

Heart Disease Classification using Ensemble Learning

*A thesis submitted in partial fulfillment of the requirements for
the award of the degree of*

Master of Science

in

Computer Science

by

Arghyadeep Mondal

(19419CMP001)



Department of Computer Science

Institute of Science

Banaras Hindu University, Varanasi – 221005

2021

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented in the report/thesis, entitled **Heart Disease Classification using Ensemble Learning**, in partial fulfillment of the requirement for the award of the Degree of **Master of Science** and submitted to the institution is an authentic record of my own work carried out during the period *March-2021* to *June-2021* under the supervision of **Dr Manoj Kumar Singh**. I also cited the reference about the text(s) /figure(s) /table(s)/equation(s) from where they have been taken.

The matter presented in this thesis as not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Date:

Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge. The Viva-Voce examination of Arghyadeep Mondal, M.Sc. Student has been held on 17/07/2021.

**Signature of
Research Supervisor**

**Signature of
Head of the Department**

ABSTRACT

Heart disease is one of the most significant causes of mortality in the world today. Number of people losing their lives due to heart disease is growing day by day, revealing the need of a model which predicts beforehand. An initiative has to be taken to aid the people by giving them a cautionary advice about the disease at the correct time. It is not easy for everyone to afford expensive treatments and medications so there is urgency of a structure which can quickly go through the information of the patient and inform them at an earlier stage if they test positive. We need a logical process that analyzes and finds unrevealed data and figures in the medical data. Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a novel method that aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several classification techniques such as single and ensembles and ultimately it will show how the ensemble learnings can be so good in case of predicting heart disease. How bagging, boosting, stacking and voting such methods will enhance the accuracy being even lighter than traditional approaches in terms of hardware requirements.

Keywords: Heart Disease Prediction, Machine Learning, Classification Algorithms, Ensemble Learning Classifier, Feature selection, Prediction model, Cardiovascular Disease (CVD).

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my supervisor DR. MANOJ KUMAR SINGH as well as our whole computer science department who gave me the golden opportunity to do this wonderful project on the topic Heart Disease Classification using Ensemble Learning, which also helped me in doing a lot of Research and i came to know about so many new things

I am really thankful to them.

Secondly i would also like to thank my friends who helped me a lot in finishing this project within the limited time.

I am making this project not only for marks but to also increase my knowledge.

THANKS AGAIN TO ALL WHO HELPED ME.

TABLE OF CONTENTS

Title	Page No.
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS.....	viii
LIST OF NOTATIONS	ix
1. INTRODUCTION	
1.1 General.....	10
1.2 Problem Description	10
1.3 Objective	11
2. LITERATURE REVIEW.....	12
3. DESIGN DETAILS	
3.1 Ensemble Learning	13
3.1.1 Bootstrap Aggregating.....	14
3.1.2 Bagging.....	15
3.1.3 Boosting	16
3.1.4 Stacking.....	17
3.2 Univariate Selection.....	17
3.3 Chi-Square Test	18
3.4 Proposed Approach.....	19
4. IMPLEMENTATION	
4.1 Dataset.....	20
4.2 Software Requirements	21
4.3 Used Tools	21
4.4 Implementation	22
4.4.1 Collecting Data	22
4.4.2 Data Preprocessing.....	23

4.4.3	Exploratory Data Analysis	24
4.4.3.1	Univariate Exploratory Data Analysis	24
4.4.3.2	Bivariate Exploratory Data Analysis	26
4.4.4	Balancing the Dataset	29
4.4.5	Feature Selection.....	30
4.4.6	Scaling and Splitting.....	31
4.4.6.1	Scaling.....	31
4.4.6.2	Splitting.....	31
4.4.7	Predictive Modeling.....	32
4.4.7.1	Single Models	32
4.4.7.1.1	Logistic Regression.....	32
4.4.7.1.2	K-Nearest Neighbor (KNN).....	32
4.4.7.1.3	Decision Tree	33
4.4.7.1.4	Support Vector Machine	33
4.4.7.1.5	Naïve Bayes	33
4.4.7.2	Ensemble Models.....	34
4.4.7.2.1	Random Forest.....	34
4.4.7.2.2	Extreme Gradient Boosting.....	34
4.4.7.2.3	Light Gradient Boosting Machine	35
4.4.7.2.4	Stacking Classifier	38
5.	RESULTS AND DISCUSSION	
5.1	Validation.....	39
5.2	Accuracy Metrics	39
5.3	Results.....	41
5.4	Comparison	43
6.	CONCLUSION AND FUTURE WORK	
6.1	Conclusions.....	44
6.2	Future Work.....	45
	REFERENCES	46
	PLAGIARISM REPORT.....	50

LIST OF FIGURES

Figure No.	Title	Page No.
1.	Ensemble Learning	13
2.	Bootstrap Aggregating	14
3.	Bagging	15
4.	Boosting	16
5.	Stacking.....	17
6.	Flowchart of the Heart Disease Prediction System.....	19
7.	Dataset in Pandas DataFrame	22
8.	Information about DataFrame	23
9.	Heatmap of the DataFrame before and after Missing Value Handling	23
10.	Univariate Analysis.....	24
11.	Bivariate Analysis	27
12.	Dataset before and after Balancing	29
13.	Feature Importance according to Chi-Square value.....	30
14.	Level-wise Tree Growth	36
15.	Leaf-wise Tree Growth	36
16.	Stacking Classifier	38
17.	Confusion Matrix	40
18.	Single Models and their Accuracy	41
19.	Ensemble Models and their Accuracy	42
20.	Every implemented model and their accuracy comparison	43

LIST OF ABBREVIATIONS

WHO	World Health Organization
CVD	Cardio Vascular Disease
IoT	Internet of Things
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
WEKA	Waikato Environment for Knowledge Analysis
IEEE	Institute of Electrical and Electronics Engineers
SMO	Sequential Minimal Optimization
KNN	K- Nearest Neighbor
SVM	Support Vector Machine
GB	Gradient Boosting
AdaBoost	Adaptive Boosting
XGBoost/XGB	Extreme Gradient Boosting
LightGBM/LGBM	Light Gradient Boosting Machine
ANOVA	Analysis of Variance
BMI	Body Mass Index
SMOTE	Synthetic Minority Oversampling Technique
MSE	Mean Squared Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
NDCG	Normalized Discounted Cumulative Gain
AUC	Area Under the Curve

LIST OF NOTATIONS

χ^2	Chi-Square Statistic
F_x	Logistic Regression Function
X_{new}	Standard Normal Score of X (Z statistic score)
$P(C_j F_i)$	Each probability of C_j given all probability of F_i
λ, γ, η	Regularization parameter in Tree based boosting algorithm
B_0	Y intercept in straight line equation
B_1	X- coordinate in straight line equation
X	Input data
Y	Actual output data
\hat{Y}	Predicted Output by a model
P	Probability

CHAPTER 1

INTRODUCTION

1.1 General

Now a days, heart disease is a most common factor in human's life after a certain age. As of 2016, according to WHO 17.9 million people which contributes 32% of total world deaths die due to heart disease. And the most surprising case is, 85% of the mentioned death was due Heart Attack and Stroke. At the turn of the century, cardiovascular diseases (CVDs) became the leading cause of mortality in India. This epidemiological transition is largely because of the increase in the prevalence of CVDs and CVD risk factors in India. In 2016, the estimated prevalence of CVDs in India was estimated to be 54.5 million. One in 4 deaths in India are now because of CVDs with ischemic heart disease and stroke responsible for >80% of this burden. These diseases tend to affect patients in the most productive years of their lives and result in catastrophic social and economic consequences.

An abnormal pumping of blood is observed in each patient who is suffering from heart disease. Though there are several factors that are supposed to cause heart disease, but there are some severer changeable factors that are strongly correlated with heart disease. Those are: smoking, high cholesterol, high blood pressure, physical inactivity, alcohol, obesity, poor diet etc. Apart from these, age, sex, family history etc... are considered as unchangeable factors.

1.2 Problem Description

Now a days, massive amount of data is continuously generated in hospital, health clinic, diagnostic centers, various surveys, and modern healthcare related IoT devices. We can extract essential information through these data which is very helpful to predict future heart disease risks. As computation evolved today much faster and less costly, to achieve this goal using the traditional diagnostic methods are less focused to early predict the heart disease, rather analytical tools like data mining, machine learning and AI are very productive to find data insights, useful underlying patterns, relationships between features and correlated data to further diagnose a patient at early stage.

1.3 Objective

As far as we have seen, Machine Learning and AI such analytical tools are very productive in case of predicting heart disease for a patient. There are several approaches or models we can use for such classification problem like traditional machine learning classifiers, Logistic Regression, K-Nearest Neighbor, Naïve Bayes, Decision Tree, Random Forest and Support Vector Machine. For better accuracy and less overfitting/underfitting issues some Deep Learning Models are deployed such as Artificial Neural Network, Deep Neural Network, Fuzzy Neural Networks. These Deep Learning models are very accurate but computationally complex. In this paper we are going use several Ensemble Learning approaches to predict heart disease which are more accurate than traditional Machine Learning models and less complex than Neural Networks.

Ensemble Learning methods is a Machine Learning technique that combines several base models in order to produce one optimal predictive model. In this particular issue the problem is solved implementing various type of Ensemble Learning approaches such as Bagging, Boosting, and Stacking which further harness the path to our goal. For this problem, Framingham dataset is chosen.

CHAPTER 2

LITERATURE REVIEW

The unstoppable evolving technology is growing with its radius of various domains including both medical and Data Science. At the present decade, Data Science field is leaded by various kind of research papers published by many individuals. Some of those are about predicting heart disease with a relevant accuracy. Some popular papers are KStar, J48, SMO, and Bayes Net and Multilayer perceptron using WEKA. Based on performance from different factor SMO (89% of accuracy) and Bayes Net (87% of accuracy) achieve optimum performance than KStar, Multilayer perceptron and J48 techniques using k-fold cross validation. But this is not sufficient enough, various approaches including Machine Learning and Deep Learning are experimented on this particular problem to get more accuracy and less overfitting like issues.

In a research conducted using Cleveland dataset for heart diseases which contains 303 instances and used 10-fold Cross Validation, considering 13 attributes, implementing 4 different algorithms, they concluded Gaussian Naïve Bayes and Random Forest gave the maximum accuracy of 91.2%.

Using the similar dataset of Framingham, Massachusetts, the experiments were carried out using 4 models and were trained and tested with maximum accuracy K Neighbors Classifier: 87%, Support Vector Classifier: 83%, Decision Tree Classifier: 79% and Random Forest Classifier: 84%.

CHAPTER 3

DESIGN DETAILS

The idea behind this project is to predict the future heart disease risk, for this I used various Ensemble Learning methods like Bagging, Boosting, and Stacking. And for selecting the best features which are supposed to strongly manipulate the future heart disease risk. I have used Univariate Selection method and some type of scaling mechanism. And I balance the dataset according to some model's requirement. All these methods are described in brief below:

3.1 Ensemble Learning

The main idea behind the Ensemble Learning is building multiple models instead of a single model to predict the target or future. We will build multiple machine learning models and we call these models as weak learners. A combination of all weak learners makes the strong learner which generalizes to predict all the target classes with a decent amount of accuracy.

Based on the model used in Ensemble Learning we can classify it into two categories.

1. **Homogeneous ensemble methods:** - All the models are built using the same machine learning algorithm. Examples: Bagging, Boosting.
2. **Heterogeneous ensemble methods:** - All the models are built using different machine learning algorithms. Examples: Stacking.

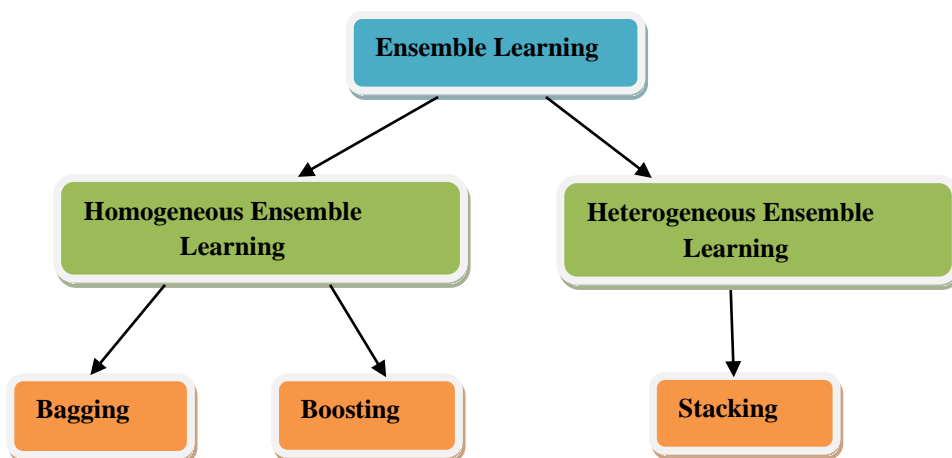


Figure -1 Ensemble Learning

3.1.1 Bootstrapping

For building multiple models whether it is a homogeneous or heterogeneous ensemble method, the dataset is the same. For each model, we need to take a sample of data, but we need to be very careful while creating these samples of data. Because if we randomly take the data, in a single sample we will end up with only one target class or the target class distribution won't be the same. This will affect model performance. To overcome this, we need a smart way to create these samples, known as bootstrapping samples.

Bootstrapping is a statistical method to create sample data without leaving the properties of the actual dataset. The individual samples of data called bootstrap samples.

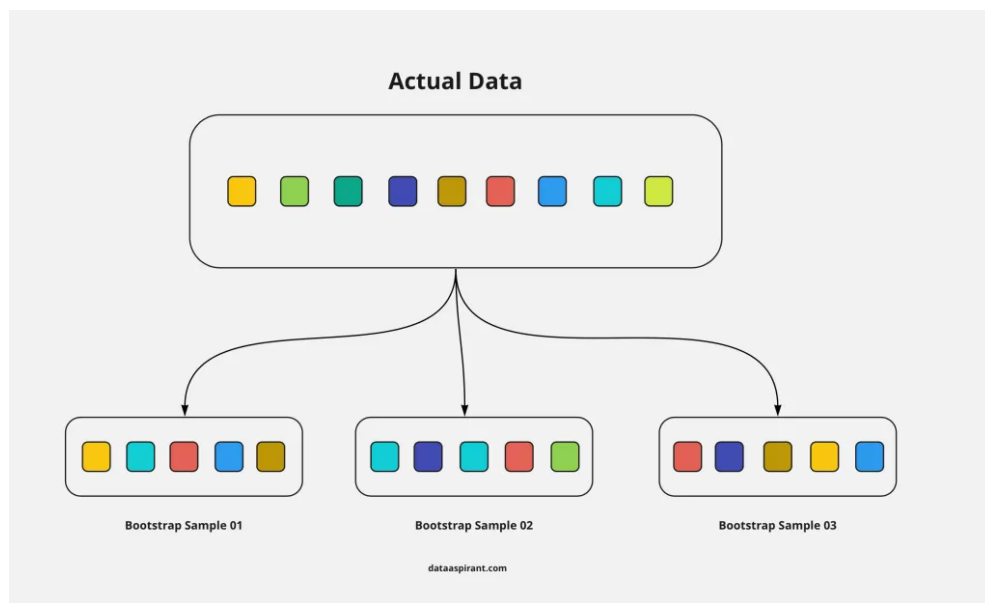


Figure – 2 Bootstrapping

3.1.2 Bagging

In the bagging method, all the individual models are built parallel, each individual model is different from one another. In this method, all the observations in the bootstrapping sample will be treated equally. In other words, all the observations will have equal at zero weightage. Because of this bagging method also called **Bootstrap Aggregating**.

As a first step using the bootstrapping method, we will split the dataset into N number of samples. Then we will select the algorithm as per requirement. Suppose if we selected a decision tree, then each bootstrap sample will be used for building one random forest model. Don't forget all the decision trees are built in parallel.

Once the training phase is completed, to predict the target outcome, we will pass the observations to all the N decision trees. Each decision tree will predict one target outcome. The final prediction target will be selected based on the majority voting.

Example: Random Forest Algorithm

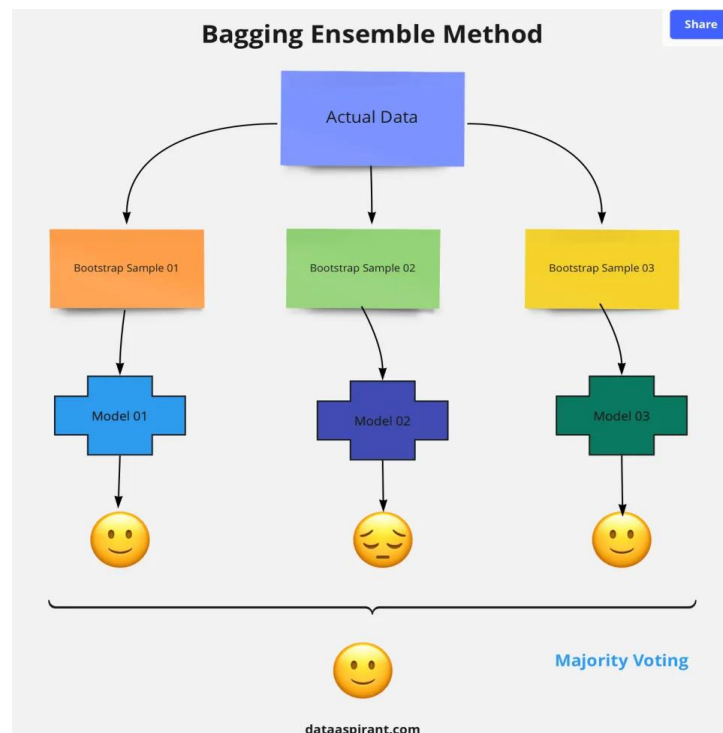


Figure – 3 Bagging

3.1.3 Boosting

In the boosting method, all the individual models are built sequentially. Which means the outcome of the first model passes to the next model and etc. In bagging the models are built parallel so we don't know what the error of each model is. Whereas in boosting once the first model built, we know the error of that model. So, when we pass this first model to the next model the intention is to reduce the error further.

Unlike bagging all the observations in the bootstrapping sample are not equally treated in boosting. Observations will have some weightage. For a few observations, the weightage will be higher whereas others may be lower.

Example: Adaboost algorithm, XGBoost, LightGBM, CatBoost, LPBoost, GradientBoost, BrownBoost etc.

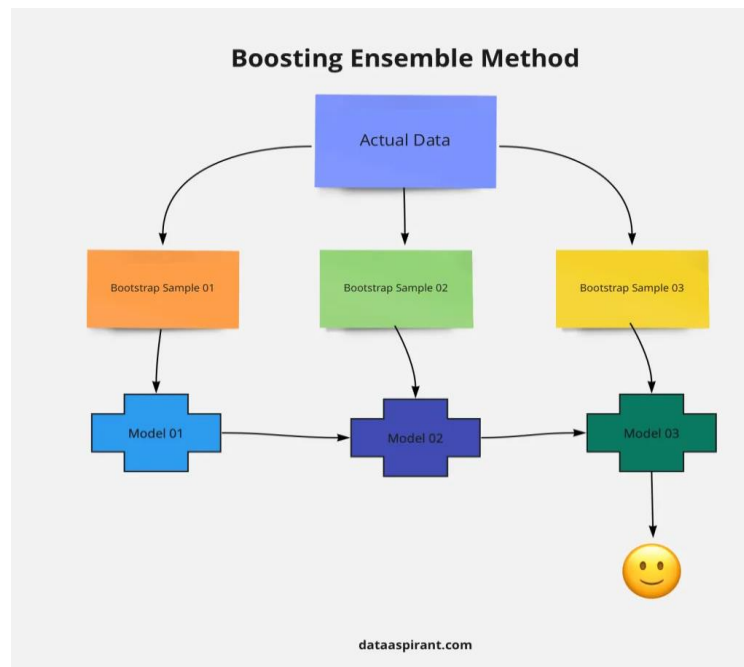


Figure – 4 Boosting

3.1.4 Stacking

Stacking is an ensemble learning technique that uses predictions for multiple models (for example KNN, Decision trees, or SVM) to build a new model. This final model is used for making predictions on the test dataset. The final model is called Meta Model.

The benefit of Stacking is that it can harness the capabilities of a range of well-performing models on a classification or regression task and make predictions that have better performance than any single model in the ensemble.

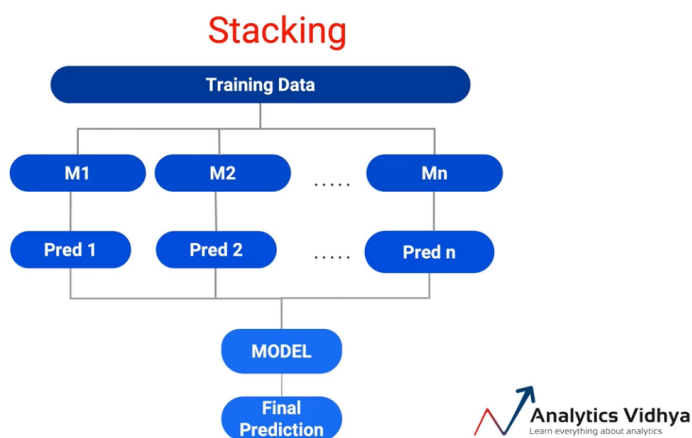


Figure – 5 Stacking

3.2 UNIVARIATE FEATURE SELECTION

Including more features in the model makes the model more complex, and the model may be overfitting the data. Some features can be the noise and potentially damage the model. By removing those unimportant features, the model may be generalized better.

Univariate feature selection works by selecting the best features based on univariate statistical tests. We compare each feature to the target variable, to see whether there is any statistically significant relationship between them. It is also called analysis of variance (ANOVA). When we analyze the relationship between one feature and the target variable, we ignore the other features. That is why it is called 'univariate'. Each feature has its test score.

Finally, all the test scores are compared, and the features with top scores will be selected.

3.3 CHI-SQUARE TEST

A chi-square test for independence compares two variables in a contingency table to see if they are related. In a more general sense, it tests to see whether distributions of categorical variables differ from each another.

$$\chi^2_c = \sum_{i=1}^k \left[\frac{(O_i - E_i)^2}{E_i} \right]$$

Where O is the observed value, E is the expected value and “i” is the “ith” position in the contingency table and c is the degrees of freedom.

It’s also used for “**Goodness of fit**” test.

A chi square test will give a p-value. The p-value will tell if the test results are significant or not.

In order to perform a chi square test and get the p-value, two pieces of information are required:

- Degrees of freedom. That’s just the number of categories minus 1.
- The alpha level(α). This is chosen by the tester. The usual alpha level is 0.05 (5%), but anyone could also have other levels like 0.01 or 0.10.

3.4 Proposed Approach

In this module, we are going to discuss about our methodology. Generally, the module's approaches are very straight forward to a specific sequence. The sequence consists of as following:

1. Collecting Data
2. Data Preprocessing
3. Exploratory Data Analysis
4. Training and Testing Models on Data
5. Evaluation

The following flowchart will describe steps those I required to design the system “Heart Disease Prediction Using Ensemble Learning”.

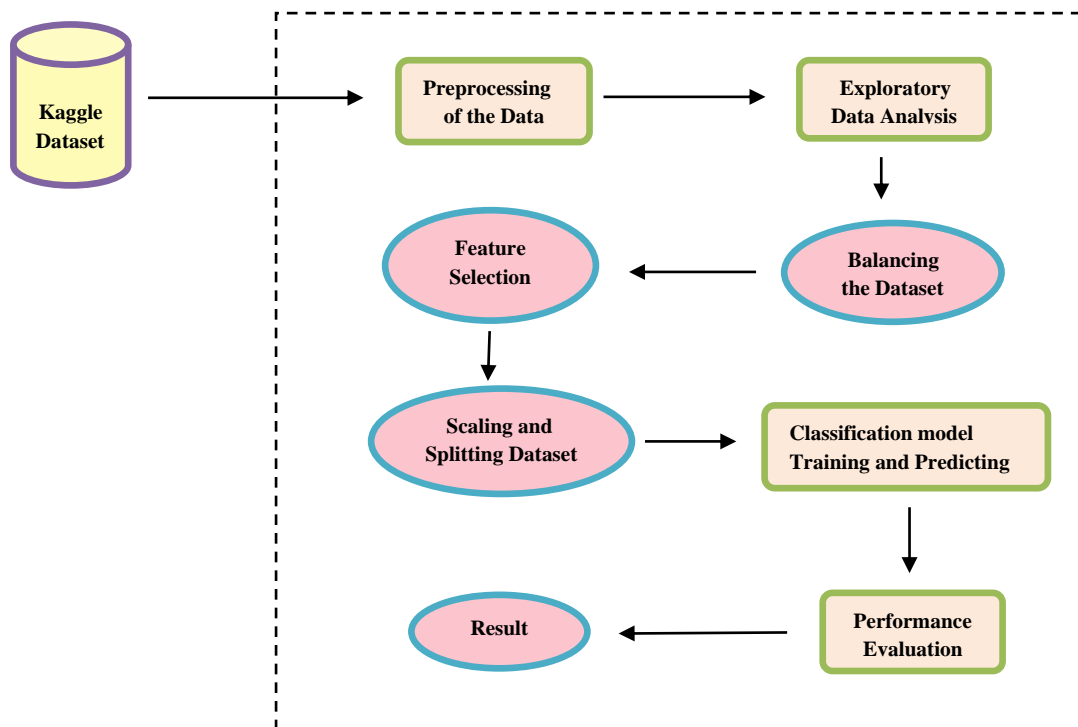


Figure – 6 Flowchart of the Heart Disease Prediction System

CHAPTER 4

SIMULATIONS AND IMPLEMENTATION

4.1 Dataset

The used dataset “framingham.csv”, which includes information about cardiovascular study performed on the residents of Framingham, Massachusetts, is publicly available at Kaggle website. This dataset contains 4000 around rows and 14 columns/attributes. Attributes are described below:

- male - 1 for male 0 for female
- age - Given age as integer
- education - Given as ordinal variable
- currentSmoker - Currently smoking or not
- cigsPerDay - Number of cigarettes per day
- BPMeds - Currently having any medication on blood pressure or not
- prevalentStroke - Family history of having stroke 1 for existence 0 for not
- prevalentHyp - Family history of having hypertension 1 for existence 0 for not
- diabetes - 1 for diabetic 0 for not
- totChol - Total cholesterol level
- sysBP - Systolic Blood Pressure
- diaBP - Diastolic Blood Pressure
- BMI - Body Mass Index
- heartRate - Pulse rate per minute
- glucose - Blood Glucose Level
- TenYearCHD - Future risk of heart disease up to 10 years.

The data set is in csv (Comma Separated Value) format which is further prepared to data frame as supported by Pandas library in python.

4.2 Software Requirements

A web browser (Mozilla Firefox) installed on Windows (10) Operating System, which is used to design this system. And it is implemented with Python Programming Language. As an IDE (Integrated Development Environment), an online cloud platform provided by Google named Colaboratory is used for interpreting Python code. It is specially designed for Data Science and AI purposes so zero configuration is required. It provides hardware facilities such as:

- 12.69 GB of Main Memory
- 107.72GB of Disk Space
- And an additional GPU for performance acceleration.

4.3 Used Tools

To satisfy the need of tools and their purposes that are required to set the process up are given below:

- NumPy, Pandas – For Dataset handling
- Seaborn, Matplotlib – Data Visualization
- Warning – Blocking various kind of unwanted warning
- Scikit-learn – One of the most useful modules for feature selection, scaling, splitting, model selection, and performance evaluation.
- XGBoost – For Extreme Gradient Boosting Model
- LightGBM – For Light Gradient Boosting Machine Model
- Mlxtend – For Stacking Classifier Model
- Imblearn – For balancing the dataset to get better accuracy for some algorithm.

4.4 Implementation

To predict heart disease, various Supervised Machine Learning models are used such as Logistic Regression, Decision Tree, K-Nearest Neighbor, Support Vector Machine, Naïve Bayes. At the very beginning, dataset is split into training and testing subset at the ratio of 70:30 after performing Data Preprocessing, Data Wrangling, EDA, and Feature Selection operation. Afore mentioned algorithms are verified with the train and test dataset for comparing their relevant accuracy. To improve accuracy of the model for predicting heart disease, various Ensemble Machine Learning approaches like Bagging, Boosting, and Stacking are deployed further. The whole process is described below sequentially.

4.4.1 Collecting Data

As Colaboratory is used, that dataset is need to be stored in Google Drive and should be mounted with Colaboratory each time I want access to the dataset. The initial CSV format dataset is converted in pandas DataFrame using Pandas module to proceed the analysis.

male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26.97	80.0	77.0	0
0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28.73	95.0	76.0	0
1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25.34	75.0	70.0	0
0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28.58	65.0	103.0	1
0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23.10	85.0	85.0	0

Figure – 7 Dataset in Pandas DataFrame

Then column information is checked to find missing values in those columns if any.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4240 entries, 0 to 4239
Data columns (total 16 columns):
#   Column              Non-Null Count  Dtype
---  -
0   male                4240 non-null   int64
1   age                 4240 non-null   int64
2   education            4135 non-null   float64
3   currentSmoker        4240 non-null   int64
4   cigsPerDay           4211 non-null   float64
5   BPMeds               4187 non-null   float64
6   prevalentStroke       4240 non-null   int64
7   prevalentHyp         4240 non-null   int64
8   diabetes             4240 non-null   int64
9   totChol              4190 non-null   float64
10  sysBP                4240 non-null   float64
11  diaBP                4240 non-null   float64
12  BMI                  4221 non-null   float64
13  heartRate            4239 non-null   float64
14  glucose              3852 non-null   float64
15  TenYearCHD           4240 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```

Figure – 8 Information about DataFrame

4.4.2 Data Preprocessing

As Machine Learning models are very sensitive towards missing values and outliers, it is very essential to make sure that the dataset is free from missing values or outliers. There are several methods that can be applied in case of dealing with the missing values in a dataset.

- Check the distribution of the attribute that contains missing values and fill those empty position with the central tendency according to variable type.
- After finding the attributes containing missing values, we need to find the related attribute/attributes if any. Following the related attributes distribution, missing values will be replaced with the central tendencies of the missing value's attribute. Which is more convenient.
- If above mentioned procedure is not applicable for any attributes, simply the rows are deleted that contains missing values. Which is least effective because we may loose our important information from the dataset.

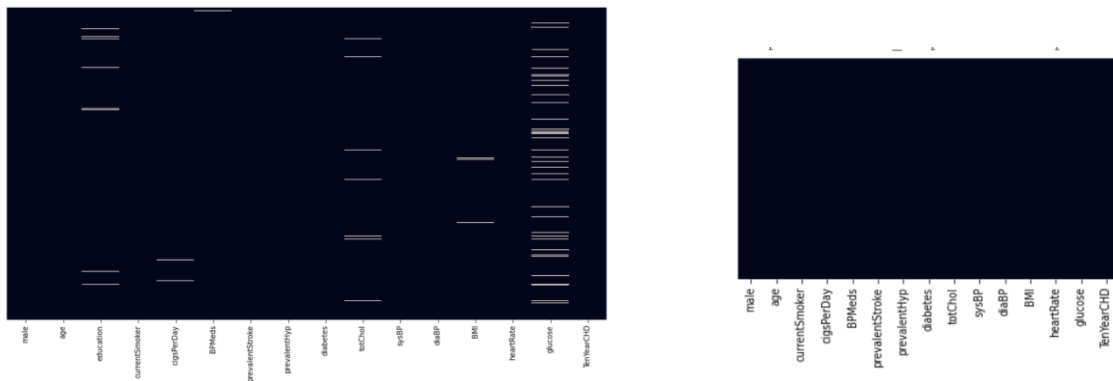


Figure – 9 Heatmap of the DataFrame before and after Missing Value Handling

There is no outlier is present in this dataset so the outlier removal part is not required here.

4.4.3 Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Two types of EDA are performed here to find the data insights.

- Univariate Exploratory Data Analysis
- Bivariate Exploratory Data Analysis

4.4.3.1 Univariate Exploratory Data Analysis

Uni means one and variate means variable, so in univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize it, and analyze the pattern present in it. In a dataset, it explores each variable separately. It is possible for two kinds of variables- Categorical and Numerical.

Some patterns that can be easily identified with univariate analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation.

Only histograms are used in this Univariate Analysis. Some of the distributions are given below:

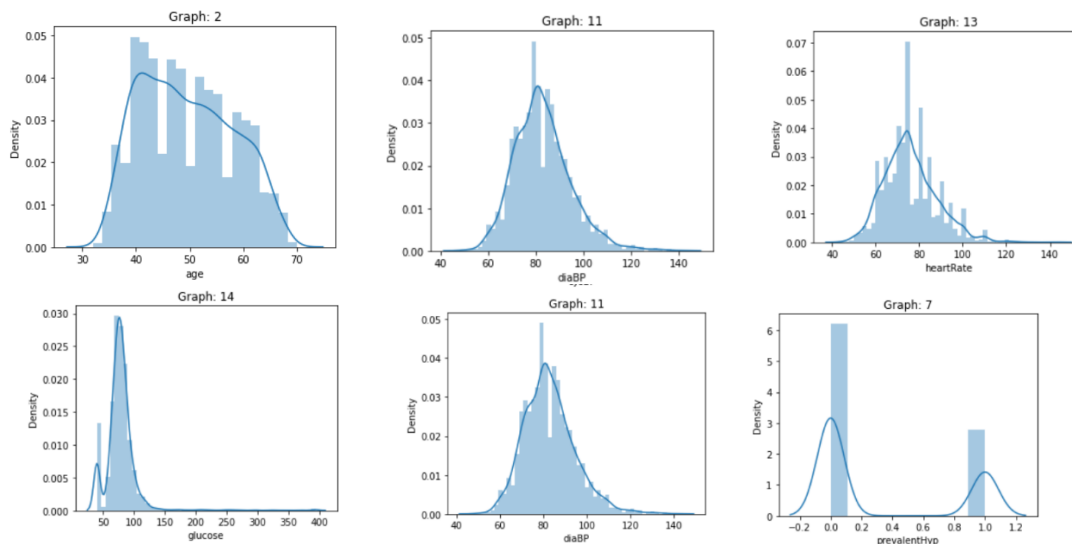


Figure – 10 Univariate Analysis

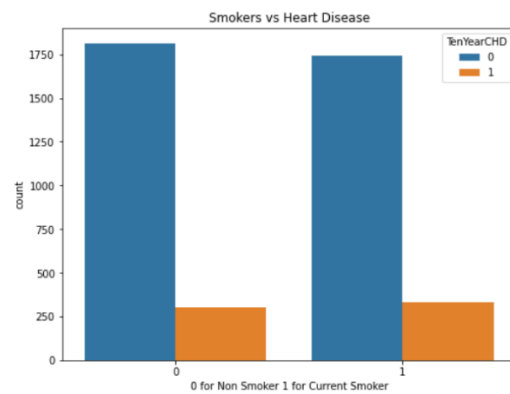
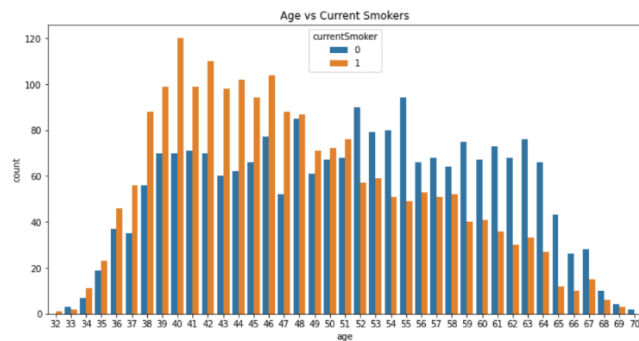
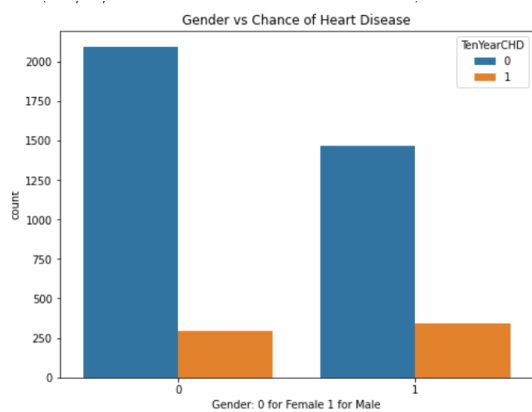
Univariate Analysis Result:

1. There are more females than males in this survey.
2. Survey participants age group is mostly in between 30 to 70.
3. There are equal number of current smokers and nonsmokers in this survey.
4. Maximum smokers in this survey consumes near about 20 cigarettes daily.
5. Very few people in this survey are in Blood Pressure Medication.
6. There are almost no previous family history of stroke in this survey.
7. Almost 1/3rd population of this survey has the record of hypertension in family history.
8. Very few people in this survey have diabetes.
9. Total cholesterol level in this survey has been observed between the range of 100-400 mainly. Also, maximum recorded cases lie between either side of 250.
10. Systolic Blood Pressure observed in this survey in between the range of 70-230 mainly. Also, maximum recorded cases lie between either side of 125.
11. Diastolic Blood Pressure observed in this survey in between the range of 55-130 mainly. Also, maximum recorded cases lie between either side of 85.
12. The range of 15-45 of BMI has been observed in this survey. Also, maximum recorded cases lie between either side of 28.
13. Heart Rate count in this survey has been observed between the range of 40-125 mainly. Also, maximum recorded cases lie between either side of 72.
14. The range of 25-150 of glucose level has been observed in this survey. Also, maximum recorded cases lie between either side of 75.

4.4.3.2 Bivariate Exploratory Data Analysis

Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them. ... It is the analysis of the relationship between the two variables.

Bar chart, Histograms, Regression Plot, Box Plot is used for Bivariate Analysis. Some the distributions are given below:



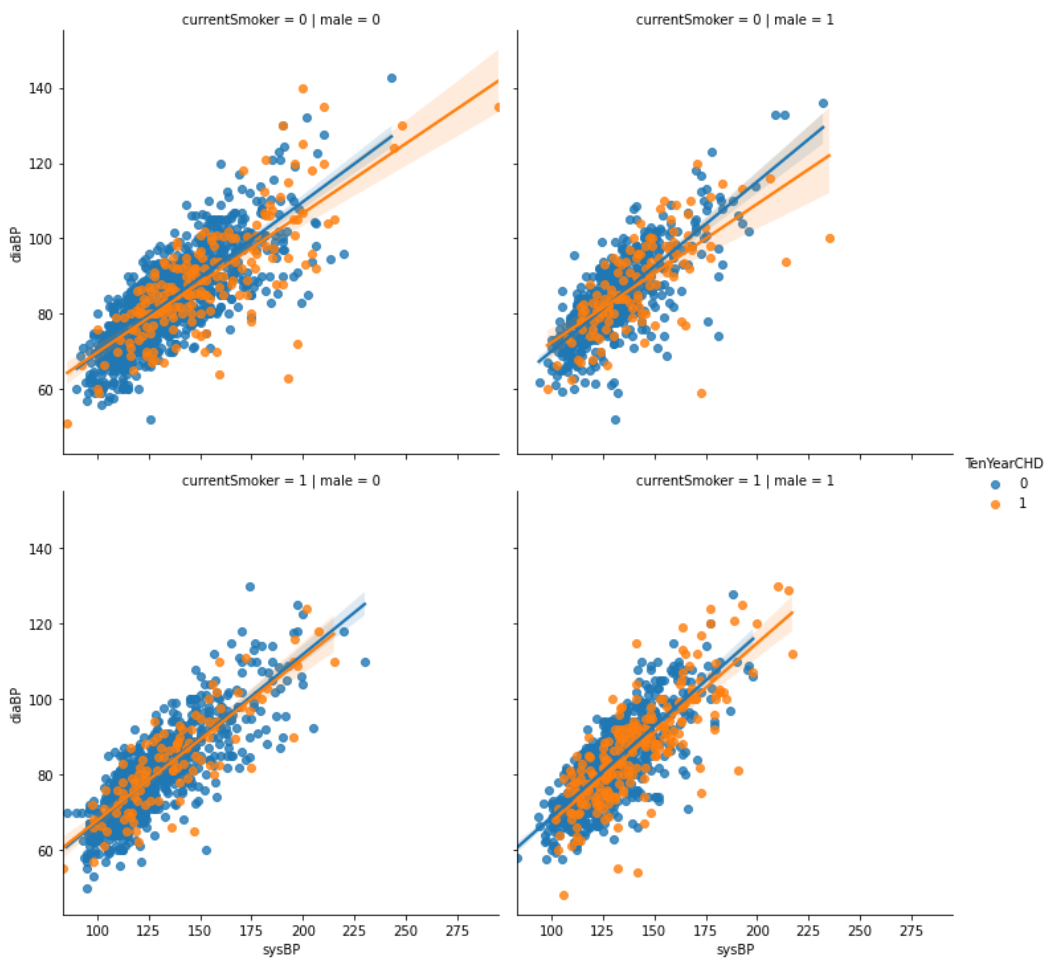
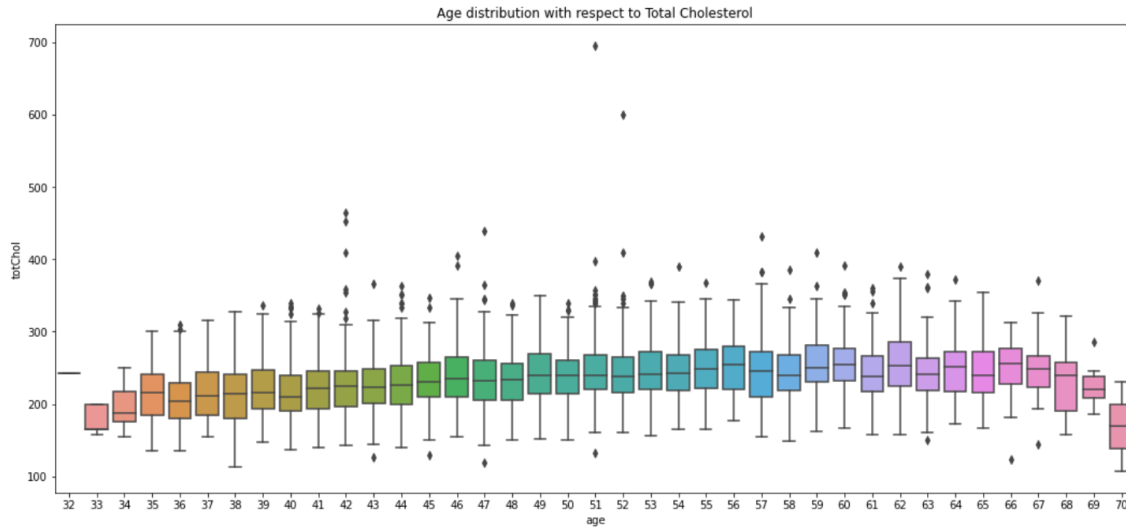


Figure – 11 Bivariate Analysis

Bivariate Analysis Result:

1. Gender vs Chance of Heart Disease: According to this survey data male are slightly prone to be future heart disease.
2. Age vs Current Smokers: More of the current smokers are in the 34-50 age group.
3. Age vs Cigarette Per day vs Heart Disease Risk: No such relation was found in this graph.
4. Smokers vs Heart Disease: Current Smokers are little bit prone to heart disease risk according to survey data.
5. Blood Pressure & Heart Rate vs Blood Pressure Medication vs Heart Disease Risk: No such direct relation was found between these variables plotted in the above graphs.
6. Age distribution with respect to Total Cholesterol: Middle aged population tends to have higher cholesterol levels in this survey.
7. Systolic Blood Pressure vs Diastolic Blood Pressure vs Age vs Current Smoker:
 - 7.1. In case of female nonsmoker as Systolic BP increases diastolic bp also increases, and risk gets higher according to this survey.
 - 7.2. In case of male nonsmoker increasing the systolic blood pressure increase the chance of getting heart disease in future according to survey data.
 - 7.3. In case of female smoker, no such relation is found in the graph.
 - 7.4 In case of male smoker increasing the systolic blood pressure increase the chance of getting heart disease in future according to survey data.

4.4.4 Balancing The Dataset

A balanced dataset is a dataset where each output class (or target class) is represented by the same number of input samples. Balancing can be performed by exploiting one of the following techniques:

- Over sampling
- Under sampling
- Class weight
- Threshold

To get unbiased result in a classification problem, balancing the dataset plays a key role (if the target variable is imbalanced). For my problem I use Over Sampling strategy to balance my target variable (TenYearCHD here). Common over sampling method or resample causes overfitting issues most of the time, to reduce this error, Synthetic Minority Oversampling Technique, or SMOTE is very effective and perhaps most widely used in terms of synthesizing new samples.

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

Specifically, a random example from the minority class is first chosen. Then k of the nearest neighbors for that example are found (typically $k=5$). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.

As my dataset contains categorical features, I used SMOTENC method instead of SMOTE from Imblearn module.

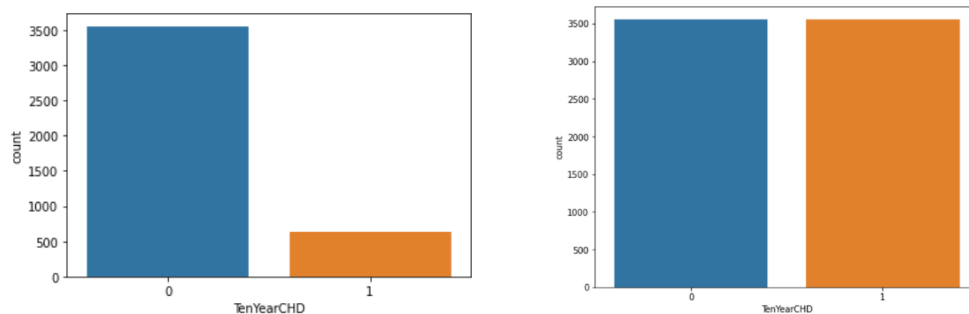


Figure – 12 Dataset before and after Balancing

4.4.5 Feature Selection

While building a machine learning model only the features which have a significant influence on the target variable should be selected. Advantages of the Feature Selection process are:

- Reduces Overfitting: Less redundant data means less possibility of making decisions based on redundant data/noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.

In this system I use SelectKBest method and Chi Square statistical method from Scikit Learn module which is a Univariate Feature Selection method. This method generates Chi Square statistics value for each variable with respect to the target variable and select the best valued attributes.

Here is the output given after the Feature Selection process:

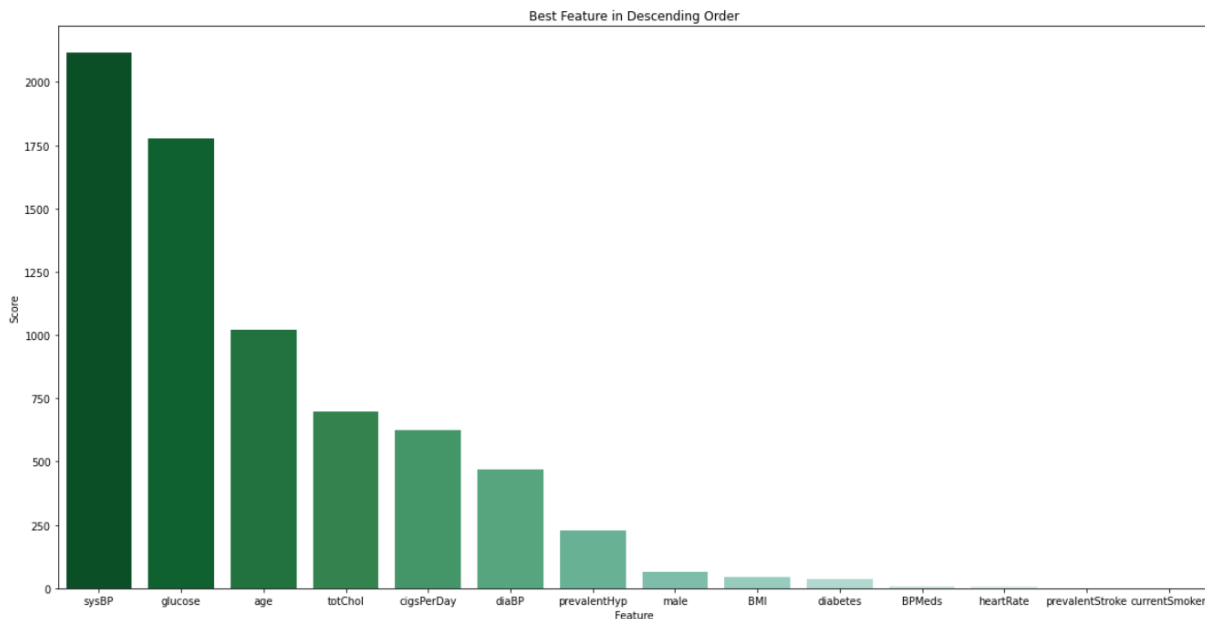


Figure – 13 Feature Importance according to Chi-Square value

Top 10 features are selected from this dataset for further operation.

4.4.6 Scaling And Splitting

4.4.6.1 Scaling

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Various types of scaling techniques are available right now. Some of the popular scaling techniques are:

- Standard Scaling
- Min-Max Scaling
- Robust Scaling

In this system Standard Scaling technique has been applied by importing Standard Scaler from Scikit Learn module.

Standardization: It is a very effective technique which re-scales a feature value so that it has distribution with 0 mean value and variance equals to 1.

$$X_{new} = \frac{X_i - X_{mean}}{Standard\ Deviation}$$

4.4.6.2 Splitting

As we are dealing with a Supervised Machine Learning problem it's essential to split our dataset into training and testing category for model training and further prediction and validation. It is obvious that any model's performance and accuracy rigorously dependable on train and test size of the respective dataset. In general, training dataset is larger than test dataset.

Train_test_split method from Scikit Learn module is used to satisfy this approach.

4.4.7 Predictive Modeling

To leverage the goal of predicting future 10-year risk of Heart Disease, predictive model building which is implemented as the final and most important section of this whole process is followed by the performance evaluation. In case of Supervised Learning model building simply means, both features and target variable are used to define train-set which is used to train the model whereas only features are used in test set to predict the target variable.

In our classification problem several Supervised Classifier models are implemented and their accuracies are compared to get the best accuracy for this particular dataset. Generally, two types of Classifier model can be observed in this system.

- Supervised Single Models
- Supervised Ensemble Models

4.4.7.1 Single Models

4.4.7.1.1 Logistic Regression

Logistic Regression is the process where the probability of discrete response variable is found with respect to explanatory variables. Mainly, Logistic Regression concludes a binary outcome such as true/false, yes/no, 0/1 and so on. Multiple Logistic Regression can have more than two discrete outcomes. Logistic Regression can resolve most of the classification problems. The formula of LR is as follows:

$$F_x = \frac{1}{1 + e^{-B_0 + B_1x}}$$

Here $B_0 + B_1x$ is similar to the linear model $y = mx + c$. The logistic function applies a sigmoid function to restrict the y value from a large scale to within the range 0–1.

In this case, Linear Regression method is used from Scikit Learn's linear module.

4.4.7.1.2 K-Nearest Neighbor (KNN)

KNN algorithm recognizes same type of existing datapoints which stays close to each other. This are called neighbors in the algorithm. Here K is used as a parameter which determines the number of closest neighbors, given as input in the algorithm. Similarity between the data points can be obtained by measuring the in between distance such as Euclidian distance, Manhattan distance or any form of Minkowski distance. KNN is resourceful to use for classification and regression problems where K is always greater than 0.

In this system, KNN method is used from Scikit Learn's neighbor module.

4.4.7.1.3 Decision Tree

A Decision Tree is a tree like structure of effective decisions that makes possible consequences of a given input, leading to the chance event and final outcome. In another definition, Decision Tree is an algorithm based on conditional probability. It is formed by taking the whole dataset as the root node of the tree, and further the dataset is broken into multiple nodes in hierarchical structure by applying some simple decision rules. Three types of nodes can be seen in a decision tree:

- The nodes that don't have any ancestors are called root nodes or decision nodes.
- The nodes they both have ancestor and successor are called internal nodes or chance nodes.
- The nodes that only have ancestor with no successor present are called leaf node or final outcome.

Decision tree deals with both categorical and numerical data and as well as classification and regression problems.

In this system, Decision Tree method is used from Scikit Learn's tree module.

4.4.7.1.4 Support Vector Machine

In Support Vector Machine a hyperplane is used to differentiate data object into two categories in case of classification problems. A decision boundary or hyperplane is chosen when the average distance is maximized between the support vectors (datapoints near to decision boundary) and decision boundary. It can be used for both classification and regression problems.

In this system, Support Vector Machine is used from Scikit Learn's svm module.

4.4.7.1.5 Naïve Bayes

Naïve Bayes is a supervised machine learning approach for categorize the datapoint into its related classes. It takes the feature attributes (F_i for $i = 1, 2, \dots, n$) and target attributes (C_j for $j = 1, 2, \dots, m$) and calculate the posterior probability of C_j , by using Bayes theorem $P(C_j | F_i)$. Then it compares the values got from each target class attributes, to find the maximum value and categorize it into that class from which it gets maximum probabilistic value.

It makes calculations simpler, easier and faster for large datasets comparatively. It makes an blind assumption that the attributes are distributed normally and independent of each other, but in case of real life data which seems quite impossible.

In this system, Naïve Bayes is used from Scikit Learn's naïve_bayes module.

4.4.7.2 Ensemble Models

4.4.7.2.1 Random Forest

Random forest is a ensemble supervised machine learning approach which generate multiple decision trees simultaneously by splitting the training data into random multiple subset. Using more decision trees give more accuracy and robustness instead of using a single decision tree model. Once the output is generated in each decision tree Random Forest algorithm takes the final output by using majority voting method.

It can be used for both classification and regression problems.

In this system, Random Forest is used from Scikit Learn's ensemble module.

4.4.7.2.2 Extreme Gradient Boosting

Extreme Gradient Boosting or XGBoost is a supervised ensemble learning approach used for both classification and regression problems. It's a advanced form of Gradient Boosting which follows also tree based structure and boosting mechanism to reduce error. Both Gradient Boosting and Extreme Gradient Boosting uses regression trees instead of classification trees. To split the nodes, MSE (Mean Square Error) is used in Gradient Boosting whereas Similarity Score and Gain resolve this problem in case of Extreme Gradient Boosting Algorithm.

Similarity Score is defined using the help of Residuals, Previous Probability and a Regularization parameter lambda.

$$\text{Similarity Score} = \frac{(\sum_{i=1}^n \text{Residual}_i)^2}{\sum_{i=1}^n [\text{Previous Probability}_i * (1 - \text{Previous Probability}_i)] + \lambda}$$

After having Similarity Score regarding each leaf, **Gain** can be calculated. The formula is as follows:

$$\text{Gain} = \text{Left Leaf}_{\text{similarity}} + \text{Right Leaf}_{\text{similarity}} - \text{Root}_{\text{similarity}}$$

Now the parameters are:

Residual: - Difference between actual output and predicted output.

Previous Probability: - Previous probability of an event is a value of possibility of this particular event at the previous tree or model. To build the first tree, by default probability is set to 0.5. Later, the previous probability is reconsidered according to all of the previous models or trees.

N.B.: For Regression problem summation of Previous Probability multiplied by 1 – Previous Probability is replaced by the number of Residuals.

Lambda (λ): - Lambda is a regularization parameter which controls over pruning the trees. Increasing the value of Lambda is more conservative towards splitting the nodes.

Gamma (γ): - Gamma is set to determine if the gain is closer to the desired value or not. If the gain is greater than gamma then splitting occurs otherwise not.

Eta (η): - Eta is learning rate similar to the Gradient Boosting algorithm. Eta is in between 0 and 1. Higher value of Eta proposes high learning rate. In some cases, overfitting may occur then Eta can be decreased to resolve this.

Evaluation Function: - Evaluation function is the objective function of XGBoost algorithm. By default, loss function is evaluated through all the predictions and regularization functions to construct the aggregate loss function. It depends on type of task being performed such as Regression and Classification. The below formula describes loss function properly where F_j means a prediction coming from Jth tree.

$$Loss = \sum_i^n l(y_i - \hat{y}_i) + \sum_{j=1}^J \Omega(F_j)$$

For classification problem sometimes Logloss performs better, which a probability-based metric which is defined below:

$$logloss = -\frac{1}{N} \sum_{i=1}^N y_i * \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i))$$

But in case of Regression problems, **MSE, MAE, RMSE** such type of metrics are popular.

In this system, Extreme Gradient Boosting is used from XGBoost module.

4.4.7.2.3 Light Gradient Boosting Machine

Light Gradient Boosting Machine or LGBM is a form of Gradient Boosting mechanism, which is known for its higher accuracy, less time complexity, less memory usage, supporting parallel and GPU learning, and efficiency of large-scale data handling. Every day when the digital world is exploiting more and more data, LGBM is outperforming other nowadays with a more optimal approach than the other algorithm which were said to be ruling the analytics world 3/4 years

before. Optimality in LGBM can be achieved in two ways:

- In terms of Speed and Memory usage.
- In terms of Accuracy.

Optimality in Speed and Memory usage:

Histogram- based algorithms are used to transform continuous features into discrete bins in LGBM algorithm, resulting enhanced speed and accuracy of the overall performance.

Its advantages are:

- It reduces cost of calculating the gain for each split by using pre-sort-based algorithm and histogram computing to attain less time complexity.
- It uses a mechanism called histogram subtraction which is responsible for boosting of speed in computing.
- The usage of memory is lessened because continuous values are replaced with discrete bins.
- When it comes to distributed learning, the communication cost is far reduced.

Optimality in Accuracy:

Most of the tree-based ML algorithms grow their trees level wise (depth wise) or horizontally and more loss is exposed generally like Gradient Boosting, XGBoost. The formation of the tree is described in the below diagram:



Figure – 14 Level-wise Tree Growth

LGBM grows trees vertically or leaf wise which means the leaf which has maximum delta loss is chosen and the tree continues to grow from there. Leaf wise algorithms are more efficient in reducing more loss than level wise algorithms when growing the same leaf. This method is called **Gradient Based One Side Sampling**.

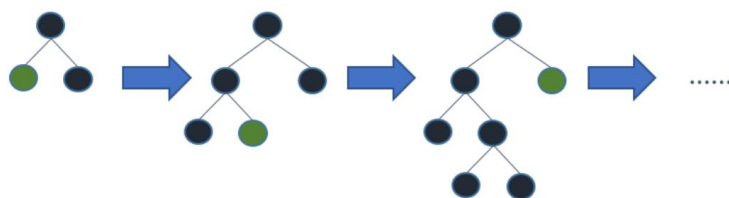


Figure – 15 Leaf-wise Tree Growth

Another novel method, **Exclusive Feature Bundling (EFB)** is introduced in LGBM which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. Its main focus is to reduce sparsity of the high dimensional data. As a result, exclusive features containing sparse values are confined to a single one. This mechanism optimizes the histogram's computational speed and time complexity.

These two methods comprise together to make the model work efficiently for LGBM model and provide it a cutting edge over other Gradient Boosting models.

Evaluation Metrics:

Some supported metrics here listed below:

- L1 loss
- L2 loss
- Log loss
- Classification error rate
- AUC etc.

Application:

Some of the renowned applications of this algorithm are as follows:

- Regression, L2 loss is the objective function.
- Binary classification, the objective function is log loss
- Cross-entropy
- Multi-classification
- Lambda rank, lambda rank with NDCG is the objective function.

Limitation:

LGBM algorithm is very prone to overfit in case of smaller datasets. That's why extra split is controlled by using `max_depth` parameter with lower value.

In this system, Light Gradient Boosting Machine is used from LightGBM module.

4.4.7.2.4 Stacking Classifier

Stacking Classifier is an ensemble learning method, which uses heterogeneous models instead of homogeneous models to predict output. For predicting mechanism, a collection **Base Models** and a **Meta-Model** is used in Stacking Classifier.

Base Models:

Initial data is split into trained dataset and test dataset, trained dataset is split further. With the help of train dataset, heterogeneous models (Logistic Regression, KNN, SVM, XGBoost etc.) are created. These models are known as Base Models. Even Base Model can be defined using previously build models of other algorithms.

Meta-Model:

A Meta-Model is produced combining all the Base Model's prediction.

The internal process of a Stacking Classifier is explained in below diagram:

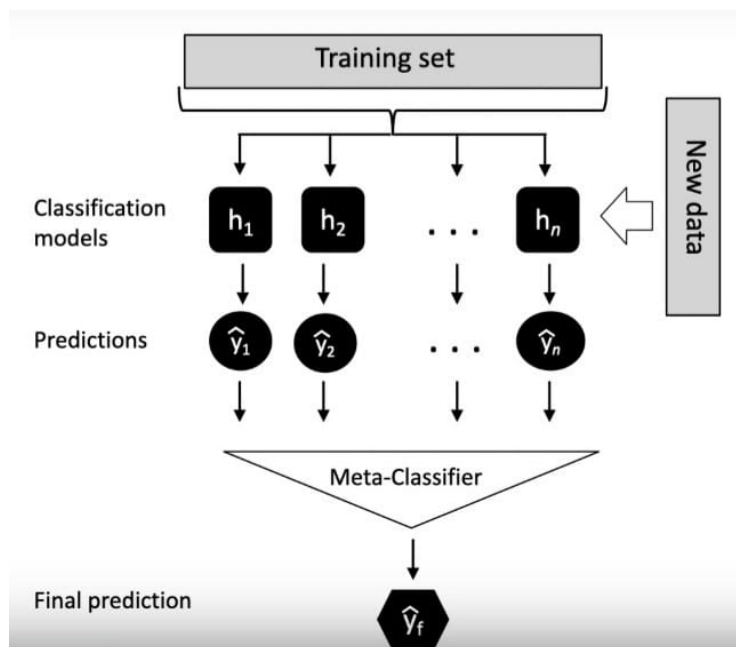


Figure – 16 Stacking Classifier

If the initial trained dataset is divided using KFold cross validation then the phenomena is known as **Stacking** and if the dataset is split normally then the process is called **Blending**.

In this system, Stacking Classifier is used from Mlxtend module.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Validation

Model validation imposes a test dataset to evaluate a prediction process. A bigger portion of the dataset is used to train the model whereas remaining portion is left for predicting and testing. Five-fold cross-validation is applied to propagate accuracy score of all the single model in this process. Those single models applied here are: Logistic Regression, Decision Tree, K-Nearest Neighbor, Support Vector Machine, Naïve Bayes. Train_test_split function from Scikit Learn library split the dataset into the ratio of 70-30 for ensemble learning algorithms (Random Forest, XGBoost, LGBM). Finally, the model validation is executed after model training is accomplished.

5.2 Accuracy Metrics

To analyze the performance of classification models, several accuracy metrics such as Accuracy Score, F1 score, Precision, Recall, Confusion Matrix are used. The key ingredient to satisfy this need is Confusion Matrix.

Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The dimension of confusion matrix increases with increasing the number of output classes. This means, the size of matrix is 2x2 for binary classification.

A binary classification confusion matrix is shown in the below diagram.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

Figure – 17 Confusion Matrix

Now some of the Classification performance measure metrics are described:

Accuracy Score:

Accuracy score is one of the most intuitive performance measure metrics, which is defined as the ratio of true predicted output to total number of outputs.

One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model.

$$\text{Accuracy Score} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision:

Precision is the ratio of correctly predicted positive to total predicted positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall:

Recall is the ratio of correctly predicted positive to total number of actual positive. Recall is also called **Sensitivity**.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Specificity:

Specificity is the ratio of correctly predicted negative to total number of actual negative.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

F1 Score:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. When the dataset is imbalanced, the Accuracy score metric becomes less trustworthy and F1 Score comes in this situation to rescue.

$$\text{F1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}$$

5.3 Results

Initially, the dataset was quite imbalanced, synthetic datapoints are used to eliminate imbalance that the Accuracy score can be considered as performance measurement. The single models and ensemble models are performed in this program. In single models, cross validation score generates the accuracy which is plotted in a Bar chart.

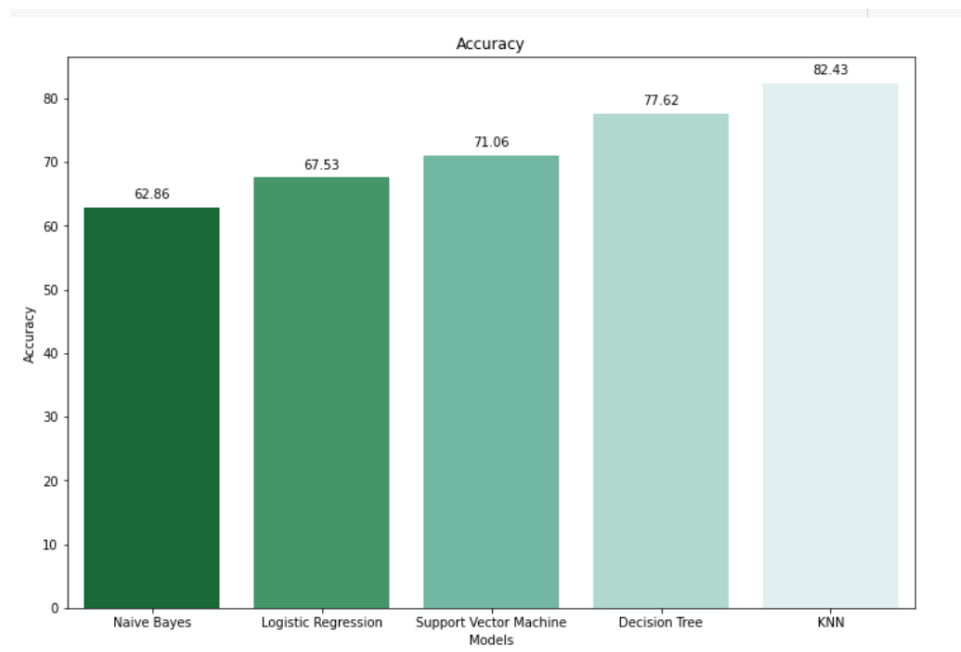


Figure – 18 Single Models and their Accuracy

In case of Ensemble models, simply training and validation is performed and the final accuracy is plotted in Bar graph.

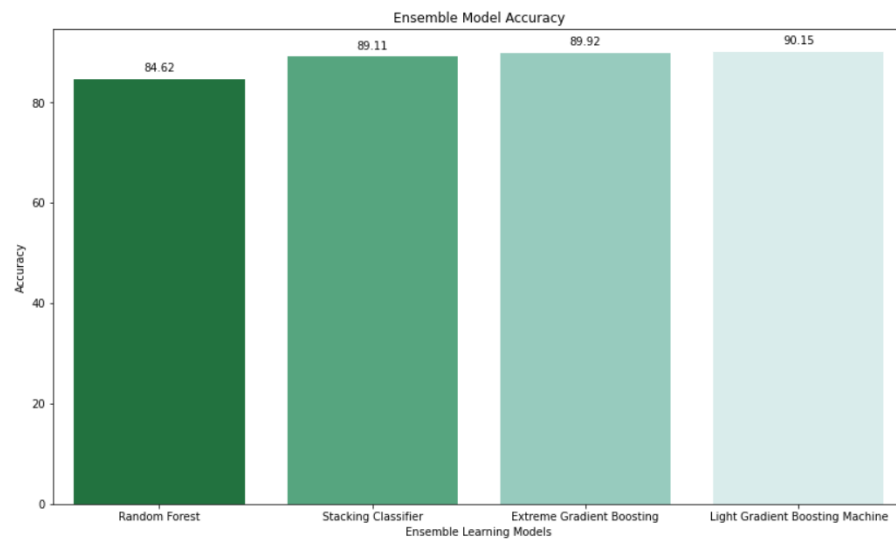


Figure – 19 Ensemble Models and their Accuracy

5.3 Comparison

Now it's time to compare the efficiency of the different model implement in this system. Every model's accuracy is represented in a Bar graph, which is shown next.

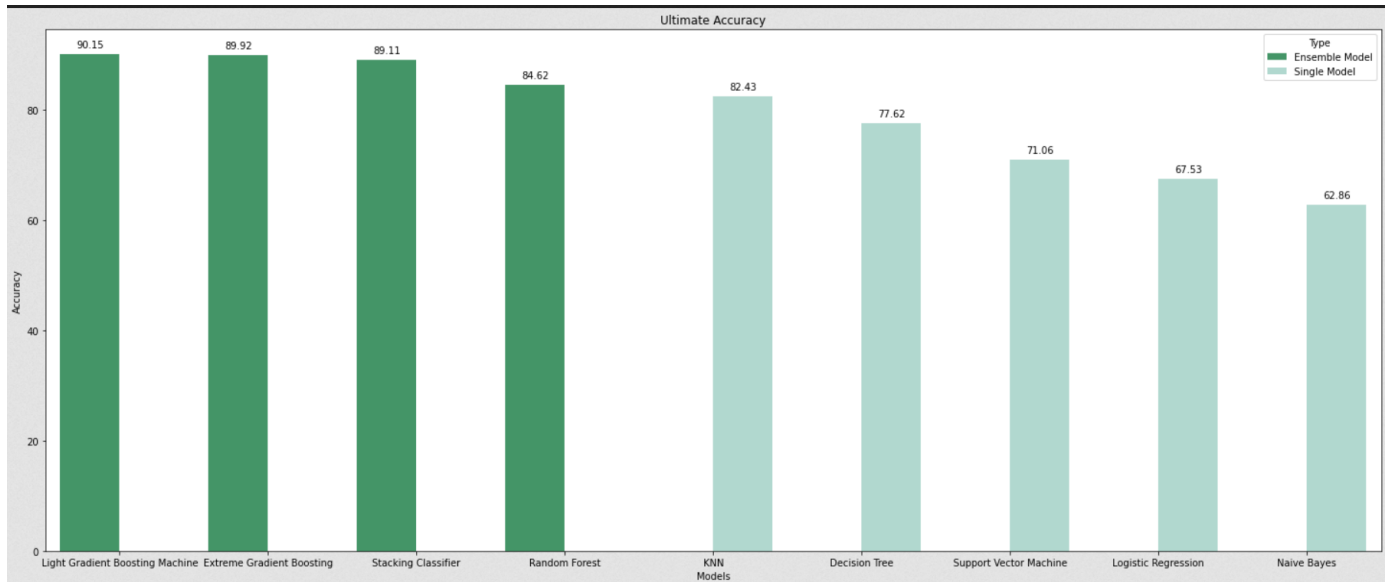


Figure – 20 Every implemented models and their accuracy comparison

It's clearly seen from this above diagram that every Ensemble models simply outperformed single models. Light Gradient Boosting Machine has the best performance for this dataset, having accuracy of 90.15%. But Extreme Gradient Boosting and Stacking Classifier come very close with the accuracy of 89.92 and 89.11 respectively. Random Forest algorithm provides 84.62% accuracy which is lowest among all ensemble models but higher than any other single models. In case of single models, KNN comes first with 82.43% accuracy followed by Decision Tree, Support Vector Machine, Logistic Regression, with the accuracy of 77.62%, 71.06% and 67.53% respectively. Naive Bayes has the worst performance in this system with 62.86% accuracy score.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this report, a Machine Learning based system is designed to predict heart disease risk for future 10years. After preprocessing, and exploratory data analysis, several models are built such as Single Model like Logistic Regression, KNN, SVM, Naive Bayes, DT and Ensemble model like LGBM, XGBoost, Stacking Classifier and Random Forest. The focus of this project is to show how Ensemble Learning model can be more accurate and robust than the Single Models. After getting the accuracy scores which are all plotted in a bar chart and compared with each other, present the ensemble models are clearly outperforming single models. The information derives from this method can predict accurately whether a person may or may not face heart disease risk in future 10 years.

This project involves in a very beneficial remedy to the medical field. In the past few years, heart disease prediction using ML/DL became one of the most concerned topics in the healthcare section of AI application primarily focused on accuracy and optimality. A model with a better accuracy can actually bring light to the problems related to the heart disease. As we discussed before those traditional approaches are less focused to predict heart disease at an early stage, hence the models of AI application not only resolve this issue by predicting it at an early stage but also, reduce the cost related to medical diagnosis and treatment. Surprisingly, Heart Disease also damages various kind of regular activity in life as well as physical fitness. An early prediction of heart disease certainly prevents all this damages by providing a healthy life style, food habit, cognitive ability and a concerned mind.

6.1 Future Work

The proposed methods of Ensemble Learning achieved higher accuracy than the traditional learning problem. But this accuracy can be more optimal by using Hyper parameter tuning mechanism. Hyperparameters are the parameters of every model passed during their instance creation. The value of hyperparameters highly differs with respect to the dataset. Although, most of the Ensemble Learning models are faster than single models, but the models being prone to overfit, actually take several extra parameters to handle this problem. Here, the refereed parameters consist of so many regularization parameters like gamma, lambda, eta, decision tree controls parameters like max depth, tree numbers, bootstrap meter, hardware consumption parameters etc. So, the parameter grid of hyperparameter tuning becomes so large with so many models and each model need to be trained to achieve the best hyperparameter combination, resulting a very costly process.

This gap should be filled up in the future by using an optimal mechanism of hyperparameter tuning to reduce cost.

Moreover, to make this system more demanding, it can be assembled with the chatbot mechanism in a cloud-based GUI application where a person can provide his/her health details by simply chatting and the model would reply if he/she has any chance of getting heart disease in next 10 years. This will be very convenient for a user.

REFERENCES

- [1] Senthilkumar Mohan, Chandrasegar Thirumalai and Gautam Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," IEEE published, vol 3, July 3 2019.V. Medina, R. Valdes, J. Azpiroz, and E. Sacristan, "Title of paper if known," unpublished.
- [2] Aditi Gavhane, Gouthami Kokkula, Isha Pandya, Prof. Kailas Devadkar, "Prediction of Heart Disease Using Machine Learning", IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1 (ICECA 2018)
- [3] Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis," Int. J. Comput. Sci. Issues, vol. 8, no. 2, pp. 150-154, 2011.J. E. Monzon, "The cultural approach to telemedicine in Latin American homes (Published Conference Proceedings style)," in *Proc. 3rd Conf. Information Technology Applications in Biomedicine, ITAB '00*, Arlington, VA, pp. 50–53.
- [4] Tülay Karayılan and Özkan Kılıç "Prediction of Heart Disease Using Neural Network", IEEE-Published 2017.
- [5] Dinesh Kumar G, Arumugaraj K, Santhosh Kumar D, Mareeswari V, "Prediction of Cardiovascular Disease Using Machine Learning Algorithms", Publisher: IEEE, Proceeding of 2018 IEEE International Conference.
- [6] J. Lopez-Sendon, "The heart failure epidemic," *Medicographia*, vol. 33, no. 4, pp. 363-369, 2011.
- [7] P. A. Heidenreich, J. G. Trogon, O. A. Khavjou, J. Butler, K. Dracup, M. D. Ezekowitz, E. A. Finkelstein, Y. Hong, S. C. Johnston, A. Khera, D. M. Lloyd-Jones, S. A. Nelson, G. Nichol, D. Orenstein, P.W. F.Wilson, and Y. J. Woo, "Forecasting the future of cardiovascular disease in the united states: A policy statement from the American heart association," *Circulation*, vol. 123, no. 8, pp. 933-944, 2011.
- [8] S. I. Ansarullah and P. Kumar, "A systematic literature review on cardiovascular disorder identification using knowledge mining and machine learning method," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6S, pp. 1009-1015, 2019.
- [9] S. Palaniappan and R. Awang, "Intelligent heart disease prediction system using data mining techniques," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Mar. 2008, pp. 108-115.
- [10] Mr Santhana Krishnan. J , Dr Geetha. S, "Prediction of Heart Disease Using Machine Learning Algorithms", 26 April, 2019.
- [11] Ching Wei Wang," New Ensemble Machine Learning Method for Classification and Prediction on Gene Expression Data", Proceeding of 2016 IEEE International Conference.
- [12] 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. 27–40, Jan. 2012.

doi: 10.1016/j.jksuci.2011.09.002.

- [13] A. S. Abdullah and R. R. Rajalaxmi, "A data mining model for predicting the coronary heart disease using random forest classifier" in Proc. Int. Conf. Recent Trends Comput. Methods, Commun.Controls, Apr.2012, pp.22–25.
- [14] J.Vijayashree and N.Ch. Sriman Narayana Iyengar, "Heart Disease Prediction System Using Data Mining and Hybrid Intelligent Techniques : A Review " , Vol.8, No.4 (2016), pp. 139-148
- [15] C. Beulah Christalin Latha, S. Carolin Jeeva," Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques ", Informatics in Medicine Unlocked 16 (2019)
- [16] E. O. Olaniyi, O. K. Oyedotun, and K. Adnan, ``heart diseases diagnosis using neural networks arbitration," Int. J. Intell. Syst. Appl., vol. 7, no. 12, p. 72, 2015.
- [17] G. G. N. Geweid and M. A. Abdallah, ``A new automatic identification method of heart failure using improved support vector machine based on duality optimization technique," IEEE Access, vol. 7, pp. 149595-149611, 2019.
- [18] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang,"Disease prediction by machine learning over big data from healthcare communities", IEEE Access, vol. 5, 2017.
- [19] Seyedamin Pouriyeh, Sara Vahid, Hamid Reza Arabnia and Giovanna Sannino, "A Comprehensive Investigation on Comparison of Machine Learning Techniques on Heart Disease Domain ", July, 2017
- [20] An Dinh, Stacey Miertschin, Amber Young and Somya D.Mohanty, "A data-driven approach to predicting diabetes and cardiovascular disease with machine learning ", 2019.
- [21] C. Beulah Christalin Latha, S. Carolin Jeeva," Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques ", Informatics in Medicine Unlocked 16 (2019)
- [22] S. Radhimeenakshi, ``Classification and prediction of heart disease risk using data mining techniques of support vector machine and artificial neural network," in Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom), New Delhi, India, Mar. 2016, pp. 3107-3111.
- [23] T. Vivekanandan and N. C. S. N. Iyengar, ``Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease," Comput. Biol. Med., vol. 90, pp. 125-136, Nov. 2017.
- [24] C. Sowmiya and P. Sumitra, ``Analytical study of heart disease diagnosis using classification techniques," in Proc. IEEE Int. Conf. Intell. Techn. Control, Optim. Signal Process. (INCOS), Mar. 2017, pp. 1-5.
- [25] M. S. Amin, Y. K. Chiam, K. D. Varathan, ``Identification of significant features and data mining techniques in predicting heart disease," Telematics Inform., vol. 36, pp. 8293, Mar. 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0736585318308876>

- [26] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems* 30 (NIPS 2017), pp. 3149-3157.
- [27] Mehta, Manish, Rakesh Agrawal, and Jorma Rissanen. "SLIQ: A fast scalable classifier for data mining." *International Conference on Extending Database Technology*. Springer Berlin Heidelberg, 1996.
- [28] Shafer, John, Rakesh Agrawal, and Manish Mehta. "SPRINT: A scalable parallel classifier for data mining." *Proc. 1996 Int. Conf. Very Large Data Bases*. 1996.
- [29] Thakur, Rajeev, Rolf Rabenseifner, and William Gropp. "Optimization of collective communication operations in MPICH." *International Journal of High-Performance Computing Applications* 19.1 (2005), pp. 49-66.
- [30] Walter D. Fisher. "On Grouping for Maximum Homogeneity." *Journal of the American Statistical Association*. Vol. 53, No. 284 (Dec., 1958), pp. 789-798.
- [31] Huan Zhang, Si Si and Cho-Jui Hsieh. "GPU Acceleration for Large-scale Tree Boosting." *SysML Conference*, 2018.
- [32] Ranka, Sanjay, and V. Singh. "CLOUDS: A decision tree classifier for large datasets." *Proceedings of the 4th Knowledge Discovery and Data Mining Conference*. 1998.
- [33] Machado, F. P. "Communication and memory efficient parallel decision tree construction." (2003).
- [34] Li, Ping, Qiang Wu, and Christopher J. Burges. "Mcrank: Learning to rank using multiple classification and gradient boosting." *Advances in Neural Information Processing Systems* 20 (NIPS 2007).
- [35] Tianqi Chen, Carlos Guestrin "XGBoost: A Scalable Tree Boosting System" (2016)
- [36] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Inf. Syst.*, vol. 2018, pp. 1-21, Dec. 2018.
- [37] A. U. Haq, J. Li, M. H. Memon, J. Khan, S. U. Din, I. Ahad, R. Sun, and Z. Lai, "Comparative analysis of the classification performance of machine learning classifiers and deep neural network classifier for prediction of parkinson disease," in *Proc. 15th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2018, pp. 101-106.
- [38] V. Bolón-Canedo, N. Sánchez-Marroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci.*, vol. 282, pp. 111-135, Oct. 2014.
- [39] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data

- mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1-37, 2008.
- [40] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1-27, Apr. 2011.
 - [41] R. Sivaranjani, V. S. Naresh, and N. V. Murthy, "4 coronary heart disease prediction using genetic algorithm-based decision tree," *Intell. Decis. Support Syst., Appl. Signal Process.*, vol. 4, p. 71, Oct. 2019.
 - [42] J. Mourão-Miranda, A. L. W. Bokde, C. Born, H. Hampel, and M. Stetter, "Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional MRI data," *NeuroImage*, vol. 28, no. 4, pp. 980-995, Dec. 2005.
 - [43] R. Detrano, A. Janosi, W. Steinbrunn, M. Psterer, 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," *Amer. J. Cardiol.*, vol. 64, no. 5, pp. 304-310, Aug. 1989.
 - [44] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Informat.*, vol. 85, pp. 189-203, Sep. 2018.
 - [45] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 2013.
 - [46] K. Larsen, J. H. Petersen, E. Budtz-Jørgensen, and L. Endahl, "Interpreting parameters in the logistic regression model with random effects," *Biometrics*, vol. 56, no. 3, pp. 909-914, 2000.
 - [47] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226-1238, Aug. 2005.
 - [48] J. Lopez-Sendon, "The heart failure epidemic," *Medicographia*, vol. 33, no. 4, pp. 363-369, 2011.
 - [49] E. O. Olaniyi, O. K. Oyedotun, and K. Adnan, "heart diseases diagnosis using neural networks arbitration," *Int. J. Intell. Syst. Appl.*, vol. 7, no. 12, p. 72, 2015.
 - [50] Heart disease Dataset- www.kaggle.com
 - [51] Used Website for further information www.analyticsvidya.com, towardsdatascience.com, dataaspirant.com, medium.com, www.geeksforgeeks.org, etc...

(Above style is based on IEEE, you may also use Harvard or Chicago styles)













PLAGIARISM REPORT



Document Information

Analyzed document	Sample pages MSc Mini-project.docx (D110463147)
Submitted	7/14/2021 9:25:00 PM
Submitted by	
Submitter email	arghyadeepm97@gmail.com
Similarity	9%
Analysis address	cenlib2014.bhuni@analysis.orkund.com

Sources included in the report

SA	FINAL THESIS RITU.pdf Document FINAL THESIS RITU.pdf (D109206336)		4
W	URL: https://www.researchgate.net/publication/344431213_Machine_Learning_for_Multiple_Stage_Heart_Disease_Prediction Fetched: 6/21/2021 7:51:10 AM		4
SA	THESIS-1.docx Document THESIS-1.docx (D108395715)		1
SA	ICISI_2021_4_100.docx Document ICISI_2021_4_100.docx (D110142263)		3
W	URL: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/ Fetched: 7/14/2021 9:26:00 PM		1
W	URL: https://www.researchgate.net/publication/351483282_Elimination_and_Backward_Selection_of_Features_P-Value_Technique_In_Prediction_of_Heart_Disease_by_Using_Machine_Learning_Algorithms Fetched: 6/6/2021 8:35:33 AM		1
SA	THESIS.docx Document THESIS.docx (D108287960)		1
SA	19001507001-dissertaion - II.docx Document 19001507001-dissertaion - II.docx (D108817868)		2
SA	THESIS.docx Document THESIS.docx (D108077506)		1
W	URL: https://analyticsindiamag.com/primer-ensemble-learning-bagging-boosting/ Fetched: 7/14/2021 9:26:00 PM		1
SA	Thesis Work.docx Document Thesis Work.docx (D108344893)		1
W	URL: https://www.ijrte.org/wp-content/uploads/papers/v8i3/B2046078219.pdf Fetched: 7/14/2021 9:26:00 PM		2