Heart Disease Prediction using Machine Learning Techniques

Kasarla Mani Kumar (MT19065), Nikhil Kolla (MT19123), Ritesh Singh (MT19044)
Indraprastha Institute of Information Technology, Delhi

Abstract - Heart Disease has become one of the most common reasons for death these days. Predicting heart disease using clinical data analysis is not a very easy job. Machine learning has become the most emerging field in computer technology for analysis and prediction techniques. Many researchers are trying to apply these Machine learning techniques in the healthcare field predicting the disease by analyzing the data. We have read a lot of research papers about the usage of Machine learning techniques for predicting heart disease and felt the topic is worth exploring. The dataset used is UCI heart disease prediction dataset. We have performed required data analysis for all the features present and applied various ML models. The models include Naïve Bayes, Logistic Regression, Decision trees, Multi layer perceptron, Random Forests, Support Vector Machines, K – Nearest Neighbours, XG Boost and we enhanced performance of models with better values for evaluation metrics..

1. Introduction

It is quite difficult to detect and identify heart related disease as there are several risk factors like high cholesterol, irregular blood pressure, abnormal pulse rates and many. Several machine learning and data analysis techniques are used in the field of heart disease prediction. The severity of the disease is classified by K Nearest Neighbours, Naive Bayes, Decision trees and genetic algorithms. As the diagnosis and identification of heart disease is difficult when compared to other related diseases, it must be taken care for predicting and classifying heart related diseases or else it may cause heavier damage to the individual or may even cause premature death. The analysis algorithms in data analysis and patient data from the medical research field can help in predicting heart related disease.

We have also come to know from our literature survey that decision trees are also used in predicting heart disease and along with the algorithms in data mining, prediction of heart disease has become one of the famous problems in the medical and machine learning field together. We have not only used various machine learning techniques for prediction of heart diseases but also tried to relate two or more techniques for getting better results. These new methods or techniques can be called hybrid techniques. We also implemented neural networks which use many clinical records of a patient in predicting the disease related to heart.The dataset is divided into training and testing parts for evaluating our developed models.

For experimental study, we used a UCI machine learning repository dataset for heart disease prediction. The further sections discuss about our models implemented and the accuracy values obtained in detail.The heart disease is predicted based on factors like age, sex, pulse rate, blood pressure levels and many. The ML algorithm implemented with neural networks looks to produce good results. Generally, neural networks are considered as the best model for heart disease prediction and brain disease prediction.

3. Materials and Methods

3.1. Dataset Description

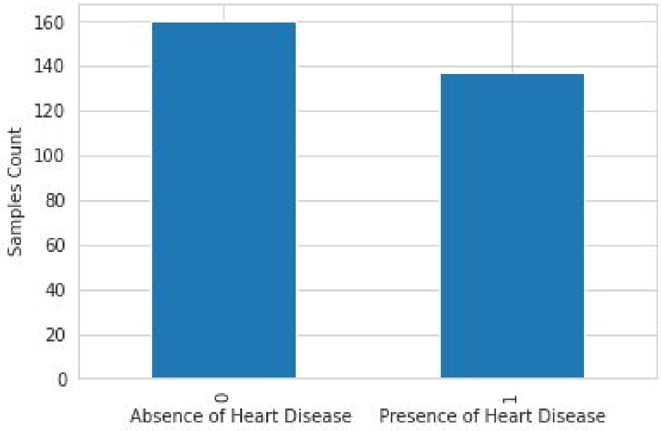
The dataset we used is the Heart Disease prediction dataset which is publicly available in UCI Machine learning repository. The dataset contains 13 features in total which are age, sex, cp(type of chest pain), blood pressure level in rest mode, cholestrol levels, fasting blood pressure levels, ECG results during rest period, Maximum heart rate, Angina induced by exercises, ST depression induced vs rest, ST depression slope is measured above, Flouoroscopy, Heart status and the target class is having binary value, where ‘0’ indicates absence of heart disease and ‘1’ indicates presence of heart disease. The dataset is having 297 samples, we have divided the dataset into 3 parts (training, CV and test) for developing and tuning the models.

3.2 Evaluation Metrics

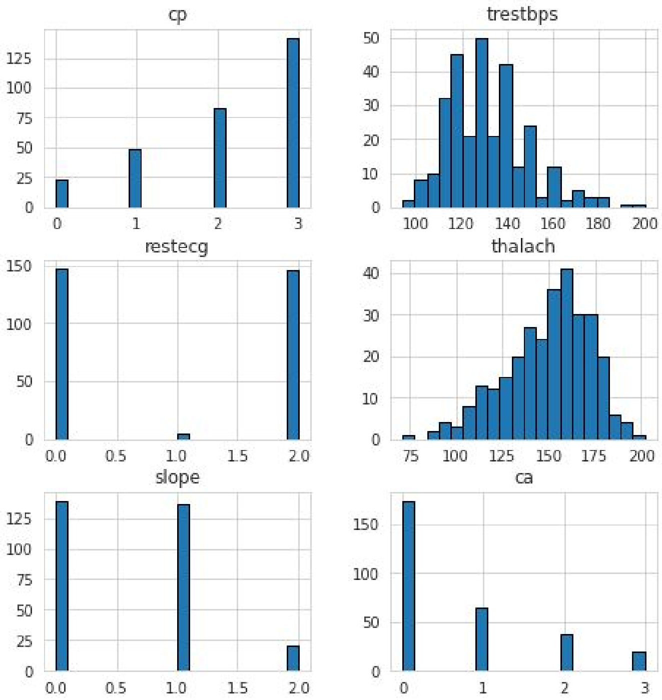
We have considered precision, recall, f1-score, specificity and accuracy as evaluation metrics/evaluation parameters. Precision is, count of true positives upon the sum of true positives (TP) and false positives (FP). In this case it can be defined as, the proportion of patients who are truly having diabetes are correctly classified as having diabetes. Recall is count of true positives upon sum of true positives and false negatives. It basically gives the true-positive rate of the model. F1-score is generally considered as a harmonic mean of both precision and recall. Specificity in this case can be defined as the proportion of patients who are not truly having diabetes and correctly classified as not having diabetes, which equals TN/(TN+FP).

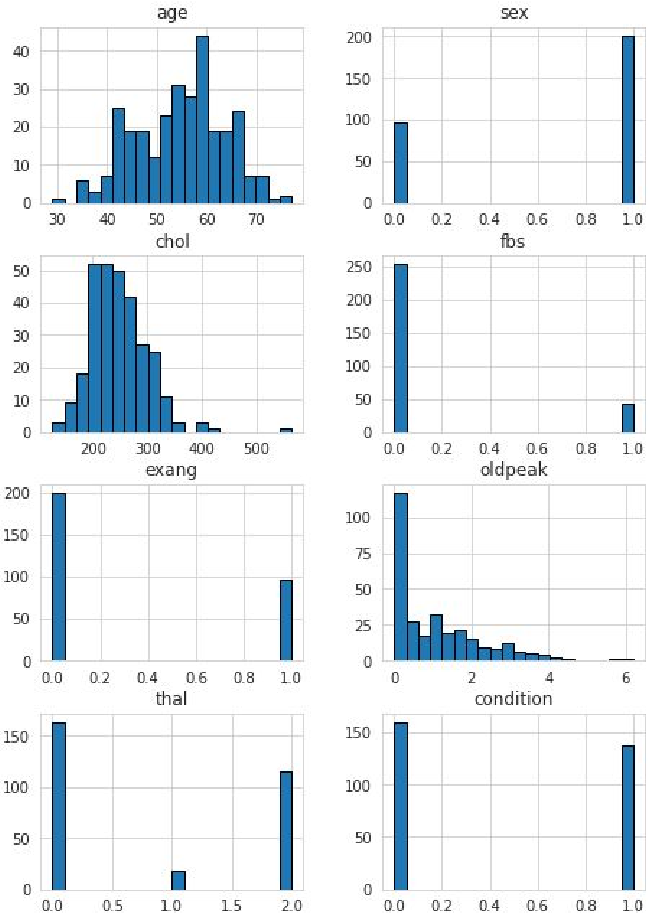
3.3 EDA (Exploratory Data Analysis)

The count of samples present per class in the dataset can be visualized using bar plot. The plot has labels on the X-axis and count of samples on the Y-axis. In the original data absence of heart disease had “0” class label whereas the presence of heart disease had class label “1”. Plot for the same can be seen here,

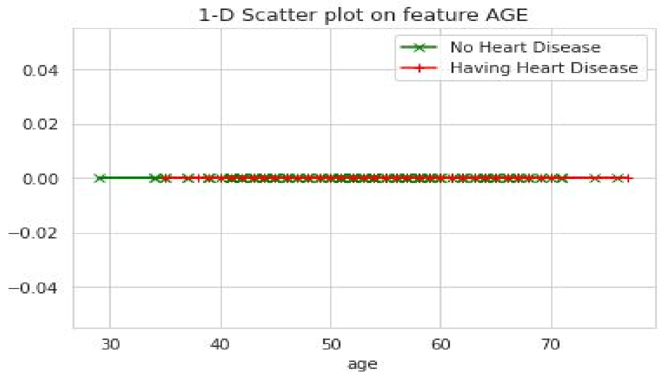


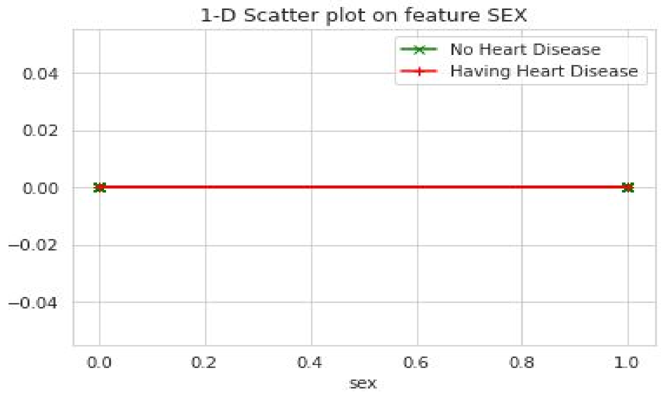
Count of non-diabetic samples is 160 whereas the count of diabetic samples is 137. A histogram is also plotted for every feature to understand the range of values present in them and to detect anomaly values. The plot for the same can be seen below,

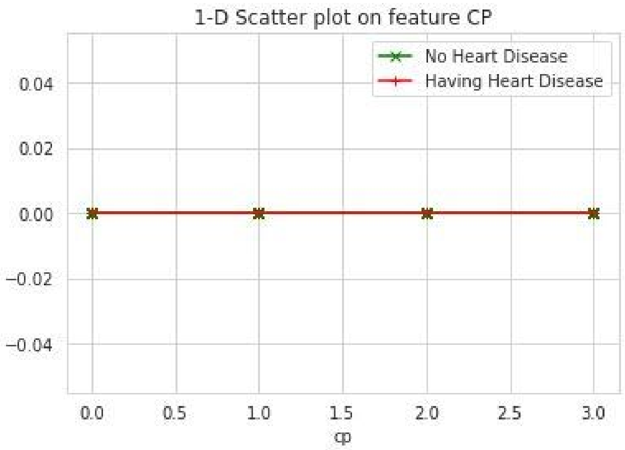




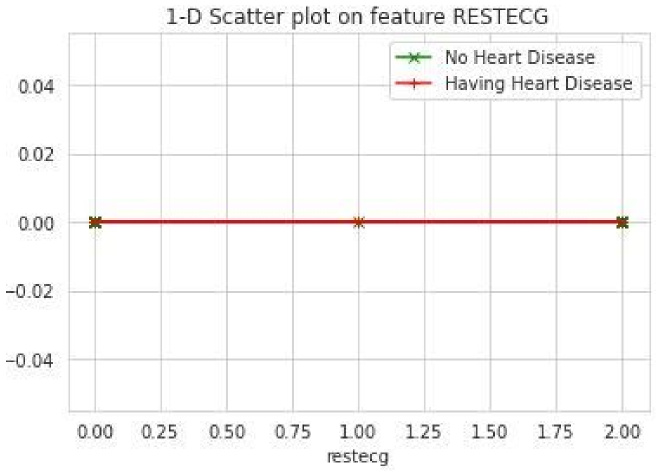
We performed univariate analysis with respect to every feature to understand their respective classification abilities. The plots with respect to every feature can be seen below,



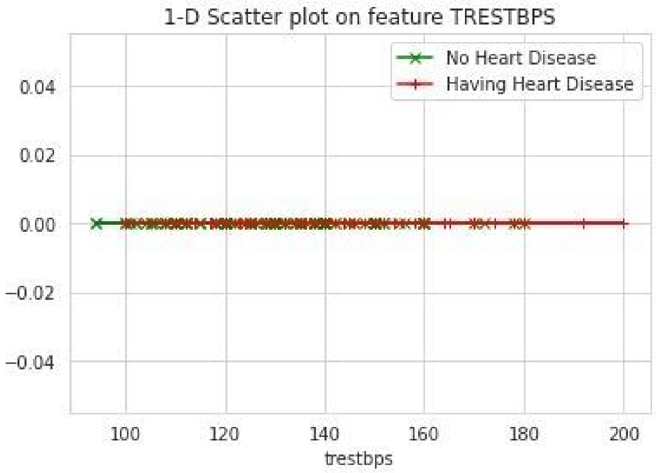




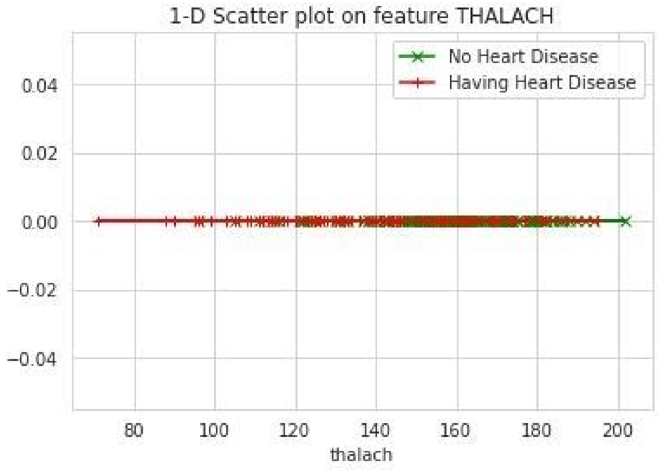
1-D Scatter plot on feature CP



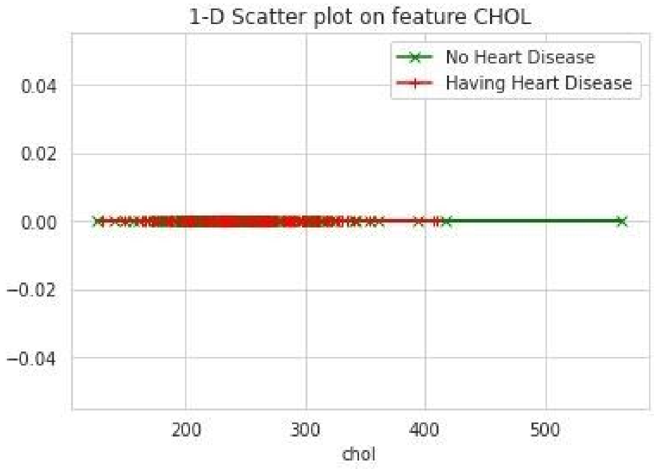
1-D Scatter plot on feature RESTECG



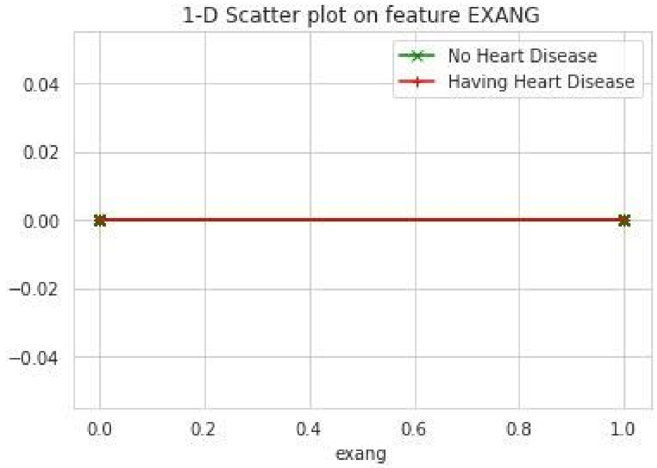
1-D Scatter plot on feature TRESTBPS



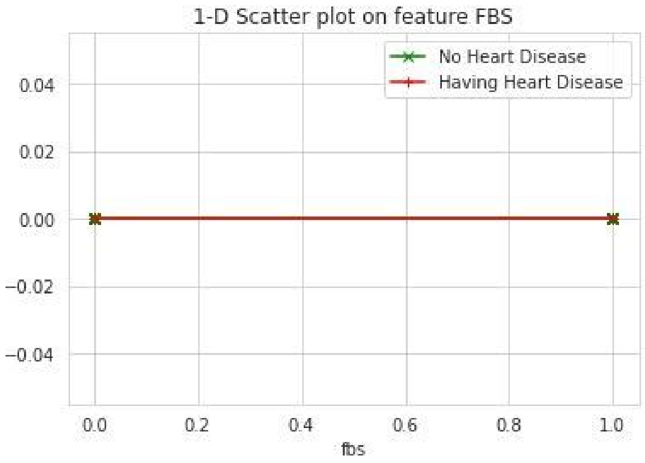
1-D Scatter plot on feature THALACH



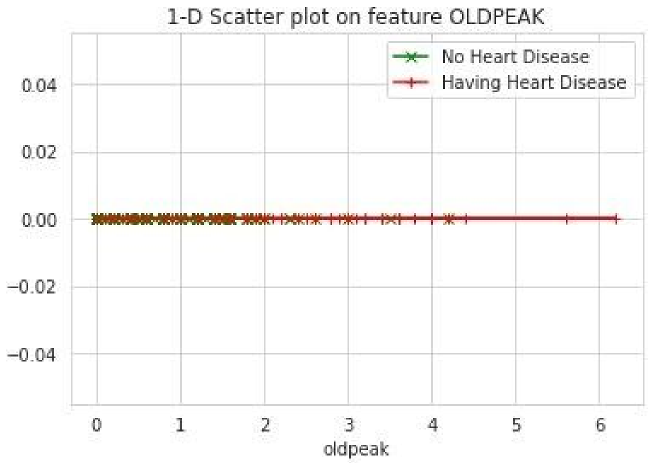
1-D Scatter plot on feature CHOL



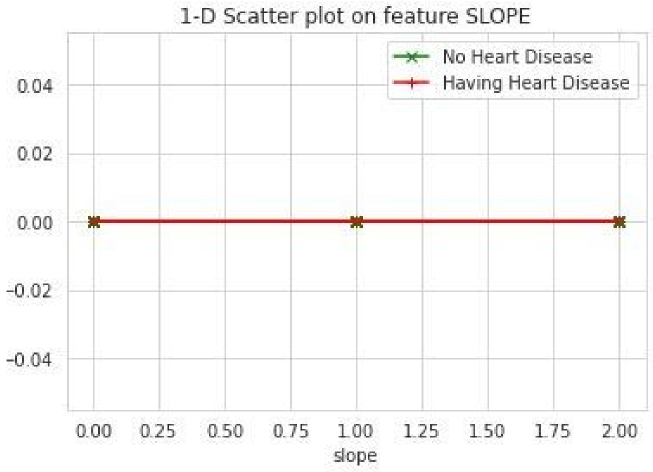
1-D Scatter plot on feature EXANG

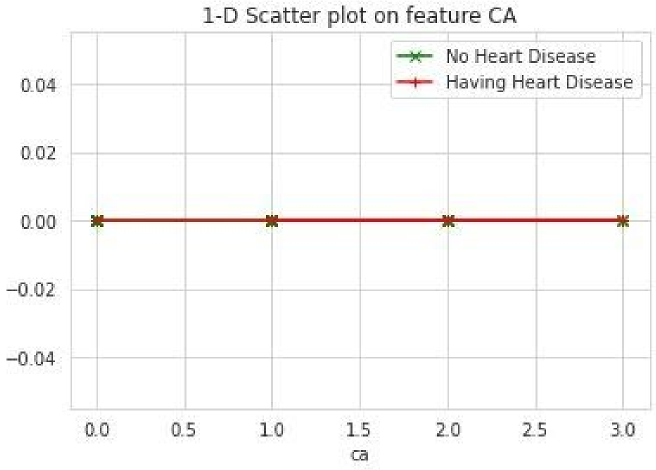


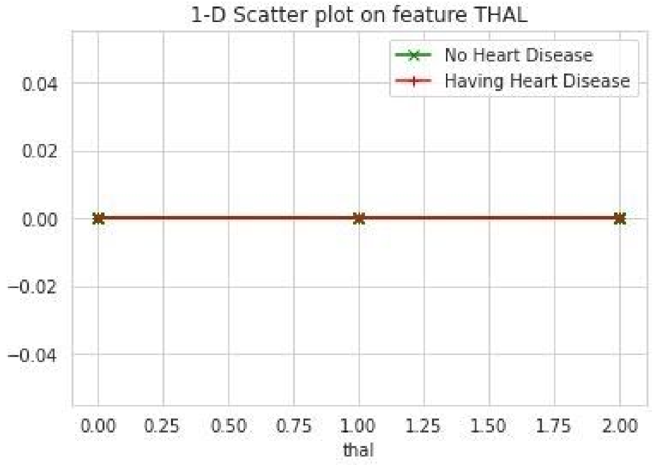
1-D Scatter plot on feature FBS



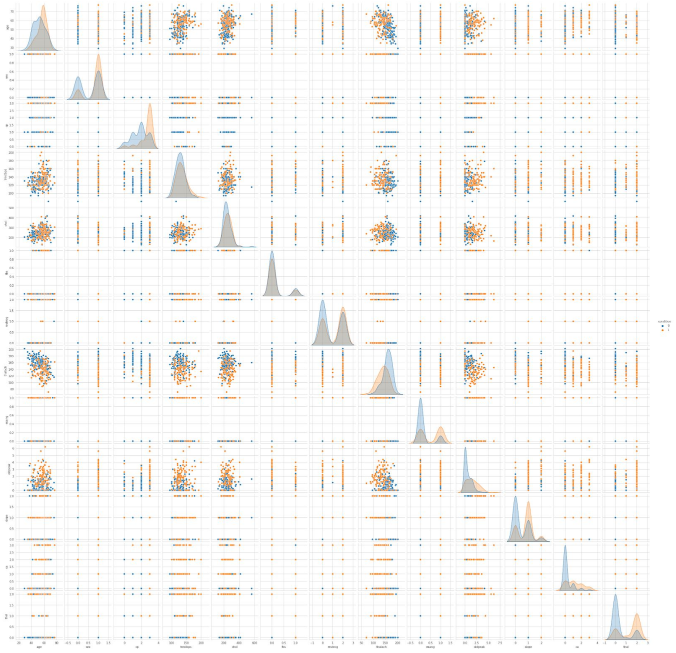
1-D Scatter plot on feature OLDPEAK





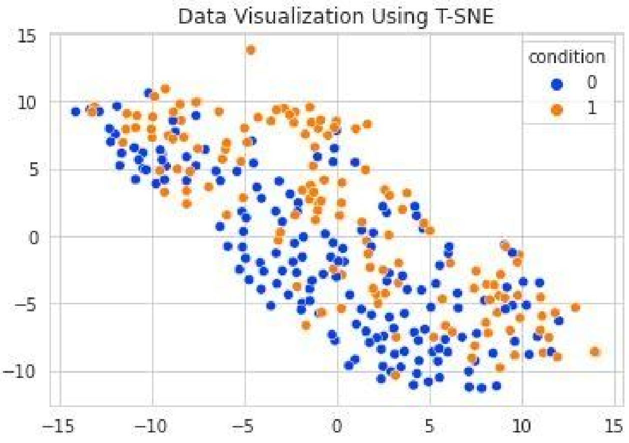


From the above plots of univariate analysis for every feature, we can clearly infer that none of a single feature individually can classify the data at least in a decent manner. As a part of Exploratory Data Analysis (EDA), the final task we had done was visualizing a pair-plot. Pair plot gives us the scatter plots for every two features in the data and also clear information of distributions for every pair of features. The plot can be seen below,



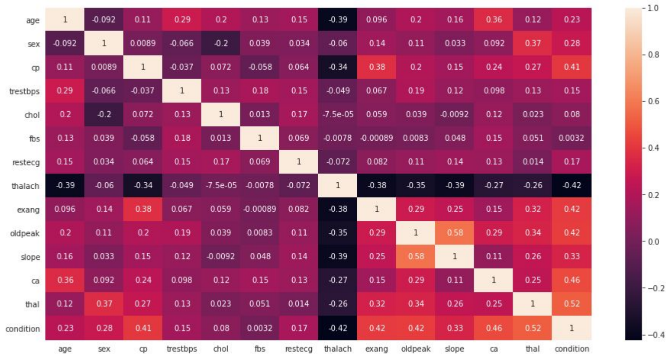
3.4 Data Visualization

We can see from the below plot drawn using tsne. We can infer that the data is not easily separated, i.eThe positive samples and negative samples both overlap each other.



3.5 Correlation Map

The correlation map gives the similarity between one feature with all the other features. For example, the first row, which is the feature “age” gives us the similarity in the data between “age” and all other features. Similarly, the last feature (condition), which is our target variable gives the similarity between the target variable and all other features. So, in the last row, no other value is above 0.7 which tells us that the feature selection will not work well for the given data.



3.6 Data Standardization

To avoid scaling problems, for distance-based learning techniques such as KNN, Logistic Regression, SVMs we have used the "MinMaxScaler()" library of "sklearn" in order to standardize the data.

3.7 Data split into train and test

We had split the data into train and test. The size of training data and test data are 70% and 30% of the total samples in the data (297 samples) respectively. The random state "42" is used for the split.

3.8 Machine learning techniques

3.8.1 Naive Bayes

In Naïve Bayes, when a test vector is given it calculates the probability of that test vector belonging to each class. P(C|Test-vector) is maximized. The class which gives maximum P(C|Test- vector) is the class label of that test vector. The smoothing factor "alpha" is the hyper-parameter in Naïve Bayes. We standardized the data using Minmaxscaler. We found out the best "alpha" using GridSearchCV technique with 5-fold cross validation on the basis of accuracy metric. Then after we predicted the labels for test data using the same "alpha" valu

3.8.2 Logistic regression

The name has regression, but this technique is only used for two-class as well as multi-class classification purposes. The logistic regression draws a decision boundary that separates two classes (might be >2 based on class labels). The logistic regression assumes that points to one side of the decision boundary or hyperplane belong to one particular class and points on the other side of the hyperplane belong to another class. We standardized the data using Minmaxscaler. The regularization coefficient "lambda" is the hyper-parameter in this technique. We used GridSearchCV with a 5-fold cross validation technique to determine the best value for lambda. The same lambda is used to predict labels on the test data.

3.8.3 Decision Trees

Decision tree is a tree-based classifier which is highly interpretable. Here each internal node represents a feature on which decision is made. Uses "Gini-impurity" (which works on the concept of entropy) technique in deciding what features to be considered as internal nodes. Entropy is defined as a measure of randomness in the data. The depth of the tree is the hyper-parameter here. We standardized the data using Minmaxscaler. We found out the best "depth" using GridSearchCV technique with 10-fold cross-validation basis of accuracy metric. Then after we predicted the labels for test data using the same "depth" value

3.8.4 K Nearest Neighbours

The KNN algorithm considers the k nearest neighbors for a test vector, takes the majority vote of class labels of its neighbors and assigns it as label to the given test vector. We standardized the data using Minmaxscaler. As, "k" is a hyper-parameter we found out the best "k" using GridSearchCV technique with 5-fold cross-validation considering accuracy as a performance metric for best estimator. Then after we predicted the labels for test data using the same "k" value that gave optimal performance on cv splits.

3.8.5 Linear SVM

SVM gives the hyperplane that best separates the data. It draws the margin maximizing hyperplane between two classes. The popular hyper parameters we have in SVM are kernel and "C" (regularization parameter). We have used only "linear" kernels for our project. Different kernels map the input data to the different feature spaces. We found out the best "C" using cross validation technique with 10-fold cross validation on the basis of accuracy metric. Then after we predicted the labels for test data using the same "C" value.

3.8.6 Gradient Boosting Decision trees

The base learners in boosting algorithms are models which have high-bias (high training error) and low variance combined with additivity. Decision stumps with maximum depth of 1 or 2 are the models which have high-bias and low variance. The base learner at a particular stage is trained on the x_i's and the errors that have been produced in the previous stage. So, as the number of base learners increases, we keep on trying to fit on the errors of the previous base learners thus reducing the training error. In gradient boosting, the negative value of gradient (derivative of loss function w.r.t x_i) at a point x_i is considered as $error_i$. The number of base learners to use is a hyper parameter in this technique. The training data considered has been up sampled before it was trained. The best value for hyper-parameter is found using cross validation with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data.

3.8.7 Multilayer Perceptron

This is a basic deep learning model, the simple neural network as the data is not with large numbers of samples, deep learning models will overfit on train data.

3.8.8 Random Forests

Random Forest is the most popular bagging technique used in current scenarios. This technique considers decision trees as base-learners with applying bagging on top and also uses column sampling strategy with aggregation. The trees considered here are of reasonable depth. The number of base learners to use is a hyper parameter in RFs. The training data considered has been up sampled before it was trained. The best value for hyper-parameter is found out using GridSearchCV with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data.

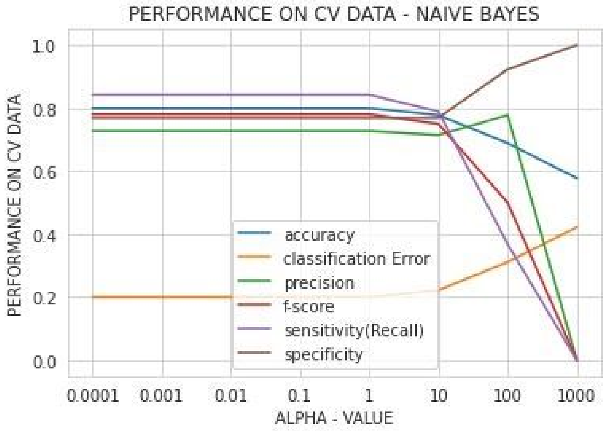
3.8.9 AdaBoost

At every stage, adaboost technique gives more weight to (can be done by up sampling) to the misclassified points. The weightage given to points exponentially increases from stage to stage. This creates a new dataset with a new plane of separation. Now, all these planes at every stage are applied on the dataset which eventually handles all the errors. The number of base learners to use is a hyper parameter in this technique. The training data considered has been up sampled before it was trained. The best value for hyper-parameter is found out using GridSearchCV with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data.

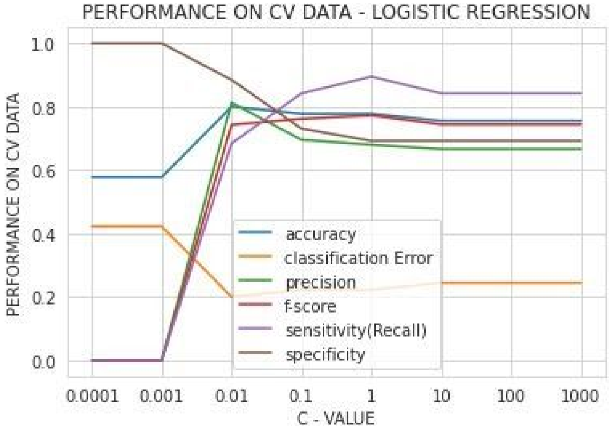
4. Results

We have used GridSearchCV technique with 5-fold cross validation and 10-fold cross-validation (for three techniques) in deciding the optimal hyper-parameters for a model. The plots are on CV data and tables of results are on test data. Every plot in this section has 4 curves each of different colors. As per the legend the blue color curve indicates how precision varies, yellow color curve indicates how recall varies, green color curve indicates how F-score varies, red color curve indicates how accuracy varies w.r.t respective hyper-parameters of the models. All plots are plotted based on performance of the model on CV data which can be seen below, here (CV) implies on "CV" data.

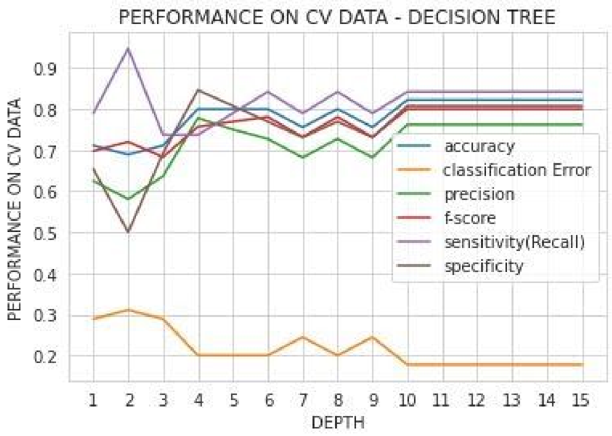
4.1 Alpha vs Performance metrics – Naïve Bayes



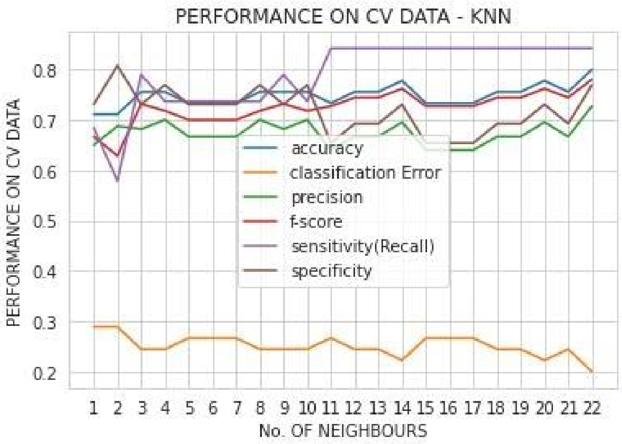
4.2 C vs Performance metrics (CV)-Logistic Regression



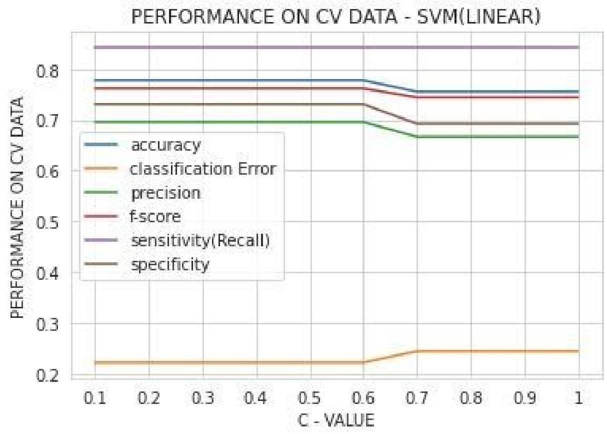
4.3 Depth vs Performance metrics (CV)-DecisionTrees



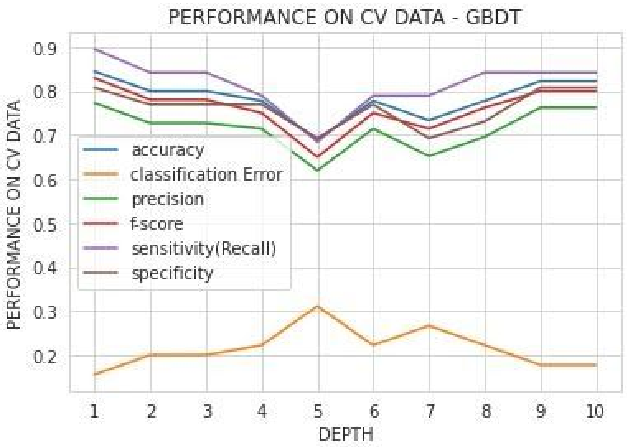
4.4 K vs Performance Metrics (CV) – KNN



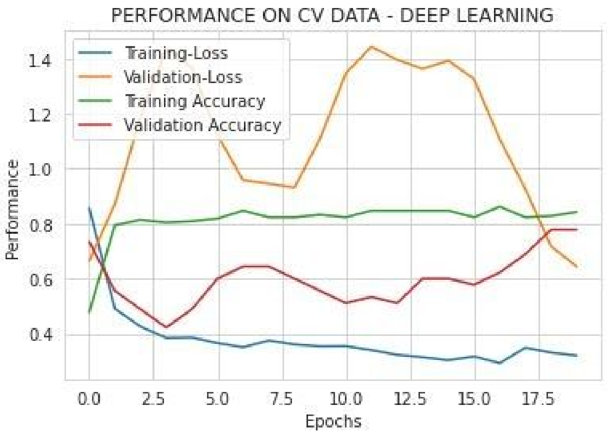
4.5 C v/s Performance metrics (CV) – Linear SVM



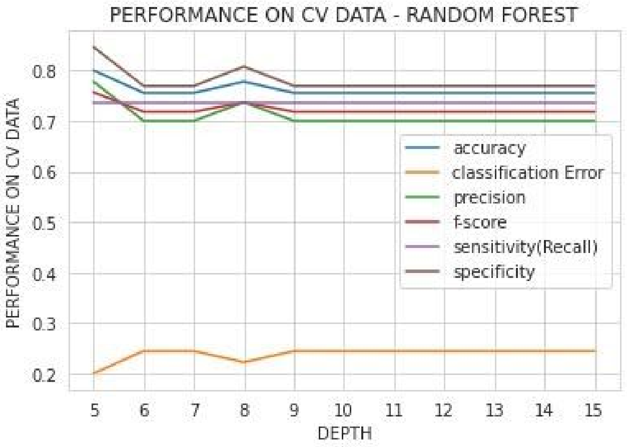
4.6 Estimator count vs Performance metrics - GBDT



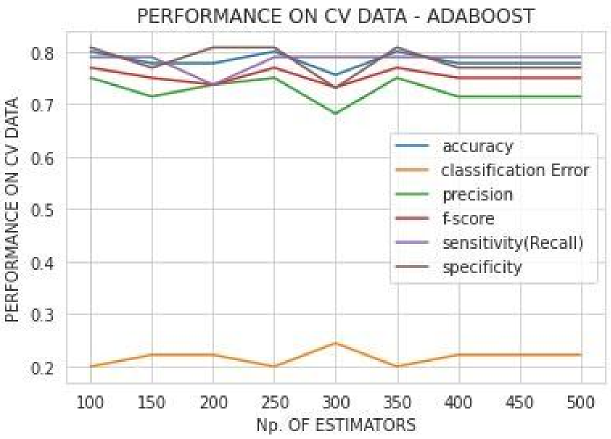
4.7 Deep learning model



4.8 Estimator count vs Performance metrics – RFs



4.9 Estimator count vs Performance metrics-Adaboost



4.10 Voting Classifier

As a part of implementing a voting classifier, the label for a test sample is computed by considering the majority vote of all predictions which were predicted by all the optimal models.

4.11 Results

S.No	Model	Accuracy in Paper	Accuracy we achieved
1.	Naive Bayes	75.8	83.7
2.	GBDT	78.3	86.0
3.	MLP	87.4	86.0
4.	KNN	Not used	83.7
5.	Linear SVM	86.1	90.6
6.	RFs	86.1	86.0
7.	Logistic Regression	82.9	83.7
8.	Ada Boost	Not used	72.0
9.	Decision Trees	85	83.7
10.	Voting Classifier	Not used	88.3

5. Analysis and Discussion

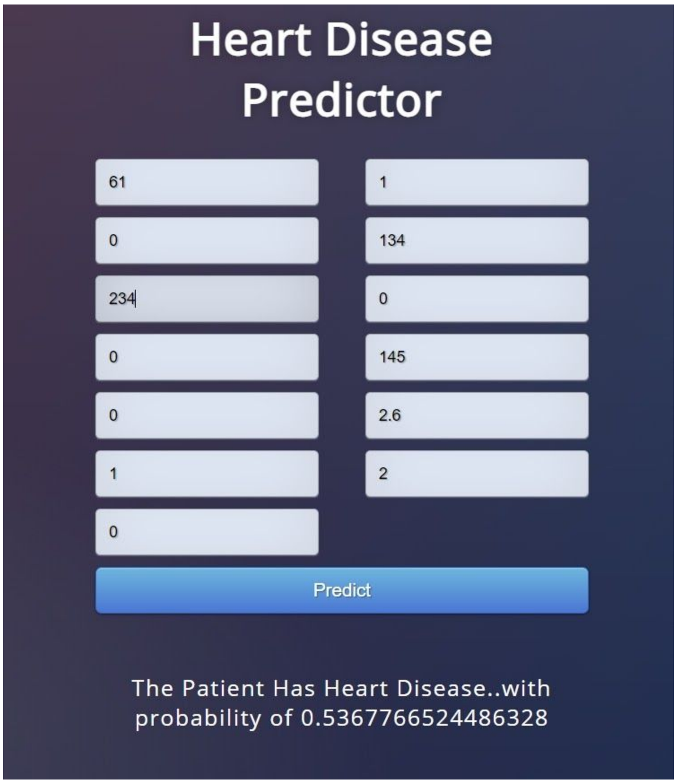
For hyper-parameter tuning we have used GridSearchCV with 5-fold cross validation. The best estimator is decided on the basis of accuracy metric.

For K Nearest Neighbors and logistic regression, the optimal values of hyper-parameters “k” and “C” are found to be 22 and 0.01 respectively. In case of gaussian process we have taken default kernel i.e. RBF and optimized it during the training in each iteration. For decision trees, Linear SVM GridSearchCV technique with 5-fold cross-validation is implemented and the optimal hyper-parameters obtained are 10 (max-depth), 0.1 (C) respectively. For gaussian naive bayes, we are able to achieve best results when variance is 1. In the case of random forest, AdaBoost classifiers we obtained an optimal number of base estimators to be 5,100 upon applying GridSearchCV technique with 5-fold cross validation and slightly even more fine-tuning the model.

6. Deployment

We have deployed all our models using flask-web framework in local host environment. The final prediction will be the majority vote on predicted labels of optimal classifiers.

Example for positive class(sample having heart disease)

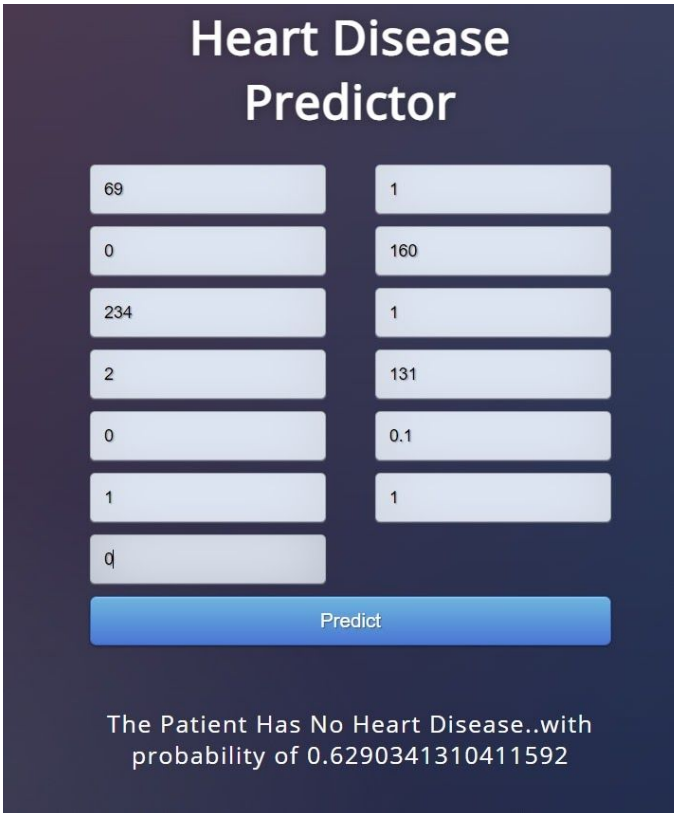


Heart Disease Predictor

61 | 1
0 | 134
234 | 0
0 | 145
0 | 2.6
1 | 2
0

The Patient Has Heart Disease..with probability of 0.5367766524486328

Example for negative class(sample not having heart disease)



Heart Disease Predictor

69	1
0	160
234	1
2	131
0	0.1
1	1
0	

The Patient Has No Heart Disease..with probability of 0.6290341310411592

References

[1] S. Mohan, C. Thirumalai and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," in IEEE Access, vol. 7, pp. 81542-81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

[2] A. H. Alkeshuosh, M. Z. Moghadam, I. Al Mansoori, and M. Abdar, ``Using PSO algorithm for producing best rules in diagnosis of heart disease," in Proc.Int.Conf. Comput. Appl. (ICCA), Sep. 2017, pp.306311.

[3] N. Al-milli, ``Backpropogation neural network for prediction of heart disease," J. Theor. Appl.Inf. Technol., vol. 56, no. 1, pp. 131135, 2013.

[4] C. A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, ``Analysis of neural networks based heart disease prediction system," in Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI),Gdansk, Poland, Jul. 2018, pp. 233239.

[5] P. K. Anooj, ``Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules," J. King Saud Univ.-Comput. Inf. Sci., vol. 24, no. 1, pp. 2740, Jan. 2012. doi:10.1016/j.jksuci.2011.09.002.

[6] L. Baccour, ``Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets," Expert Syst. Appl., vol. 99, pp. 115125, Jun. 2018. doi: 10.1016/j.eswa.2018.01.025.

[7] C.-A. Cheng and H.-W. Chiu, ``An artificial neural network model for the evaluation of carotid artery stenting prognosis using a national-wide database," in Proc. 39th Annual. Int. Conf. IEEE Eng. Med. Biol. Soc.(EMBC), Jul. 2017, pp. 25662569.

[8] H. A. Esfahani and M. Ghazanfari, ``Cardiovascular disease detection using a new ensemble classier," in Proc. IEEE 4th Int. Conf. Knowl.-Based Eng. Innov. (KBEI), Dec. 2017, pp. 10111014.

[9] F. Dammak, L. Baccour, and A. M. Alimi, ``The impact of criterion weights techniques in TOPSIS method of multi-criteria decision making in crisp and intuitionistic fuzzy domains," in Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE), vol. 9, Aug. 2015, pp. 18.

[10] R. Das, I. Turkoglu, and A. Sengur, ``Effective diagnosis of heart disease through neural networks ensembles," Expert Syst. Appl., vol. 36, no. 4,pp. 76757680, May 2009. doi: 10.1016/j.eswa.2008.09.013.