Heart Disease Prediction

Priya Sudharsanan, Rutuja Pisal, Prithika Vijayakumar, Aparna Siripurapu

Abstract – Heart Diseases also known as Cardiovascular disease (CVDs) are the leading cause of death globally. A large population suffer from heart diseases and many of them die annually from CVDs than from any other diseases. These diseases are usually silent and cannot be identified until the symptoms are very serious and the patient is likely to have a heart attack. In such cases anticipating the heart condition plays a key role in its diagnosis. To address this issue, various machine learning models were used in this project to predict whether or not a person has a heart disease. We implemented a variety of machine learning models, analyzed their drawbacks, and fine-tuned them to achieve the higher accuracy. Additionally we have also implemented a Stack regressor to obtain the best from every model and Neural Network to get better accuracy. Further we compared the models using various evaluation metrics and found Stack Regressor and Neural Network to be the best models.

I. INTRODUCTION

An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% are due to heart attack and stroke. Nearly half (48 percent, 121.5 million in 2016) of all adults in the United States have some type of heart disease. Fig. 1 shows the statistics of causes of death by various diseases per 100,000 population. The death rate for Heart Disease is the highest followed by Cancer and Accidents.

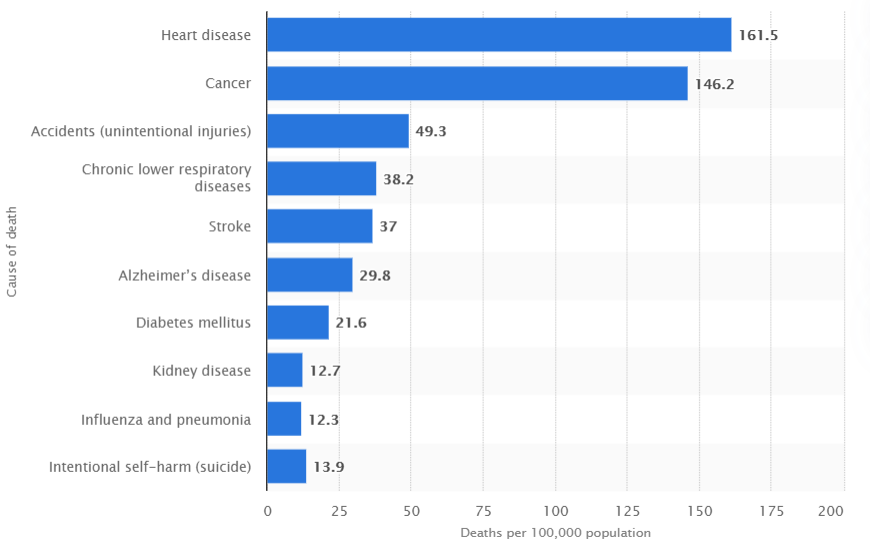

Fig. 1 : Heart Disease death ratio per 100,000 population in US

Diseases under the heart disease umbrella include blood vessel diseases (such as coronary artery disease, heart rhythm problems and heart defects you're born with), cerebrovascular disease, peripheral arterial disease, rheumatic heart disease, congenital heart disease, deep vein thrombosis and pulmonary embolism, etc. It is difficult to identify such diseases because of several contributory risk factors such as diabetes, high blood pressure, high cholesterol,

abnormal pulse rate, and many other factors. In such cases predicting the heart disease plays a crucial role in its treatment. Prediction can facilitate early detection and help avoid the death of many patients, enabling them to take accurate and efficient treatments. This increases the need for improving the medical diagnosis system day by day i.e making them cost effective.

In order to address this issue, machine learning was used in this project to predict the presence of heart disease. The goal of this project is to build a significantly efficient system to predict if an individual has heart disease or not, so that depending on the prediction one can take necessary steps to improve a person's health. For the scope of this project, different traditional classification models were explored. Later Stack Regressor and a Neural Network was used to build a better prediction model. All technical details and methodology used are described further in the report.

II. PROBLEM STATEMENT

To provide an early and on time diagnosis, we propose to build a classifier that can identify whether a person is prone to heart disease or not. We are going to explore a number of machine learning algorithms to build the classifier that gives the best results.

III. OVERVIEW OF PROPOSED APPROACH

The pipeline of our project is depicted in Fig. 2. From the figure it can be noted that the initial step is to load the dataset. To load the dataset we have used the pandas framework. The second step is to perform a data quality check. This was done by checking for missing values, outliers and also checking if data is balanced. The next step was to preprocess the dataset depending upon the data quality check analysis. In preprocessing we do data cleaning and obtain the cleaned data using which feature selection process is done next. In order to implement feature selection, we performed data analysis and dataset exploration. Here we drew relations between each feature and the target variable, also we check for any inter feature relationships. Using correlation mapping and other analysis we selected the features suitable for this dataset. Now using this refined and cleaned dataset we performed model training using 9 different classifiers. These include Logistic regression, KNN, SVM, Naive bayes, Decision trees, Random forest,Ensemble learning methods Adaboosting & Bagging and also using neural networks. Each model provides an accuracy and these accuracies were varied for different models. Now to obtain the best from all the above mentioned classifiers, we used the stack regressor to get better prediction results. Lastly, we train the neural network using the Automated Machine Learning(AutoML) approach, which gave us an accuracy of 99%. The obtained output is one of the best models, with the highest accuracy for this dataset along with the Stack regressor.

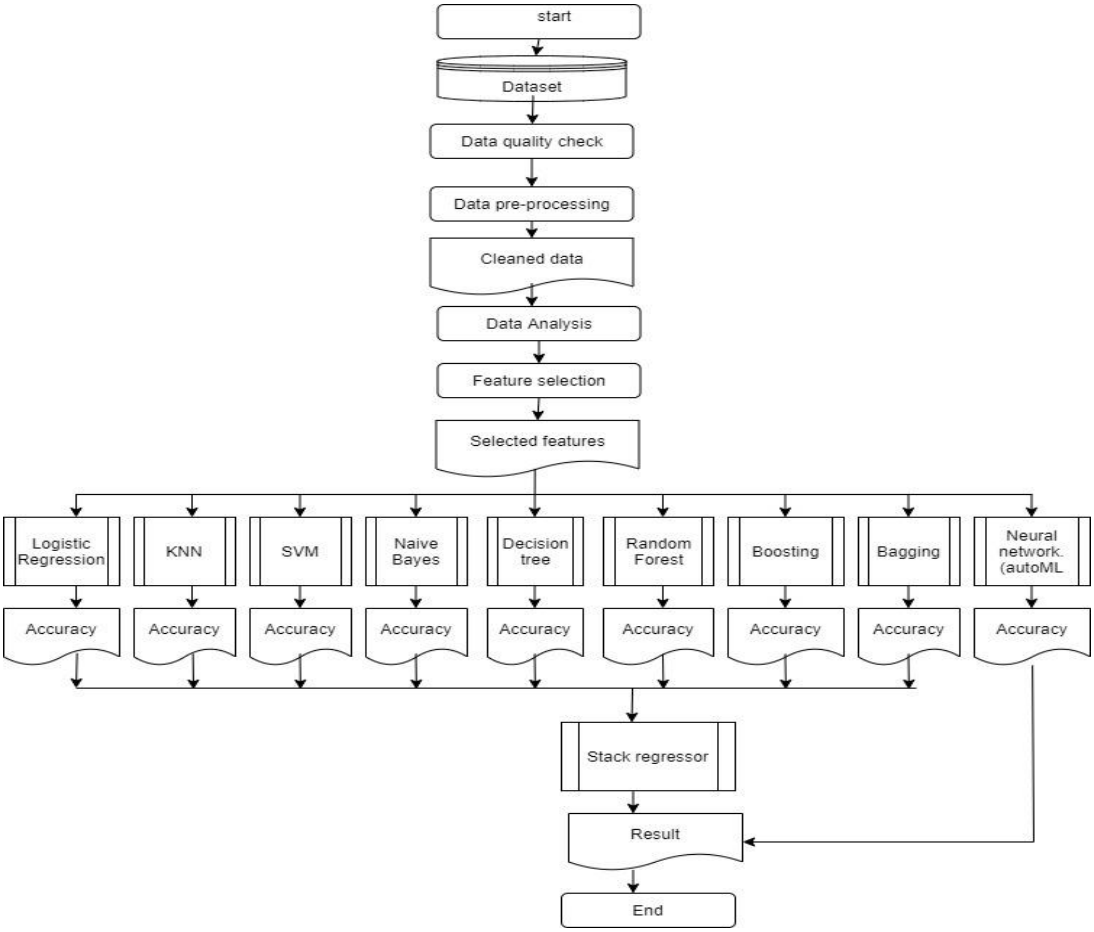

Fig 2: Pipeline of the Proposed Approach

IV. TECHNICAL DETAILS OF PROPOSED APPROACH

A) Data Description

The dataset used in this project is downloaded from kaggle. Along with data from UCI ML repository (Cleveland) three other datasets were combined from Hungary, Switzerland, and Long Beach V databases in this dataset, which dates back to 1988. The size of the dataset is around 37.22 KB. The dataset contains 1025 rows and 14 columns. While there are 76 attributes in total, including the predicted attribute, all reported experiments only use a subset of 14 of them. The "target" field indicates whether or not the patient has heart disease. It is integer-valued, with '0' indicating no disease and '1' indicating disease.

The 13 attributes in the dataset are used as features in the model training. Below are the attributes of data :

1. age
2. sex
3. chest pain type (4 values)

4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by fluoroscopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
14. The names and social security numbers of the patients were recently removed from the database,
15. replaced with dummy values.
16. target : 0 = no disease, 1 = disease

The dataset is examined to see if it is balanced or not. The target attribute is checked to see if there are approximately the same number of 0's and 1's. The heart disease dataset is balanced data with 51% of target attribute values as 1 and 49% of target attribute values as 0.

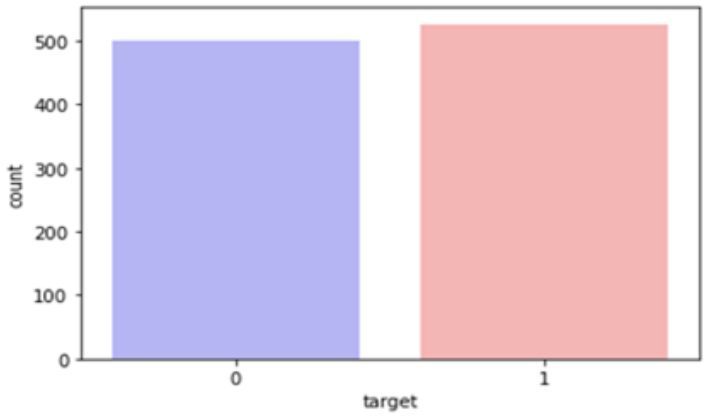

Fig. 3: Distribution of target attribute in the dataset

B) Data Preprocessing

The raw data acquired needs to be filtered so that the resulting data set can be used for building the models. We first checked if the dataset has any duplicate or missing values. Since the dataset does not have any missing or duplicate values we further checked if the dataset contained any outliers. We calculated the standard deviation of all the data points and the one's which std. deviation greater than 3 were considered outliers in our case. We identified 56 such outliers and replaced them with the median values of the existing data points.. The dataset was then clean enough to do

feature selection and further train the models.

C) Feature Selection

Feature selection is the process of selecting a subset of relevant features for use in model construction. We plotted a correlation matrix for all 14 features (Fig 4), wherein each cell in the table shows the correlation between two variables. A graph representing how each feature affects a person with heart diseases (Fig 5) is also plotted.

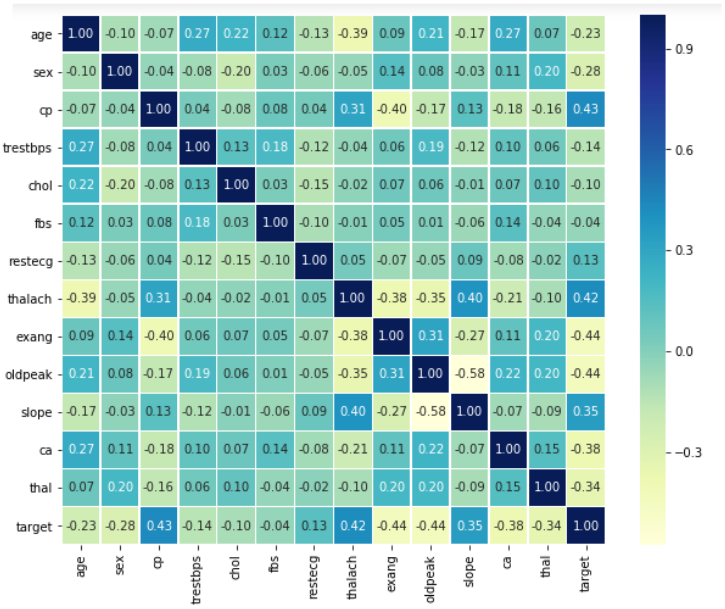

Fig 4: Correlation matrix showing correlation between every pair of features

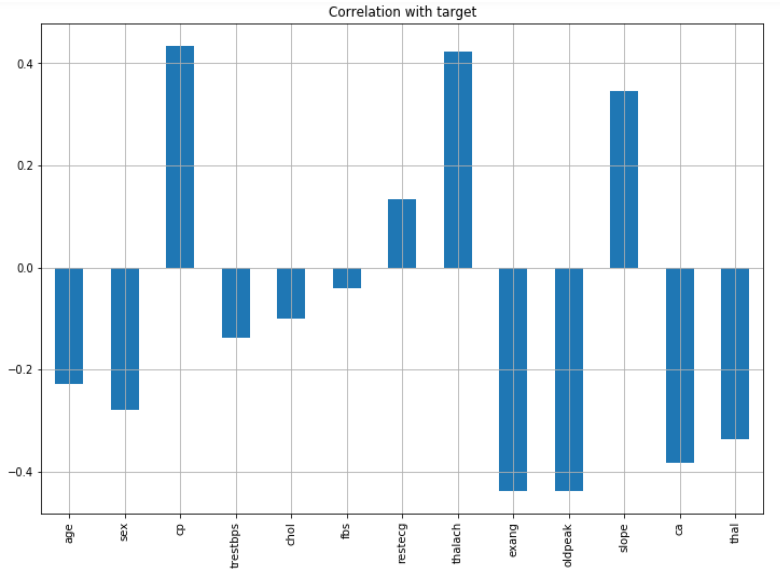

Fig 5: Correlation between every feature with respect to the target feature

We can see that fbs and chol are the lowest correlated with the target variable. All other variables have a significant correlation with the target variable. Further, using the train_test_split library from python sklearn library the data was split into training data and testing data. The size of training dataset is 70% and size of testing data is 30% of the dataset.

D) Algorithms

(a) Decision Tree

Decision tree is most widely used because of its robustness to noise, tolerance against missing information, handling of irrelevant, redundant predictive attribute values, low computational cost, interpretability, fast run time and robust predictors. We performed initial experiments on some parameters of the decision tree and got an accuracy of 85% on the test dataset. Then we tuned the model by doing Hyperparameter tuning using gridsearchCV to get the best decision tree among all the possible decision trees.

The initial accuracy for Decision Tree is 85.71%.

After application of parameter tuning approach, optimal parameters obtained are the following :

'criterion': 'gini',
'max_depth': 7,
'max_features': 'auto',
'min_samples_leaf': 1,
'min_samples_split': 2,
'splitter': 'best'

Once hyperparameter tuning was applied, the accuracy increased to 89.61%.

(b) Random Forest:

Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest outputs a class prediction, and the class with the most votes becomes our model's prediction. The initial accuracy obtained for Random forest on the testing data set is 95.78%. After application of parameter tuning approach, optimal parameters obtained are the following :

Criterion: gini
max_depth: 8
max_features: auto
n_estimators: 100

Once hyperparameter tuning was applied, the accuracy increased to 96.10%.

(c) Bagging

Bagging is an Ensemble Learning technique which aims to reduce error through the implementation of a set of homogeneous machine learning algorithms. The key idea of bagging is to use multiple base learners which are trained separately with a random sample from the training set, which through a voting or averaging approach, produce a more stable and accurate model. The accuracy obtained for the Bagging classifier on the test data set is 95.45%. Hyper parameter tuning was applied in order to increase the accuracy of the model. After application of parameter tuning approach, optimal parameters obtained are the following :

max_features: 5

max_samples: 500

n_estimators: 100

The accuracy plots for these parameters are plotted.Figure 6a shows the accuracy with respect to the parameter 'maximum feature', we can see that the accuracy is highest for max_features=5. Figure 6b shows the accuracy with respect to the parameter 'N estimators', we can see that the accuracy is highest for N_estimators=100. Figure 6c shows the accuracy with respect to the parameter 'maximum samples', we can see that the accuracy is highest for max_samples=500.

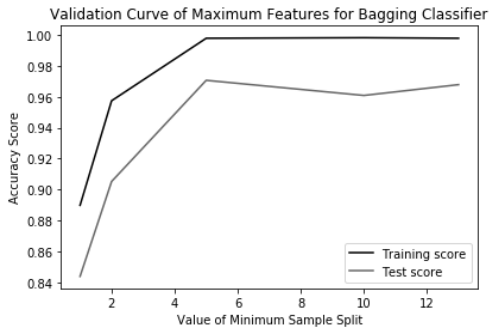

Fig 6a: Accuracy plot for "max_feature"

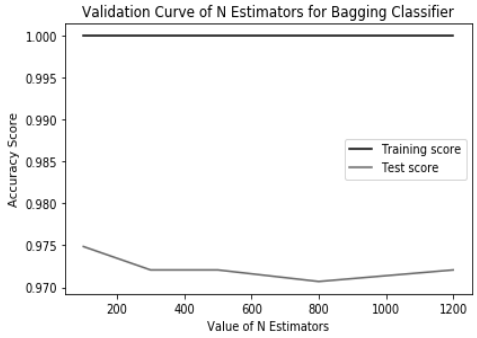

Fig 6b: Accuracy plot for "n estimators"

The accuracy obtained after hyperparameter tuning is 96.10%

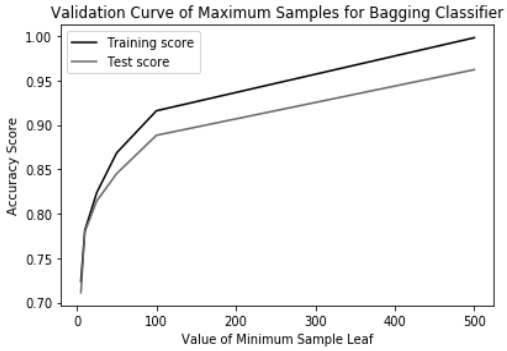

Fig 6c: Accuracy plot for "max_samples"

(d) Logistic Regression

One of the statistical regression models is the logistic regression model that could measure the relationship between a categorical dependent variable and one or more independent variables. The target variable is the dependent variable, rest of the features are independent variables. The model obtained an accuracy of 84% without any hyper tuning. There are a total of 15 parameters in the logistic regression model. To achieve better results hyper tuning was performed. With different solvers, differences in performance or convergence could be observed. Regularization (penalty) specifies the norm used in the penalization which when tuned provides different results, and lastly the C-value parameter controls the penalty strength. In logistic regression, the trade-off parameter c is the inverse regularization parameter, which is $1/\lambda$ and also defines the regularization strength.

We used GridSearchCV to find the key hyper parameters for logistic regression. The hypertunned parameter results were {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'} with the best train accuracy of 86%. On using this hypertuned model we obtained a test accuracy of 87%. As can be seen from fig. 7 below, the higher the value of c (c=100), the less regularization and accuracy. The value c=0.1 was chosen on hypertunning and it worked better.

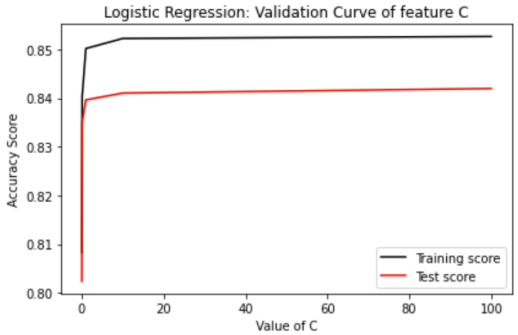

Fig. 7 :Value of C vs Accuracy

After hyper parameter tuning, the accuracy obtained is 87%.

(e) Naive Bayes

Naive Bayes or Bayes' Rule is the basis for many machine learning. The algorithm is used to create models that have predictive capabilities. Naive Bayes algorithm is based on the application of Bayes law of probability.

$$P(A|B)=P(B|A)*P(A)/P(B) \qquad (A \text{ and } B \text{ denotes events.})$$

Joint probability is determined when Bayes theorem is applied to classification problems. As a result the Calculation and storage become intractable. To make it easier, the Naive Bayes algorithm makes an assumption that all of the features are independent of one another. Conditional independence is assumed between attributes. This makes Naive Bayes a very simple classification algorithm. There are different types of naive bayes algorithms (Gaussian, Multinomial and Bernoulli). Since our features have real values, the gaussian model was chosen. Without any hyperparameter tuning an accuracy of 82% was obtained. On tuning the hyperparameter var_smoothing using GridSearchCV algorithm we obtained an accuracy of 83.3%. The best parameter var_smoothing values was found to be 2.848035868435799e-05 that gave the best accuracy. It could be seen from fig. 8 that higher the value of Var_smoothing lesser the accuracy. After hyper parameter tuning, the accuracy obtained is 83.36%.

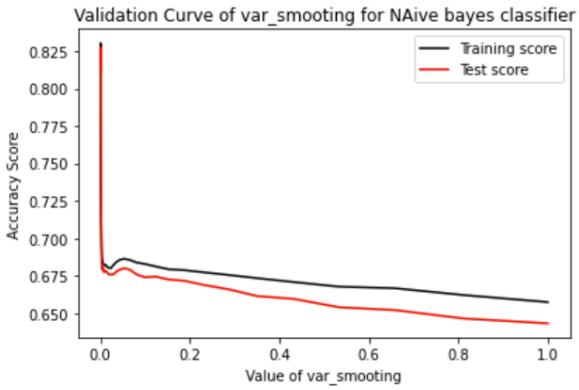

Fig. 8:Value of var_smoothing vs Accuracy

(f) AdaBoost

Boosting is a class of ensemble machine learning algorithms that involve combining the predictions from many weak learners. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost with decision trees as the weak learners is often the best

classifier but we performed experiments with Logistic Regression and SVC as well. The initial accuracy we got with Decision Tree as Base Classifier was 87%. We then experimented with features n_estimators and learning_rate to get best value for parameters and plotted accuracy graphs for the same. In Fig. 9a the plot is of accuracy vs n_estimators and second plot Fig.9b is of accuracy vs learning rate. From both the plots we can say that we get best accuracy when n_estimators is 700 and learning rate=0.1.

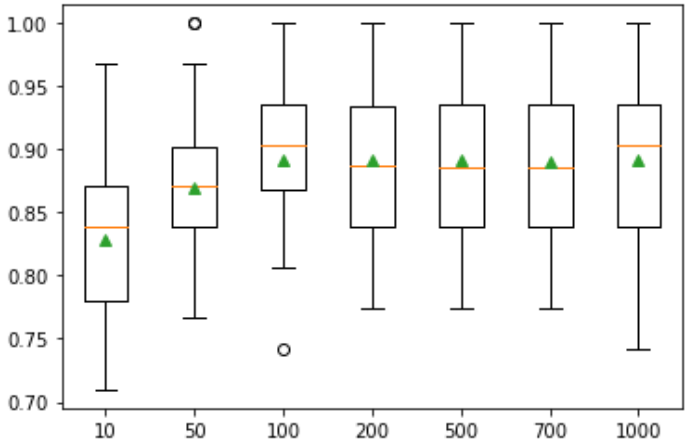

Fig. 9a: Accuracy vs N_estimtors

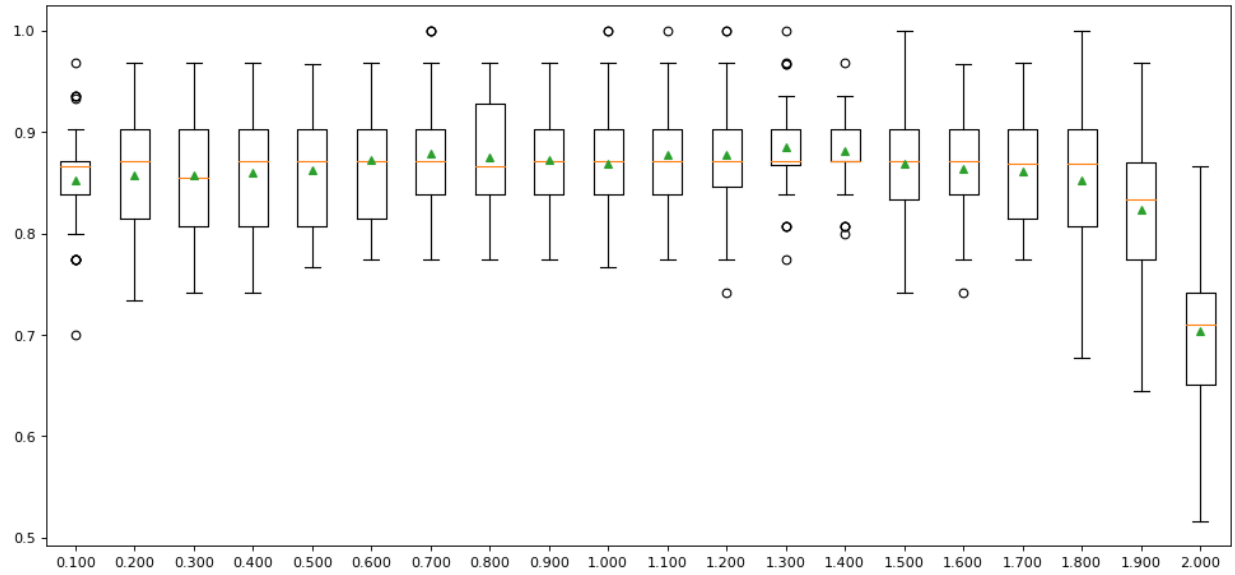

Fig. 9b: Accuracy vs Learning rate

Fig. 10 shows Decision Tree performs better than Logistic Regression and SVC and gives best accuracy. Thus for final

model training we selected n_estimators = 700, learning_rate = 0.1 and base_estimator = Decision Tree and calculated the accuracy on testing dataset.

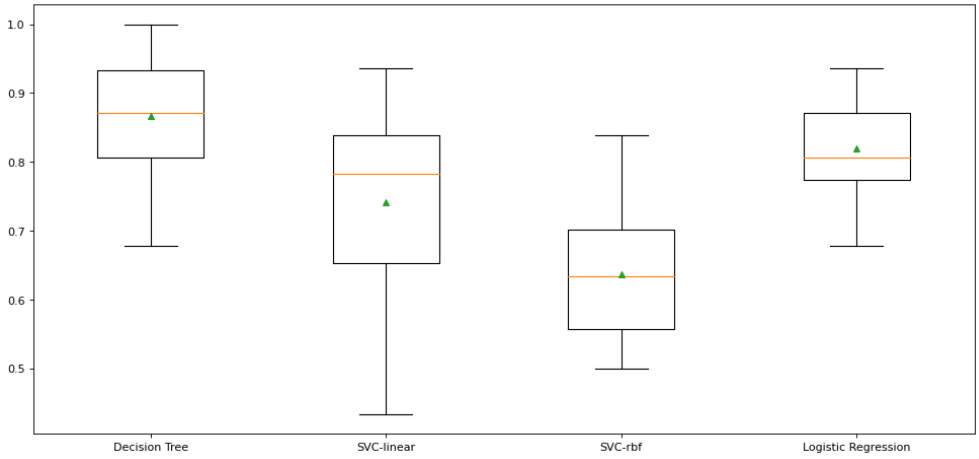

Fig. 10: Accuracy vs base_estimators

Thus Accuracy after parameter tuning is 89%.

(g) K-Nearest Neighbors:

KNN is a simple algorithm which can be used as both classification and regression algorithm. KNN is based on the assumption that any data point that is next to another belongs to the same class. In other words, it uses similarity to classify a new data point.K-value vs Error Mean graph is plotted with k=1 to 40 ,to check for which values of K there is a high error value. It can be noted from the graph below that as K value increases the error mean increases from K=0. Fig. 11 shows the change in Error Mean with K-values. Low error value means the model gives higher accuracy. Hence to train the KNN model, K values are taken as K=1,2 and 3.

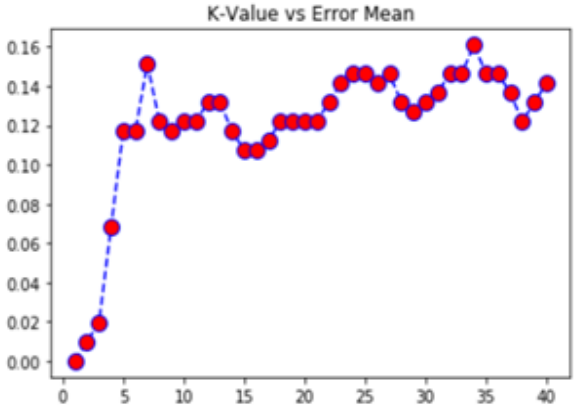

Fig. 11: K-value vs Error Mean

Five-fold cross validation is performed using K = 1, 2, and 3. The dataset is divided into two sets at each fold. The first set contains 820 data points for training the model, while the second set contains 205 data points for testing the model. Fig. 3. Shows all the data point distribution of all the five folds of the data. Accuracy, precision, recall, and F1 scores were computed across all folds during five-fold cross validation (with K = 1, 2, and 3). These values reported in Table 1. The mean values of Accuracy, Precision, Recall and F1 score were reported in Table 2.

TABLE 1

Accuracy, Precision, Recall, F1 Score over Five folds for K=1,2 and 3

K value	Folds	Accuracy	Precision	Recall	F1 Score
K=1	Fold 1	0.944444	0.944402	0.944402	0.944402
	Fold 2	0.965278	0.964928	0.965557	0.965196
	Fold 3	0.979021	0.979167	0.97973	0.979017
	Fold 4	0.972028	0.974026	0.971429	0.971961
	Fold 5	0.944056	0.94598	0.945456	0.944053
K=2	Fold 1	0.772277	0.789959	0.778105	0.770839
	Fold 2	0.8	0.826818	0.814141	0.799277
	Fold 3	0.75	0.796346	0.768687	0.746938
	Fold 4	0.77	0.788588	0.786822	0.769977
	Fold 5	0.83	0.839485	0.796689	0.809225
K=3	Fold 1	0.742574	0.743524	0.743907	0.742549
	Fold 2	0.79	0.792468	0.794949	0.789811
	Fold 3	0.66	0.663589	0.664646	0.659864
	Fold 4	0.68	0.676435	0.679315	0.676768
	Fold 5	0.74	0.731884	0.744482	0.733169

TABLE 2

Mean accuracy, mean precision, mean recall and mean F1 score when k=1,2 and 3 over all five folds

K values	Mean Accuracy	Mean Precision	Mean Recall	Mean F1 score
K=1	0.961	0.9617	0.9613	0.9609
K=2	0.7845	0.8082	0.7889	0.7793
K=3	0.7225	0.7216	0.7255	0.7204

After the analysis is done on the training dataset, testing dataset is used to make predictions. The highest accuracy of the testing dataset is when k=1 is 92.8571%.

(h) SVM:

Support-vector machine builds a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for tasks like classification, regression, outliers detection etc.,Using scikit learn module, both linear and non-linear support vector machines (SVM) models are implemented.

Linear kernel is used to train the SVM model with the training data set. To make predictions, a test data set is used. Accuracy and the confusion matrix are computed.The accuracy of the model is 85.71%. Most of the time, real-time data is not linearly separable. In such instances, a non-linear SVM model can be used. In non-linear SVM models, using kernel trick, the data is moved to a higher dimensional feature space where it is linearly separable. Non-linear SVM model with RBF kernel is built. Hypertuning of parameter C and gamma is performed using GridSearchCV method to obtain better results. Accuracy plots are drawn with C and gamma values as [1,10,100] and [0.01,0.001,0.0001] respectively. Figure shows the C vs Accuracy plots. For gamma=0.0001 the accuracy values are very low for C=1. The values of C and gamma for highest accuracy are selected as 100 and 0.01 respectively.

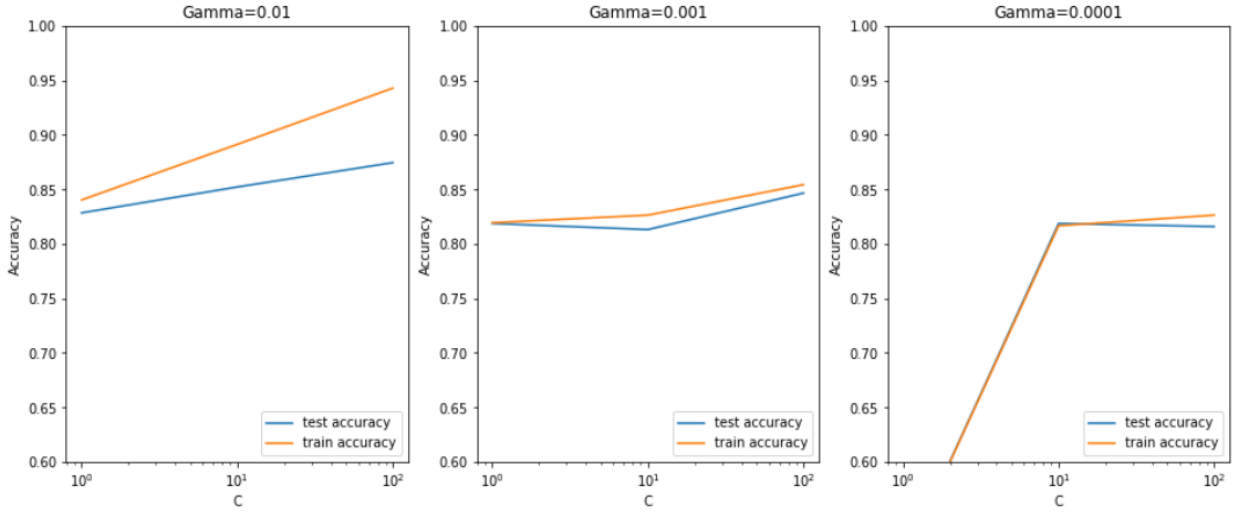

Fig 12 : C vs Accuracy Plots

The accuracy of the non-linear SVM model with rbf kernel on the testing dataset is 95.45% which is higher than that of the linear SVM model.

(i) Stacking Regressor:

Stacking is a way of combining multiple models that introduces the concept of a meta learner. Stacking is normally used to combine models of different types. The point of stacking is to explore a space of different models for the same problem. The idea is to attack a learning problem with different types of models which are capable of learning some part of the problem, but not the whole space of the problem. So here we build multiple different learners and use

them to build an intermediate prediction, one prediction for each learned model. Then we add a new model which learns from the intermediate predictions. The final model is said to be stacked on the top of the others, hence the name. In our project we have used Decision Tree, Random Forest, Bagging, Boosting,Naive Bayes, SVM, KNN as classifiers and Logistic Regression as MetaClassifier. After 5 fold cross validation we get Bagging, Boosting and KNN as the best models with an accuracy of 99%.

5-fold cross validation:

Accuracy: 0.91 (+/- 0.02) [Decision Tree,Random Forest]

Accuracy: 0.99 (+/- 0.01) [Bagging]

Accuracy: 0.99 (+/- 0.01) [AdaBoost]

Accuracy: 0.90 (+/- 0.02) [Naive Bayes]

Accuracy: 0.84 (+/- 0.03) [SVM]

Accuracy: 0.99 (+/- 0.01) [KNearest Neighbors]

Accuracy: 0.98 (+/- 0.01) [MetaClassifier]

(j) Neural Networks:

Deep learning has a significant impact on predictive analytics. It is a complex but effective tool for constructing predictive functions and accurately performing classification tasks. The main advantage of neural networks is high accuracy. To find the best neural network for the given dataset we use Automated Machine learning. AutoML is the new way of doing deep learning and it is associated with the development of Machine Learning solutions for data scientists without the need to conduct endless investigations into data preparation, model selection, model hyperparameters, and model compression parameters. Neural architecture search under automl runs several possible neural networks to pick out the one that gives the best accuracy. We have used Autokeras (An AutoML system based on Keras) to perform the Neural architecture search. Initially automl was used only by google and google charged $20 USD per usage. But now it has become accessible to all.

Many different architectures are considered in parallel, trained, and evaluated. Following that, each can be tweaked based on a chosen algorithm to try out a different architecture. The end result is the model that has best accuracy. For our project we have used an automl model that tries 250 different neural network structures and each structure goes through 100 epochs. The output obtained is a model with the structure as shown in figure 13.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 13)]              0
_________________________________________________________________
multi_category_encoding (Mul (None, 13)                0
_________________________________________________________________
normalization (Normalization (None, 13)                27
_________________________________________________________________
dense (Dense)                (None, 32)                448
_________________________________________________________________
re_lu (ReLU)                 (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                1056
_________________________________________________________________
re_lu_1 (ReLU)               (None, 32)                0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33
_________________________________________________________________
classification_head_1 (Activ (None, 1)                 0
=================================================================
Total params: 1,564
Trainable params: 1,537
Non-trainable params: 27
_________________________________________________________________
```

Fig. 13: Neural Network Model summary

The obtained model has 1 multi-category encoding layer, output of which was given to the normalisation layer for normalising the data and then to an alternating of 3 dense layers and 2 re-lu layers(activation). This model was then used to perform prediction on the test set and an accuracy of 99% was obtained.

E) EVALUATION METRICS:

Evaluating a model is a major part of building an effective machine learning model. The key classification metrics are **Accuracy, Recall, Precision, and F1- Score.** The table 3 provides the F1 score, Precision and Recall values for each of the machine learning models we have used in this project.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

TABLE 3:

Evaluation metrics of all the models used

ML Model	Target	F1 Score	Precision	Recall
Decision tree	0	0.89	0.86	0.92
	1	0.89	0.92	0.86
Random forest	0	0.96	0.95	0.97
	1	0.96	0.97	0.95
Bagging	0	0.96	0.95	0.97
	1	0.96	0.97	0.95
Boosting	0	0.88	0.92	0.85
	1	0.89	0.86	0.92
Logistic Regression	0	0.86	0.90	0.81
	1	0.88	0.85	0.92
Naive Bayes	0	0.83	0.90	0.78
	1	0.87	0.82	0.92
SVM	0	0.84	0.91	0.77
	1	0.87	0.82	0.93
KNN	0	0.91	0.92	0.90
	1	0.92	0.91	0.93
Neural Network	0	0.99	0.99	1.00
	1	0.99	1.00	0.98

V. RELATED WORKS

Lots of research work have been done for assessment of the classification accuracies of different machine learning algorithms. Authors of [2] achieved 77% prediction accuracy by applying a logistic regression algorithm on this dataset. In this study, authors [3] did an enhanced work by doing comparison of global evolutionary computation approaches and thus they observed higher prediction accuracy. Data mining models have been optimized by

introducing new concepts like ensemble-learning models that improved the classification performance. This has been suggested by developing a predictive ensemble-learning model on different datasets in order to diagnose and classify the presence and absence of coronary heart disease and achieved promising accuracy that exceeded the state-of-the-art results [4]. The idea of ensemble-learning was also applied by aggregating the predictions of different classifiers instead of training an individual classifier. An example of application was conducted for predicting coronary heart disease using bagged tree and AdaBoost algorithms [5]. From this basis, an ensemble method based on neural networks has been suggested for creating a more effective classification model and showed promising classification accuracy in [6,7]. An example of ensemble-learning model was also proposed for heart failure detection using an LSTM-CNN-based network [8,9].

VI. CONCLUSION

In conclusion, each algorithm was implemented and tested on the heart disease dataset, and the results were analyzed to determine how well each algorithm has performed. Hyperparameter tuning using GridSearchCV was used to improve the accuracy of the models. Comparing the results of the models implemented we see that Random Forest and Bagging gives us an accuracy of 96%. To increase precision of the prediction system we then used Stack Regressor for all the models implemented and got an accuracy of 99% for Bagging, Boosting and KNN. And from the results of the neural network we can see that we get an accuracy of about 99%. Thus these 2 are the best models for predicting heart disease. So, it is clear that in healthcare ML is a kind of miracle to solve multiple kinds of problems.

Link for the code: https://drive.google.com/drive/folders/1v4jzFRQwN_knSVi6s58tYMXOVdP6L7Pn?usp=sharing

VII. REFERENCES

[1] https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid,S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, “International application of a new probability algorithm for the diagnosis of coronary artery disease,” The American journal of cardiology, vol. 64, no. 5, pp. 304–310, 1989.

[3] B. Edmonds, “Using localised ’gossip’ to structure distributed learning,” 2005.

[4] Miao, K.H.; Miao, J.H.; Miao, G.J. Diagnosing coronary heart disease using ensemble machine learning. Int. J. Adv. Comput. Sci. Appl. (IJACSA) 2016, 7, 30–39.

[5] Yekkala, I.; Dixit, S.; Jabbar, M.A. Prediction of heart disease using ensemble learning and Particle Swarm Optimization. In Proceedings of the 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), Bengaluru, India, 17–19 August 2017; pp. 691–698.

[6] Das, R.; Turkoglu, I.; Sengur, A. Effective diagnosis of heart disease through neural network ensembles. Expert Syst. Appl. 2009, 36, 7675–7680.

[7] Das, R.; Turkoglu, I.; Sengur, A. Diagnosis of valvular heart disease through neural network ensembles. Comput. Methods Programs Biomed. 2009, 93, 185–191.

[8] Wang, L.; Zhou, W.; Chang, Q.; Chen, J.; Zhou, X. Deep Ensemble Detection of Congestive Heart Failure using Short-term RR Intervals. IEEE Access 2019, 7, 69559–69574

[9] Altan, G.; Kutlu, Y.; Allahverdi, N. A new approach to early diagnosis of congestive heart failure disease by using Hilbert–Huang transform. Comput. Methods Programs Biomed. 2016, 137, 23–34.

[10] https://www.statista.com/statistics/1110699/percentage-cvd-deaths-worldwide-by-risk/

[11] https://autokeras.com/