Heart Failure Prediction

Lakshya Kumawat, Hemant
B20AI019, B20EE024

1. Introduction:

Our project is on heart failure prediction which is a critical thing that can be a matter of life of a person. So we want that algorithms designed by us do not predict the wrong result especially when a person has heart failure and it predicts that the person does not have any condition of heart failure and this thing can even take the life of a person. This will come under the condition of false negatives. So we want false negatives as low as possible and accuracy of the model as high as possible. On this idea we are going to work in this model.Thus preventing Heart diseases has become more than necessary. Good data-driven systems for predicting heart diseases can improve the entire research and prevention process, making sure that more people can live healthy lives. This is where Machine Learning comes into play. Machine Learning helps in predicting Heart diseases, and the predictions made are quite accurate.

2. Dataset:

Our Dataset contains 12 features and 1 target variable. These are:

- **Age**: age of the patient [years]
- **Sex**: sex of the patient [M: Male, F: Female]
- **ChestPainType**: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- **RestingBP**: resting blood pressure [mm Hg]
- **Cholesterol**: serum cholesterol [mm/dl]
- **FastingBS**: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
- **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]
- **MaxHR**: maximum heart rate achieved [Numeric value between 60 and 202]
- **ExerciseAngina**: exercise-induced angina [Y: Yes, N: No]
- **Oldpeak**: oldpeak = ST [Numeric value measured in depression]
- **ST_Slope**: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- **HeartDisease**: output class [1: heart disease, 0: Normal]

3. Methodology:

Overview

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

- Decision Tree Classifier
- Gaussian Naive Bayes
- K-Mean Clustering
- Neural Networks: MLP
- Random Forest
- XGBoost Classifier

Exploring the Dataset and pre-processing:

There is no null value present in the dataset.

- Do the label encoding of all categorical features.
- Scale the data using *StandardScaler*.
- Split the data into a testing and training set.

Implementation of various regression algorithms:

- **Decision Tree Classifier**: The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute. I tune various hyperparameters like: *criterion, max_depth, max_leaf_nodes, min_sample_split, random_state=42*.

- **Gaussian Naive Bayes**: Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution. An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions. As we can see some of our continuous variables follow Gaussian distribution. So, we use this algorithm.

- **K-Mean Clustering**: K-Means Clustering is an unsupervised learning algorithm which groups the unlabeled dataset into different clusters. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

- **Neural Networks: MLP:** Neural networks, also known as artificial neural networks (ANNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. We are using shallow neural networks, not the deep, with only two hidden layers and tune hyperparameters.

- **Random Forest:** Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Hyperparameters taken are:
  max_depth=6, random_state=0,n_estimators=500

- **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. Boosting is an ensemble modeling technique which attempts to build a strong classifier from the number of weak classifiers. It is done by building a model using weak models in series. First, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

We have defined a loss function, to determine which hyper-parameter works better for a particular algorithm. That loss function is: (100*FN) + FP.
Because in our case we want False negatives minimum, because it is related to a person's life. So, in loss function we give more weightage to false negatives than false positives.

4. Evaluation of Models:

We can evaluate our models on different parameters. Here I am going to evaluate using the Accuracy, CV Score and False negative rate. False Negative Rate is important for us.

Table 1.1 : Comparison of Different Models on the basis of different parameters.

Model	Accuracy	CV Score	False Negative Rate
Decision Tree	81.52%	84.58%	7.84%
Naive Bayes	87.23%	84.73%	13.07%
K-Mean Clustering	77.90%	54.84%	20.9%
MPL	81.15%	83.47%	11.76%
Random Forest	84.42%	88.15%	7.20%
XGBoost (Shallow)	84.78	95.15%	9.15%
XGBoost (Deep)	82.6%	100%	9.15%

Plots:

Fig. 1.1 Bar graph of Accuracy vs Models

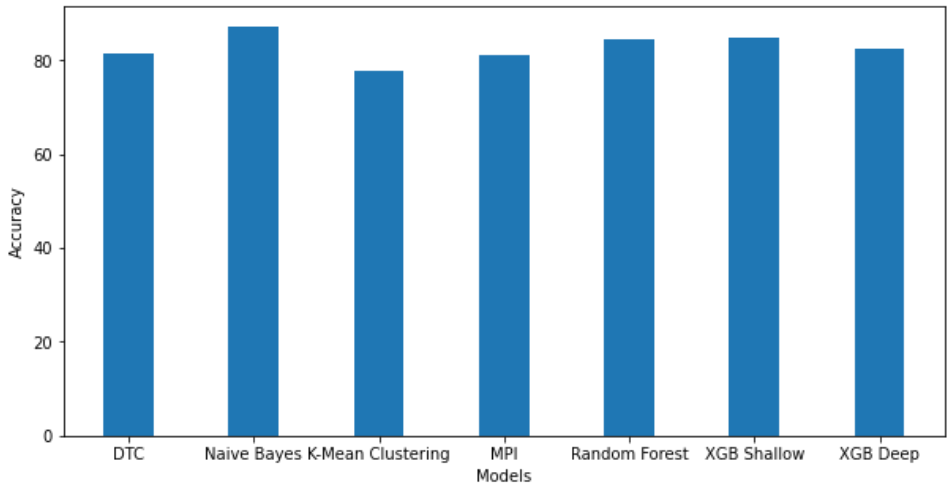
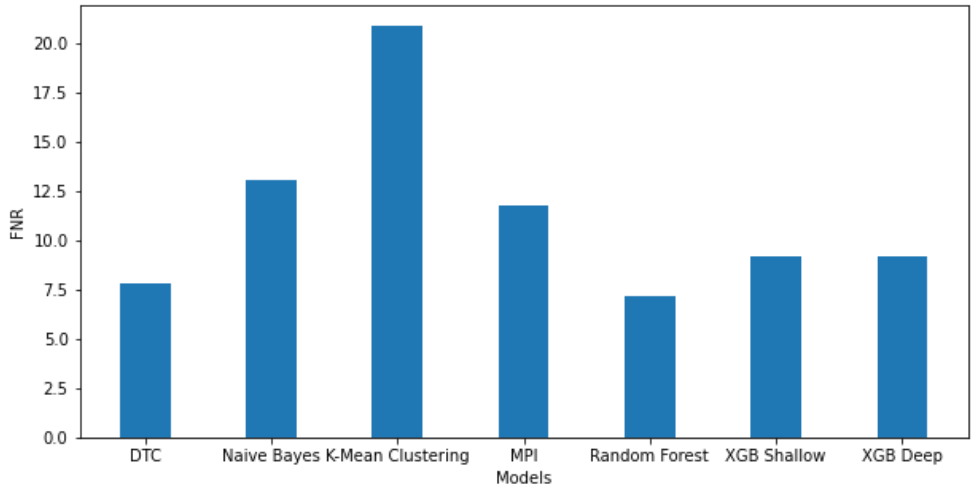


Fig. 1.2 Bar graph of FNR vs Models



5. Result & Analysis:

- We can see all the implemented models give similar results if we compare accuracies as in Table 1.1., means there is very little difference in their accuracy.
- In terms of accuracy Gaussian Naive Bayes is working best (87.23%), but False Negative Rate is high (13.07%). It gives the highest accuracy, because we can see that continuous features follow gaussian distribution and this is the main assumption we took in it.

Example

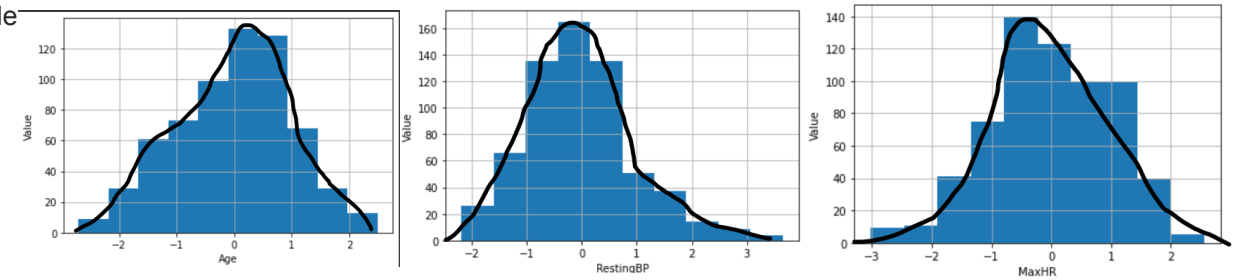


Fig 1.3 Data following Gaussian distribution

- XGBoost gives a very high CV score and decent accuracy and FNR, so it also works fine.
- The K-mean clustering model gives very less accuracy(77.9%) and CV score(54.84%) and also gives very high FNR(20.9%). So, this is the worst model in this case. We can also see from the distribution of data that classes are not forming proper clusters in 2-D(there may be a chance that they form proper clusters in higher dimensions as our data is of 12 dimensions), due to which this algorithm is not working properly.

Example

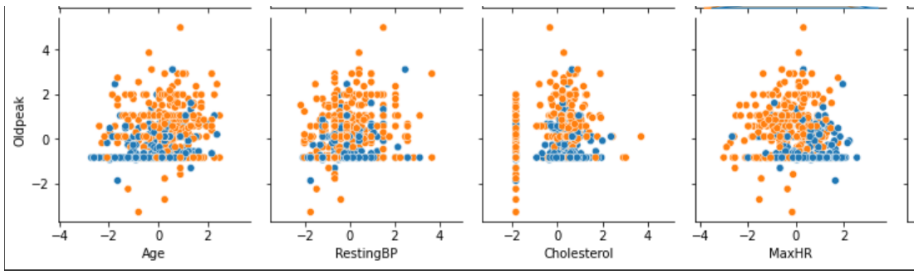


Fig 1.4 Data not following Gaussian distribution

- In XGBoost the overall model seems to underfit, as it performs very well in CV score but overall accuracy is very less.
- Random forest algorithm gives minimum FNR, which is 7.2% only.
- **Final Result**: We can conclude that overall Random Forest is working best, because:
    - It gives the highest accuracy after Naive Bayes, but naive bayes give high FN rate, so we didn't choose it.
    - CV score is also better, even more than overall accuracy, so, model is also not overfitting nor underfitting.
    - Gives Lowest False Negative Rate, which is the most important factor for us.

Contributions:

The learning and planning is done as a team. And individual contribution is given as:

- **Lakshya Kumawat** => Data preprocessing and exploratory analysis, K-mean Clustering, Random Forest, Neural Networks and Report
- **Hemant** => Decision Tree Classifier, Gaussina Naive Bayes, XGBoost Classifier, Report

References:

- Sklearn documentation
- Cross Validation Explained: Evaluating estimator performance
- Understanding Random Forest.
- Understanding K-means Clustering in Machine Learning
- Deep Learning with Python: Neural Networks (complete tutorial)
- Class Notes, Slides and Lectures