PRML COURSE PROJECT

HEART FAILURE PREDICTION

📎 Abstract - Cardiovascular disease (CVD) is one of the most common causes of death killing approximately 17 million people annually. The main reason behind CVD is the failure of the heart to pump blood normally. This dataset was created by combining different datasets already available independently but not combined before. In this dataset, 5 heart datasets are combined with over 11 common features which makes it the largest heart disease dataset available so far for research purposes. We proposed a machine-learning-based approach that distinguishes the most important correlated features amongst patients' electronic clinical records. We applied Optimised hyperparameter classification algorithms like SVM, KNN, Light GBM, Random Forest, and Neural Network to obtain the best possible accuracy, F1 score, recall, and precision.

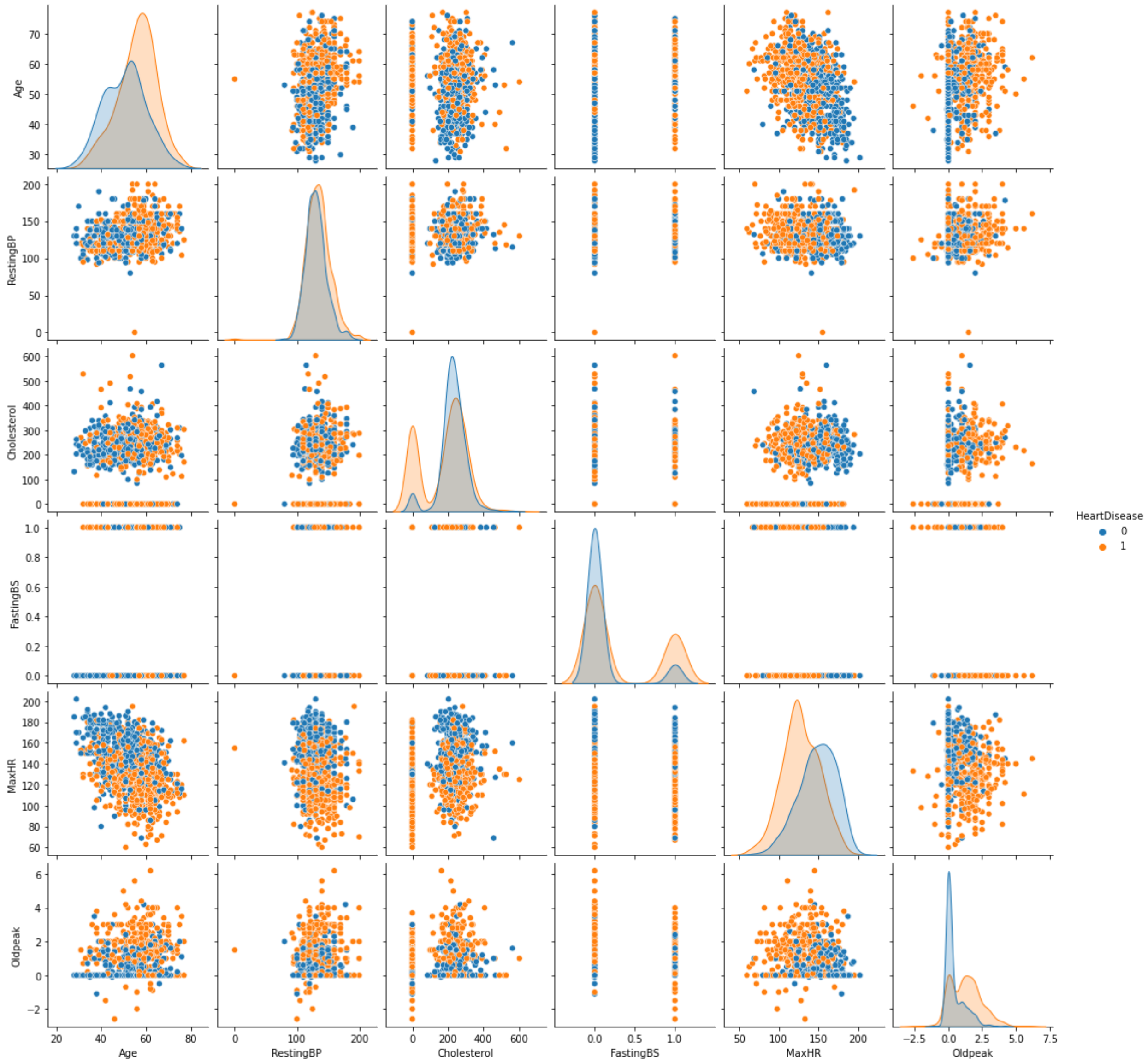
▼ **Group Members :**

Nishant Sharma (B20EE039)

Neielotpal Rao (B20EE038)

Shashwat Singh (B20CS066)

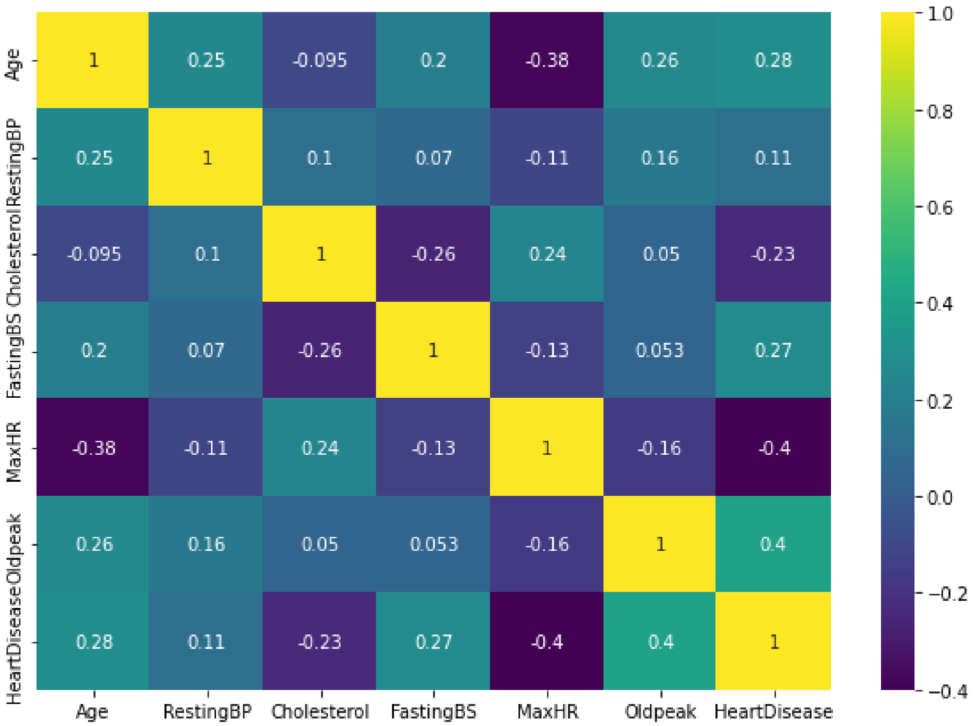
Exploratory data analysis and visualization



Here we observed Feature Importance in seperating classes

1. **Pearson Correlation Coefficient**

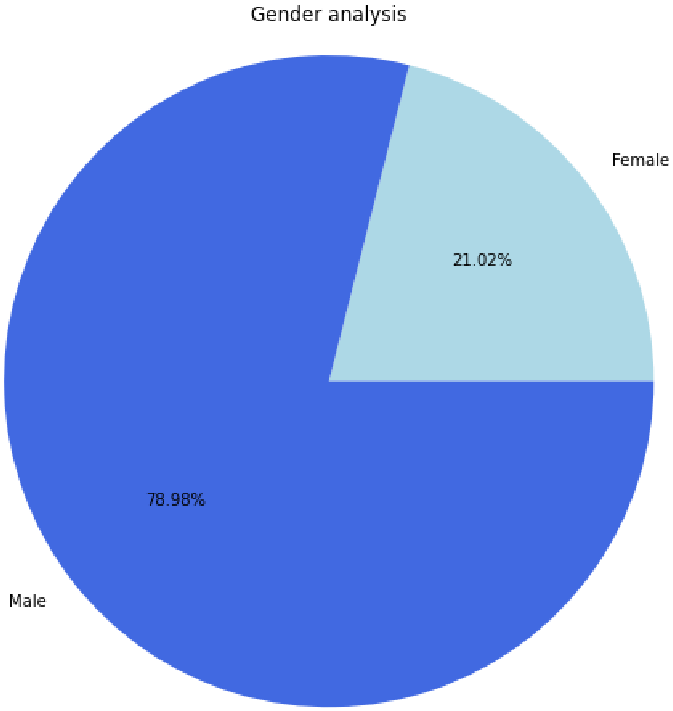
$$r = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum\left(x_i - \bar{x}\right)^2 \sum\left(y_i - \bar{y}\right)^2}}$$



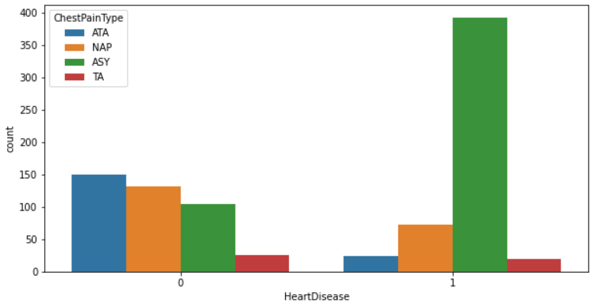
$$r = \Sigma((x_i - \bar{x}) * (y_i - \bar{y}))/\sqrt{(\Sigma(x_i - \bar{x})^2 * \Sigma(y_i - \bar{y})^2)}$$

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 normalized measurement of the covariance, such that the result always has a value between −1 and 1.

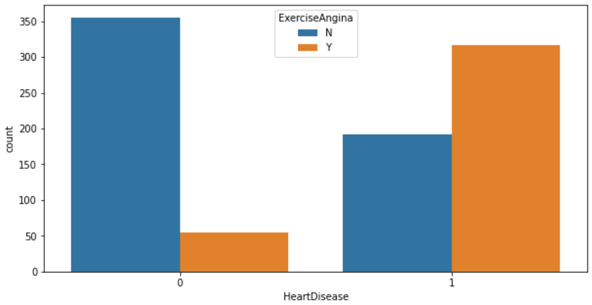
2. **Feature Analyzation**



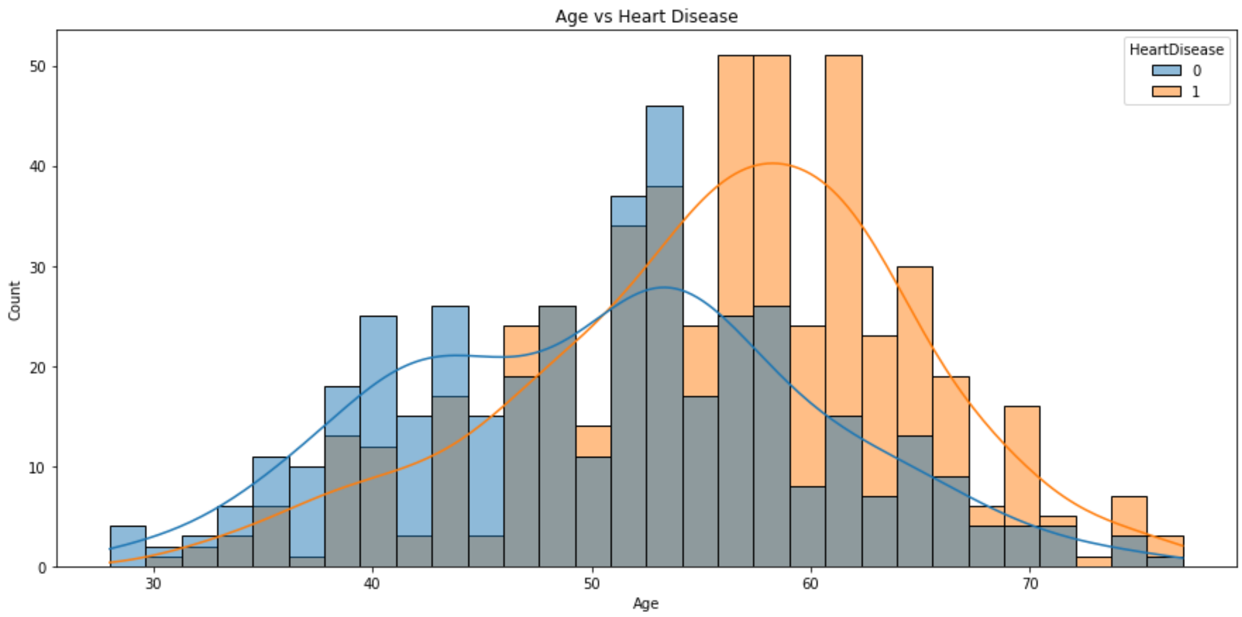
In the provided data 78.98% are males while leftover are female.



It’s the distribution of `ChestPainType` for each value of `HeartDisease`.



It’s the distribution of `ExerciseAnigma` for each value of `HeartDisease`.



Distribution plot of `Age` for each value of `HeartDisease`.

Feature Engineering

1. **Encoding**

   One hot encoding makes our training data more useful and expressive, and it can be rescaled easily, by using numeric values, we more easily determine a probability for our values.

   Encoded the `Sex`, `ChestPainType`, `ExerciseAngina` and `ST_Slope` using `pandas.get_dummies()`.

2. **Scaling**

   `StandardScaler` removes the mean and scales each feature/variable to unit variance.

   Scaled the data in order to perform SVC as the score won't change in the case of other classifiers.

Model Training

1. **function to find the classification table and accuracy**

   made a function to print the `classification_report` and accuracy using `sklearn.metrics`.

2. **Model Training**

   We Selected Trained the following classifiers:

   a. Decision Tree Classifier - It consists of root node, inner nodes, branches, and leaf nodes.Decisions are made on the basis of features selected in the dataset. The algorithm compares the value of the root feature with the feature’s values of the dataset, and in accordance with the comparison, it moves to the next nodes. The process continues until the leaf node is obtained.

b. Support Vector Classifier - This model is similar to neural networks in its objective of adjusting a set of parameters, which allow to establish boundaries in a dimensional space and approximate functions or separate patterns in different regions of the attribute space.SVMs base their training on maximizing the margin between the hyperplane and the instances of two classes.Hyperplane is the decision boundary that separates the class data. Support vectors are the data points that form close to the hyperplane.

c. XGB Classifier - This Algorithm implements machine learning algorithms under the Gradient Boosting Framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

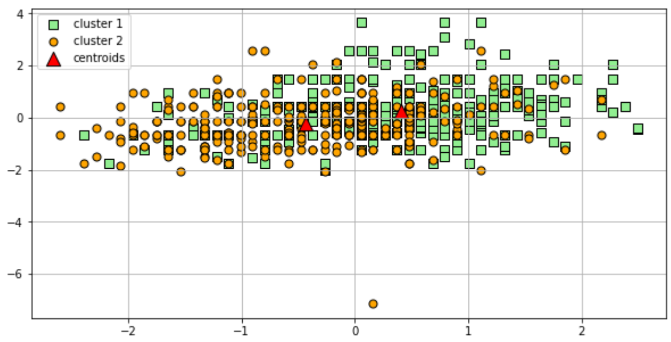
d. LGBM Classifier - Tt is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise

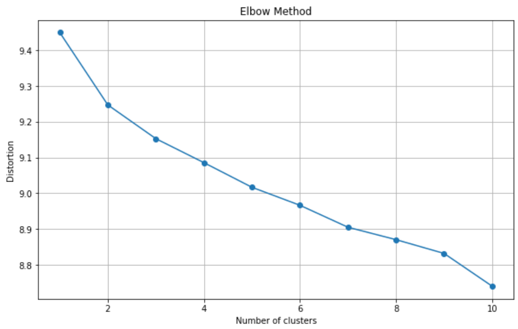
e. Random Forest Classifier - It works on the basis of ensemble learning, and solves the problem by combining several classifiers to improve the performance of the algorithm. The algorithm contains several classifiers of Decision Trees.

f. K Neighbors Classifier - This algorithm works on the basis of similarity of the state between the new data point and the stored dataset and classifies new data in accordance with the most similar class on the basis of the value of K and the closest one on the basis of Euclidean distance.

g. Deep Neural Network -Deep neural network represents the type of machine learning when the system uses many layers of nodes to derive high-level functions from input information. It means transforming the data into a more creative and abstract component.

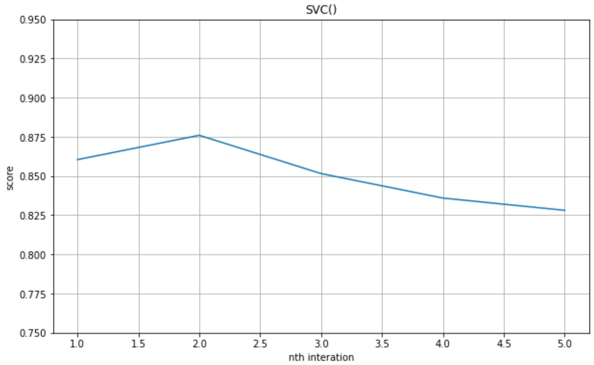
1. K Means Clustering



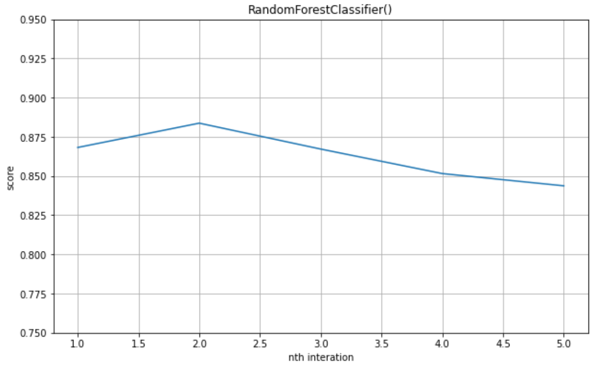


Cross-Validation

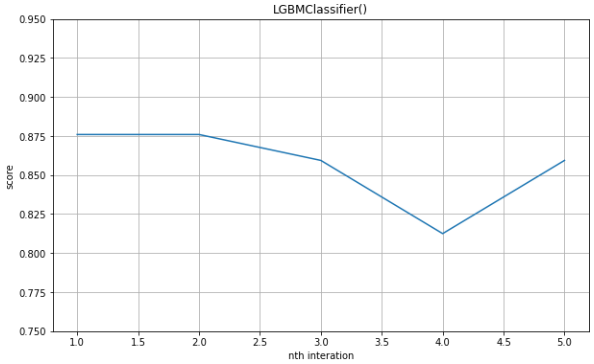
It uses a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model, this also significantly reduces bias as we are using most of the data for fitting, and also significantly reduces variance as most of the data is also being used in the validation set. Interchanging the training and test sets also adds to the effectiveness of this method.



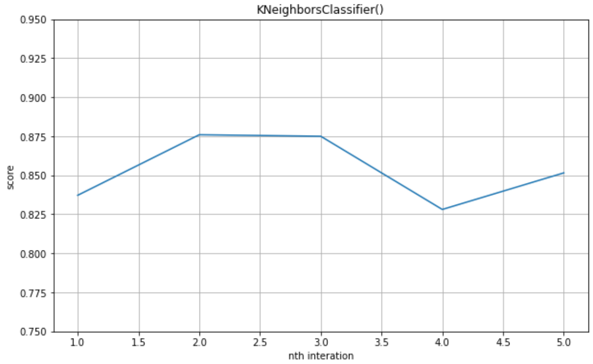
SVC() Cross Validation Score :
[0.86 0.87 0.85 0.83 0.82]



RandomForestClassifier() Cross Validation Score :
[0.87 0.89 0.85 0.82 0.86]



LGBMClassifier() Cross Validation Score :
[0.87 0.87 0.85 0.81 0.85]



KNeighborsClassifier() Cross Validation Score :
[0.83 0.87 0.87 0.82 0.85]

Model Comparison

Model	Accuracy Score	F1 Score	Recall Score	Precision Score
Decision Tree Classifier	0.7572463768115942	0.7641196013289037	0.725609756097561	0.8439716312056738
Support Vector Classifier	0.894927536231884	0.9113149847094801	0.9085365853658537	0.9141104294478528
XGB Classifier	0.8695652173913043	0.8853503184713376	0.8475609756097561	0.9266666666666666

LGBM Classifier	0.8623188405797102	0.8812500000000001	0.8597560975609756	0.9038461538461539
Random Forest Classifier	0.8768115942028986	0.8944099378881988	0.8780487804878049	0.9113924050632911
K Neighbors Classifier	0.8804347826086957	0.897196261682243	0.8780487804878049	0.9171974522292994
Deep Neural Network	0.894927536231884	0.9118541033434651	0.9146341463414634	0.9090909090909

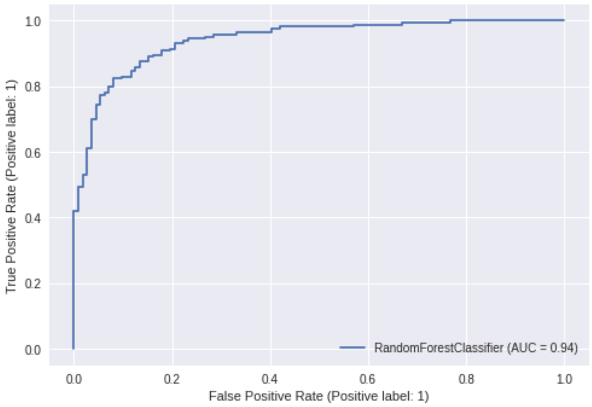
Hyper-parameter Optimization

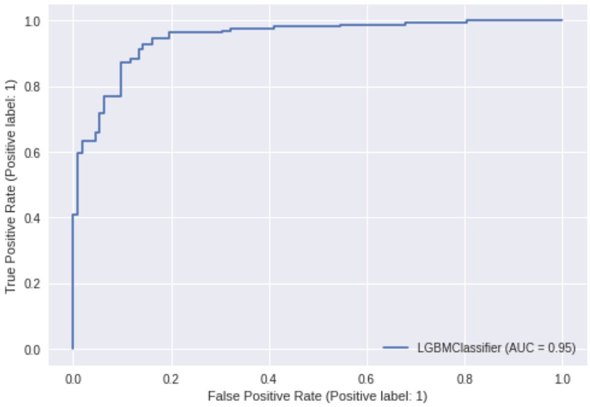
We use Hyper-parameter Optimization in order to find the best parameters to improve a given model.

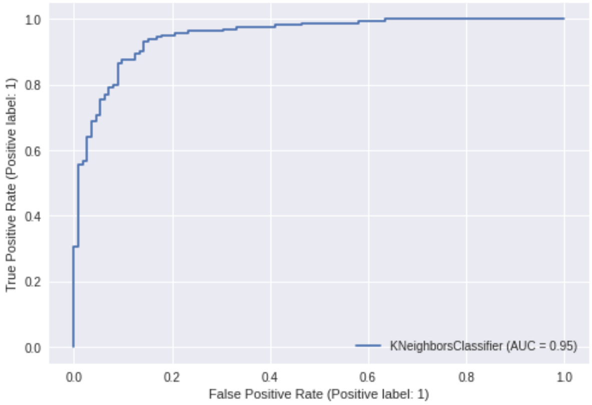
It is done by using `RandomizedSearchCV` and GridSearchCV

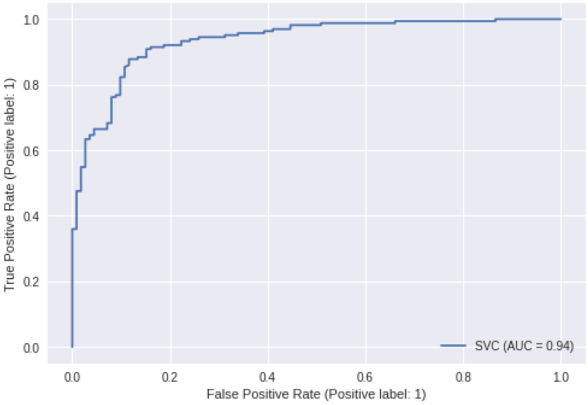
```
#best parameters for SVC()
{'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

#best parameters for RandomForestClassifier()
{'max_depth': 20,
 'max_features': 'auto',
 'min_samples_leaf': 10,
 'min_samples_split': 5,
 'n_estimators': 700}

#best parameters for LightGBMClassifier()
{'colsample_bytree': 0.6,
 'learning_rate': 0.01,
 'max_depth': 50,
 'n_estimators': 500}

#best parameters for KNN
{'metric': 'minkowski', 'n_neighbors': 30, 'p': 1, 'weights': 'distance'}
```

ROC PLOTS OF BEST MODELS

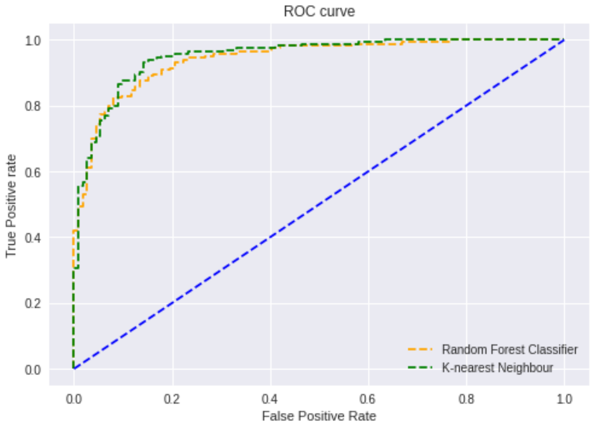


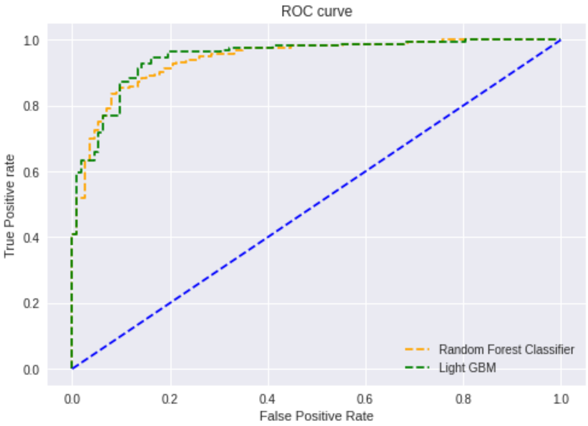


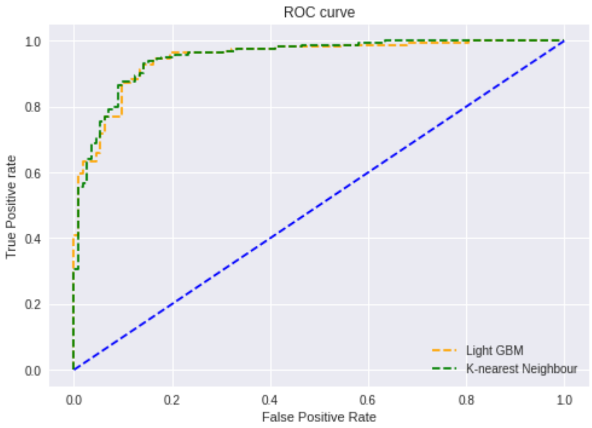


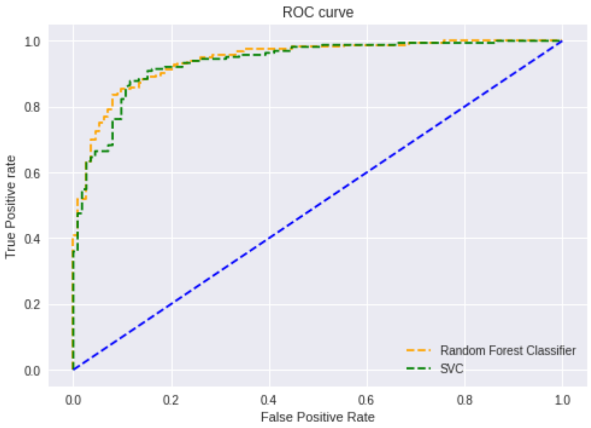


ROC CURVE COMPARISON









RESULTS AND ANALYSIS:

The Exploratory Data Analysis was used to observe the features with a strong correlation with the target feature, which gave us the relation between each feature and the target feature. also Feature engineering methods like One-hot Encoding, and Standard Scaling was also applied to increase the number of correlated features between them and trained machine learning models to obtain reliable results that were better than the results obtained from the original dataset. The Hyperparameter Tuning of some Machine learning algorithms improved the performance significantly, All the algorithms reached superior results during the training and testing phases. However,as we can observe from The above table and plots that most of the classifiers classifiers had a nearly equally efficient performance.During the testing phase, SVM and KNN achieved better results than the rest of the algorithms. Deep Neural network also performed very well,compaing to other decision tree based algorithms. During testing of model we have to be carefull about precsion and recall, as these models will be implemented in Healthcare sector.we have to be specially carefull about prediction of True Negative cases as here danger of life involved. There are some limitations to the our models, that is limited number of data due to it may be not be generalized for real world application.