BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

Inspiring Excellence

Comparative Analysis between Machine Learning Algorithms in Efficiency of Coronary Heart Disease (CHD) Prediction

AUTHORS

Fayza Rezwana Oishi
Mehnaj Al Mahadi
Omar Bin Parvez

SUPERVISOR

Hossain Arif

Assistant Professor
Department of CSE

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements of the degree of
B.Sc. in CS/ B.Sc. Engineering in CSE

December 2018

To our ever supporting faculty body, seniors and well wishers of the department and beyond. Love goes out to our friends and family for giving us the latent energy we always needed to get through this.

Declaration

It is hereby declared that this thesis/project report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Authors

Fayza Rezwana Oishi	Mehnaj Al Mahadi	Omar Bin Parvez
Student ID: 18201215	Student ID: 13201076	Student ID: 18241031

Supervisor

Hossain Arif
Assistant Professor, Department of Computer Science and Engineering
BRAC University

December 2018

The thesis titled .....
Submitted By:
Fayza Rezwana Oishi, Student ID: 18201215
Mehnaj Al Mahadi, Student ID: 13201076
Omar Bin Parvez, Student ID: 18241031
of Academic Year ..... has been found as satisfactory and accepted as partial fulfillment of the requirement for the Degree of .....

1. _____ Chairman
Hossain Arif
Assistant Professor

2. _____ Member
Md. Abdul Mottalib, PhD
Chairperson

Acknowledgments

We would like to acknowledge the patience of our Supervisor, for bearing with us through the full year and guiding us in any way possible towards the right directing whenever we found ourselves derailed; for giving us the chance to try and do what we wanted to do.

Acknowledgment also goes out to our senior brother for questioning us during meetings and helping us strive to know better and do better.

Thank you for being there always!

Abstract

The world of Machine Learning is expanding everyday through its implementations in modern day healthcare. Researchers have sketched out many ways to implement Machine Learning algorithms and droned into ways to make them work in their utmost efficiencies. As there will always be the need for healthcare in the world, we believe that there will always be a need of comparison between Machine Learning algorithms in terms of their performance and relevance to make healthcare more reliable through Machine Learning. For this study, we have picked up the most commonly used Machine Learning algorithms, Logistic Regression, Support Vector Machine, Decision Tree and Random Forest to produce a comparative analysis on a dataset of Framingham Heart Study which is dedicated to the prediction of risk of Coronary Heart Disease (CHD). We have used a combination of Data Preprocessing and Feature Selection methods, namely The Row Elimination method and Recursive Feature Elimination respectively. To understand the impact of each prevailing features in the dataset on the target feature, we have applied the Chi Squared Technique which is a highly recommended technique when it comes to classification problems. To compare and analyze performance of the algorithms, we applied concepts of the Confusion Matrix, Precision, Recall and F1 Scores; we have plotted ROC curves using Sensitivity and Specificity scores to categorize the algorithms' behavior. We have found out that the highest average accuracy in our study was given by the Logistic Regression algorithm (83.9%) while the other algorithms have come fairly close.

Table of Contents:

List of Figures

List of Tables

List of Equations

Nomenclature

Acronyms / Abbreviations

CHD	Coronary Heart Disease
DT	Decision Tree
FS	Feature Selection
LR	Logistic Regression
ML	Machine Learning
P	Probability
RF	Random Forest
ROC	Receiver Operating Characteristics
SVC	Support Vector Classification/Classifier
SVM	Support Vector Machine

Chapter 1
OVERVIEW

1.1 Introduction

Machine Learning (ML) algorithms have been a part of the scientific field of exploration for a long time now. The usage of ML in healthcare has increased significantly in recent years. Applications of such algorithms have been used in many real life situations to help us in making accurate predictions and performing statistical analysis on levels of diverse dimensions. Among the popular classification based Machine Learning algorithms, Logistic Regression, Decision Tree, Random Forest and Support Vector Machine Classifications are notably the most widely used ones when it comes to predictive problems. In this study, we are going to analyze the efficiencies between these four algorithms in regard to a prediction problem that predicts the risk of an individual having Coronary Heart Disease.

Coronary Heart Disease (CHD) is a type of heart disease where the arteries that supply blood flow to the heart muscles are damaged by plagues that block the blood flow. These plagues are caused by factors like smoking, hypertension, systolic and diastolic blood pressure, diabetes and cholesterol levels, just to name a few. It is most likely to affect in the long run, however, one of the most common ways to detect CHD is unfortunately a sudden heart attack [1].

The importance of this study is to understand how each of these Machine Learning algorithms work, when they are put in a level playing field in terms of dataset and features. It is important to note that in order to have a fair comparison between these

algorithms, the dataset and the data preprocessing methods need to be kept constant to figure out how each of these algorithms perform on a common ground.

Logistic Regression is a classification algorithm that is best known for its efficiency in probabilistic output to separate instances to one of two classes. Simply put, it determines the probability of whether a particular instance is a part of a particular class in the presence of multivariate features with a good correlation between its predicted values [2].

Decision Tree, popularly known for its tree-like breakdown of elements in the dataset to reach a certain conclusion about which particular way should be taken given the values of the features that are highly relevant to the conclusive target [2].

Random Forest, a very interesting classification algorithm that uses clusters of Decision Trees to build its own tree-like structure to determine the belonging class of the target element or feature in the dataset based on majority votes [3].

Support Vector Machine is another classification algorithm that is thoroughly used for disease risk prediction in the field of medical science. Its ability to build a very optimum separating boundary through consistent estimators, make it stand out as a leading research content in the world of Machine Learning algorithms [2].

Further into the paper, we would be able to recognize pros and cons of all the above mentioned algorithms through precision, recall and accuracy scores and also discuss why a certain algorithm behaved the way it did in terms of results.

1.2 Literature Review

Comparison between Machine Learning algorithms is probably never going to get old as a topic of research in computer science. Many a times, we have come across results where one performs better than the others and there is always a justified reason for it; and the very next time, when done a little differently, there comes another algorithm leading the chart with higher precision and accuracy readings. Such a fact is best justified by certain Machine Learning guidelines that is true and applies to all kinds of applications where Machine Learning algorithms are required.

Peter W.F. Wilson et. al. [16] proposed some guideline and identified risk factors of coronary heart disease (CHD). His paper was to examine blood pressure and cholesterol categories with coronary heart disease risk and to incorporate them into coronary prediction algorithms.

Data Preprocessing is one of the most important aspects of Machine Learning. Staying relevant is absolutely vital and the data being used for training and testing must be important to the result so that the output of the prediction is more accurate and meaningful. The data should not also have missing values and incomprehensible feature readings. Such data will result in flawed prediction by any Machine Learning algorithm. Imputing the dataset is an essential part of preparing the data which means that if there are any values missing in the dataset features, those values will be replaced by a suitable method to make sure the dataset does not have any missing values at the end [4].

Heart Diseases affect a large population in today's world, where the lifestyle is moved from active to comfort-oriented. According to the World Health Organization Cardiovascular Diseases (CVD) or Heart Diseases cause more death than any other diseases globally [18]. Peter W.F. Wilson et. al. [16] proposed some guideline and identified risk factors of coronary heart disease (CHD). His paper was to examine blood pressure and cholesterol categories with coronary heart disease risk and to incorporate them into coronary prediction algorithms. The authors, Sabarinathan, V. and Sugumaram in their work, have tried to find out the cause of heart disease by analyzing different parameters (age, gender, chest pain etc.) using decision tree algorithm [17]

Shouman et al. [19] cited the statistically identified risk factors to be age, blood pressure, cholesterol, smoking, total cholesterol, diabetes, hypertension, hereditary, obesity, lack of physical activity. A family ancestry of early coronary illness is a hazard factor for creating coronary illness.

Additionally, genetic markers involved in lipoprotein handling, endothelial integrity, arterial inflammation, and thrombosis formation have been linked to increased risk of CVD in families [20, 21].

Feature Selection has also been looked at very closely by researchers and experts in the field of ML. The main benefit of feature selection is small sample classification problems is to overcome over-fitting problems to improve prediction performance. In general, feature selection methods can be categorized as being of three types: filters, wrappers and embedded methods [11]. Small sample classification is a common problem raised in many practical applications. For example, in automatic target recognition, high dimensional data are obtained from multi/hyperspectral sensors [12-13]; in microarray based cancer classification, there are typically several tens of samples and several thousands to tens of thousands of features [14-15]. It is given a very careful attention because of its impact on getting the right prediction as output. Feature Selection in Machine Learning means to be able to select the most relevant features from the dataset which may contain a huge number of features that are irrelevant to the final output, which is the prediction. The idea is to eliminate the most irrelevant and redundant features from the dataset leaving more space for the relevant features to have an impact on the final outcome which then becomes more concrete and error free [5].

Further studied thoroughly by Guyon et al [5], is a method of Feature Selection called Recursive Feature Elimination which models dependencies among features in the dataset eliminating features with smaller criteria (less impact on final outcome). The Recursive Feature Elimination method is mostly coupled with Support Vector Machine and is found most valued in problems related to gene selection.

As discussed earlier in the introduction, explained by the authors of Journal of Biomedical Informatics, Dreiseitl, Stephan, and Lucila Ohno-Machado [2], Logistic Regression calculates the probability of class membership for one of the two categories in the dataset. A hyperplane is generated based on the probability of all categories belonging to only one class to separate the rest of the instances in the dataset through

this threshold.

Decision Tree classification operates by splitting data into partitions across which the rest of the testing data is classified. A Decision Tree algorithm can handle data in categorical form so for such an algorithm to perform at its best, all continuous data should be converted to categorical data [7].

Closely related to a Decision Tree is the adaptation that is, Random Forest. It is a very popular ensemble algorithm which means that it is a combination of Decision Trees. It is very handy and accurate when dealing with smaller datasets, high dimensional feature sets and complicated data structures coming with an advantage of being able to measure importance of features during training [8].

Support Vector Machine classifies a dataset by using an optimal separating hyperplane that differentiates between the two classes for a dataset. The hyperplane is generated by a subset of training samples which is surrounded by data that lie on either side of the hyperplane. The smaller the margin between the hyperplane and the data, the more difficult it is to classify it. SVM is very effective in today's research work because of its design to support noise-free classification and resistance towards over-fitting [9].

Chapter 2
Data Analysis

2.1 Dataset

The dataset used in this study is from the Framingham Heart Study data repository. Designed to monitor and assess the risk of Coronary Heart Disease (CHD), the dataset has a total of 4241 samples with 16 features. The data is fairly clean with very few missing values which is very good for implementing Machine Learning algorithms, however, the missing data needs to be dealt with properly so that at there are no missing values to interrupt the prediction process.

Sex_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	195	106	70	26.97	80	77	0
0	46	2	0	0	0	0	0	0	250	121	81	28.73	95	76	0
1	48	1	1	20	0	0	0	0	245	127.5	80	25.34	75	70	0
0	61	3	1	30	0	0	1	0	225	150	95	28.58	65	103	1
0	46	3	1	23	0	0	0	0	285	130	84	23.1	85	85	0
0	43	2	0	0	0	0	1	0	228	180	110	30.3	77	99	0
0	63	1	0	0	0	0	0	0	205	138	71	33.11	60	85	1
0	45	2	1	20	0	0	0	0	313	100	71	21.68	79	78	0
1	52	1	0	0	0	0	1	0	260	141.5	89	26.36	76	79	0
1	43	1	1	30	0	0	1	0	225	162	107	23.61	93	88	0
⋮															
0	51	3	1	20	0	0	1	0	251	140	80	25.6	75	NA	0
0	56	1	1	3	0	0	1	0	268	170	102	22.89	57	NA	0
1	58	3	0	0	0	0	1	0	187	141	81	24.96	80	81	0
1	68	1	0	0	0	0	1	0	176	168	97	23.14	60	79	1
1	50	1	1	1	0	0	1	0	313	179	92	25.97	66	86	1
1	51	3	1	43	0	0	0	0	207	126.5	80	19.71	65	68	0
0	48	2	1	20	NA	0	0	0	248	131	72	22	84	86	0
0	44	1	1	15	0	0	0	0	210	126.5	87	19.16	86	NA	0
0	52	2	0	0	0	0	0	0	269	133.5	83	21.47	80	107	0
1	40	3	0	0	0	0	1	0	185	141	98	25.6	67	72	0
0	39	3	1	30	0	0	0	0	196	133	86	20.91	85	80	0

Table 1 Dataset Sample

2.2 Feature Analysis

Among the 16 features in the dataset [Sex_male, age, education, currentSmoker, cigsPerDay, BPMeds, prevalentStroke, prevalentHyp, diabetes, totChol, sysBP, diaBP, BMI, heartRate, glucose, TenYearCHD], there are some features that may not have no to very little relevance to the risk of an individual having Coronary Heart Disease, for

example, education or prevalentStroke which is a factor relevant with the individual's brain and not the heart. While education is inevitable to be disregarded in the matter, we may consider otherwise for the 'prevalentStroke' feature in the dataset and will have to justify its irrelevance in the matter banking on the values and its impact on the feature TenYearCHD which is our target feature of prediction.

The feature Sex_male is a categorical feature (in this study, a categorical feature has only two outlets: 0 and 1) where 0 means female and 1 stands for male.

Age is a key feature for this study because the risk of having CHD increases with age but may not be conclusively stated that a younger person may not be at risk.

Smoking has a direct relationship with the lungs and heart of any individual who has the habit of smoking; however, it may not be the case all the time. Thus, we may bank on the two features 'currentSmoker' which is also a categorical data and 'cigsPerDay' which is the number of cigarettes an individual consumes per day that might generate risk factors to Coronary Heart Diseases.

The feature BPMeds stands for Blood Pressure Medication, which in this study is a categorical data where 0 stands for not on BP medication and 1 stands for on BP medication. It is a control measure in terms of biological prescriptions to keep blood pressure in the body stable. Its subjective nature creates a cloudy assumption of its impact on the risk of CHD, as a result of which it will be addressed and inspected in the feature selection section of the implementation phase.

Hypertension is a very important feature when it comes to Coronary Heart Disease prediction. In this dataset, the feature 'prevalentHyp' is a categorical data which narrows down to 0 for not hypertensive and 1 for hypertensive individual. Hypertension relates to blood pressure and is usually called positive when the blood pressure is high in general for an individual. Looking closely into the dataset features, we also have other blood pressure related features, such as, sysBP and diaBP, both of which are not categorical data. The feature 'sysBP' stands for Systolic Blood Pressure and the feature 'diaBP' stands for Diastolic Blood Pressure.

The BMI (Body Mass Index), biologically, indicates the mass of an individual discarding body fat and bone ratio. It is correlated to an individual's weight but is calculated with factors of age and height taken into consideration. In terms of the risk of CHD, it might or might not have an impact on the final prediction.

The feature ‘heartRate’ is the number of complete heart beats per minute which is very subjective to every individual. In this study, it is not a categorical feature and it might have an impact in the predictive results.

Glucose is a measure of diabetes in biological terms. Blood sugar has higher impact on Coronary Heart Diseases in general which means that there is a possibility of this feature playing a key role in the study.

The feature ‘TenYearCHD’ in the study is the target feature of the study that we will try to predict correctly using the four aforementioned Machine Learning algorithms, which is the goal of our study.

2.3 Data Exploration

In this section, we are going to explore the dataset features thoroughly before going into further processing of Machine Learning techniques. It will be helpful for us to understand the final results of the study as we tie back our conclusion to this section later into the paper.

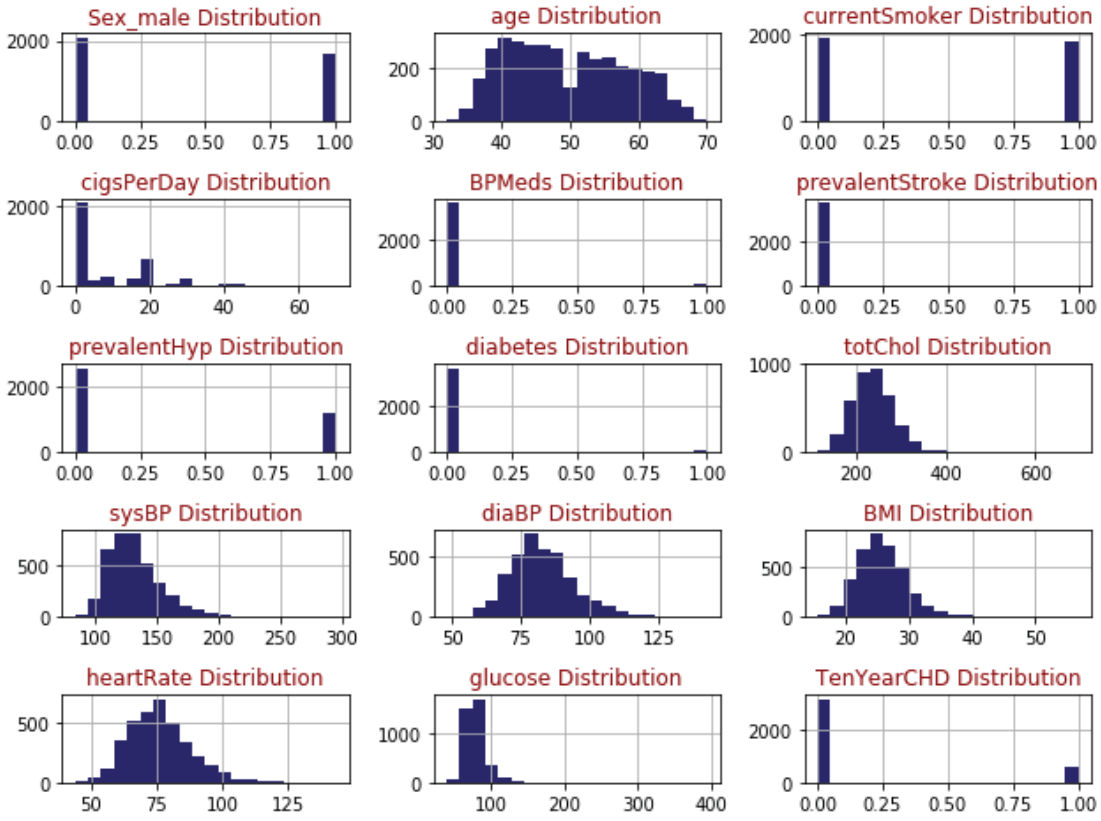


Figure 2.1 Feature Distribution Histograms

Figure 2.1, above, shows the distribution histograms of the features in the dataset. We can see from this which the categorical features are and which are not categorical.

Gender distribution seems to be well balanced as is the age distribution. Smokers are also evenly balanced in the distribution while the off balanced features are 'BPMeds', 'prevalentStroke' and 'diabetes' features with very low variance in their distributions. The features 'cigsPerDay', 'totChol' and 'prevalentHyp' are moderately balanced.

One key factor to notice from the graph is that our target feature 'TenYearCHD' is also quite off balanced in distribution with majority of the individuals with no risk of Coronary Heart Disease according to the dataset.

Almost 85% of the total number of samples in our target feature is not at risk of CHD. However, this might still test the efficiencies of our algorithms in predicting correctly, the number of individuals who are not at risk in more percentage than others.

Analyzing relevance within features

As we go through the features of the dataset, we can see that there are some features that might overlap each other in terms of their realistic impacts on the human body.

We have intuitively chosen some features from the dataset that might be really relevant with each other. We would do a pairplot analysis to understand relevance between these features. A pairplot shows us how a certain feature behaves on its own in the domain, as well as how it behaves with the other features (in this case, its pair) in the dataset.

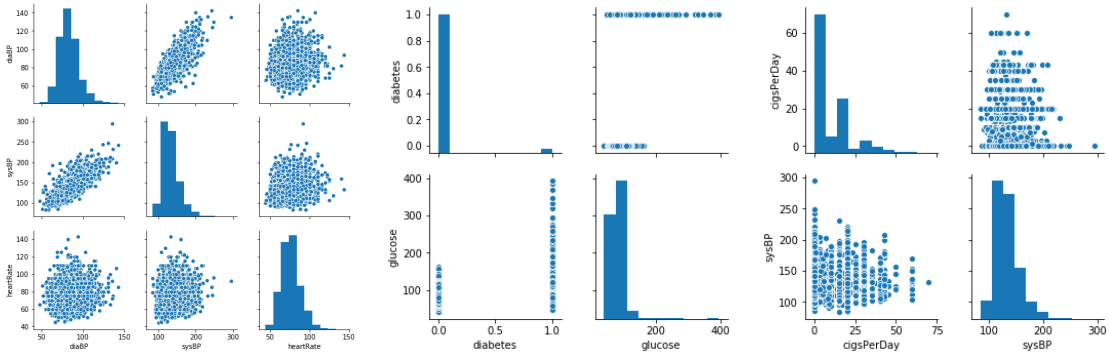


Figure 2.2 Feature Pairplot

In Figure 2.2(a), we have placed the three features ‘diaBP’, ‘sysBP’ and ‘heartRate’ together in a pairplot to see how they interact with each other as they are very similar with each other biologically.

We can see that these features are very dense at the center of their plots which means that the values they have are more or less of the same range for all the individuals in the dataset. This can lead us to a thought that may be these values are redundant for the study which will compromise the prediction of the Machine Learning algorithms. We want to get rid of data redundancies in order to have better predictive results.

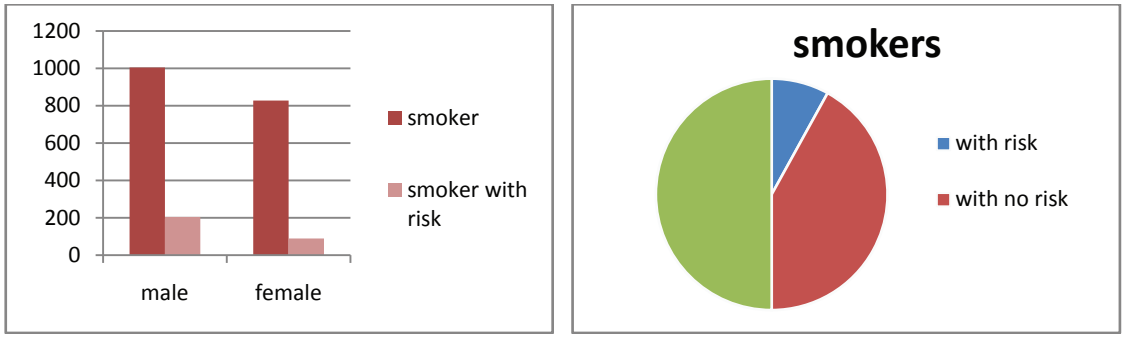
In Figure 2.2(b), the two features ‘diabetes’ and ‘glucose’ are placed for analysis. The ‘diabetes’ feature is a categorical feature, while the ‘glucose’ feature is a continuous feature.

The reason to put these two features together is to understand their relationship with each other in the dataset. Glucose is a measure for diabetes. From the graph, we can clearly see that as the number of individuals having higher glucose levels is also showing positive values for diabetes; this renders our understanding clearer and makes the dataset more believable and true.

In Figure 2.2(c), we try to understand the relationship between ‘sysBP’ and ‘cigsPerDay’ features.

Our concern with smoking here is its impact in blocking the heart arteries and ventricles with harmful chemicals. And Systolic Blood Pressure directly relates to the heart’s blood pumping abilities. That is why we thought that there might be a relationship with heavy smokers and their systolic blood pressure readings.

We can see that the reading of Systolic Blood Pressure is more than 120 for about 60% of the individuals smoking more than 10 cigarettes per day. These are features seem to have a very strong impact on the risk of CHD.



(a) (b)

Figure 2.3 Impact of smoking on CHD

The Figure 2.3(a) above, shows how smoking and gender combined relate to risks of CHD. We see that the number of male smokers at risk is more than the number of female smokers at risk of CHD. Smoking however does not have much of an impact on the risk of Coronary Heart Disease as there are a huge number of smokers, both male and female, who do not find themselves at risk of future CHD. Figure 2.3(b) goes on to show that a huge number of smokers are risk free (83.95%).

2.4 Data Preprocessing

In order to have an effective and efficient prediction by any Machine Learning algorithm, we need to ensure that there is no missing data in the dataset. Missing values interrupt with the final outcome and makes it weaker and inconclusive. There are a few ways in which missing data may be replaced effectively. One of them is data imputation [4].

When the amount of missing values is more in volume in comparison to the sample dataset, it is wise to impute the data using processes like, mean, median and mode so that it can help in filling the void for the Machine Learning algorithms to be trained properly.

One other way to handle missing values when they are smaller in volume in comparison to the total sample is to eliminate rows that contain such null values.

Data Preprocessing in this study has been explored by the Pandas library of Python. It is a very handy library to analyze a dataset and its missing values.

Dealing with Missing Values: Row Elimination

First we would have looked into the dataset and searched in all the features, the total number of missing values in the dataset. We need to know it so that we can have an idea of the percentage of missing values in the dataset.

If the percentage of the missing values is more than 20%, removing the row data from the dataset will result in a less efficient prediction. However, if the percentage is less than 20%, it is safe to eliminate the rows to clear all the missing values or null values from the dataset.

We found that there are a total of 489 rows with missing values in the dataset. The total number of rows in the dataset is 4241. The number of rows with missing values is about 12% of the total dataset.

This makes our choice much easier and now we can eliminate all the rows with missing values with the Pandas function ‘dropna ()’ to make the dataset cleaner.

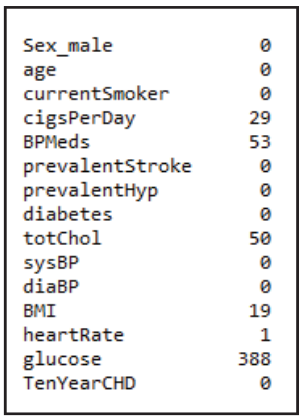


Figure 2.4 Null value look up

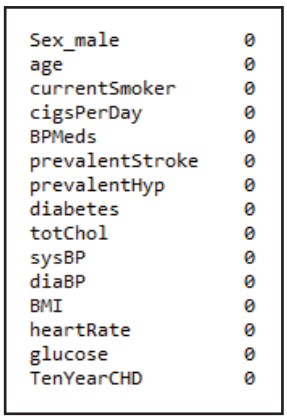


Figure 2.5 No null values

2.5 Feature Selection

Now, after elimination of the missing valued rows, we have a clean data that has no missing values for any features in the dataset. We are now set to move on to the next important phase of the Machine Learning cycle, that is Feature Selection.

2.5.1 The Chi Squared Technique

As we have observed in the data so far, we understand that there are a few features that might be redundant in nature, there might also be some features that are not important to the final outcome from observation. To find out which features are statistically relevant to our target feature, we apply the Chi Squared statistical model.

With the Chi Squared model, we try to find out how much of the feature has occurred by chance. This means that the higher the probability, the higher the chances of it occurring by fact and not by chance. It calculates a Chi Squared value which is then compared to the critical value of Chi Square which in medical studies is considered to be at the 5% range. So in our calculations we will also follow this standard.

First we add a constant to make all continuous features into categorical features. This will ensure the consistency of values being processed in the Chi Squared model.

Our goal from this is to find out how many of the features have an actual impact on the target feature; in other words, what are the features to which our target feature is most sensitive to.

The Null Hypothesis

To determine this we have to first make a null hypothesis. The null hypothesis in this case is that the impact of all the features on the target feature is equal. However, observations and expectations are two different things that play an important role in proving the trueness of the hypothesis.

The Degrees of Freedom

The Degrees of Freedom is all possible outcomes of all the features minus 1. We need this value to find out our critical value which we don’t want to exceed or, if any feature does exceed the critical value, it means that it should be eliminated from the list because it adds impurity to the prediction.

The Accuracy

To be sure that the features to predict from are indeed the best ones, we choose the 0.05 threshold for the Chi Squared table which means that we will be 95% sure that the features are naturally involved and there are no external influences that tamper their impact.

$$X^2 = \sum \frac{(O - E)^2}{E} \qquad (1)$$

Here, X is the Chi Squared Value, O stands for Observed Values, E stands for Expected Values

Equation 1 The Chi Squared Equation

With the above Equation 1 the Chi Squared value is calculated which is then compared to the critical value from the Chi Squared table. We tabulate the Chi Squared values for all the features through the summary () function of scipy, a statistical tool of python and for the features that exceed the probability of 5% are going to be excluded from the list of features to get better results of our prediction.

	coef	std err	z	P>|z|	[0.025	0.975]
const	-8.6532	0.687	-12.589	0.000	-10.000	-7.306
Sex_male	0.5742	0.107	5.345	0.000	0.364	0.785
age	0.0641	0.007	9.799	0.000	0.051	0.077
currentSmoker	0.0739	0.155	0.478	0.633	-0.229	0.377
cigsPerDay	0.0184	0.006	3.000	0.003	0.006	0.030
BPMeds	0.1448	0.232	0.623	0.533	-0.310	0.600
prevalentStroke	0.7193	0.489	1.471	0.141	-0.239	1.678
prevalentHyp	0.2142	0.136	1.571	0.116	-0.053	0.481
diabetes	0.0022	0.312	0.007	0.994	-0.610	0.614
totChol	0.0023	0.001	2.081	0.037	0.000	0.004
sysBP	0.0154	0.004	4.082	0.000	0.008	0.023
diaBP	-0.0040	0.006	-0.623	0.533	-0.016	0.009
BMI	0.0103	0.013	0.827	0.408	-0.014	0.035
heartRate	-0.0023	0.004	-0.549	0.583	-0.010	0.006
glucose	0.0076	0.002	3.409	0.001	0.003	0.012

Figure 2.6 Results of Chi Square method

From the Figure 2.6 above, we can see that among all the features, there are only some which did not cross the threshold of 0.05. We want to keep such features so that the prediction will be more accurate.

2.5.2 Recursive Feature Elimination

We use this feature selection technique to eliminate one by one, the features that exceed the 0.05 probability threshold.

First we take in as input the data frame, feature names and the target feature. We then create a model and check which feature crosses the 5% threshold. We eliminate one feature with the highest p value and recursively run the model with the remaining features.

We continue this process until we reach a point where no features in the dataset have a p value larger than the 5% threshold which is our goal.

From hind sight, we can make an educated guess of which features are going to make it for the testing phase. For example, the 'currentSmoker' feature in Figure 2.6 is not fit for the model and might be eliminated in the process along with 'diabetes' and 'BMI'.

	coef	std err	z	P>|z|	[0.025	0.975]
const	-9.1264	0.468	-19.504	0.000	-10.043	-8.209
Sex_male	0.5815	0.105	5.524	0.000	0.375	0.788
age	0.0655	0.006	10.343	0.000	0.053	0.078
cigsPerDay	0.0197	0.004	4.805	0.000	0.012	0.028
totChol	0.0023	0.001	2.106	0.035	0.000	0.004
sysBP	0.0174	0.002	8.162	0.000	0.013	0.022
glucose	0.0076	0.002	4.574	0.000	0.004	0.011

Figure 2.7 Results of Recursive Feature Elimination

The above Figure 2.7 shows the results after the recursive feature elimination method. We can see that the features with less that 5% probability value only survived the screening and now, we have narrowed down the feature set to only 6 features.

This promises to enrich the prediction of the Machine Learning algorithms and bring out efficient and meaningful results, which has always been our goal from the beginning.

2.6 Further Data Exploration

Now that we know which features to consider, we are ready to train the four Machine Learning models with these features. But before going into that, we can further explore the features to see more closely why these features may have been selected by the Feature Selection method.

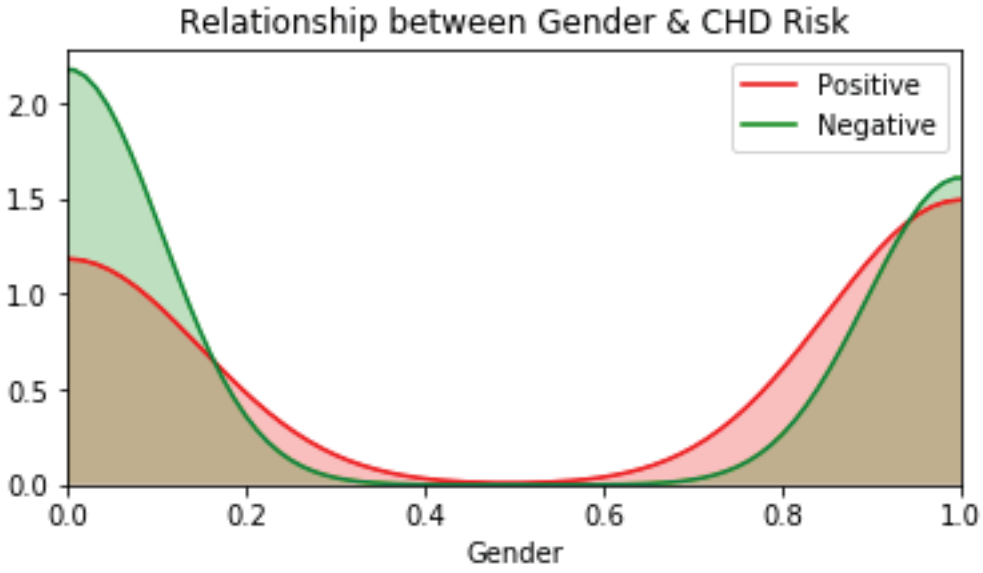


Figure 2.8 Relationship between Gender & CHD Risk

We can see from the above Figure 2.8 that the possibility of men being at risk of CHD is more in ratio than women. This is, according to specialists, a good assumption based on practice and experience.

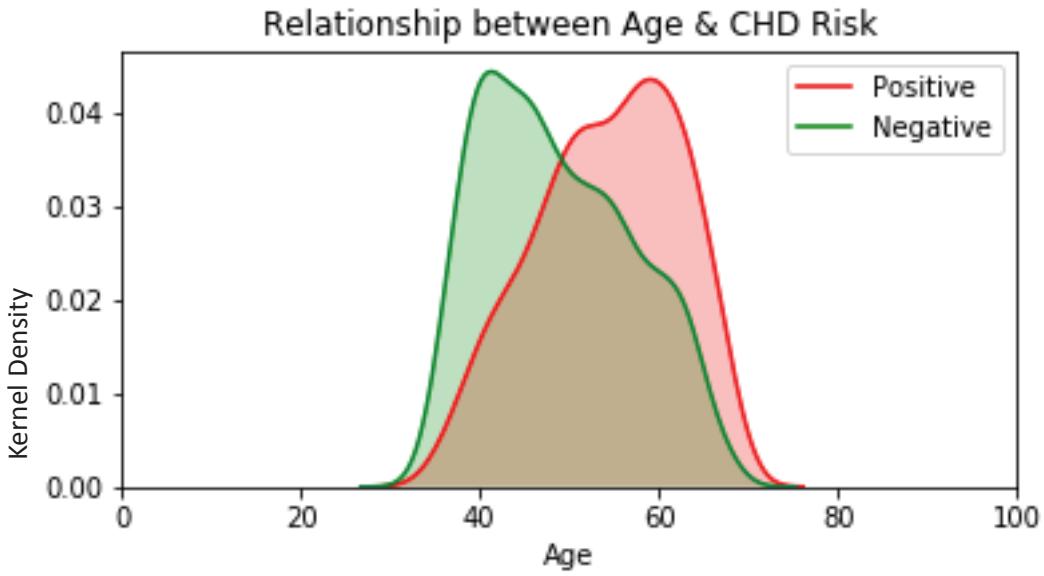


Figure 2.9 Relationship between Age & CHD Risk

As we can see from the above relationship graph of the 'age' feature against our target, we can see that as the individual grows older, the risk of CHD increases.

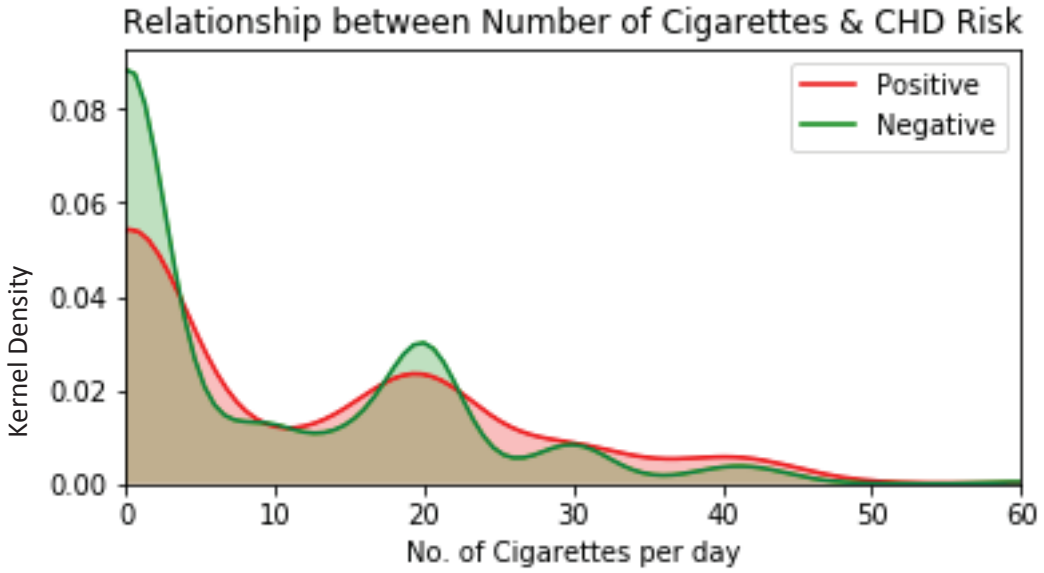


Figure 2.10 Relationship between Number of Cigarettes & CHD Risk

The possibilities of having CHD is very evident in smoking individuals but it is not the most conclusive factor as we can see that even non smoking individuals may be at risk of CHD. However, it is seen that with the increase in number of cigarettes per day, the chances of getting CHD increase.

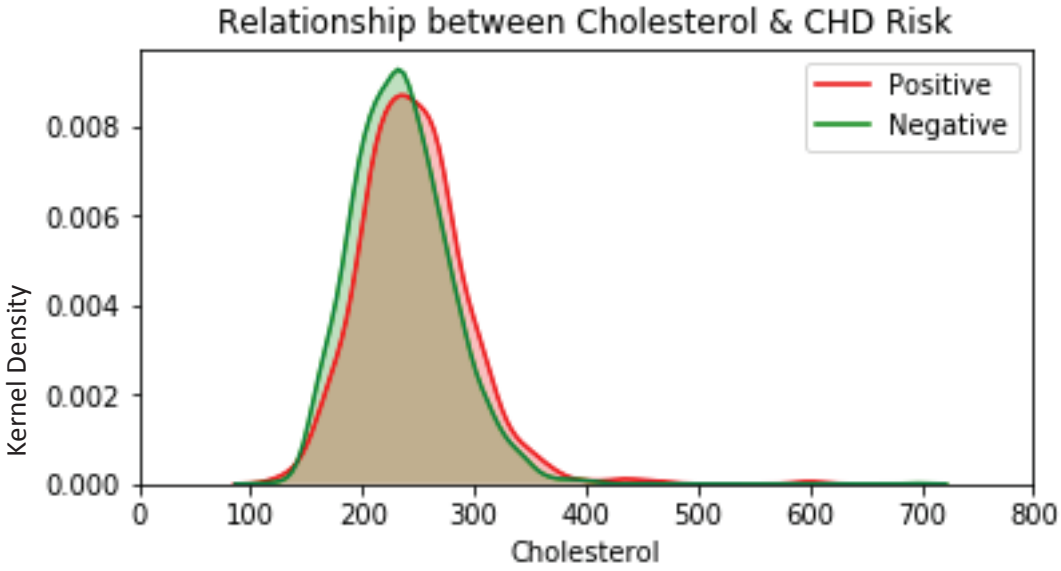


Figure 2.11 Relationship between Cholesterol & CHD Risk

Among the selected features, the 'totChol' feature relates to the risk of CHD like the above graph. We see that the distribution almost even out at the 250 mark but as it increases from that point, the possibility of getting CHD gets higher.

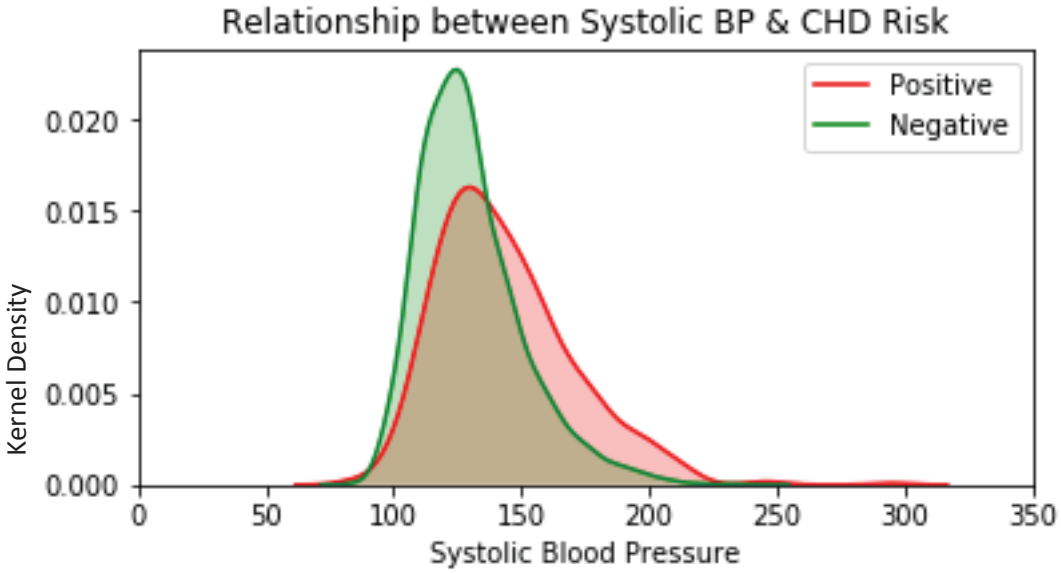


Figure 2.12 Relationship between Systolic Blood Pressure and CHD Risk

The normal reading of Systolic Blood Pressure is generally between the ranges of 90 to 110. From the graph we see that with the increase in the systolic blood pressure, the risk of catching CHD increases. There is almost no individual who has a systolic blood pressure beyond the 200 mark and is not at risk of CHD.

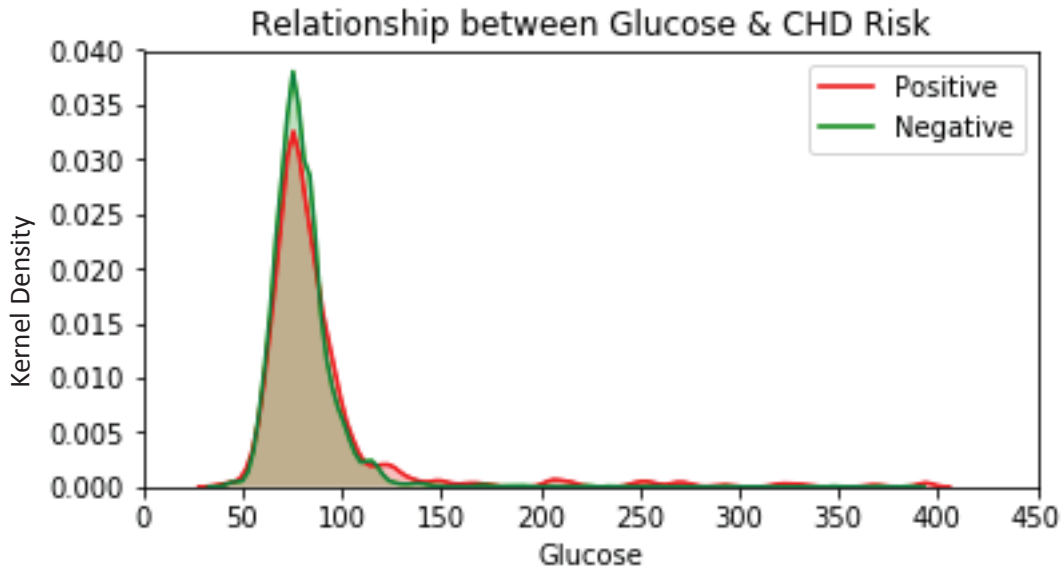


Figure 2.13 Relationship between Glucose & CHD Risk

Glucose relates to CHD in an expected way as it shows in the graph. The expectation was that it should play a very key role in causing a CHD. According to the reading from the dataset, the relationship between glucose and CHD risk is going fairly hand to hand.

2.7 Preparing to Train the Models

Now it is time to train the models with the selected features. The whole dataset is split into two parts, randomly extracting samples to form a training data and testing data based on test size percentage. It is worthy to notice that each sample dataset is randomized across the whole dataset and not biased towards anything.

Some key factors to be kept in mind in this phase is that no features should have any missing values and there should be substantial samples so that training selections may have the opportunity to vary from each other every time.

It is also important to make sure that an average of the final reading is taken into consideration for each algorithm. For this, we are going to use a K-fold cross validation process where we are going to change the test size from a range of 10% to 90% of the dataset which will consequently vary the training size from 90% to 10% and the model will be well versed in the training process and help in increasing the efficiencies to achieve high accuracy in predictions, which is our final goal.

The SciKit-Learn library has a very smart method to split the dataset into training and testing parts. We can assign the test size when using the method.

We start off by training the method with a test size of 10%. This means that 90% of the whole dataset will be used to train the algorithm and 10% will be used to test the performance of the Machine Learning algorithm.

We gradually increase the test size along the way which means that simultaneously the training data size will decrease. The test size is increased by 10% every time.

We keep on doing so until we reach a test size of 90% implying that with the learning from only 10% of the dataset model, the algorithm will be tested in the last phase.

It is important to note that all the splits are chosen randomly to make sure that there is the highest chance of producing a unique list for every split.

Chapter 3
Implementation

3.1 Programming Platform

SciKit-Learn Python Library

The SciKit-Learn Python library has been used to implement the Machine Learning algorithms in this study. It is one of the leading libraries when it comes to performing operations or applications of Machine Learning algorithms.

It has very well written Machine Learning algorithms ready for implementation with input parameters to be set by us; which is the task of assigning the training and testing sets of data properly in the proper sections for every algorithms.

Each Machine Learning algorithm is trained with a data split before it is ready to be tested on a new set of split from the dataset. It is also known as fitting the data into the model.

For each Machine Learning algorithm an accuracy reading is taken and stored in a variable which is then added to the sum variable for future mean that will establish the trueness of the final prediction. Every time the test size changes in the training phase, there is a new set of training data to be trained from and consequently a new set of test data to be tested with. Sequentially, after every complete prediction, the accuracy reading is stored and added to the sum variable. It is trained with a data split before it is ready to be tested on a new set of split from the dataset. It is also known as fitting the data into the model.

With every split, the model is trained and tested, a classification report is generated that takes the predictive accuracy and the test split.

3.2 Implementing the Algorithms

Logistic Regression Classification

The Logistic Regression classifier takes as input, the training set of data before moving into the test phase. The rest of the dataset is then fed into the predictive function of LR which iterates through all the data in the testing set and predicts the target feature result from what it has learned and understood from the data of the training set.

First, we import the Logistic Regression Classifier from the SciKit-Learn library. Then we create a model that is based on the Logistic Regression method to train and test the data set.

Once the LR model is ready to receive the data, we feed into it the training dataset after the first split. The training dataset consists of 2 parts, the x-train and the y-train. These are fed into the logistic regression method as parameters.

Once the training is complete, we create a predictive variable. This variable is an array of predictions.

The LR model is then fed with the test dataset, which also has 2 parts, the x-test and the y-test. With the help of the predict () method from sklearn library, we store all the predictions from the logistic regression part into the predictive variable.

A score function is then called on the test dataset to check the accuracy of the model.

We take 8 scores (ranged from 10% to 90% of test sizes) and make an average of the scores for final evaluation results.

Support Vector Machine Classification

The Support Vector Classifier also takes as input, the training set of data before moving into the test phase. The rest of the dataset is then fed into the predictive function of SVM which iterates through all the data in the testing set and predicts the target feature result from what it has learned and understood from the data of the training set.

Firstly, we import the Support Vector Machine class from the sklearn library before we create a model that applies the Support Vector Classification method.

After the SVC model is ready, it is time to pass to it the training dataset. To train the SVC model, we need to send as parameters, the training rows and the training columns selected at random by the train test split module of python.

Soon after the SVC model is trained, we test the performance of the model and keep the test results in a prediction variable. This variable will be used to check how much of it matches the actual values that it predicted.

In the score function, we compare the prediction variable against the true values to find how many of it was correct. We also keep the accuracy scores in a score variable which we use later on to extract the mean of the accuracy for the SVC model.

Decision Tree Classification

The Decision Tree Classifier take as input, a split point criteria called Gini which is recommended when splitting categorical data. It also requires the assignment of the depth of the tree and the number of minimum leaf samples for the tree.

At first, we import the Decision Tree Classifier from the tree class of sklearn library. We create a decision tree model that will be trained and later tested on the train and test splits we have performed earlier.

The Decision Tree algorithm splits the dataset into branches based on the maximum depth allowed and on the total number of data samples. The mode of the leaf nodes are always taken as the predicted value for categorical data.

The training dataset is fed into the decision tree model and after the process is over, we create a prediction variable, and save all the predicted results based on the testing rows that are given to the model after training.

A decision tree prediction variable is created that is used to store the score of all the predictions and then the average of the score value is taken as the final accuracy score for Decision Tree's performance.

Random Forest Classification

The Random Forest classifier uses the ensemble function of Sci-Kit Learn library to accumulate and generate decision trees creating a barrage of leaves and split points. Training and testing data are subsequently fitted into the model's prediction function to generate prediction of the target feature.

After importing the Random Forest Classifier from the sklearn library, we are ready with the model for the RFC.

Fitting training data is easy as we use the fit () function of the classifier to train the model with the training dataset. The parameters for the fit () function are x_train and y_train, which are the rows and columns of the dataset respectively.

After the model is tested, we use the testing rows as input data to for the model to let it predict the results. In the process, we store all the results in a variable so that it can be used later on to find accuracy of the model.

The score () function is used to test the accuracy of the model which is kept in a accuracy variable that adds up all the accuracy scores for an average of prediction accuracy score.

3.3 Results

After each Machine Learning algorithm has been run k times (where k is 8) to make a comprehensive conclusion about the accuracy results in the study, the confusion matrix are generated at each step to understand individual predictive performances [10].

3.3.1 The Confusion Matrix

The Confusion matrix is a matrix that describes the performance of a Machine Learning algorithm with parameters like True Positive, True Negative, False Positive and False Negative.

- True Positives stand for the number of accurately predicted true values (actually 1, predicted 1).
- True Negatives stand for the number of accurately predicted false values (actually 0, predicted 0).
- False Positives stand for the number of wrongly predicted true values (actually 0, predicted 1).
- False Negatives stand for the number of wrongly predicted false values (actually 1, predicted 0).

Ideally, the algorithm with the highest True Positives and True Negatives is the most suitable algorithm for a particular predictive solution. Let us look at the confusion matrices for the four Machine Learning algorithms of our study.

a) Logistic Regression Confusion Matrix

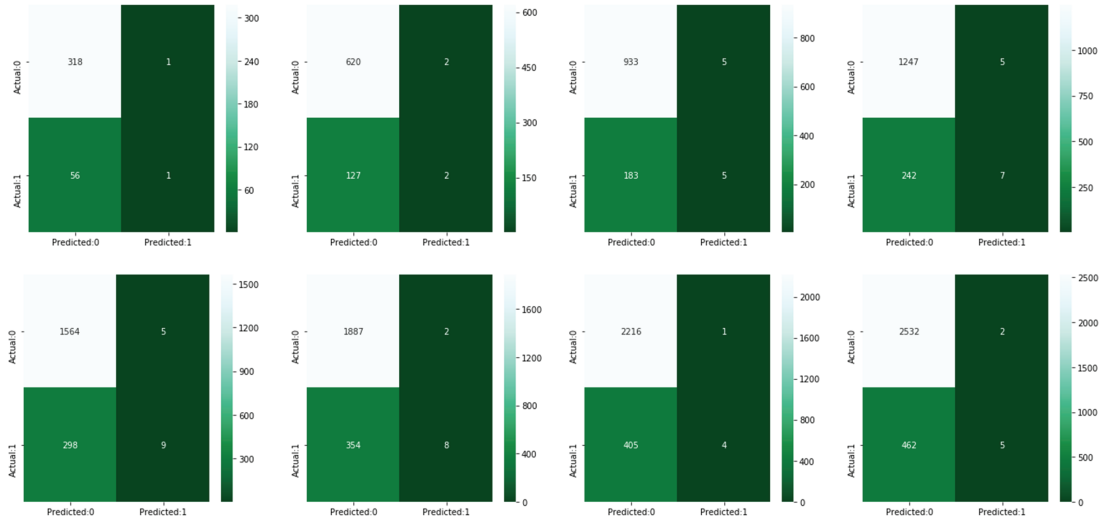


Figure 3.1 Logistic Regression Confusion Matrix

Figure 3.1 represents the confusion matrix of the LR algorithm. We observe a very high percentage of correct True Negative values, which is very important for our study because it our dataset has a lot of True Negative values.

b) Support Vector Machine Confusion Matrix

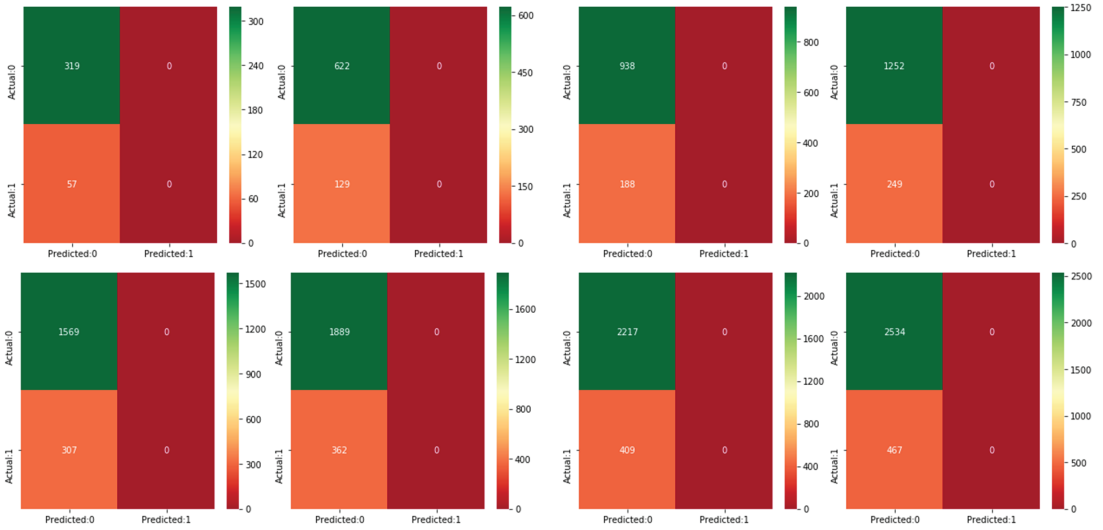


Figure 3.2 Support Vector Machine Confusion Matrix

From the above Figure 3.2, we can see that the SVM has failed to predict almost all of the True Positive values. This might be an alarming if there were a huge number of True Positives in the whole dataset.

c) Decision Tree Confusion Matrix

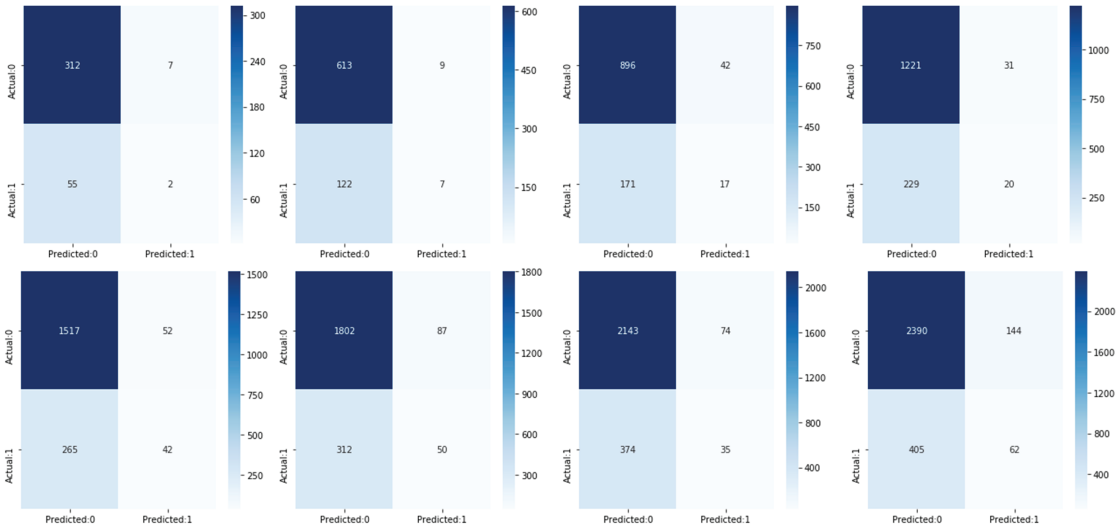


Figure 3.3 Decision Tree Confusion Matrix

Figure 3.3 shows the confusion matrix of the Decision Tree algorithm. From its readings, we can understand that in comparison to all the other algorithms used in this study, it has correctly predicted the most number of True Positive values.

d) Random Forest Confusion Matrix

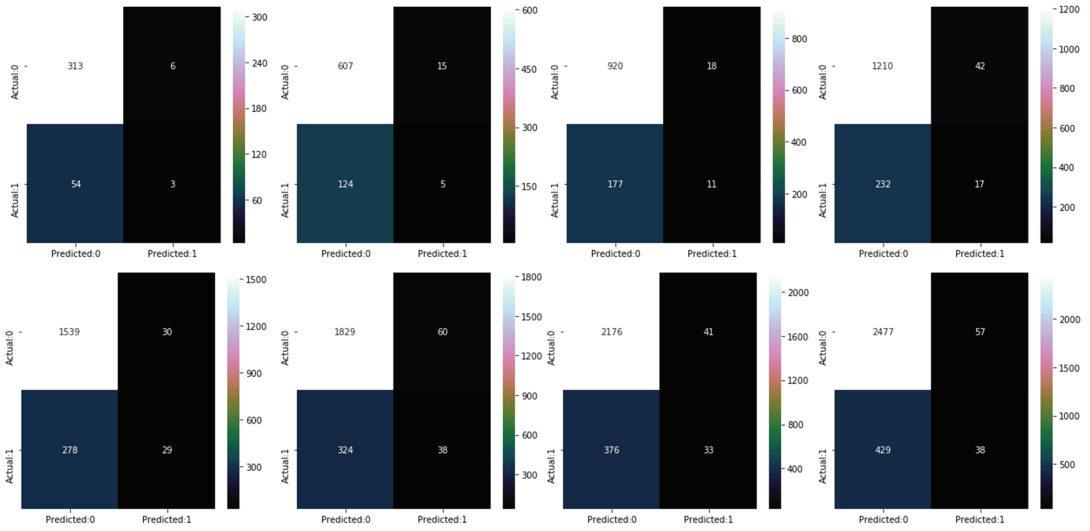


Figure 3.4 Random Forest Confusion Matrix

In Figure 3.4 the confusion matrix of RF shows a high percentage of True Positive value prediction, when noticed closely. It has done a very good performance in predicting True Negative values as well.

3.3.2 Observation and Performance Study

An interesting observation from the confusion matrix is that all the algorithms have predicted True Negatives with the highest percentage while the second highest percentage goes to the prediction of False Positives.

Logistic Regression acc: 83.95
Support Vector Machine acc: 83.84
Decision Tree acc: 82.48
Random Forest acc: 83.05

Figure 3.5 Accuracy Scores of the Algorithms

The average accuracy score for each of the Machine Learning algorithm is given in the Figure 3.5 above. We can see that all the algorithms have performed quite well under the given circumstances after being tested on 8 different sets of data samples.

The score function of Sci-Kit Learn computes the number of accurately predicted samples. It ranges from 0-1 where 1 is the ideal value giving us 100% accuracy in predicting the target.

Precision, Recall and F1 Scores

	precision	recall	f1-score	support
0	0.83	1.00	0.91	622
1	0.50	0.02	0.03	129
micro avg	0.83	0.83	0.83	751
macro avg	0.66	0.51	0.47	751
weighted avg	0.77	0.83	0.76	751

	precision	recall	f1-score	support
0	0.83	1.00	0.91	622
1	0.00	0.00	0.00	129
micro avg	0.83	0.83	0.83	751
macro avg	0.41	0.50	0.45	751
weighted avg	0.69	0.83	0.75	751

	precision	recall	f1-score	support
0	0.83	0.99	0.90	622
1	0.44	0.05	0.10	129
micro avg	0.83	0.83	0.83	751
macro avg	0.64	0.52	0.50	751
weighted avg	0.77	0.83	0.76	751

	precision	recall	f1-score	support
0	0.83	0.98	0.90	622
1	0.25	0.04	0.07	129
micro avg	0.81	0.81	0.81	751
macro avg	0.54	0.51	0.48	751

Figure 3.6 Classification Report Sample of the Algorithms

The above Figure 3.6 is a classification report that was generated after every test on the algorithms. This means that there are eight classification reports generated for each of the algorithms. The classification report summarizes the performance of the algorithms presenting their Precision, Recall and F1-Scores. The above figure is the classification report of the four algorithms with the test size of 20%.

When we notice the F1-Score of the algorithms, we can see that it is quite high. The F1-Score is a combination of the Precision and Recall readings of an algorithm. This means that there is a huge number of Actual Negatives in the dataset. We know this is true from figure 1 where we see the distribution of the features of the dataset.

The results for each algorithm seem to be quite fair, but Logistic Regression and Support Vector Machines have the slightest of edges over Decision Tree and Random Forest Classifiers.

The ROC Curves

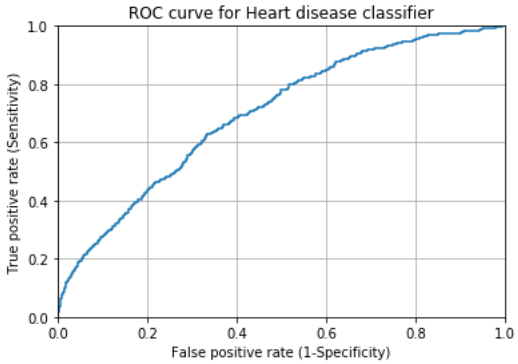


Figure 3.7 ROC Curve for LR

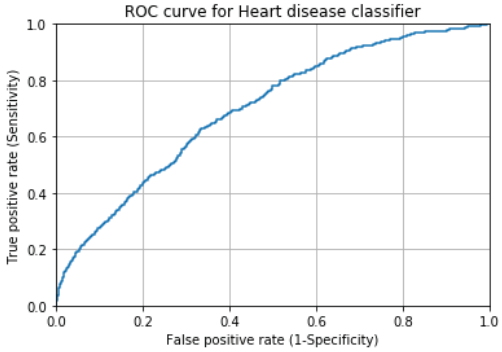


Figure 3.8 ROC Curve for SVC

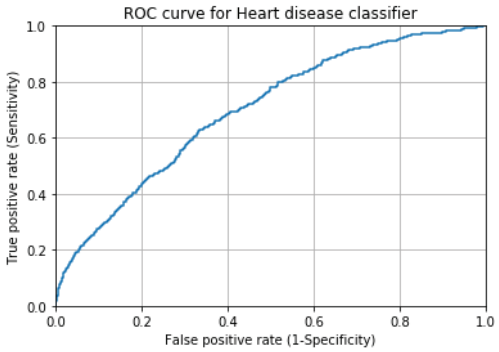


Figure 3.9 ROC Curve for DT

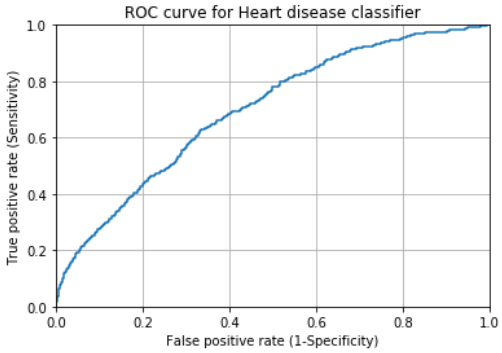


Figure 3.10 ROC Curve for RF

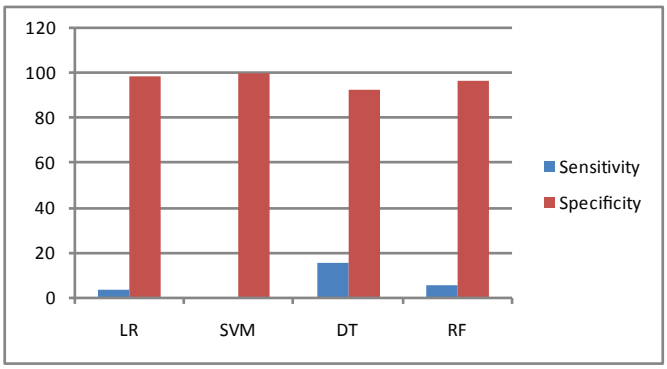


Figure 3.11 Bar Graph for Sensitivity vs Specificity for all algorithms

It is noticed that the all the algorithms show high specificity as the dataset has a high range of True Negatives. This is not a bad result in terms of accuracy because the results depict the accurate prediction of True Negatives. Had there been a more balanced ratio of True Positives and True Negatives in the dataset, the algorithms could show a higher percentage of sensitivity.

Chapter 4
Conclusion

In conclusion, we would like to draw attention towards the results and the method of preprocessing data before implementing the Machine Learning algorithms. It would be worth noticing that we have used a single technique of Feature Selection and we have used the most basic data imputation and completion techniques to get to the decent results.

It is critical to know that there are many ways that a dataset can be prepared for prediction and it depends on the type of prediction we are looking for in our study. A good, clean and well preprocessed dataset can generate high accuracy for Machine Learning algorithms and especially in the field of predicting medical components, high accuracy is a must.

We have used the Row Elimination method to clean the null valued rows and now imputed any data as it was convenient in this case. In any other case where the number of missing rows exceed recommended threshold, we have to use the imputation method to preserve every data sample possible to make a more efficient environment for machine learning algorithms to function.

We have used the Chi Squared Technique and Recursive Feature Elimination, to find out the more suitable features among all the features, and eliminate redundant or less effective features to ensure that the best ones are used to train the ML models thus, be able to produce good results in prediction.

Had there been more Actual Positives in the dataset, the results of the algorithms could have been further enhanced.

Result Summary

- The Decision Tree algorithm could predict the highest number of True Positives
- The Support Vector algorithm could predict the highest number of True Negatives but predicted the least number of True Positives
- The Logistic Regression algorithm has the highest average of accuracy predictions
- The Random Forest algorithm performed with a better average than the Decision Tree algorithm
- Sensitivity of an algorithm depends on the amount correct True Positive Predictions
- Specificity of an algorithm depends on the amount correct True Negative Predictions
- The algorithms had an average of 69.7% ROC Area Under Curve score which inclines them all towards behaving more Specifically than Sensitively

Future Work

We have tried to produce a comprehensive study on four of the top Machine Learning algorithms today that deal with categorical data. We have come across fair results in terms of accuracy and F1-Score but with a more balanced set of data, we would probably end up having a higher accuracy scores for these algorithms.

What we have tried to show is not the only way to perform a comparative analysis between Machine Learning algorithms and we believe that there is a lot of room for improvement and analysis between these algorithms and their performances on a level playing ground.

References

[1] Torpy, Janet M., Alison E. Burke, and Richard M. Glass. "*Coronary heart disease risk factors*." Jama 302.21 (2009): 2388-2388.

[2] Dreiseitl, Stephan, and Lucila Ohno-Machado. "*Logistic regression and artificial neural network classification models: a methodology review.*" Journal of biomedical informatics 35.5-6 (2002): 352-359.

[3] Liaw, Andy, and Matthew Wiener. "*Classification and regression by randomForest.*" R news 2.3 (2002): 18-22

[4] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "*Supervised machine learning: A review of classification techniques*." Emerging artificial intelligence applications in computer engineering 160 (2007): 3-24.

[5] Granitto, Pablo M., et al. "*Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products*." Chemometrics and Intelligent Laboratory Systems 83.2 (2006): 83-90.

[6] Soleimani, Paria, and Arezoo Neshati. "*Applying the Regression Technique for Prediction of the Acute Heart Attack*." World Academy of Science, Engineering and Technology, International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering 9.11 (2015): 767-771.

[7] Srinivas, K., B. Kavihta Rani, and A. Govrdhan. "*Applications of data mining techniques in healthcare and prediction of heart attacks*." International Journal on Computer Science and Engineering (IJCSE) 2.02 (2010): 250-255.

[8] Qi, Yanjun. "*Random forest for bioinformatics*." Ensemble machine learning. Springer, Boston, MA, 2012. 307-323.

[9] Veropoulos, Konstantinos, Colin Campbell, and Nello Cristianini. "*Controlling the sensitivity of support vector machines*." Proceedings of the international joint conference on AI. Vol. 55. 1999.

[10] Story, Michael, and Russell G. Congalton. "*Accuracy assessment: a user's perspective*." Photogrammetric Engineering and remote sensing 52.3 (1986): 397-399.

[11] Weston, Jason, et al. "Use of the zero-norm with linear models and kernel methods." Journal of machine learning research3.Mar (2003): 1439-1461.

[12] Kuo, Bor-Chen, and David A. Landgrebe. "A covariance estimator for small sample size classification problems and its application to feature extraction." IEEE Transactions on Geoscience and Remote Sensing 40.4 (2002): 814-819.

[13] Casasent, David, and Xue-Wen Chen. "Feature reduction and morphological processing for hyperspectral image data." Applied Optics 43.2 (2004): 227-236.

[14] Xiong, Huilin, and Xue-wen Chen. "Kernel-based distance metric learning for microarray data classification." BMC bioinformatics 7.1 (2006): 299.

[15] Chen, Xue-Wen. "Margin-based wrapper methods for gene identification using microarray." Neurocomputing 69.16-18 (2006): 2236-2243.

[16] P. Wilson, R. D'Agostino, D. Levy, A. Belanger, H. Silbershatz and W. Kannel,"Prediction of Coronary Heart Disease Using Risk Factor Categories", Circulation, vol. 97, no. 18, pp. 1837-1847, 1998.

[17] Sabarinathan, V. and Sugumaram, V. (2014)."Diagnosis of Heart Disease Using Decision Tree". International Journal of Research in Computer Applications and Information Technology, 2(1).

[18] WHO, "Cardiovascular diseases (CVDs)," Published by WHO, 2013. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs317/en/. [Accessed: 30-May-2017].

[19] Mai Shouman, Tim Turner, Rob Stocker 2012 "Using Data Mining Techniques In Heart Disease Diagnoses And Treatment" Electronics, Communications and Computers (JECECC), 2012 Japan-Egypt Conference March 2012, pp 73-177.

[20] Topol EJ Smith J, Plow EF, Wang QK. Genetic Susceptibility to Myocardial Infarction And Coronary Artery Disease. Human Molecular Genetics. 2006;15(suppl 2): R117–R123.

[21] Scheuner MT. Genetic Evaluation for Coronary Artery Disease.Genetics in Medicine. 2003;5(4): 269–285. doi: 10.1097/01.GIM.0000079364.98247.26`