Heart Disease Analysis and Prediction

Abstract: According to recent survey by WHO (World health organization) 17.9 million people die each year because of heart related diseases and it is increasing rapidly. With the increasing population and disease, it is become a challenge to diagnosing disease and providing the appropriate treatment at the right time. But there is a light of hope that recent advances in technology have accelerated the public health sector by developing advanced functional biomedical solutions. This paper aims at analysing the various Machine Learning techniques namely Naive Bayes, Random Forest Classification, Decision tree, KNN classification Logistic Regression and Support Vector Machine by using a qualified dataset for Heart disease prediction which is consist of various attributes like gender, age, chest pain type, blood pressure, blood sugar etc. The research includes finding the correlations between the various attributes of the dataset by utilizing the standard data mining techniques and hence using the attributes suitably to predict the chances of a heart disease. These machine learning techniques take less time for the prediction of the disease with more accuracy which will reduce the dispose of valuable lives all over the world.

1. Introduction: One of the prominent diseases that affect many people during middle or old age is heart disease, and in many cases, it eventually leads to fatal complications. Heart diseases are more prevalent in men than in women. According to statistics from WHO, it has been estimated that 24% of deaths due to non-communicable diseases in India are caused by heart ailments. One-third of all global deaths are due to heart diseases. Half of the deaths in the United States and in other developed countries are due to heart ailments. Around 17 million people die due to cardiovascular disease (CVD) every year worldwide, and the disease is highly prevalent in Asia. Heart disease defines several healthcare conditions that are vast in nature which is related to the heart and has many basic causes that affect the entire body. This study's goal is to predict the presence of heart disease in patients where this presence is valued from no presence to likely presence. Machine Learning provides the methodology and technology to convert these data mounds into useful decision-making information. This predication system for heart disease would facilitate Cardiologists in taking quicker decisions so that more patients can receive treatments within a shorter period of time, resulting in saving millions of life.

2. Objectives: The main objective of this study is to predict whether a patient is affected with heart disease or not using different machine learning algorithms on a qualified dataset. Find out the co-relations between different attributes. Obtaining clear idea of our proposed Machine Learning techniques and analyse the result and comparing between the results of different data mining techniques. We will analyse our techniques if there is any possibility to bring improvement for our results.

3. Methods: Machine Learning provides the methodology and technology to convert data mounds into useful decision-making information. In this research the comparison of different machine learning techniques like- Support Vector Machine, Logistic Regression, Decision Tree, Random Forest, Naive Bayes algorithm, K-nearby neighbour algorithm (KNN) are implemented to predict heart disease [1]. Naïve mathematician used probability for predicating heart disease, SVM used on classification and regression technique, Random Forest works with varied decision Tree. These algorithms show different accuracy. We will try to tuning our techniques to obtain better accuracy which will be beneficial for more accurate prediction.

4. Literature Review: We will be discussing about various machine learning classifiers and previous work on the heart disease. In machine learning we can use different algorithms otherwise known as classifiers to help us predict. Here in our project we are looking forward to predict the number of patient that have heart disease and the number of patient that do not have heart disease running six algorithms to our data set. The reason we are going to use six is that it will allow us to get better and more reliable prediction. Because if we are using one algorithm or classifier and do not have anything else to compare it with then we cannot say that it a reliable prediction because it might be giving us a very good accuracy but this algorithm might not be the best or more appropriate one to use for our scenario. Whereas if we use more than one algorithm or classifier in our case four of them, we can compare them with one another and if we find one classifier is giving us accuracy that is not even in the ball park of the other algorithm provided accuracy we can understand that something is going wrong. It can be that the algorithm itself is not suitable for the job or we made a mistake in our coding . So using more than one algorithm is essential for any prediction based system. Now the algorithms that we have chosen to use in our project are: 1. Decision tree, 2. Naïve Bayes, 3.SVM (support vector machine), 4.Logistic Regression 5. KNN Algorithm and lastly 6. Random Forest. We will be discussing each of those algorithms below. And finally we are also going to discuss about the previous work that has been done and

show how it improved over time and what improvements we were able to bring in our project. Many research has been done on blood test in order to predict heart disease. Our blood offer us with many clues about our heart condition. For example, if our cholesterol in our blood is high that is a clear sign that we are at the increased risk of having a heart attack. Other substances in our blood can also help our doctor to determine if we have heart failure or are at risk of developing plaque deposits in our arteries also known as atherosclerosis. So it is very important to remember that one blood test alone is not enough to determine our risk of heart disease. The most vital risk factors for cardiopathy square measure smoking, high blood pressure high cholesterol and diabetes. Now let us look at some of the blood test that we can do to diagnosis and manage heart disease. First of all we can do the cholesterol test. A cholesterol test also known as lipid panel or lipid profile, measures the flats (lipids) in our blood. The measurements can indicate our risk of having a heart attack or other heart disease. The test is typically including measurements of – (1) total cholesterol. This is a sum of our blood cholesterol content. If it is high than it puts us at a high level risk of having a heart attack. In an ideal state, the total cholesterol should be below 200 mg per deciliter (mg/dL) or 5.2 mill moles per liter (mmol/L). (2) Low-density lipoprotein (LDL) cholesterol. This is sometimes called the 'bad' cholesterol. Too much of it in blood causes the accumulation of fatty deposits in our arteries, which reduces blood flow. These plaque deposits typically rupture and cause major heart and tube issues. Our LDL cholesterol level should be less than 130 mg/dL in order for us the stay fit. More desirable level should be under 100 mg/dL, especially if we have diabetes or a history of heart attack, heart stents, heart bypass surgery or other heart/vascular conditions. (3) High-density lipoprotein (HDL) cholesterol. This is typically referred to as the 'good' cholesterin as a result of it helps take away cholesterin, keeping arteries open and your blood flowing more freely. Ideally, your HDL cholesterol level should be over 40 mg/dL for a man, and over 50 mg/dL for a woman. (4) Triglycerides. It is another type of fat in the blood. High lipoid levels typically mean you frequently eat a lot of calories than you burn. If it go too high it can increase our risk of heart disease. Ideally, our triglyceride level should be less than 150 mg/dL. The American heart association (AHA) states that a triglyceride level of 100 mg/dL or lower is considered 'optimal'. (5) Non-HDL cholesterol. Non- high density compound protein cholesterol is that the distinction between total cholesterol and HDL cholesterol (HDL-C). Non-HDL-C contains of cholesterol in lipoprotein particles that are involved in hardening of the arteries (atherosclerosis). This includes beta-lipoprotein (LDL), compound protein (a), intermediate-density compound protein and very-low-density compound protein. In some cases Non-HDL-C fraction can be considered a better marker of risk than LDL cholesterol [3]. Now let us talk a little bit further about High-sensitivity C reactive protein. C-reactive protein otherwise known as CRP is a protein your liver produces as part of your body's response to injury or infection (inflammatory response). CRP is a sign of inflammation somewhere in the body. But high sensitivity CRP tests cannot figure out where exactly in the body this may be happening or why it is happening. Inflammation plays a central role in the process of atherosclerosis where fatty deposits clog our arteries. Now measuring CRP alone will not tell our doctor our risk of heart disease. Natriuretic peptide (BNP), is a protein that our heart and blood vessels produce. BNP can help us by eliminating our body fluids and relaxing our blood vessels and funnels sodium into our urine [3]. When our heart is damaged our body secretes high levels of BNP into our blood stream to try to ease the strain on our heart [3]. BNP levels may also rise if we have a new or increased chest pain (unstable angina) or after a heart attack. Now our BNP level can also help in the diagnosis and evaluation of heart failure and other heart conditions. Normal levels do vary according to age and gender and whether we are overweight. One of the foremost vital uses of BNP is to do to map out whether or not shortness of breath is because of failure. Now for people who have heart failure, establishing a baseline BNP can be helpful and future tests can be used to help gauge how well our treatment works. In addition to that a variation of BNP called N-terminal BNP also is useful in diagnosing heart failure and in some laboratories is used instead of BNP N- terminal BNP may also be useful in evaluating. our risk of a heart attack and other problems if we already have heart disease. Finally A high level of BNP alone is not enough to diagnose a heart

problem in that case our doctor will also consider our risk factors and other blood test results [5]. So as we can see due to the demand of having a system that can predict heart disease prediction many studies have taken place.

5. Methodology: These are the algorithms used for prediction of the Heart Condition. These are all machine learning algorithms. For giving the best results PCA (Principle Component Analysis).

1)Decision Tree: Decision Tree Algorithm is an ensemble learning of methods like classification, regression and other tasks. This is based on the classification of the attributes based on their Info and gain. The height gain attribute is split into its respective values. The new split data is set into new data sets and the applied as new decision tree. This is repeated until we get pure subsets. This is one of the best algorithms for linearly distributed datasets. [4] Tree models with a discrete set of values that the target variable can take are called classification trees; In these tree structures, leaves represent class labels and branches represent combination of features that lead to class labels.

2) Random Forest Algorithm: Random Forest algorithm or Random forests are the same algorithms These are ensembles of methods for classification, regression and other tasks. These work on the basis of construction of multiple decision trees. This will be better than decision tree because here we are using multiple trees to predict the results, as there are more trees to predict the accurate results. Random decision forests correct the flaw of decision tree algorithm's habit of overfitting to their training set. Random forest algorithm is more adaptable and easier to use machine learning algorithm that predicts and produces, even without hyper-parameter tuning. It is mostly used algorithm because of its simplicity.

3) Naïve Bayes: Naïve Bayes classifier is a mere "probabilistic classifier" within the family.This is based on applying the theorem of Bayes with strong independent assumptions among the features of the attributes. Naïve Bayes classifiers are exceptionally adaptable, requiring different parameters in the amount of factors in a learning problem. By evaluating iterative approximately the most extreme probability training can be performed.

4) SVM Kernel (RBF and POLY): In Machine Learning, kernel methods are a class of algorithms for pattern analysis the Support Vector Machine (SVM) being the best-known member. The pattern analysis task is to study and find the general relationships in clusters, rankings, in any given data sets. Radial Basis Function kernel, or RBF kernel, is widely used in various learning algorithms. In precise, it is ordinarily used in Support Vector Machine Classification. The polynomial kernel is a kernel function widely used with support vector machines (SVMs) and other kernelized models, that exemplifies the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.

5) K-Nearest Neighbour: One of the simplest of all Machine learning algorithms is the k - NN algorithm. The algorithm k-nearest neighbour is a method used to classify and regress. In both cases, the input consists in the feature space of the k closest training examples. The output depends on whether classification or regression k-NN is used K-NN is a lazy learning method in which the function is only locally approximated and the computation is postponed until final classification.

6) Logistic Regression: Logistic Regression is a method which analyses a dataset which has a one or more independent variable and gives an outcome. The goal of the Logistic Regression is to predict the best relationship between the dependent and independent variables. Figure 3 shows the Logistic Regression of the model and the accuracy of the test and train model.

Method of applying the algorithm:
1) First importing the libraries
2) Reading the dataset, splitting the dataset into training and test of 80% and 20%
3) Applying Feature Scaling
4) Setting the Classifier for all the ML algorithms one at a time
5) Predicating the results
6) Finding the accuracy by using Confusion Matrix
7) Applying the PCA algorithms and taking the new training and test datasets.

8) Setting the Classifier for all the ML algorithms one at a time
9)Predicating the results
10)Finding the accuracy by using

5.1 Design of System : In this portion of our report we are going to discuss how we prepared or designed the whole system. In terms of how we executed the system it will be discussed later in the paper.

5.2 Data Collection: We found our data set that has been used in our paper from kaggle (https://www.kaggle.com/ ronitf/heart-disease-uci/version/1) [2]. The dataset that we used in our project has in total 14 columns and 303 rows. First 13 of those columns are the features that we will be using later on in order to predict the final column 'diagnosis' which will tell us if the patient is going to be affected by heart disease or not. The 303 rows represents data of 303 patients that we found from the dataset.

S.No	Attribute Name	Description
1	Age	Age of the person in years
2	Sex	Gender of the person [1: Male, 0: Female]
3	Cp	Chest pain type [1-Typical Type 1 Angina 2- Atypical Type Angina 3-Non-angina pain 4-Asymptomatic)
4	Trestbps	Resting Blood Pressure in mm Hg
5	Chol	Serum cholesterol in mg/dl
6	Fbs	Fasting Blood Sugar in mg/dl
7	Restecg	Resting Electrocardiographic Results
8	Thalach	Maximum Heart Rate Achieved
9	Exang	Exercise Induced Angina
10	OldPeak	ST depression induced by exercise relative to res
11	Slope	Slope of the Peak Exercise ST segment
12	Ca	Number of major vessels colored by fluoroscopy
13	Thal	3 – Normal, 6 – Fixed Defect, 7 – Reversible Defe
14	Num	Class Attribute

Fig 1: Shows the Dataset from Kaggle website.

5.3 Data Pre-processing: Before we start let us give a brief information about what data pre-processing actually is. Data pre-processing may be a data processing technique that involves re-modelling data into a lucid format. Real-world data is often incomplete, inconsistent and lacking in certain behaviours or trends and is likely to contain many errors. Data pre-processing may be a tried technique of partitioning such problems. Data pre-processing prepares raw data for further processing. Data pre-processing is used in database-driven applications such as customer relationship management and rule-based applications. For our project we are using standard scaler from the sklearn library for pre-processing our data. We choose this one over the many other ones because it suits very well with our system.

Load Data: We created an array called col names and put down all our columns on that array. Then we read the csv file also known as the dataset file as shown in the below figure

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	ta
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Fig 2.Dataset Downloaded

We notice that the dataset has no null values as shown in the below figure Fig 3. This saved us from converting the null values into some data or dropping then altogether. Which also shows that the data is of same datatype.

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
```

Fig 3.Dataset Non-null entry

As the null values are very less we can either drop them or impute them. I have imputed the mean in place of the null values however one can also delete these rows entirely.

After loading the data, best Data Science practice is to check whether the data is balance or not. Below figure shows the data is balanced.

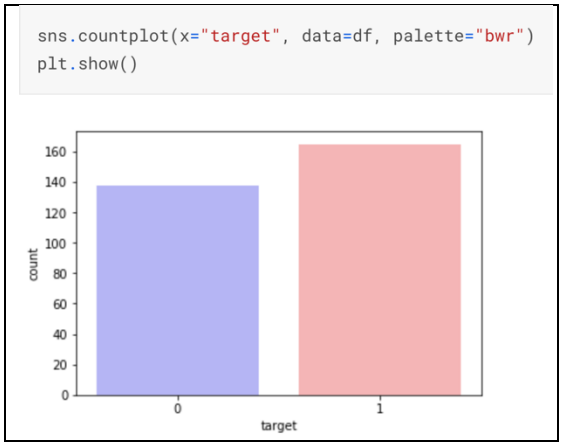


Fig 4.Dataset Balanced

Below figure shows the count of male and female present in the dataset.

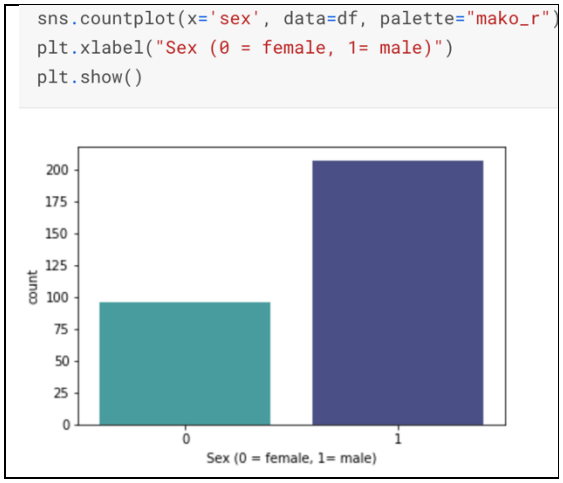


Fig 5.Male vs. Female Data count

5.4 Analyse Features: In this section we are going to distribute the target value is vital for choosing appropriate accuracy metrics and consequently properly assess different machine learning models. First of all we are going to count values of explained variable otherwise known as the determining variable which is going to give us the prediction of a patient being affected by heart disease or not. Second of all we are going to separate numeric features from categorical features. Then we are going to show the relation between the categorical features in various plots and try to figure out or rather observe the influence of those categorical features in the actual determining variable "diagnosis".

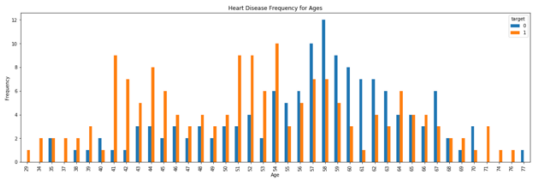


Fig 6. Frequency of heart diseases based on age

The above figure shows the frequency of heart disease in patients based on the Age factor. According to the figure shown , we can make out that people with age ranging from 50-67 are more prone for heart diseases. In the similar way we analyse the data which includes many contributory risk factors.
Statistical Details Describe provides us with statistical information in the numerical format. we can infer that in the AGE column the minimum age is 29yrs and maximum is 77yrs mean of age is 54yrs. The quartiles details are given in form of 25%, 50% and 75%. The data is divided into 3 quartiles or 4 equal parts. so 25% values lie in each group. standard deviation and mean are statistical measures which give us an idea of the central tendency of the data set. However , mean is effected by outliers and hence we need more information to make accurate decisions.

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve.

We have used correlation matrix with heatmap as it is considered to be efficient technique as shown in the below figure Fig 6.

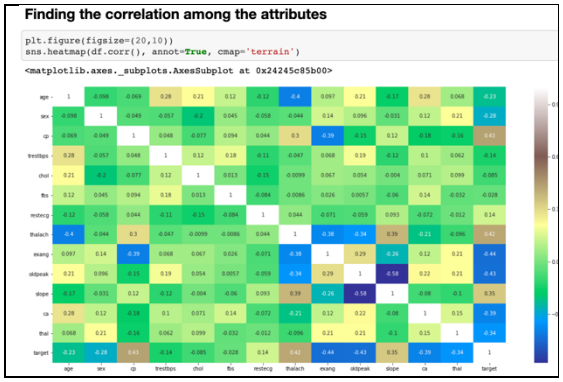


Fig 7.Co-relation matrix using HeatMap

How to select features and what are Benefits of performing feature selection before modelling your data?

· **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
· **Improves Accuracy**: Less misleading data means modeling accuracy improves.
· **Reduces Training Time**: fewer data points reduce algorithm complexity and algorithms train faster.
There are 3 Feature selection techniques that are easy to use and also gives good results.

1. Univariate Selection
2. Feature Importance
3. Correlation Matrix with Heatmap

Correlation states how the features are related to each other or the target variable. Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable). Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)

The Below figure Fig.8 shows how each and every. Attribute has been related with each other.

Data Processing: After exploring the dataset, I observed that I need to convert some categorical variables into dummy variables and scale all the values before training the Machine Learning models. First, I'll use the get_dummies method to create dummy columns for categorical variables.

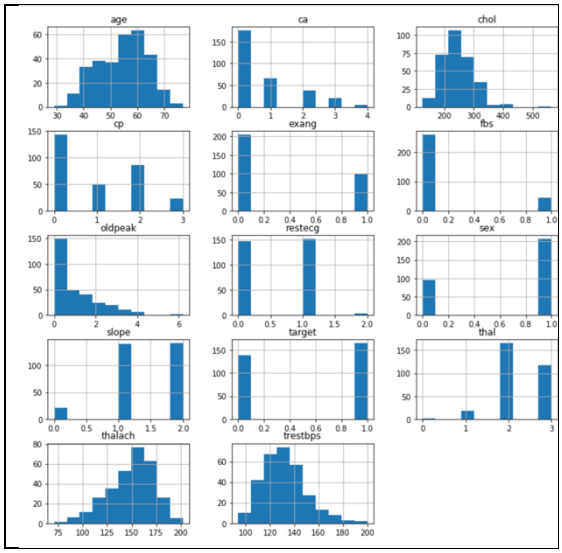


Fig 8.Relation with Data Attributes

It is a step of Data Pre Processing which is applied to independent variables or features of data. It basically helps to normalise the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.
Package used is sklearn.pre-processing.

Why and Where to apply Feature Scaling?
Real world dataset contains features that highly vary in magnitudes, units, and range. Normalisation should be performed when the scale of a feature is irrelevant or misleading and not should Normalise when the scale is meaningful.

The algorithms which use Euclidean Distance measure are sensitive to Magnitudes. Here feature scaling helps to weigh all the features equally.

Formally, If a feature in the dataset is big in scale compared to others then in algorithms where Euclidean distance is measured this big scaled feature becomes dominating and needs to be normalized.

We have done standard scaling that means when we compare the dataset where age, chol, restbps. These are something that algorithm wouldn't understand.
So, we need to scale those values first as our dataset has features which have lot of variables

and they're measured in different units. Based on the standard distribution the values will be scaled down as shown in the below figure:

```
In [71]: from sklearn.model_selection import train_test_split
         from sklearn.preprocessing import StandardScaler
         StandardScaler = StandardScaler()
         columns_to_scale = ['age','trestbps','chol','thalach','oldpeak']
         df[columns_to_scale] = StandardScaler.fit_transform(df[columns_to_scale])

In [72]: df.head()

Out[72]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	0.952197	1	3	0.763956	-0.256334	1	0	0.015443	0	1.087338	0	0	1	1
1	-1.915313	1	2	-0.092738	0.072199	0	1	1.633471	0	2.122573	0	0	2	1
2	-1.474158	0	1	-0.092738	-0.816773	0	0	0.977514	0	0.310912	2	0	2	1
3	0.180175	1	1	-0.663867	-0.198357	0	1	1.239897	0	-0.206705	2	0	2	1
4	0.290464	0	0	-0.663867	2.082050	0	1	0.583939	1	-0.379244	2	0	2	1

```
In [73]: X= df.drop(['target'], axis=1)
         y= df['target']

In [74]: X_train, X_test,y_train, y_test=train_test_split(X,y,test_size=0.3,random_state=40)
```

Fig 9.Standard Distribution

5.5 Feature Engineering: 1. A lot of features can affect the accuracy of the algorithm. So working with the features is very important. There are few reasons for which some may want to work with some selected features.
2. Choosing less features helps us to train faster.
3. By picking up the most important features, we can use interactions between them as new features. Sometimes this gives surprising improvement.
4. Some features are linearly related to others. This might put a strain on the model.
5. Feature Selection means to select only the important features in-order to improve the accuracy of the algorithm.
6. It reduces training time and reduces over fitting.

5.5.1 Feature Importance: A very basic question that we might ask of a model is what features have the biggest impact on predictions? This concept is called feature importance. In dataset there may be some attributes which don't effect the prediction that much [1]. In some cases, few attributes may decrease the accuracy level of a model. So, it is important to work with the correct attributes. So far we have worked with all the features of the dataset and listed [4] the accuracy of different models. Now, we want to see the change of accuracies of different classifiers after selecting a subset of the attributes.

5.6 Modeling and predicting with machine learning: The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. I have chosen several algorithms typical for solving supervised learning problems throughout classification methods. First of all, let's equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model over fits or under fits the data (so-called bias/variance tradeoff). Then we are going to split the data then test and train them in the ratio of 70:30. Then we are going to create a model where we are going to run all our algorithms.

As mentioned above in the methodology we check the accuracy using the confusion matrix. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error 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. It allows the visualization of the performance of algorithm.

It allows easy identification of confusion between classes e.g. one class is commonly mis-labelled as the other. Most performance measures are computed from the confusion matrix.
A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.

Definition of the Terms:

- Positive (P) : Observation is positive (for example: is an apple).
- Negative (N) : Observation is not positive (for example: is not an apple).

- 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.

The same thing is applied on all the algorithms to check the accuracy of all the algorithms that's been used in our project to predict that one efficient algorithm.

After this, the original data is compared to the predicted data and then the accuracy is calculated. Sometimes referred to as testing data, the holdout data provide the final estimation of the machine learning model performance after it has been trained. In this model, we divided the 303 patients into two parts. In the hold, an experiment, we almost used 2/4 data for training and build the classification model.

```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()

model1=lr.fit(X_train,y_train)
prediction1=model1.predict(X_test)
```

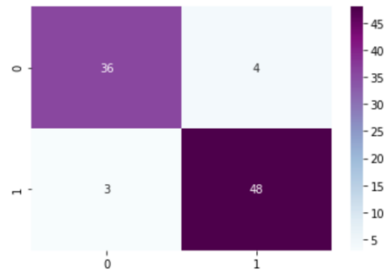
```
from sklearn.metrics import confusion_matrix

cm=confusion_matrix(y_test,prediction1)
cm
```

```
array([[36,  4],
       [ 3, 48]], dtype=int64)
```

```
sns.heatmap(cm, annot=True,cmap='BuPu')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1608a0eec48
```




```
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
print('Testing Accuracy:',(TP+TN)/(TP+TN+FN+FP))
```

```
Testing Accuracy: 0.9230769230769231
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,prediction1)
```

```
0.9230769230769231
```

Fig 10.Confusion Matrix for Logistic Regression

The above figure shows the computation of Logistic Regression and the accuracy that has been obtained from the Confusion Matrix.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction1))
```

	precision	recall	f1-score	support
0	0.92	0.90	0.91	40
1	0.92	0.94	0.93	51
accuracy			0.92	91
macro avg	0.92	0.92	0.92	91
weighted avg	0.92	0.92	0.92	91

Fig 11.Classification report

The above figure shows the classification report of the accuracy obtained.
The same has been computed for all the other algorithms which is shown below:

```
from sklearn.naive_bayes import GaussianNB

NB = GaussianNB()
model5 = NB.fit(X_train, y_train)
prediction5 = model5.predict(X_test)
cm5= confusion_matrix(y_test, prediction5)
```

```
cm5
```

```
array([[35,  5],
       [ 6, 45]], dtype=int64)
```

```
accuracy_score(y_test, prediction5)
```

```
0.8791208791208791
```

Fig 12. Confusion Matrix for Naïve Bayes

Above figure shows the computation for Naïve Bayes Algorithm.

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

dtc=DecisionTreeClassifier()
model2=dtc.fit(X_train,y_train)
prediction2=model2.predict(X_test)
cm2= confusion_matrix(y_test,prediction2)
```

```
cm2
```

```
array([[33,  7],
       [16, 35]], dtype=int64)
```

```
accuracy_score(y_test,prediction2)
```

```
0.7472527472527473
```

```
print(classification_report(y_test, prediction2))
```

	precision	recall	f1-score	support
0	0.67	0.82	0.74	40
1	0.83	0.69	0.75	51
accuracy			0.75	91
macro avg	0.75	0.76	0.75	91
weighted avg	0.76	0.75	0.75	91

Fig 13. Confusion Matrix for Decision tree

The above figure Fig 13, shows the accuracy and classification report of Decision Tree.

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier()
model3 = rfc.fit(X_train, y_train)
prediction3 = model3.predict(X_test)
confusion_matrix(y_test, prediction3)
```

```
array([[34,  6],
       [ 6, 45]], dtype=int64)
```

```
accuracy_score(y_test, prediction3)
```

```
0.8681318681318682
```

```
print(classification_report(y_test, prediction3))
```

```
              precision    recall  f1-score   support

           0       0.85      0.85      0.85        40
           1       0.88      0.88      0.88        51

    accuracy                           0.87        91
   macro avg       0.87      0.87      0.87        91
weighted avg       0.87      0.87      0.87        91
```

Fig 14. Confusion Matrix for Random Forest

The above figure shows the accuracy and classification report of Random Forest Algorithm.

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

```
from sklearn.svm import SVC

svm=SVC()
model4=svm.fit(X_train,y_train)
prediction4=model4.predict(X_test)
cm4= confusion_matrix(y_test,prediction4)
```

```
cm4
```

```
array([[33,  7],
       [ 2, 49]], dtype=int64)
```

```
accuracy_score(y_test, prediction4)
```

```
0.9010989010989011
```

Fig 15. Confusion Matrix for KNN

The above figure shows the accuracy of SVC classification Algorithm.

```
print('KNN :', accuracy_score(y_test, prediction6))
print('lr :', accuracy_score(y_test, prediction1))
print('dtc :', accuracy_score(y_test, prediction2))
print('rfc :', accuracy_score(y_test, prediction3))
print('NB: ', accuracy_score(y_test, prediction4))
print('SVC :', accuracy_score(y_test, prediction5))
```

```
KNN : 0.8351648351648352
lr : 0.9230769230769231
dtc : 0.9010989010989011
rfc : 0.8681318681318682
NB:  0.9010989010989011
SVC : 0.8791208791208791
```

Best accuracy is given by Logistic Regression : 9

Fig 16.Efficiency of all the algorithms

The above algorithm shows the accuracy details for all the algorithms and by the computation we can analyse that best accurate results is given by Logistic Regression Algorithm followed by Naïve Baye's and Decision Tree Algorithm compared to other algorithms.

6. Conclusion and Future Work: In this paper, we have presented a system which is suitable for real-time heart diseases prediction and can be used by the users who have coronary disease. Different from many other systems it is able to both monitor and prediction. The diagnosis system of the system is able to predict the heart disease by using ML algorithms and the prediction results are based on the heart disease dataset instance. To prove the effectiveness of the system we have carried out experiments for both monitoring and diagnosis system . we ran experiments with some popular algorithms like KNN, Decision Tree, Random Forest, Naive Bayes, SVM, Logistic Regression. The experiment was carried out with the holdout test and the accuracy of the proposed system was 92% achieved with the Logistic Regression algorithm. Heart Disease is one of the major concerns for society today.
It is difficult to manually determine the odds of getting heart disease based on risk factors. However, machine learning techniques are useful to predict the output from existing data.

7. References:

[1].https://www.researchgate.net/publication/319393368_Heart_Disease_Diagnosis_and_Prediction_Using_Machine_Learning_and_Data_Mining_Techniques_A_Review

[2]. Predicting and Diagnosing of Heart Disease Using Machine Learning Algorithms Sanjay Kumar Sen Asst. Professor, Computer Science & Engg. Orissa Engineering College, Bhubaneswar, Odisha – India

[3]. Heart disease prediction using machine learning techniques as shown in : a survey V.V. Ramalingam*, Ayantan Dandapath, M Karthik Raja

[4]. Effective Diagnosis and Monitoring of Heart Disease Ahmed Fawzi Otoom1 , Emad E. Abdallah2 , Yousef Kilani3 , Ahmed Kefaye4

and Mohammad Ashour5 1,2,3,4,5 Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology The Hashemite University, Zarqa, Jordan { 1 bottom, 2 emad,3 ymkilani}@hu.edu.jo,4a.kefaye@alpha-hub.com,m.ashour@teleogx.com [https://pdfs.semanticscholar.org/f4ec/b47e080001d8ea08bab686acdb5a741a7159.pdf]

[5]. Disease Prediction Using Heart rate Variability Analysis - IoT Dharmik Jampala 1, Venkat Naidu Mittapalle2, Sitanshu Nandan3 1, 2, 3 School of Information Technology and Engineering, VIT University, Vellore