Pattern Recognition & Machine Learning (CSL 2050)

Heart Failure Prediction

Abstract

This report brings our analysis of building a Heart Failure Prediction classifier wherein the dataset contains the information of people suffering from heart diseases due to the presence of one or more risk factors. We implemented various classification algorithms and compared their results based on various evaluation metrics. We also thought of a new model and implemented our own class which we named as DRY(), which can combine the predictions & results of chosen models and can bring out new classification results based on some mathematical equations.

MAJOR PROJECT Github Link

Riyanshu Jain (B20AI060)
Viradiya Dhruvkumar (B20CS079)
Maniya Yash RajeshBhai (B20CS033)

Problem Statement

Heart failure is a common event caused by cardiovascular diseases (CVDs). 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, hyperlipidemia or already established disease) need **early detection and management wherein a machine learning model can be of great help.**

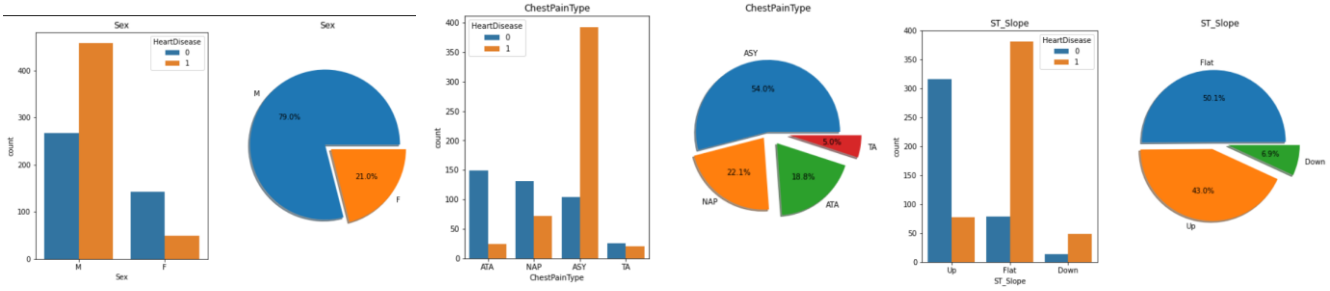
Analysis of the Dataset

Dataset Link

- The **dataset contains 11 features** that can be used to predict a possible heart disease
- It contains **6 categorical features** (Sex, ChestPainType, RestingECG, ST_Slope, FastingBS, ExerciseAngina) and **5 continuous features** (Age, RestingBP, Cholesterol, MaxHR, Oldpeak) along with a **label column named as HeartDisease** which is to be classified
- The description of the 918 rows and 12 columns is as follows –

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
count	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000	918.000000
mean	53.510893	0.789760	0.781046	132.396514	198.799564	0.233115	0.989107	136.809368	0.404139	0.887364	1.361656	0.553377
std	9.432617	0.407701	0.956519	18.514154	109.384145	0.423046	0.631671	25.460334	0.490992	1.066570	0.607056	0.497414
min	28.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	60.000000	0.000000	-2.600000	0.000000	0.000000
25%	47.000000	1.000000	0.000000	120.000000	173.250000	0.000000	1.000000	120.000000	0.000000	0.000000	1.000000	0.000000
50%	54.000000	1.000000	0.000000	130.000000	223.000000	0.000000	1.000000	138.000000	0.000000	0.600000	1.000000	1.000000
75%	60.000000	1.000000	2.000000	140.000000	267.000000	0.000000	1.000000	156.000000	1.000000	1.500000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	603.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	1.000000

Data Visualization –



- From the plots of Sex column, we can see that the values female, have more chances of no heart disease, whereas the values male, have a higher chance of suffering from heart disease.
- Similarly, ChestPainType ASY has a high chance of being suffering with heart disease, and ST_slope with value flat has higher chance of being suffering with a heart disease.
- Hence, we can say that these features may have a high importance while doing the classificaion.

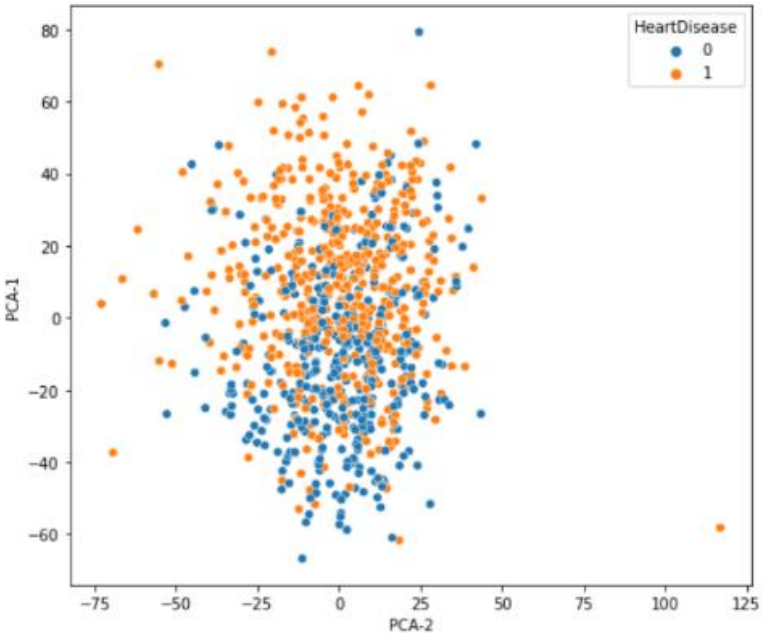
Preprocessing –

Firstly, we found the correlations of the features with respect to target column and the results are as follows –

```
HeartDisease      1.000000
ExerciseAngina    0.494282
Oldpeak           0.403951
Sex               0.305445
Age               0.282039
FastingBS         0.267291
RestingBP         0.107589
RestingECG        0.057384
Cholesterol      -0.232741
ChestPainType    -0.386828
MaxHR            -0.400421
ST_Slope         -0.558771
Name: HeartDisease, dtype: float64
```

We can conclude that the features ExerciseAngina, Oldpeak and Sex are the most correlated among all the features.

After this, we encoded all the categorical columns with the help of LabelEncoder().



Then we applied PCA to figure out the principal components that can store upto 99.9 percent of variance, and the following was the result –

- The data got reduced to 4 features
- 1 feature showed upto 90 percent variance, 2 features showed upto 95 percent variance, 3 features showed upto 99 percent variance
- The accuracy score was about 71 % on the Random forest classifier with precision scores also about 70%

The plot clearly shows that the data is not linearly separable, neither it gives any infromation so that we can apply the KMeans clustering and the chances are high that clustering also may not give appropriate results.

- **Next, we divided the data into training, validation and testing set in a ratio of 20 percent testing, 70 percent training and 10 percent validation.**

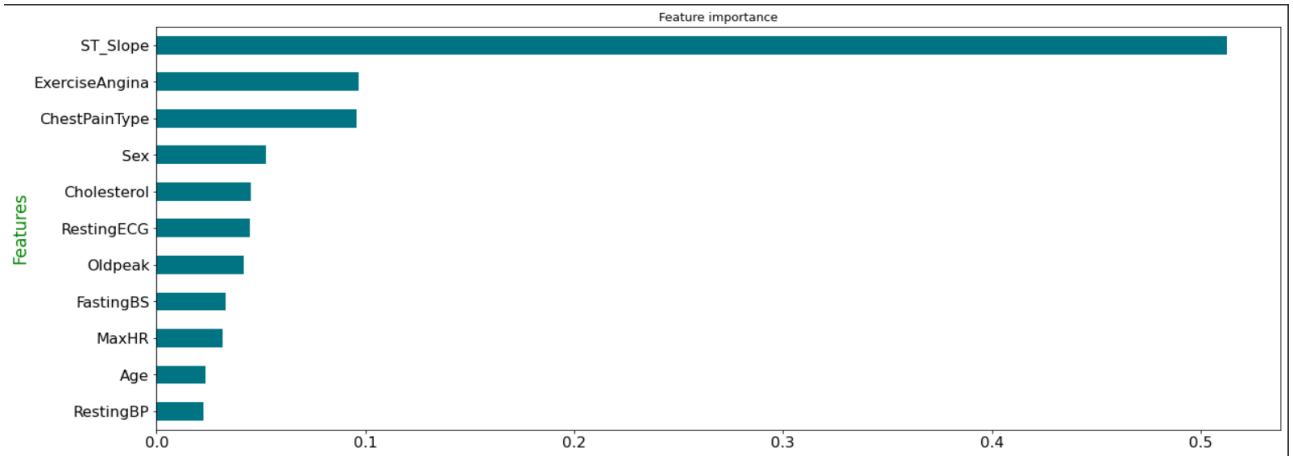
Methodology Adopted

- We applied the following models on the training dataset and then tested the scores on validation set –

Models	Accuracy score	Precision (Class 0)	Precision (Class 1)	Recall (Class 0)	Recall (Class 1)	F1score (Class 0)	F1score (Class 1)
LightGBM	0.91	0.95	0.88	0.86	0.96	0.90	0.92
Random Forest	0.90	0.93	0.88	0.86	0.94	0.89	0.91
XGBoost	0.89	0.90	0.88	0.86	0.92	0.88	0.90
Gradient Boosting	0.88	0.88	0.88	0.86	0.90	0.87	0.89
GaussianNB	0.85	0.83	0.87	0.86	0.83	0.84	0.85
AdaBoost	0.83	0.82	0.83	0.82	0.83	0.82	0.83
Logistic Regression	0.83	0.84	0.84	0.82	0.85	0.83	0.85
Decision Tree	0.80	0.81	0.80	0.77	0.83	0.79	0.82

Considering the above table we have decided the best 5 models based on the accuracy score and F1 score as LightGBM, Random Forest, XGBoost, Gradient Boosting and Gaussian Naïve Bayes.

- The feature importance from one of the models is shown below and as we have predicted from the data visualisation that ST_slope, Sex and ChestPainType will play a major role in predicting the heart failure chances –



Next, we concatenated both the train and validation data to create a new training data and implemented a class based on this training data and tested the scores on the testing data. The functioing of the implemented class is as follows –

- **Input to the class**

  List of models (best 5 models selected above), metric (any of the metrics such as accuracy score, precision, f1-score etc.)

- **Methods implemented inside the class**

1. **Fit()** – Used to train all the models which are given as input and then it is storing the input metrics, class (0,1) predictions and predicted probabilites.

2. **Mode_accuracy()** – It will consider the predictions of all the models and then calculates the mode of the predictions.

   For eg : If 3 models predict the value as 1, and 2 models predict the value as 0, then it will consider the final prediction based on the majority and hence will store the final prediction as 1 in this case.

3. **Mean_proba_accuracy()** – Calculates the mean of the proabablities of all 5 models and then makes the prediction on the basis of the final calculated probability.

   For eg : If the predicted probabilities from the 5 models are [p1, p2, p3,, p4, p5] then the final probability will be -

$$p = \frac{p1 + p2 + p3 + p4 + p5}{5}$$

   Then this value will get rounded to find the prediction as class 0 or class 1, and then calculates the input metrics based on the test data.

4. **Weighted_mean_accuracy()** – Taking the prediction array of all the models as input and then calcualting the weighted mean of the predictions by taking the weights as the accuracy score of that model.

   *For eg : If the predicted accuracy from the 5 models are [a1, a2, a3, a4, a5] and the predicted classes are **[p1, p2, p3, p4, p5]** then the final probability will be –*

$$p = \frac{p1(a1) + p2(a2) + p3(a3) + p4(a4) + p5(a5)}{a1 + a2 + a3 + a4 + a5}$$

   Then this will be considered as the final probability to calculate the class 0 or class 1 prediction.

5. **Weighted_proba_accuracy()** – Taking the probability prediction array of all the models as input and then calcualting the weighted mean of the probabilites by taking the weights as the accuracy score of that model.

   *For eg : If the predicted accuracy from the 5 models are [a1, a2, a3, a4, a5] and the predicted probas are **[p1, p2, p3, p4, p5]** then the final probability will be –*

$$p = \frac{p1(a1) + p2(a2) + p3(a3) + p4(a4) + p5(a5)}{a1 + a2 + a3 + a4 + a5}$$

Then this will be considered as the final probability to calculate the class 0 or class 1 prediction.

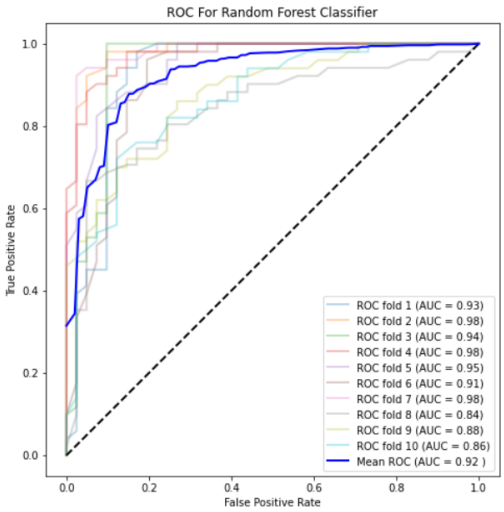
Results and Analysis

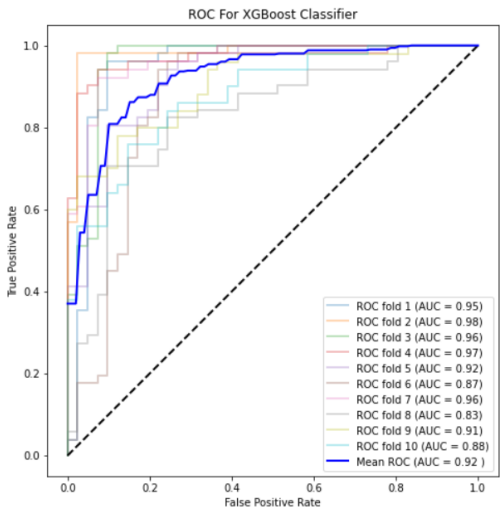
The results we got by implementing the above class on the best 5 selected models are –

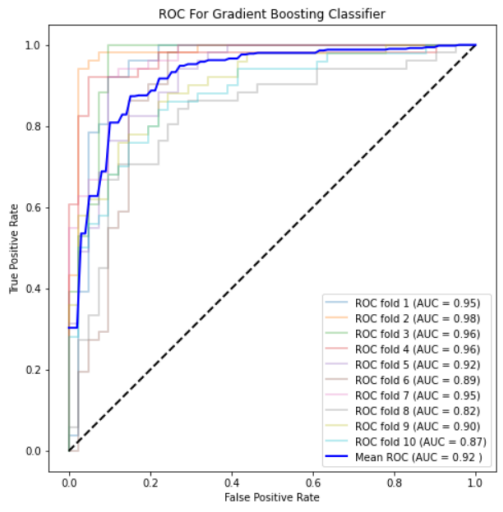
Models	Mode_ accuracy	Mean_proba_ accuracy	Weighted_mean_ accuracy	Weighted_proba_ accuracy
Accuracy score	91.30	90.00	91.30	90.00
Precision score	92.13	91.94	92.13	91.94
Recall score	92.13	89.76	92.13	89.76
F1-score	92.13	90.84	92.13	90.84

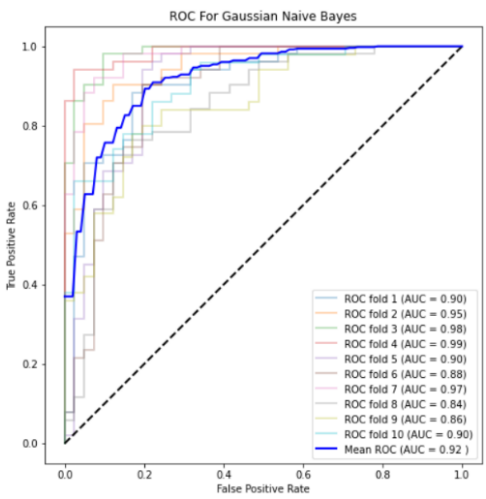
As we can see that, the final accuracy score increased by a slight amount, as in the previous table the max accuracy was about 91 percent and here the max accuracy is about 91.30 percent.

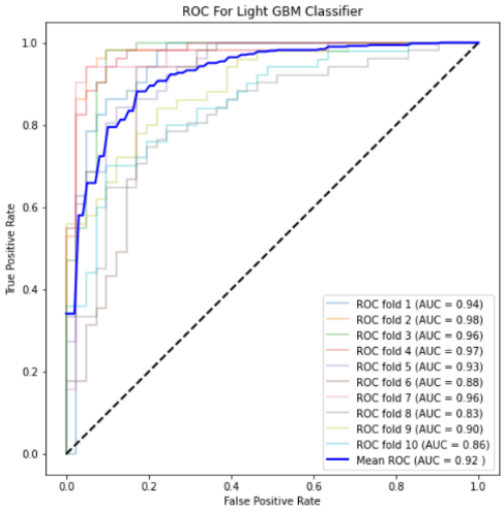
ROC plots and 10 fold cross validation scores –

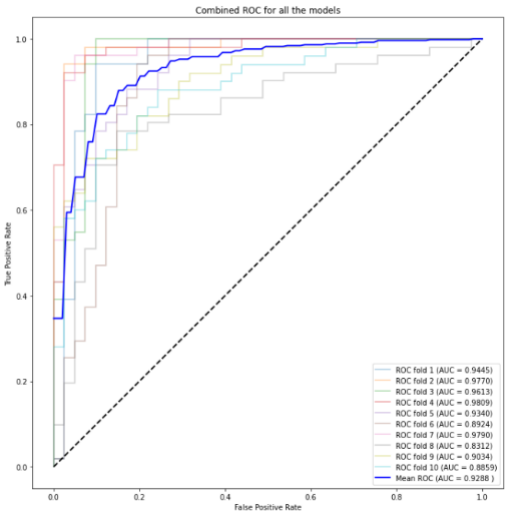












Contributions –

- **Riyanshu Jain (B20AI060) –** Report making, Exploratory Data Analysis & visualization, Pre-processing, Implemented XGBoost, Logistic Regression, Random Forest, Ideation of the new concept of DRY() class
- **Viradiya Dhruvkumar (B20CS079) –** Implementation of the DRY() class, Discriminant Analysis, ROC & AUC, Report, Ideation of the new concept of DRY() class
- **Maniya Yash Rajeshbhai (B20CS033) –** Implementation of Gaussian NB, Gradient Boosting, Light GBM, Decision Tree, AdaBoost, Report, Deployment of the final model

Major work in the coding section is done by Dhruv along with some work in deployment & preparing the presentation, **major work in the report content and video presentation is done by Riyanshu** along with some work in code domain, and **the deployment work was solely completed by Yash** along with a good amount of work in colab too.
At the end, we have contributed equally in every aspect of our expertise by collaborating with each other and looking on each other's work.

---------END OF REPORT---------