blood vessels, such as coronary artery disease, heart failure, and arrhythmias. Risk factors for heart disease include high blood pressure, high cholesterol, smoking, diabetes, obesity, and a family history of the condition.Early prediction and diagnosis of heart disease can improve patient outcomes and prevent complications. Machine learning techniques, particularly gradient boosting algorithms, can be used to analyze large amounts of patient data and identify patterns that may indicate an increased risk of heart disease.Overall, this study highlights the potential of machine learning methods for predicting heart disease and improving patient care.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 HEART DISEASE:

Heart disease, also known as cardiovascular disease, encompasses various conditions and illnesses that impact the heart and circulatory system. This ailment is a significant contributor to disability worldwide, and because the heart is a crucial organ, its complications can affect other parts of the body. Heart disease takes on multiple forms, with common examples including blockage or narrowing of coronary arteries, valve malfunctions, heart enlargement, and other issues that can result in heart failure and heart attack.

Key facts according to WHO for heart diseases presented in bullet points:

1. Cardiovascular diseases are the leading cause of death globally
2. An estimated 17.9 million people die from cardiovascular diseases each year
3. This accounts for approximately 31% of all deaths worldwide
4. 80% of premature heart disease and stroke can be prevented through lifestyle changes such as a healthy diet, regular physical activity, and avoiding tobacco use

1.2 MACHINE LEARNING

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible Heart Diseases in people using Machine Learning algorithms.

1.3 OBJECTIVE:

This project has two main objectives:

1. Explore the heart disease dataset using exploratory data analysis (EDA)
2. Exercise with classification algorithms for prediction (modeling)

CHAPTER 2
METHODOLOGY AND DESIGN

2.1 DATASET AND COLLECTION METHOD

A collection of data points that a computer may use to analyze and anticipate a situation as a whole. Internet information was gathered for the Kaggle.com website. The test data set used in this study comprises 303 rows as well as 14 categories, which have been trained to deliver the most accurate prediction outcomes. A brief introduction of the dataset is shown below:

1. age: age in years
2. sex: sex
    - 1 = male
    - 0 = female
3. cp: chest pain type
    - Value 0: typical angina
    - Value 1: atypical angina
    - Value 2: non-anginal pain
    - Value 3: asymptomatic
4. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholestoral in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl)
    - 1 = true;
    - 0 = false
7. restecg: resting electrocardiographic 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
8. thalach: maximum heart rate achieved

9. exang: exercise induced angina
    - 1 = yes
    - 0 = no
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment
    - Value 0: upsloping
    - Value 1: flat
    - Value 2: downsloping
12. ca: number of major vessels (0-3) colored by flourosopy
13. thal:
    - 0 = error (in the original dataset 0 maps to NaN's)
    - 1 = fixed defect
    - 2 = normal
    - 3 = reversable defect
14. target (the lable):
    - 0 = no disease,
    - 1 = diseases

2.2 METHODOLOGY:

The methodology for building machine learning models involves several steps:

1. Problem Definition: The problem to be solved and the goals to be achieved needs to be defined clearly.
2. Data Collection: Relevant data needs to be collected to for training and testing the model as the quality and quantity of data will have an impact on the model's performance.
3. Data Preparation: Clean and preprocess the data, which includes tasks such as handling missing values, handling outliers, normalizing data, and transforming data.
4. Feature Engineering: Select or create features that will be used to train the model. Feature engineering is a crucial step in building accurate and efficient models.
5. Model Selection: Choose the appropriate machine learning algorithm for the problem you want to solve.
6. Training of Model: The chosen model should be trained on the training data in order to grasp the underlying patterns and relationships that exist within the data.

7. Model Evaluation: Evaluate the performance of the model on a separate validation or test dataset. This helps to estimate the model's accuracy and identify potential issues.
8. Model Deployment: Once the model is trained and validated, deploy it to a production environment where it can be used to Perform predictions on novel data.

Monitoring and Maintenance: Continuously check the model's behavior and update the model as needed to maintain its accuracy and effectiveness over time. The flow diagram of Methodology is shown in Figure

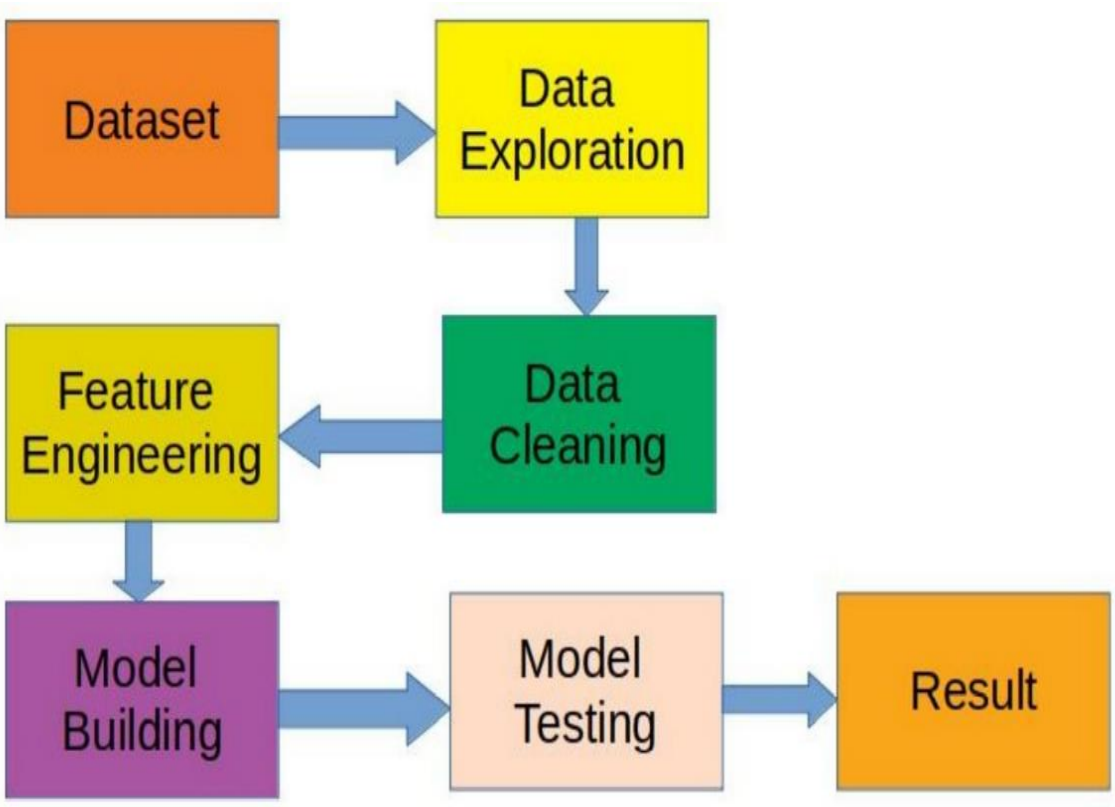


Figure 2.1 steps followed for obtaining result

CHAPTER 3
DATA PREPROCESSING

3.1 DROP FAULTY DATA

Based on the investigation will drop 7 rows.

```
data = data[data['ca'] < 4] #drop the wrong ca values
data = data[data['thal'] > 0] # drop the wong thal value
print(f'The length of the data now is {len(data)} instead of 303!')
```

```
The length of the data now is 296 instead of 303!
```

3.2 RENAME COLUMNS FOR THE SAKE OF CLARITY

- The feature names in the dataset are abbreviated and hard to understand their meaning. A full medical/technical name is hard enough to understand for most of us let alone their short form. So to make them a little bit easier to read we will, here under, change the column names of the data frame using information from the UCL data repository.
- Also replace the coded categories (0, 1, 2,..) to their medical meaning ('atypical angina', 'typical angina', etc. for example)

```
data = data.rename(
    columns = {'cp':'chest_pain_type',
               'trestbps':'resting_blood_pressure',
               'chol': 'cholesterol',
               'fbs': 'fasting_blood_sugar',
               'restecg' : 'resting_electrocardiogram',
               'thalach': 'max_heart_rate_achieved',
               'exang': 'exercise_induced_angina',
               'oldpeak': 'st_depression',
               'slope': 'st_slope',
               'ca':'num_major_vessels',
               'thal': 'thalassemia'},
    errors="raise")
```

CHAPTER 4

DATA VISUALIZATION

4.1 TARGET VARIABLE VISUALIZATION

It is observed that the target is fairly balanced with ~46% with no heart disease and ~54% with heart disease. So no need to worry about target imbalance. Target variable distribution shown below in figure 4.1

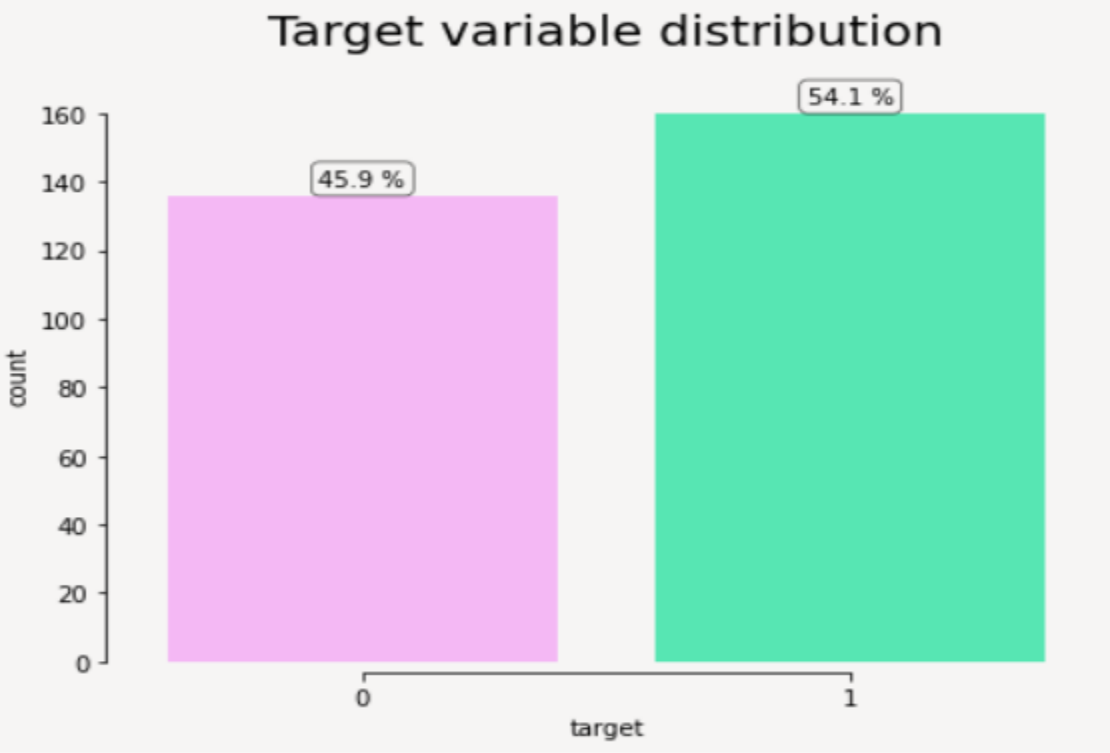


Figure 4.1 Target variable distribution

4.2 NUMERICAL FEATURE DATA VISUALIZATION:

4.2.1 Distribution Density plots- Distribution of numerical features shown in figure 4.2

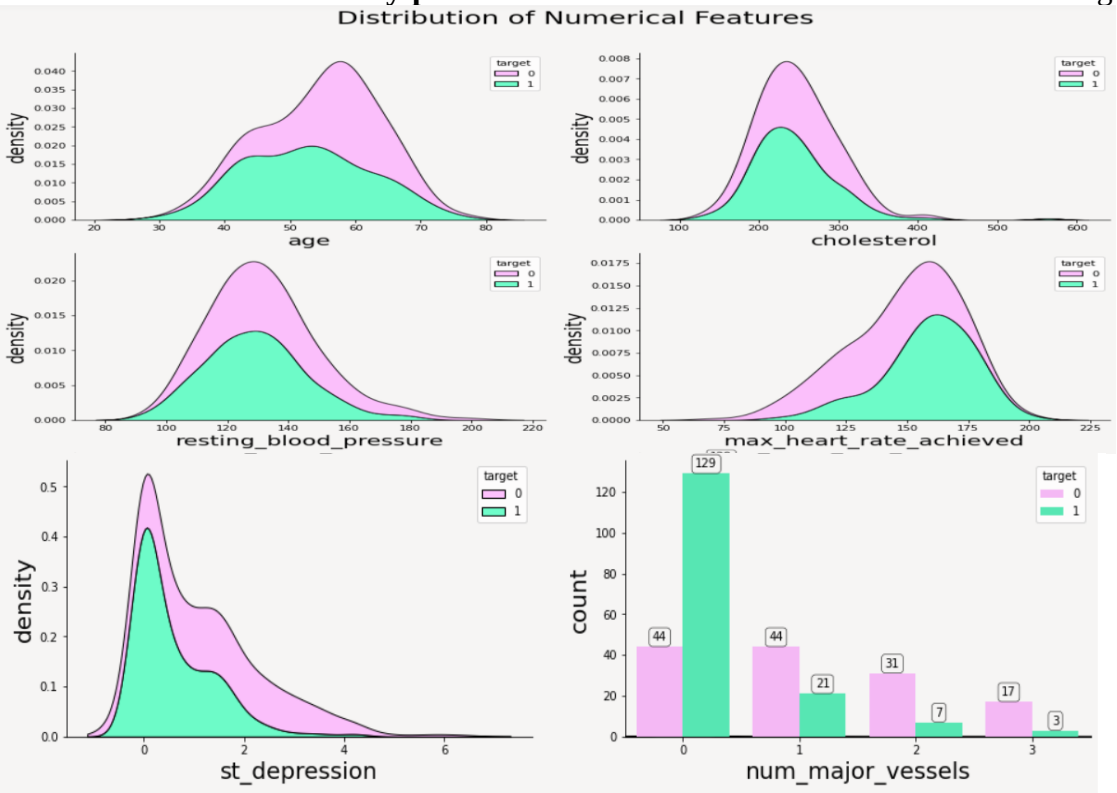
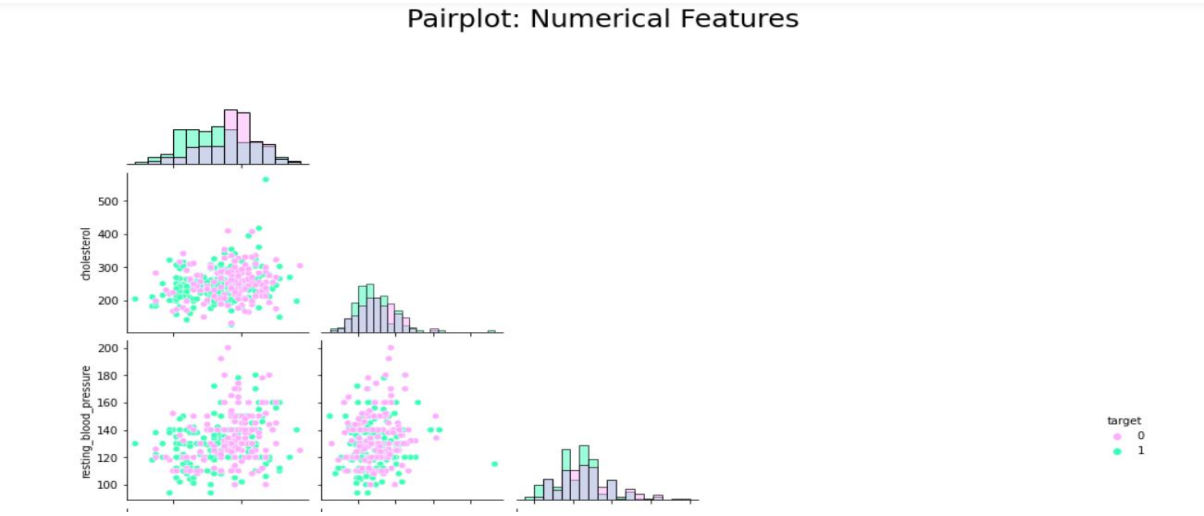


Figure 4.2 Distribution of numerical features

4.2.2 Pair-plots: Pair plot of numerical data shown below in figure 4.3



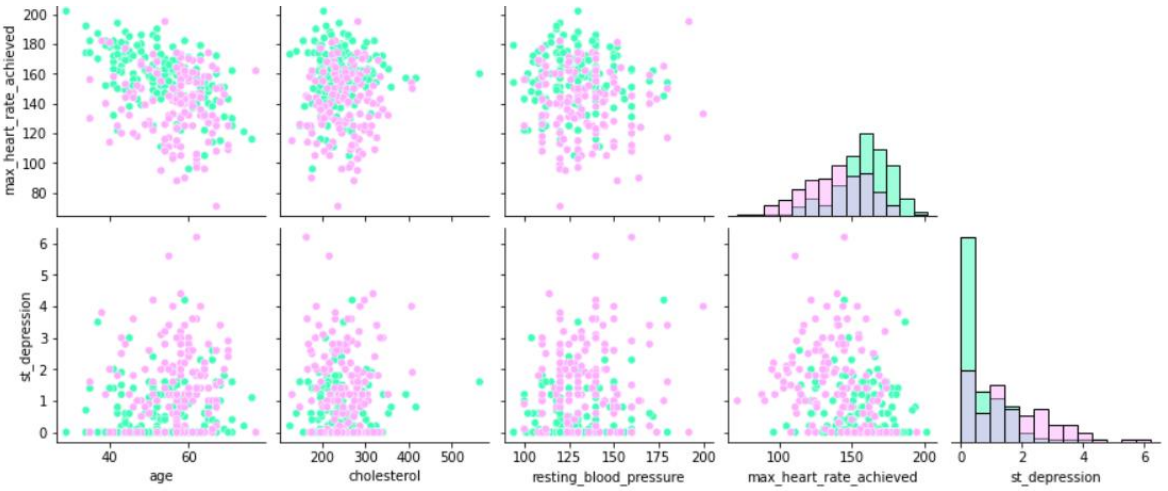


figure 4.3 Pair plot of numerical data

4.3 CATEGORICAL FEATURE DATA VISUALIZATION

4.3.1 Distribution Count plots: Distribution of categorical features shown below in figure

Figure 4.4 Distribution of categorical feature

CHAPTER 5

EXPLORATORY DATA ANALYSIS

5.1 Correlation Heatmaps

Correlation heatmap is a useful tool to graphyically represent how two features are related to eachother. Depending upon the data types of the features, project need to use the appropriate correlation coefficient calculation methods. Examples are pearson's correlation coefficient, point biserial correlation, cramers'V correlation and etc.

5.1.1 Pearson's Correlation

- The Pearson correlation coefficient — is a measure of linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations; thus it is essentially a normalised measurement of the covariance, such that the result always has a value between −1 and 1.Shown below figure5.1

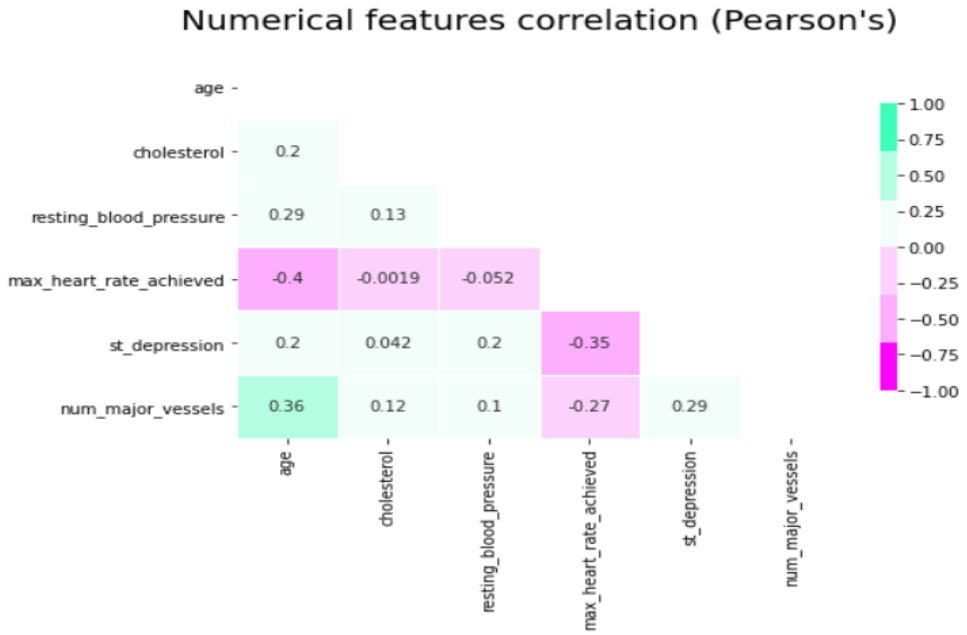


Figure 5.1 pearson’s correlation heatmap

5.1.2 Point Biserial Correlation

- A point-biserial correlation is used to measure the strength and direction of the association that exists between **one continuous variable and one dichotomous variable**. It is a special case of the Pearson's product-moment correlation, which is applied when there have two continuous variables, whereas in this case one of the variables is measured on a dichotomous scale , correlation heatmap shown below 5.2

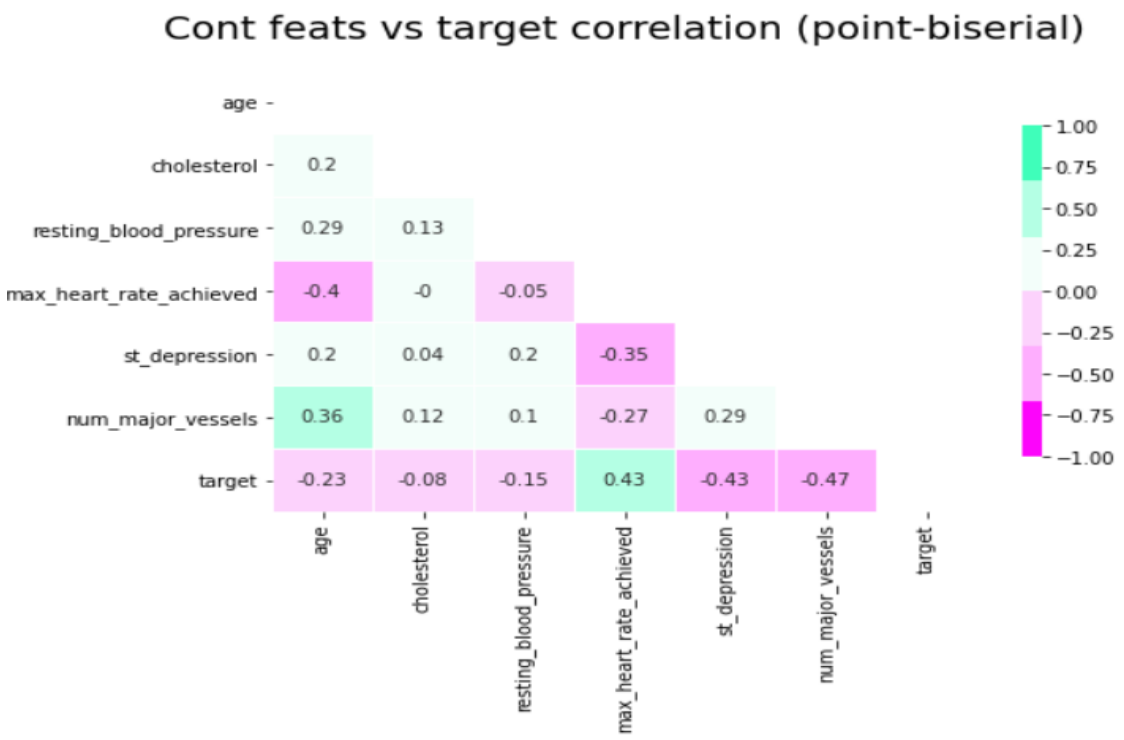


Figure 5.2 Point biserial correlation heatmap

5.1.3 Cramer's V Correlation

- In statistics, Cramér's V is a measure of association between **two nominal variables**, giving a value between 0 and +1 (inclusive). It is based on Pearson's chi-squared statistic and was published by Harald Cramér in 1946**.**Cramer's V correlation heatmap shown below in figure 5.3

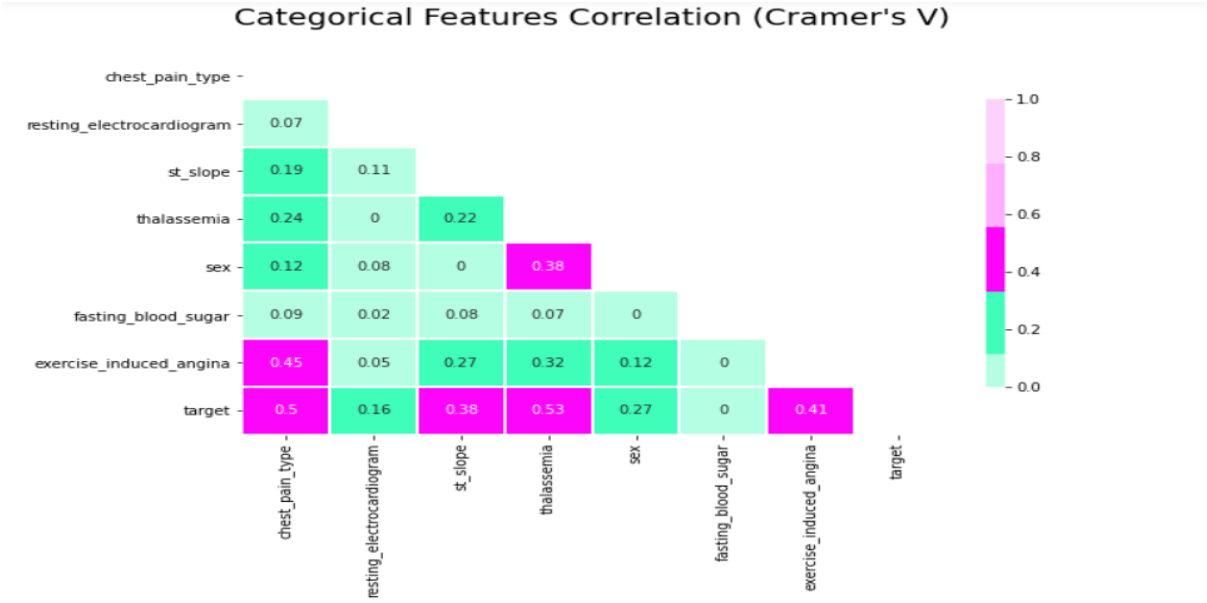


Figure 5.3 Cramer's V correlation heatmap

5.1.4 **EDA Summary**:

- Data size: 303 rows and 14 columns (13 independent + one target variable) > later reduced to 296 after removing faulty data points!
- Data has no missing values
- Features (columns) data type:
  - Six features are numerical
  - The rest (seven features) are categorical variables
- Target variable is fairly balanced, 54% no-disease to 46% has-disease
- Correlations:
- Correlation between features is weak at best
- From the numerical features num_major_vessels, max_heart_rate_achieved and st_depression are reasonabily fairly correlated with the target variable at -0.47, 0.43 and -0.43 correlation coefficient respectively.
- From the categorical features chest_pain_type, num_major_vessels, thalassemia, and exercise_induced_angina are better correlated with the target variable, thalassemia being the highest at 0.52.
  - Cholestrol (to my surprize, but what do I know?) has less correlation with heart desease.

CHAPTER 6

PREDICTIONS

6.1 SCIKIT LEARN CLASSIFIERS

This is a binary classification problem (has-disease or no-disease cases). Scikit learn offers a wide range of classification algorithms and is often the starting point in most/traditional machine learning challenges, so we start by exploring few of the classification alorithms from the sklearn libarary such as Logistic Regression, Nearest Neighbors, Support Vectors, Nu SVC, Decision Tree, Random Forest, AdaBoost, Gradient Boosting, Naive Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis and Neural Net. Let's first build simple models using the above mentioned ML algorithms and later we will optimize them by tuning the parameters.

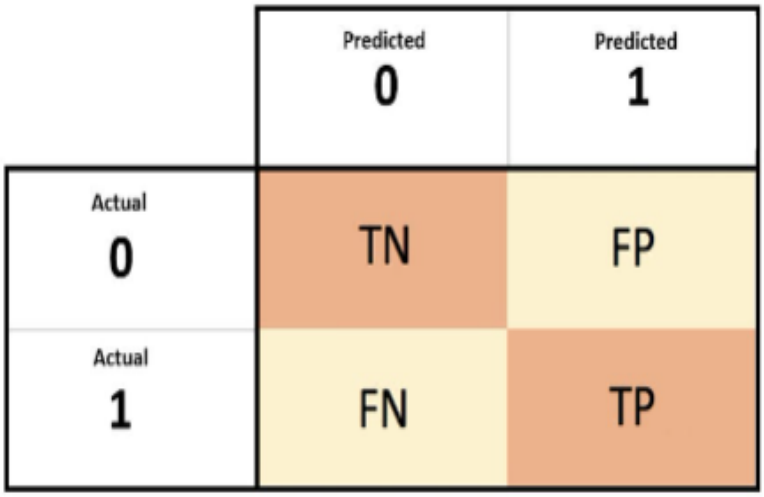
```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC, LinearSVC, NuSVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis

from sklearn.metrics import confusion_matrix, plot_confusion_matrix, classification_report
from sklearn.metrics import recall_score, accuracy_score,roc_curve, auc
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier

from sklearn.preprocessing import LabelEncoder
```

6.2 PERFORMANCE METRICS

There are several metrics that can be used to gauge the performance of a given classification algorithm. The choice of the 'appropriate′ metrics is then dependent on the type of problem. There are case where, for example, accuracy can be the right choice and in some other case a recall or precision could be more fitting to the purpose. Since this project is dealing withmedical case (classify if a case is positive for heart disease or not), project could use recall (true positive rate or sensitivity) as performance metrics to choose our classifier.

6.2.1 CONFUSION MATRIX :

A confusion matrix (aka an error matrix) is a specific table layout that allows visualization of the performance of a supervised learning algorithm. Each row of the matrix represents the instances in an *actual* class while each column represents the instances in *a* predicted *class* . The table below is an example of a confusion matrix for a binary classification from which other terminologies/metric can be derived. Some of the metrics are described below.



Term	Meaning	Descriptions
TP	True Positive	Positive cases which are predicted as positive
FP	False Positive	Negative cases which are predicted as positive
TN	True Negative	Negative cases which are predicted as negative
FN	False Negative	Positive casea which are predicted as negative

Accuracy : Measures how many of the cases are correctly identified/predicted by the model, i.e correct prediction divided by the total sample size.$(TP + TN)/(TP + TN + FN + FP)$

Recall: Measures the rate of *true positives*, i.e how many of the *actual* positive cases are *identified/predicted* as positive by the model.

$TP/(TP+FN)$

Precision: Measures how many of the positive predicted cases are actually positive.

$TP/(TP+FP)$

F1-Score : Combines the precision and recall of the model and it is defined as the harmonic mean of the model's precision and recall.

$2(recall*precision)/(recall+precision)$

ROC curves : A receiver operating characteristic (ROC) curve, is a graphical plot which illustrates the performance of a binary classification algorithm as a function of ture positive rate and false positive rate.

Performance metrics summary table 6.1

	Classifier	Accuracy	ROC_AUC	Recall	Precision	F1
0	Logistic Regression	86.490000	0.920000	0.910000	0.820000	0.860000
9	Linear DA	85.140000	0.920000	0.890000	0.820000	0.850000
10	Quadratic DA	85.140000	0.900000	0.830000	0.850000	0.840000
5	Random Forest	83.780000	0.920000	0.830000	0.830000	0.830000
4	Decision Tree	82.430000	0.820000	0.830000	0.810000	0.820000
6	AdaBoost	82.430000	0.860000	0.910000	0.760000	0.830000
7	Gradient Boosting	82.430000	0.900000	0.890000	0.780000	0.830000
8	Naive Bayes	82.430000	0.920000	0.860000	0.790000	0.820000
3	Nu SVC	81.080000	0.910000	0.910000	0.740000	0.820000
11	Neural Net	78.380000	0.880000	0.940000	0.700000	0.800000
2	Support Vectors	64.860000	0.800000	0.890000	0.580000	0.700000
1	Nearest Neighbors	55.410000	0.600000	0.310000	0.550000	0.400000

6.2.2 **ROC CURVES**

```
roc_auc_curve(names, classifiers)
```

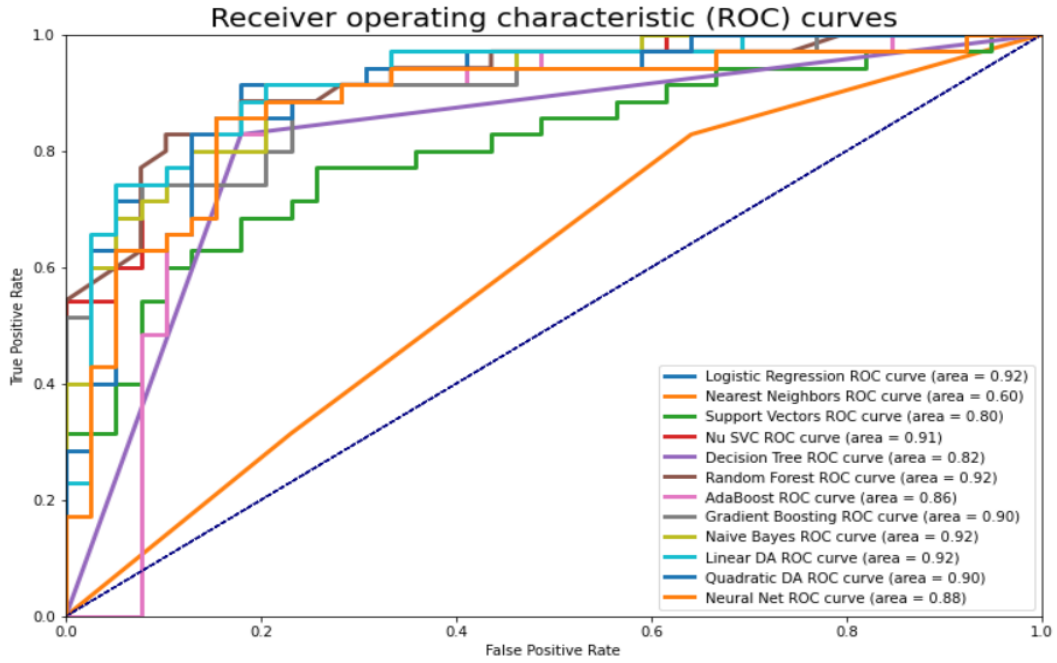
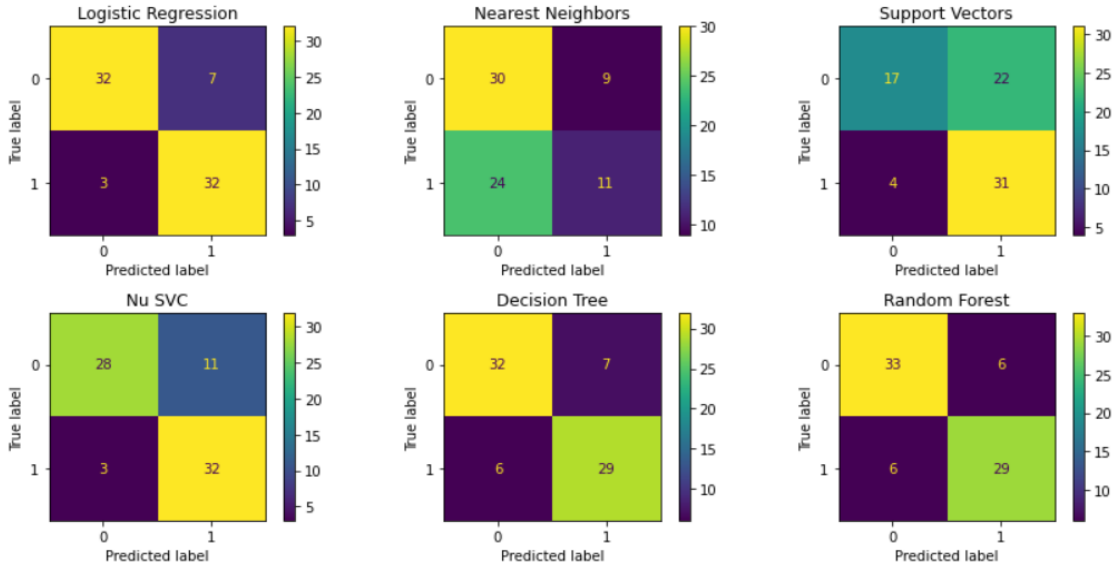


Figure 6.1 ROC curves

6.2.3 CONFUSION MATRIX OF SCIKIT LEARN CLASSIFIERS

```
plot_conf_matrix(names, classifiers, nrows=4, ncols=3, fig_a=12, fig_b=12)
```



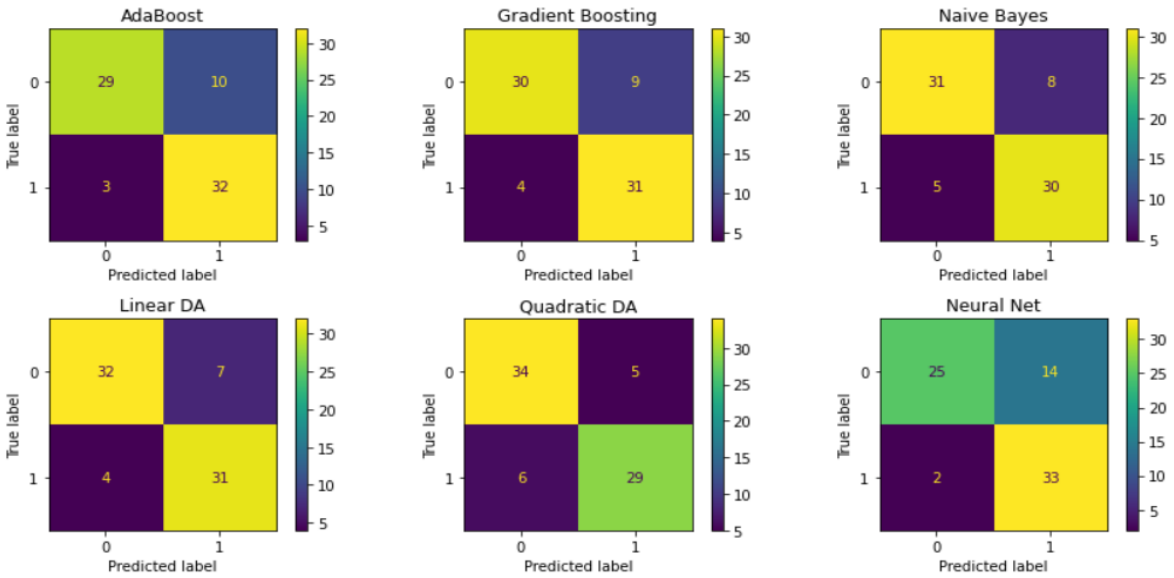


Figure 6.2 Confusion Matrix of Scikit Learn Classifiers

Now all the performance metrics of the classifiers are seen, it is decision time to choose the best possible classifier algorithm. Based on precision LR ranks first (86%); whereas if we see the recall, Neural Nets ranks first with 94%. In the case of precision, QDA ranks first with 85%. So which one to choose? The F1-score can give us a balance between recall and precision. LR happens to have the best F1-score so we choose Logistic Regression as our best classifier.

6.3 MACHINE LEARNING ALGORITHMS BASED ON THE GRADIENT-BOOSTING TECHNIQUE

6.3.1 CATBOOST, LGBM AND XGBOOST

In the above section we have seen classifiers out of the scikit-learn library. Now we will try the modern (boosted trees) ML algorithms such as the **catboost**, **xgboost** and **lgbm**. They are optimized machine learning algorithms based on the **gradient-boosting** technique. Depending on the problem at hand, one algorithm is may be better suited than others. For detailed info one can easily refer to their documentations.

```
from catboost import CatBoostClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier

names_boost =[
    'Catboost',
    'xgbbost',
    'light GBM'
]
classifiers = [
    CatBoostClassifier(random_state=seed, verbose=0),
    XGBClassifier(objective= 'binary:logistic', random_state=seed),
    LGBMClassifier(random_state=seed)
   ]
```

6.3.2 PERFORMANCE METRICS SUMMARY TABLE

```
score_summary(names_boost, classifiers).sort_values(by='Accuracy' , ascending = False)\
.style.background_gradient(cmap='coolwarm')\
.bar(subset=["ROC_AUC",], color='#6495ED')\
.bar(subset=["Recall"], color='#ff355d')\
.bar(subset=["Precision"], color='lightseagreen')\
.bar(subset=["F1"], color='gold')
```

```
[23:11:42] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'b
inary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
```

	Classifier	Accuracy	ROC_AUC	Recall	Precision	F1
0	Catboost	82.430000	0.920000	0.830000	0.810000	0.820000
2	light GBM	82.430000	0.910000	0.860000	0.790000	0.820000
1	xgbbost	79.730000	0.920000	0.830000	0.760000	0.790000

6.3.3 CONFUSION MATRIX OF BOOSTING TECHNIQUES

```
plot_conf_matrix(names=names_boost, classifiers=classifiers, nrows=1, ncols=3, fig_a=12, fig_b=3);
```

```
[23:11:44] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used
inary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restor
```

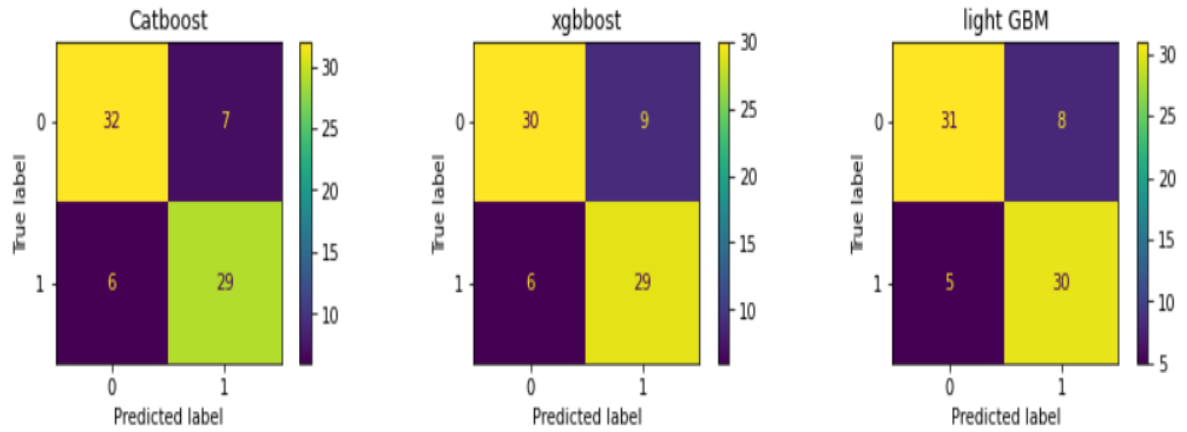


Figure 6.3 Confusion Matrix of Boosting Techniques

Remark : Here it can be seen that the lgbm calssifier is marginally better than the other two algorithm. Following the same procedure will try to tune the parameters in the next section.

6.3.4 PARAMETER TUNING (RANDOMIZEDSEARCH): LGBMCLASSIFIER

```
from sklearn.model_selection import GridSearchCV
rs_params = {
    'num_leaves': [20, 100],
    'max_depth': [5, 15],
    'min_data_in_leaf': [80, 120],
}
rs_cv = GridSearchCV(estimator=LGBMClassifier(random_state=seed, verbose=-1),
                     param_grid=rs_params,
                     cv = 5)

rs_cv.fit(X_train, y_train)
params = rs_cv.best_params_
params
```

```
lgbm = LGBMClassifier(**params);

lgbm.fit(X_train, y_train,
        eval_set=(X_val, y_val),
        verbose=False,
);

print(classification_report(y_val, lgbm.predict(X_val)))
```

```
              precision    recall  f1-score   support

           0       0.94      0.79      0.86        39
           1       0.80      0.94      0.87        35

    accuracy                           0.86        74
   macro avg       0.87      0.87      0.86        74
weighted avg       0.88      0.86      0.86        74
```

```
plot_confusion_matrix(lgbm, X_val, y_val);
```

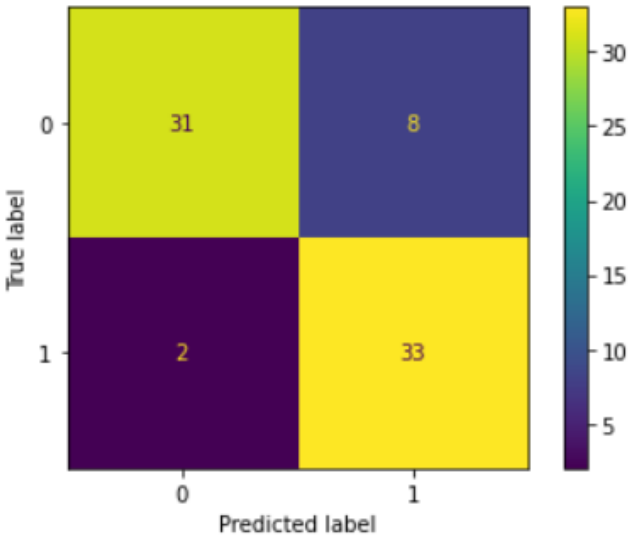


Figure 6.4 Confusion Matrix Of LGBM

CHAPTER 7

CONCLUSION

At the beginning of the notebook, the objective was to investigate the heart disease dataset through exploratory data analysis (EDA) and binary classification modeling. In the first part of the project, the dataset was examined, data sanity checks were performed, the data was pre-processed by removing faulty data, and correlations between features and the target variable were identified. In the second part, binary classifiers were created, starting with base models and refining them through hyper-parameter tuning, which resulted in identifying the best model. Some of the key takeaways from the project include:

1. The best model for this project was the lgbm classifier, which was fine-tuned using randomizedSearch.

2. Surprisingly, contrary to initial assumptions, the model revealed that cholesterol was not an important feature.

REFRENCES

1. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
2. https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9