HEART FAILURE PREDICTION

Kartik Chhipa (B20CS084), Rushil Ashish Shah(B20AI036), Ruthvik K(B20AI037)

Abstract - *This paper reports our experience with building a classifier that predicts if a person will have heart failure or not depending on a given set of features. We are given a dataset containing several feature vectors that provide complete information about the person. We are supposed to apply machine learning based models for classification and predict if that person will have heart failure or not.*

Introduction

Cardiovascular diseases kill approximately 17 million people globally every year, and they mainly exhibit myocardial infarctions and heart failures. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure. Most cardiovascular diseases can be prevented by addressing behavioural risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help

Dataset

We have used the dataset present on the Kaggle for this project. It contains 12 columns and 918 Rows.
Explanation of Columns in the dataset

1. Age - age of the person in years
2. Sex - sex of the person (M: Male, F: Female)
3. ChestPainType - chest pain type
4. (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic)
5. RestingBP: resting blood pressure (mm Hg) Cholesterol: serum cholesterol (mm/dl)
6. FastingBS: fasting blood sugar (1: if FastingBS > 120 mg/dl, 0: otherwise)
7. RestingECG: resting electrocardiogram results (Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria)
8. MaxHR: maximum heart rate (Numeric value between 60 and 202)
9. ExerciseAngina: exercise-induced angina (Y: Yes, N: No)
10. Oldpeak: oldpeak = ST (Numeric value measured in depression)
11. ST_Slope: the slope of the peak exercise ST segment (Up: upsloping, Flat: flat, Down: downsloping)
12. HeartDisease: output class (1: heart disease, 0: Normal)

Methodology

Overview

There are various classification algorithms present out of which we shall implement the following

- *Random Forest Classification*
- *KNN*
- *Gradient Boosting*
- *Gaussian Naive Bayes*
- *XGBoost*
- *Neural Network*

Exploring the dataset and pre-processing

On counting the number of NULL values in the train dataset , it was found that there are no NULL values present. The columns were of two types continuous and nominal the nominal features were onehot encoded because of no relative ordering between them. The outliers were identified using the box plots. And removed. A comparison is made between the results

Implementation of classification algorithms

A better approach for implementing classifiers is to objectively search different values for model hyperparameters and choose a subset that results in a model that achieves the best performance on a given dataset. This is called **hyperparameter optimization** or hyperparameter tuning and here we use RandomizedSearchCV for the same.

• ***Random Forest Classifier :*** Random Forest Classifiers use boosting ensemble methods to train upon various decision trees and produce aggregated results.It is one of the most used machine learning algorithms.
After applying RandomizedSearchCV for both the data's i.e original one and the one after removing the outliers we get the best parameters for the classifier and now we implement the models with those parameters. Best parameters obtained for the data with outliers:
roc_auc_score for the model with outlier data : **0.9424**
roc_auc_score for the model without outliers : **0.9567**
As expected we can clearly see the increase in roc_auc_score of the model after removing the outliers.

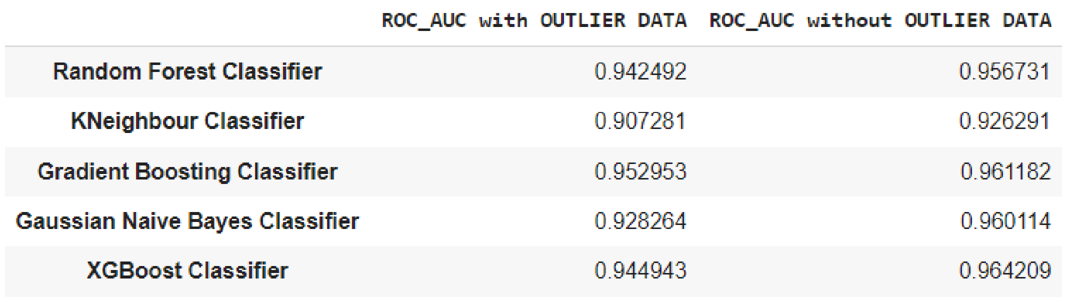
• ***KNN (k - nearest neighbors)*** *:* KNN are supervised algorithms which classify on the basis of distance from similar points. Here k is the number of nearest neighbors to be considered in the majority voting process.
After applying RandomizedSearchCV for both the data's i.e original one and the one after removing the outliers we get the best parameters for the classifier and now we implement the models with those parameters.
roc_auc_score for the model with outlier data : **0.9072**
roc_auc_score for the model without outliers : **0.9262**
As expected we can clearly see the increase in roc_auc_score of the model after removing the outliers.

• ***Gradient Boosting*** : Gradient Boosting Classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets
After applying RandomizedSearchCV for both the data's i.e original one and the one after removing the outliers we get the best parameters for the classifier and now we implement the models with those parameters.
roc_auc_score for the model with outlier data : **0.9529**
roc_auc_score for the model without outliers : **0.9611**
As expected we can clearly see the increase in roc_auc_score of the model after removing the outliers.

• ***Gaussian Naive Bayes :*** GNB is a type of Naive Bayes classifier that assumes that the distribution of data is gaussian and classifies data based on this assumption.
After applying RandomizedSearchCV for both the data's i.e original one and the one after removing the outliers we get the best parameters for the classifier and now we implement the models with those parameters.
roc_auc_score for the model with outlier data : **0.9282**
roc_auc_score for the model without outliers : **0.9601**
As expected we can clearly see the increase in roc_auc_score of the model after removing the outliers.

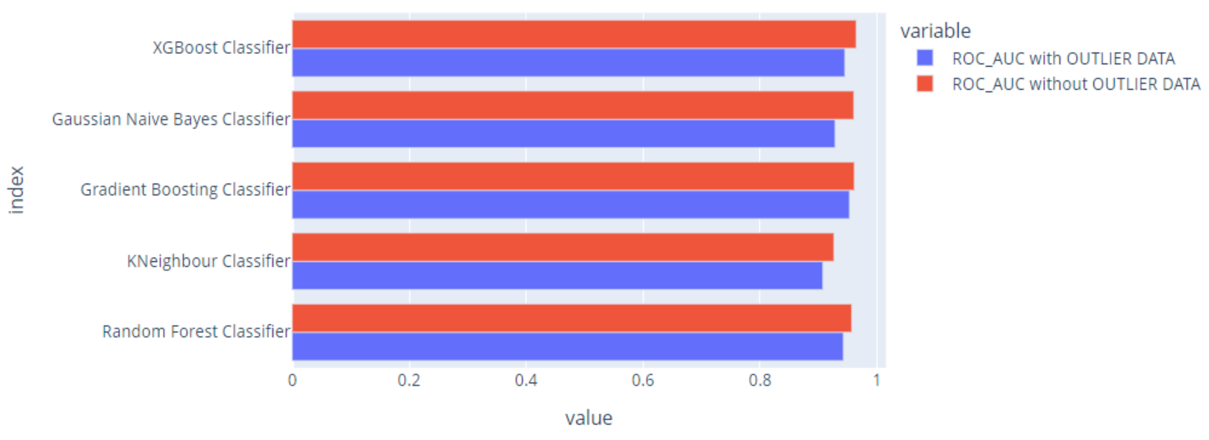
• ***XGBoost Classifier*** *: XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way*
After applying RandomizedSearchCV for both the data's i.e original one and the one after removing the outliers we get the best parameters for the classifier and now we implement the models with those parameters.
roc_auc_score for the model with outlier data : **0.9449**
roc_auc_score for the model without outliers : **0.9642**

Visualization of the above numeric results in dataframe



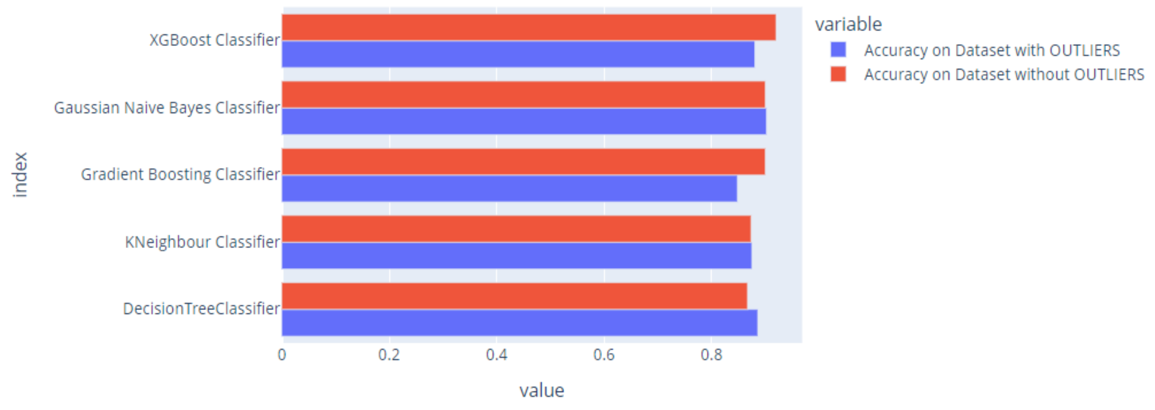
	ROC_AUC with OUTLIER DATA	ROC_AUC without OUTLIER DATA
Random Forest Classifier	0.942492	0.956731
KNeighbour Classifier	0.907281	0.926291
Gradient Boosting Classifier	0.952953	0.961182
Gaussian Naive Bayes Classifier	0.928264	0.960114
XGBoost Classifier	0.944943	0.964209

Visualization of the above dataframe

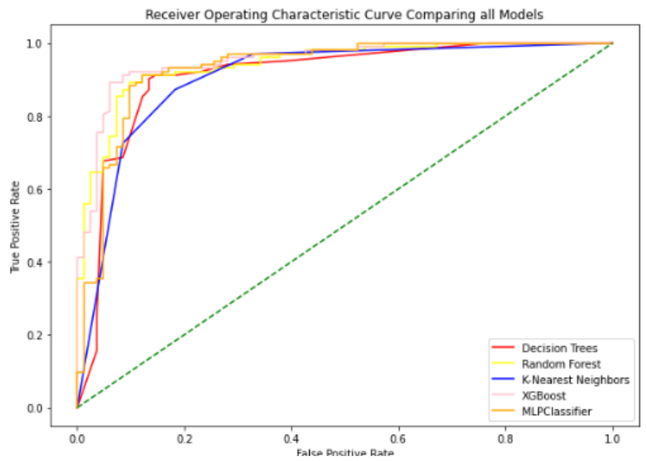


Visualizing the Results of various models implemented through Pipeline

	Accuracy on Dataset with OUTLIERS	Accuracy on Dataset without OUTLIERS
DecisionTreeClassifier	0.885870	0.866667
KNeighbour Classifier	0.875000	0.873333
Gradient Boosting Classifier	0.847826	0.900000
Gaussian Naive Bayes Classifier	0.902174	0.900000
XGBoost Classifier	0.880435	0.920000



ROC Curve for model comparison



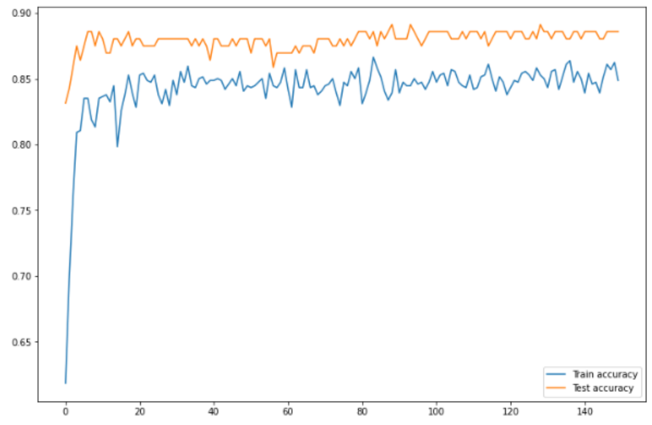
Classification Report

```
Classification Report of Dataset with Outliers
Decision Trees
                 precision    recall  f1-score   support

   HeartDisease       0.88      0.87      0.87        82
No HeartDisease       0.89      0.90      0.90       102

       accuracy                           0.89       184
      macro avg       0.88      0.88      0.88       184
   weighted avg       0.89      0.89      0.89       184

Random Forest
                 precision    recall  f1-score   support

   HeartDisease       0.89      0.82      0.85        82
No HeartDisease       0.86      0.92      0.89       102

       accuracy                           0.88       184
      macro avg       0.88      0.87      0.87       184
   weighted avg       0.88      0.88      0.87       184

K-Nearest Neighbors
                 precision    recall  f1-score   support

   HeartDisease       0.84      0.82      0.83        82
No HeartDisease       0.86      0.87      0.86       102

       accuracy                           0.85       184
      macro avg       0.85      0.84      0.85       184
   weighted avg       0.85      0.85      0.85       184

XGBoost
                 precision    recall  f1-score   support

   HeartDisease       0.90      0.88      0.89        82
No HeartDisease       0.90      0.92      0.91       102

       accuracy                           0.90       184
      macro avg       0.90      0.90      0.90       184
   weighted avg       0.90      0.90      0.90       184

MLPClassifier
                 precision    recall  f1-score   support

   HeartDisease       0.89      0.83      0.86        82
No HeartDisease       0.87      0.92      0.90       102

       accuracy                           0.88       184
      macro avg       0.88      0.88      0.88       184
   weighted avg       0.88      0.88      0.88       184
```

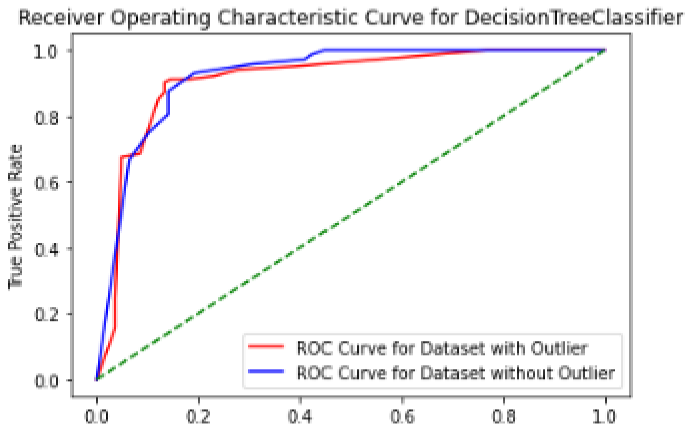
```
Classification Report of Dataset without Outliers
Decision Trees
                 precision    recall  f1-score   support

   HeartDisease       0.88      0.86      0.87        78
No HeartDisease       0.85      0.88      0.86        72

       accuracy                           0.87       150
      macro avg       0.87      0.87      0.87       150
   weighted avg       0.87      0.87      0.87       150

Random Forest
                 precision    recall  f1-score   support

   HeartDisease       0.93      0.82      0.87        78
No HeartDisease       0.83      0.93      0.88        72

       accuracy                           0.87       150
      macro avg       0.88      0.88      0.87       150
   weighted avg       0.88      0.87      0.87       150

K-Nearest Neighbors
                 precision    recall  f1-score   support

   HeartDisease       0.96      0.85      0.90        78
No HeartDisease       0.85      0.96      0.90        72

       accuracy                           0.90       150
      macro avg       0.90      0.90      0.90       150
   weighted avg       0.91      0.90      0.90       150

XGBoost
                 precision    recall  f1-score   support

   HeartDisease       0.92      0.88      0.90        78
No HeartDisease       0.88      0.92      0.90        72

       accuracy                           0.90       150
      macro avg       0.90      0.90      0.90       150
   weighted avg       0.90      0.90      0.90       150

MLPClassifier
                 precision    recall  f1-score   support

   HeartDisease       0.97      0.87      0.92        78
No HeartDisease       0.88      0.97      0.92        72

       accuracy                           0.92       150
      macro avg       0.92      0.92      0.92       150
   weighted avg       0.93      0.92      0.92       150
```

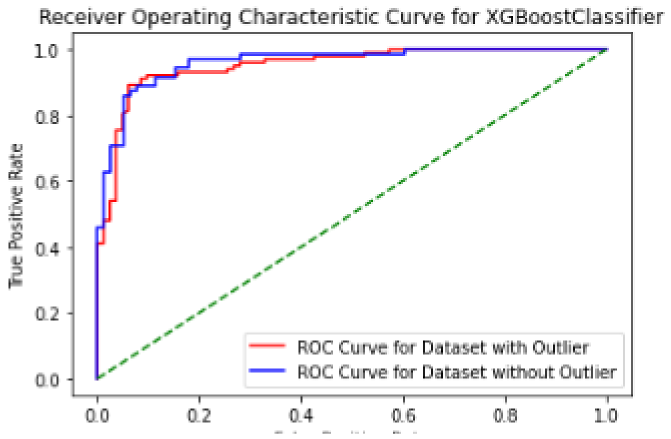
Neural Network Training and Testing Accuracy Curve

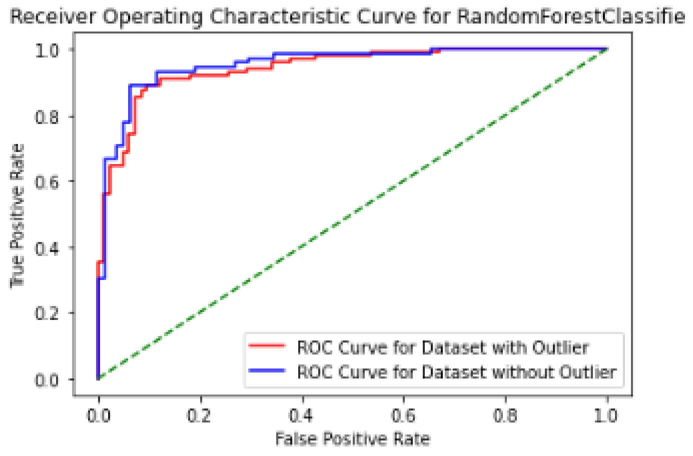
```
6/6 [==============================] - 0s 2ms/step - loss: 0.3451 - accuracy: 0.8859
Neural Network score     ========>>> 0.886
```

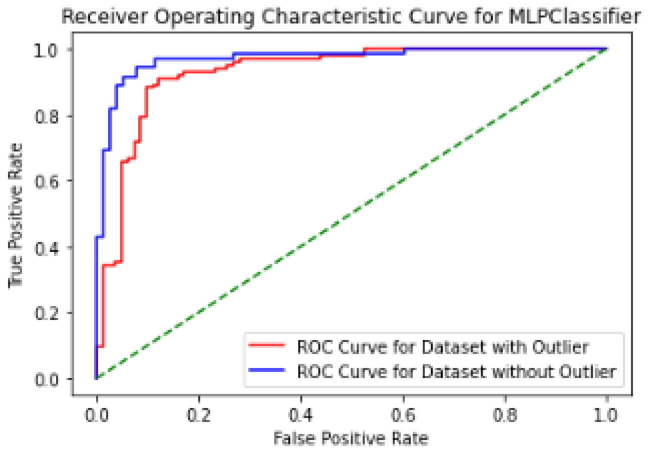


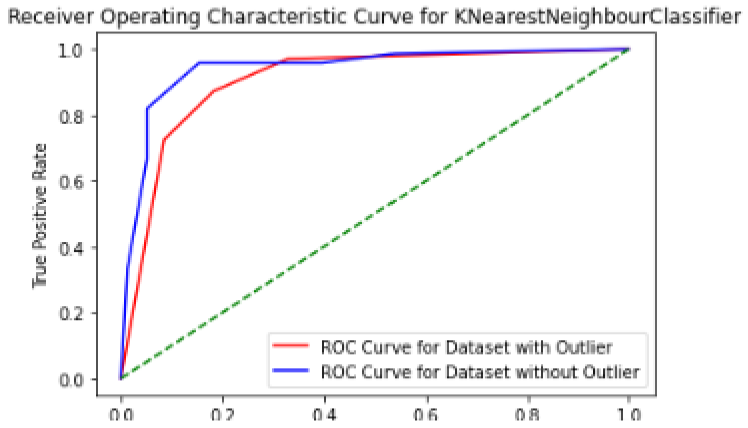
ROC Curve for comparing improved accuracy after removing outliers











Results and Analysis

The table shows that all classifiers have had nearly equally efficient performance. Also we have observed that the results have significantly improved after applying exploratory data analysis techniques and removing the outliers from the dataset outliers are false values and significantly impact the machine learning model performance. We can see a significant amount of increase in roc_auc_score after removing the outliers from the data for all the classifier models that we have implemented.
XGBoost has turned out to be the best classifier when evaluated on the basis of roc_auc score which is equal to 96.642%. and also performed marginally better than other classifier when evaluated against accuracy which came out to be 92.00%.

Link to the webapp : https://share.streamlit.io/kartikchhipa01/heartfailureprediction/main.py

Contributions

The learning and planning was done as a team. The individual contributions are as given

- Kartik Chhipa (B20CS084): Exploratory Data Analysis (EDA), Web app, Compiling individual Work, Report.
- Rushil Ashish Shah(B20AI036) : Artificial Neural Network, Pipeline, Gaussian Naive Bayes, Report
- Ruthvik K (B20AI037): Random Forest Classifier, KNeighbors, Gradient Boosting, *XGBoost,* Report.

References

[1] K Nearest Neighbor | KNN Algorithm | KNN in Python & R (analyticsvidhya.com)
[2] Pattern Classification -Book by David G. Stork, Peter E. Hart, and Richard O. Duda
[3] Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science
[4] Link for image dataset : Heart Failure Prediction | Kaggle