CSE422: Artificial Intelligence

Lab Project Report

Group: 10

Title: Predicting Heart Disease

Kamran Hassan Shomrat	21101010
Aditya Barua	21101004
Jubayer Hossain	21101205

Table of Content

Introduction

Heart disease and other cardiovascular illnesses continue to be a major global health concern, raising rates of morbidity and death. Using machine learning to address complicated medical issues has become increasingly important as we stand on the brink of technological progress. This project, "Predicting Heart Disease," aims to provide an accurate and effective tool for estimating an individual's risk of developing heart disease by utilizing machine learning algorithms. The report presents the results of our 422-lab project, which involved the use of four algorithms (Logistic Regression, Naive Bayes, Random Forest Classifier, K-NeighborsClassifier) to develop a model for heart disease prediction. Additionally, we have thoroughly demonstrated several methods for assessing the dataset's components: count plot, heatmap, and other three categories of data visualization approaches boxplot. Before training the model, we have evaluated the data using preprocessing methods for numerous datasets to get better outcomes.

i) Motivation Behind the Project:

No model can accurately predict every dataset's result. Overfitting is the attempt to use the same model to predict diverse datasets. Thus, our goal was to observe how various models lead to diverse outcomes. In actuality inconsistencies in datasets are unavoidable. Thus, mastering large data management. To make the data suitable for training was another aspect of this endeavor.

ii) Aims and Objective:

The main goal of this project is to see which model works best among the four for the particular heart disease dataset. Some key objectives were handling and pre-fixing data, and visualizing data.

Dataset Description:

- **Source:**

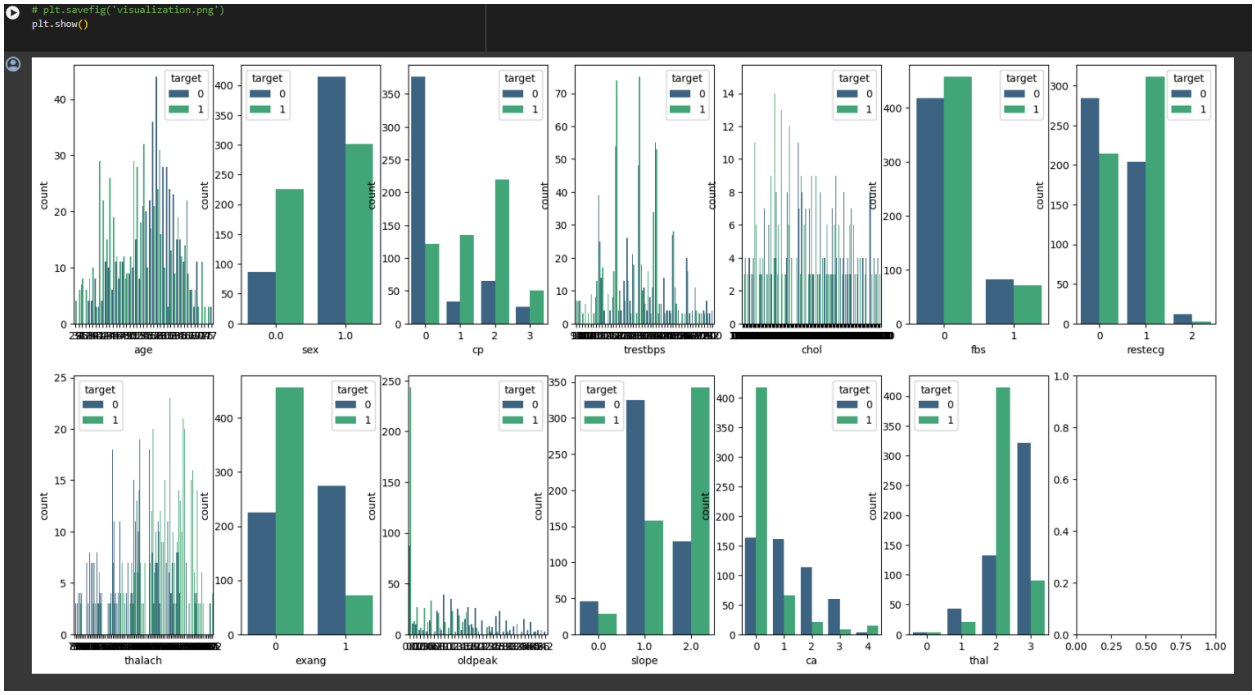
  Link: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

  Reference: *Heart disease dataset*. (2019, June 6). Kaggle. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

- **Dataset Description:** This dataset, which dates back to 1988, is divided into four databases: Long Beach V, Cleveland, Hungary, and Switzerland. It has 76 features in total, including the one that was predicted, but only 14 of these features have been used in studies that have been published. An integer value (0 indicates absence; 1 indicates presence) in the "target" field indicates whether or not patients have heart disease. The "Heart Disease Dataset" on Kaggle contains thousands of records with fourteen fields filled in with crucial patient information. The last column predicts the presence or absence of potential cardiac conditions ("Target": No illness = 0; Disease presence = 1). Gender data entries are identified as 'sex', where male denotes the number code 'One'; While female representatives are zero in terms of numbers. Growth age data is represented as integers under the label "age." The acronym 'cp' is used to represent pain around the chest area, and it is further classified into four distinct codes."trestbps" monitors resting blood pressure, which is measured in millimeter-Hg and provided upon hospital admission. Serum cholesterol is represented by the term "chol," which is equivalent to mg/dl figures. On the other hand, fasting glucose levels exceeding 120 mg/DL are labeled as "fbs" (one equals trues; false-zero). Results of the ECG test taken during the rest phase line up with "restecg," while the results of the manual cardiogram follow "thalach," showing the peak HR (maximum beats per minute reached), the onset

of angina caused by exercise (oldpeak), and the difference between stable stillness and jog-formed depression. Complete The term "slope" refers to peak inclination/progression, while "ca" refers to the sum of vessel visualization numbers (from the range marked lowest at no x-ray detectability, or zero up to three maximum visibility) via projection radiography. Along with thalassemia, observed condition flags down parameters (normalcy defines nil figure touches on permanent abnormality refreshes into temporary linings).

- **Imbalanced Dataset:** Carefully selected, the Balanced Classification Dataset is a perfect tool for testing and training machine learning models in a classification setting. This dataset stands out for a number of reasons, including its careful attention to attaining an ideal balance between classes, which always worries about unbalanced data. By guaranteeing that every class is fairly represented, this equilibrium promotes impartial and equitable model training.



Data Pre-processing:

- **Facts:** We must also check for Null values and categorical values to detect unnecessary rows of data. We must properly process those values.

  Null values: In the dataset we intentionally included null values distributed among different features. These missing values simulate scenarios where data may be unavailable or not recorded. This deliberate inclusion allows us to observe the effect of null value deletion on the overall dataset, which we handled using data.dropna.

  Categorical Values: For this project, the dataset is made up only of numerical features that correspond to different quantifiable properties of the target variable. The lack of category inputs makes data preparation easier and highlights the importance of numerical features in identifying trends and connections.

```
Rows with null values and their counts:
      age  sex  cp  trestbps   chol  fbs  restecg  thalach  exang  oldpeak  \
1025   47  NaN   0       110  275.0    0        0      118      1      1.0
1026   50  0.0   0       110    NaN    0        0      159      0      0.0
1027   54  1.0   0       120  188.0    0        1      113      0      1.4

      slope  ca  thal  target
1025    1.0   1     2       0
1026    2.0   0     2       1
1027    NaN   1     3       0
Total number of rows with null values: 3
Cleaned DataFrame without rows containing null values:
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1028 entries, 0 to 1030
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1028 non-null   int64
 1   sex       1028 non-null   float64
 2   cp        1028 non-null   int64
 3   trestbps  1028 non-null   int64
 4   chol      1028 non-null   float64
 5   fbs       1028 non-null   int64
 6   restecg   1028 non-null   int64
 7   thalach   1028 non-null   int64
 8   exang     1028 non-null   int64
 9   oldpeak   1028 non-null   float64
 10  slope     1028 non-null   float64
 11  ca        1028 non-null   int64
 12  thal      1028 non-null   int64
 13  target    1028 non-null   int64
dtypes: float64(4), int64(10)
memory usage: 120.5 KB
```

Feature Scaling:

The project utilizing the Standardized Numeric Dataset aims to demonstrate the significance of standardization in improving the robustness and efficiency of machine learning models. The resulting insights contribute to best practices in scaling numeric data for optimal model performance.

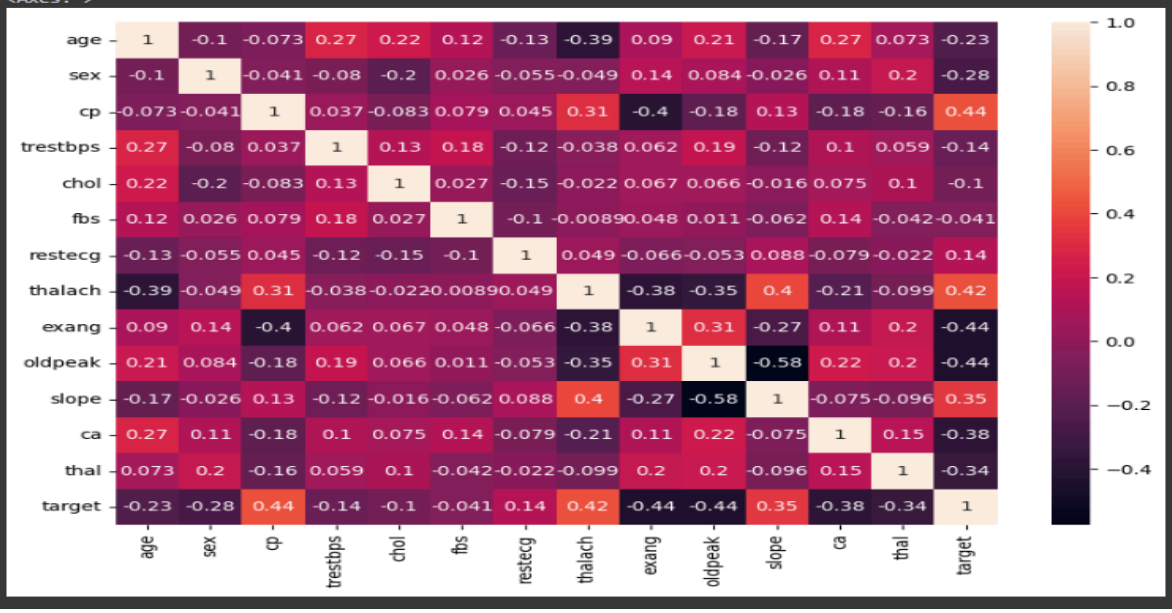
```
# Creating a StandardScaler instance
feature_scaler = StandardScaler()

# Scaling the training features
X_train_scaled = feature_scaler.fit_transform(X_train)

# Scaling the testing features
X_test_scaled = feature_scaler.transform(X_test)
```

Data Splitting:

Here, “age”, “thalanch”, “trestbps” and “oldpeak” columns have too many values, therefore, understanding those fields from a bar graph would be difficult. However, we can observe that every other row is somewhat contributing to the change of our target field “target”.

```
cor= data.corr()
plt.subplots(figsize=(10,7))

sns.heatmap(cor,annot=True)
```

<Axes: >



To detect unnecessary or redundant rows, heatmap is a great tool. If any two rows have very low temperatures, we can delete one of the rows. Since our heatmap does not show such a case, we don't need to change anything.

Model Training & Testing:

We have used 4 models:

1. Logistic Regression

2. Naive Bayes

3. Random Forest Classifier &

4. K-Neighbors Classifier

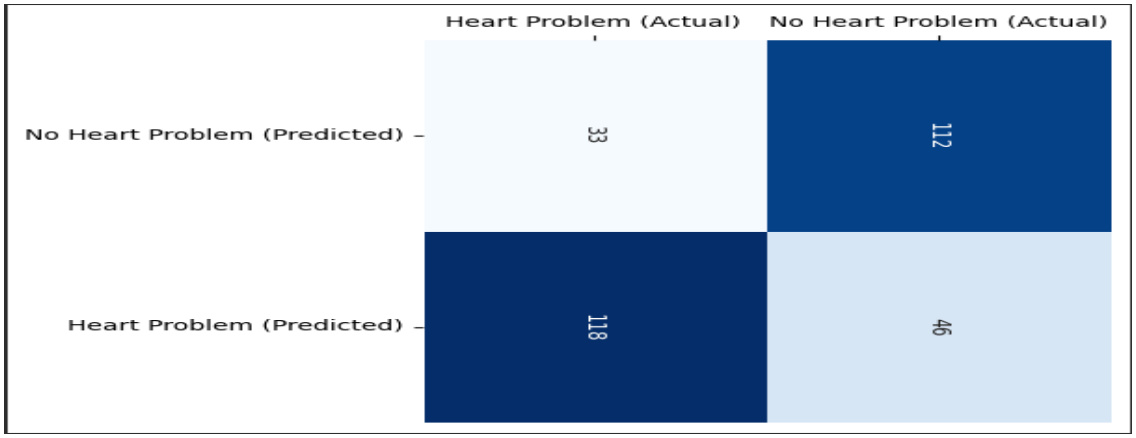
For each method, the data was fitted by a scaling and training model trained on 70% of the data set and 30% of the test set. The results that we found are discussed in the latter paragraphs.

Model Selection/Comparison analysis:

1) KNeighbours Classifier:

```
Accuracy of K-Neighbors Classifier: 74.4336569579288

              precision    recall  f1-score   support

           0       0.71      0.77      0.74       145
           1       0.78      0.72      0.75       164

    accuracy                           0.74       309
   macro avg       0.75      0.75      0.74       309
weighted avg       0.75      0.74      0.74       309
```

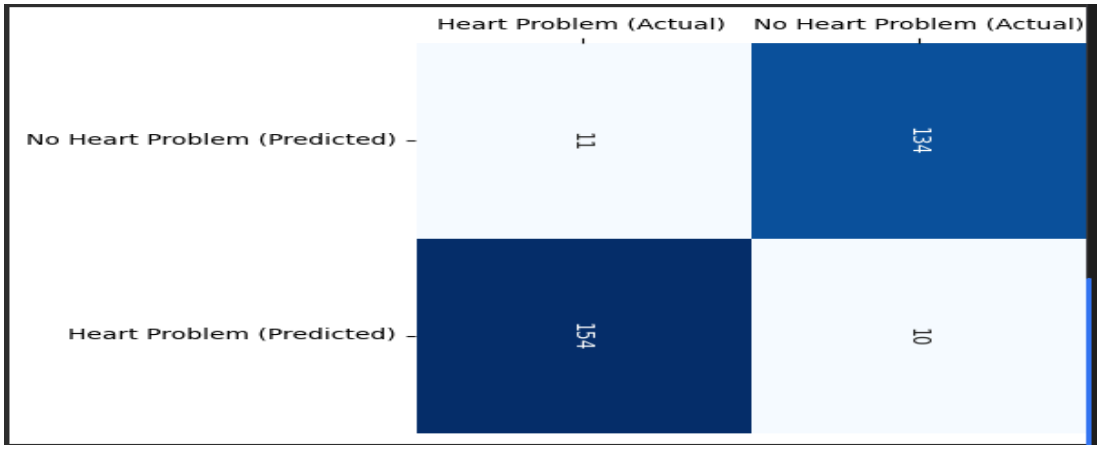
Confusion Matrix:



2) Random Forest Classifier:

```
Accuracy of Random Forest Classifier: 93.20388349514563

              precision    recall  f1-score   support

           0       0.93      0.92      0.93       145
           1       0.93      0.94      0.94       164

    accuracy                           0.93       309
   macro avg       0.93      0.93      0.93       309
weighted avg       0.93      0.93      0.93       309
```

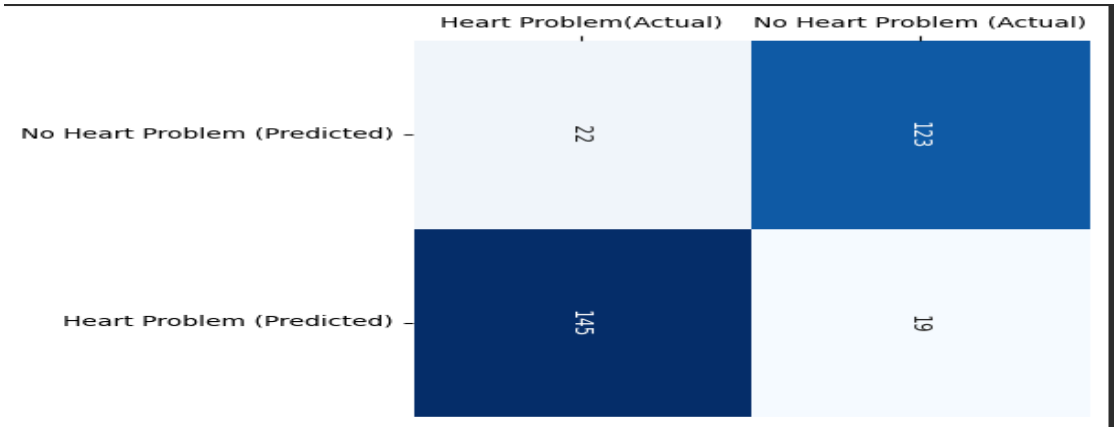
Confusion Matrix:



3) Naive Bayes:

```
Accuracy of Naive Bayes model: 86.73139158576052

              precision    recall  f1-score   support

           0       0.87      0.85      0.86       145
           1       0.87      0.88      0.88       164

    accuracy                           0.87       309
   macro avg       0.87      0.87      0.87       309
weighted avg       0.87      0.87      0.87       309
```

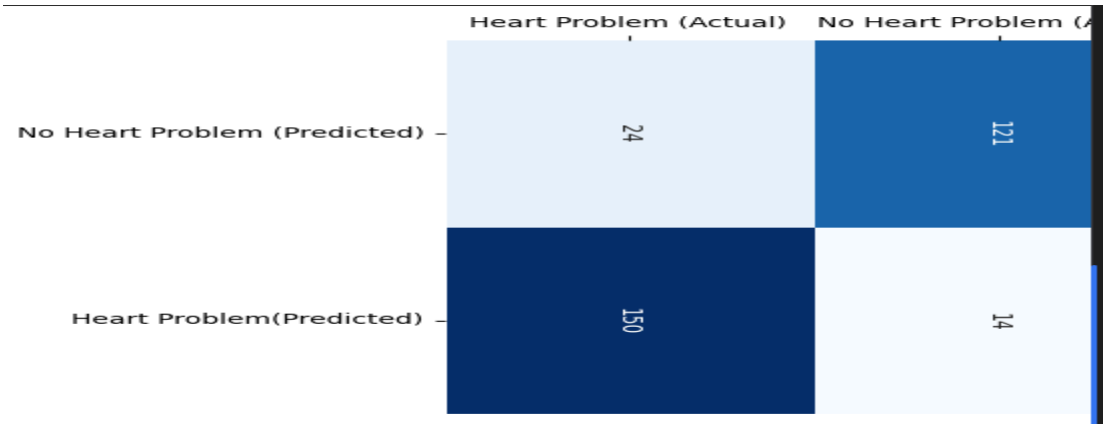
Confusion Matrix:



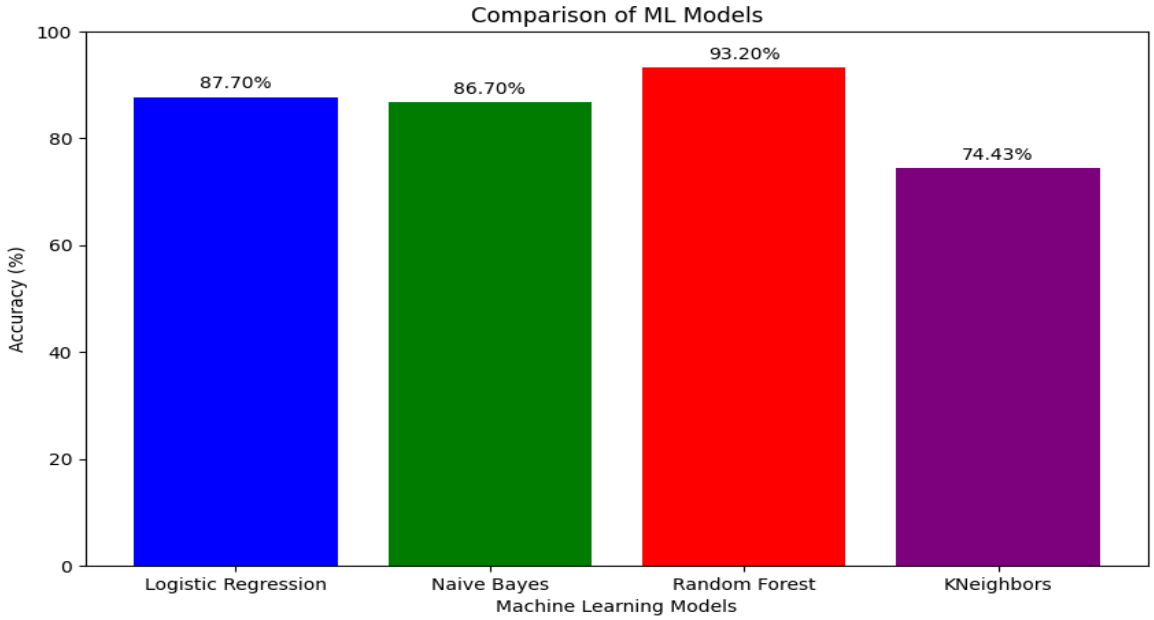
4) Logistic Regression:

```
Accuracy of Logistic Regression: 87.70226537216828

              precision    recall  f1-score   support

           0       0.90      0.83      0.86       145
           1       0.86      0.91      0.89       164

    accuracy                           0.88       309
   macro avg       0.88      0.87      0.88       309
weighted avg       0.88      0.88      0.88       309
```

Confusion Matrix:



Conclusion:



Random forest models performed better among the other three models and predicted results with 93.20388349514563 percent accuracy.