**Predicting Heart Disease using Machine learning techniques**

**Application in Bank Telemarketing Scenario**

Project Report Submitted in Partial Fulfillment of the requirements for the Degree of

**Bachelors of Technology in**

**Computer Science and Engineering By**

**Ali Hussain (Roll No. 2029) Moonis Ali (Roll No. 2050) Iqra shafi (Roll No. 2033) Mannan Shah (Roll No. 2022)**

**Under the Supervision of Khalid Hussain (Assistant Professor)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY OF KASHMIR**

## NORTH CAMPUS

**DELINA, BARAMULLA-193101 (SESSION 2018)**

**Department of Computer Science and Engineering**

**University of Kashmir**

## North Campus

**Delina, Baramulla 193101**

**(SESSION 2018)**

****

**Certificate**

This is to certify that this project report entitled *PREDICTIVE ANALYTICS FOR CUSTOMER TARGETING: Application in Bank Telemarketing Scenario* by Moonis Ali (Roll No. 14048112050), Ali Hussain (Roll No. 14048112029) , Iqra Shafi (Roll No. 14048112033) and Mannan Shah (Roll No. 14048112022), submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering of the University of Kashmir, North Campus, during the academic year 2018, is a bonafide record of work carried out under our guidance and supervision.

(Guide) (External Examiner) (Head of Department)

## STUDENTS’ DECLARATION

We,Moonis Ali (Roll No. 14048112050), Ali Hussain (Roll No. 14048112029) , Iqra Shafi (Roll No. 14048112033) and Mannan Shah (Roll No. 14048112022), hereby declare that the work, which is presented in the project report entitled “*PREDICTIVE ANALYTICS FOR CUSTOMER TARGETING: Application in Bank Telemarketing Scenario.*” submitted in partial fulfilment of the requirements for the award of “Bachelor of Technology in Computer Science & Engineering ” degree in the session 2018, is an authentic record of our own work carried out under the supervision of Mr.

Khalid Hussain, Department Of Computer Science & Engineering, University Of Kashmir North Campus, Delina, Baramulla.

|  |  |  |  |
| --- | --- | --- | --- |
| **Moonis Ali** | **Ali Hussain** | **Iqra Shafi** | **Mannan Shah** |
| **Roll No.** | **Roll No.** | **Roll No.** | **Roll No.** |
| **14048112050** | **14048112029** | **14048112033** | **14048112022** |

**Dated: October 2018**

## ACKNOWLEDGMENT

The progress of the project ***PREDICTIVE ANALYTICS FOR CUSTOMER TARGETING: Application in Bank Telemarketing Scenario*** could not have been made without the help of following people to whom we would like to express our most sincere gratitude.

First of all we are grateful to our teacher & guide Mr.Khalid Hussain whose immense motivation and knowledge have been a driving force behind this project.

We are also thankful to Dr.Muheet Ahmed Butt, Scientist ‘D’, Department of Computer Science, University of Kashmir for his constant encouragement and support.

We would like to put on record the support of our parents who always stood by us in situations of grim affliction and patted us when this work was going on.

## ABSTRACT

Throughout the world, banking institutions employ diverse marketing strategies to advertise their products, gain new customers and/or retain the old ones. Telemarketing is one such method utilized by banks where individual customers are contacted by bank representatives with offers. The work undertaken in the project deals with finding solution to such problems which may help to re-design telemarketing strategies. This can be done when telemarketing strategies are based on data science and machine learning methods which can predict the outcome- whether a customer will buy the product or not. The work focuses on the various aspects of the dataset and endeavors to come up with a structured approach to derive proper insights from the data that would help in solving problems which are similar to this. In the end extensive comparative analysis is presented to aid in the understanding of different classification algorithms and various performance parameters that were employed for this problem.

**Keywords:** Telemarketing, data science, machine learning, classification algorithms.

## TABLE OF CONTENTS

Page No. LIST OF TABLES viii

[LIST OF FIGURES ix](#_TOC_250032)

CHAPTER

1. INTRODUCTION… 10
   1. [Motivation 11](#_TOC_250031)
   2. [Problem description 12](#_TOC_250030)
   3. [Outline… 13](#_TOC_250029)
2. LITERATURE SURVEY… 15
   1. [Introduction. 15](#_TOC_250028)
   2. Examination of Related Work 15
3. BACKGROUND 17
   1. [Introduction 18](#_TOC_250027)
   2. [Machine learning 18](#_TOC_250026)
   3. [Data Analytics 19](#_TOC_250025)
   4. [Algorithms 19](#_TOC_250024)
      1. [Logistic Regression 19](#_TOC_250023)
      2. [Support Vector Machine 20](#_TOC_250022)
      3. kNN (k-Nearest-Neighbour) 21
      4. [Decision Tree 21](#_TOC_250021)
      5. [Random Forest 22](#_TOC_250020)
      6. [Extra Tree 22](#_TOC_250019)
      7. [AdaBoost 23](#_TOC_250018)

[3.4.8 Voting Classifier …...................................................................](#_TOC_250017)

24

* 1. [Evaluation Metrics… 24](#_TOC_250016)

1. DATASET EXPLORATION 29
   1. [Introduction… 30](#_TOC_250015)
   2. [Overview of Dataset 30](#_TOC_250014)
   3. [Segmentation of Features 31](#_TOC_250013)
   4. [Exploratory Data Analysis 33](#_TOC_250012)
      1. [Descriptive Statistics 33](#_TOC_250011)
      2. Exploratory Data Visualization. 35
2. IMPLEMENTATION… 44
   1. [Introduction 45](#_TOC_250010)
   2. [Data Preprocessing Overview 45](#_TOC_250009)
      1. [Data Aggregation 47](#_TOC_250008)
      2. [Data Discretization 48](#_TOC_250007)
      3. [Imputation Analysis 50](#_TOC_250006)
   3. [Establishing Feature Importance 51](#_TOC_250005)
3. RESULTS AND COMPARISONS… 53
   1. [Introduction 54](#_TOC_250004)
   2. [Classification Results 54](#_TOC_250003)
   3. [Impact of Feature Selection. 56](#_TOC_250002)
4. Conclusion and Future Scope 59

[REFERENCES 61](#_TOC_250001)

[APPENDIX 65](#_TOC_250000)

No.

## LIST OF TABLES

Table Page

1. Snapshot of raw dataset 30
2. Description of features in the raw dataset. 32
3. Descriptive Summary of Dataset. 34
4. Feature engineering of education attribute… 47
5. Feature engineering of job attribute… 47
6. Cross-tabulation of pdays and poutcome… 48
7. Discretization of pdays attribute 49
8. Cross tabulation of education and job… 50
9. Cross-tabulation result after distribution… 50
10. Comprehensive analysis - I 54
11. Comprehensive analysis - II. 56

## LIST OF FIGURES

Figure Page No.

* 1. Overview of the problem… 12
  2. Types of Machine learning techniques… 18
  3. Confusion Matrix… 25
  4. Confusion Matrix of a Model 26
  5. Precision and recall… 27
  6. Contribution of all dataset features to each particular category. 31
  7. Total count vs Type of Job. 35
  8. Type of Job vs Total Count w/target variable distribution. 35
  9. The distribution of ‘yes’ and ‘no’ in target variable 35
  10. Total count vs (type of) Contact 36
  11. (Type of ) Contact vs Total Count w/ target variable distribution… 36
  12. Total count vs Default 36

13.Default vs Total Count w/ target variable distribution. 36

1. Total count vs Education 37
2. Education vs Total Count w/ target variable distribution. 37
3. Total count vs poutcome 38
4. Total Count w/ target variable distribution. 38
5. Probability density function for pdays. 38
6. Histogram for pdays. 38
7. Probability density function for age & its histogram. 39
8. Histogram for age… 39
9. Probability density function of duration (last contact duration in seconds). 40
10. Histogram for duration (last contact duration in seconds) 40
11. Probability density function of Euribor rate 40
12. Histogram for Euribor rate 40
13. Probability density function of Consumer Confidence Index… 41
14. Histogram for Consumer Confidence Index… 41
15. Probability density function of consumer price index 42
16. Histogram for consumer price index… 42
17. Probability density for Previous 42
18. Histogram for Previous 42
19. Data Preprocessing… 46
20. Number of features versus information gain. 51
21. Contribution of features to output… 52
22. Execution time of algorithms without feature selection. 55
23. Execution time of various algorithms with and without feature selection… 57
24. Comparison of algorithms vs various performance parameters 58

**CHAPTER 1**

# INTRODUCTION

## Motivation

Over the course of the last 2-3, decades marketing has become an essential part of any thriving business establishment. Banking institutions utilize marketing techniques to increase client subscriptions to investments. In turn, these strategies increase customer retention and improve customer relationship. One of the most widely used marketing technique is telemarketing, where phone calls are made by the banks to customers in order to gain new investments and increase company profits. Although a prominent working strategy, there is more that can be done to maximize profits using telemarketing methods. To improve efficiency and decrease operational costs, these marketing strategies can be further improved with use of statistical techniques that can predict in advance, whether a customer will buy a new product/scheme or not. Through the use of Machine Learning classification algorithms, various organizations can make these predictions of client interest to refine and customize their marketing strategies according to their customer base.

Machine Learning helps in the derivation of the meaningful information from the dataset by identifying patterns in it. These algorithms use the historical data in providing us with the probable picture of the future events. Such futuristic outlooks can serve as a guide for making beneficial decisions in the present. Classification is a type of Machine Learning problem wherein we classify the problem into various classes.Machine learning classification algorithms help in creating models on which future records can be evaluated. Dataset is initially divided into two parts, one part of the dataset is the training set and the other portion is the testing set. The training set will be used to generate a Machine learning model that is used to predict the future values. The test set consists of the data that is unseen by the model, it is used to evaluate the model with the idea that it is representative of the population and eventually also future instances.

A classification model can be utilized to improve bank decision-making. For example, predicting clients who are most and least likely to subscribe will allow a bank to prioritize the customers to contact for each subscription offer in order to maximize the total number of subscriptions in less time. Overall, it will increase the

bank’s focus to areas that are likely to cause the most efficient use of company resources.

## Problem Description

This project relates to the marketing campaign of Banco de Portugal, a renowned Portuguese banking institution. It provides a solution in terms of YES/NO prediction to the bank’s marketing division, to sell long-term (fixed) deposits to clients based on various parameters such as their socio-economic indicators, their marital status, type of job, age, etc. The aim of the project is to make use of machine learning techniques to build an effective predictive-analytical model which can forecast, whether a client is likely to subscribe to a long-term (fixed) deposit account with the bank or not. Since the resulting variable (outcome) is either a Yes or No and more than two variables are used as input to make the prediction, the problem attains the shape of a multivariate binary classification problem.

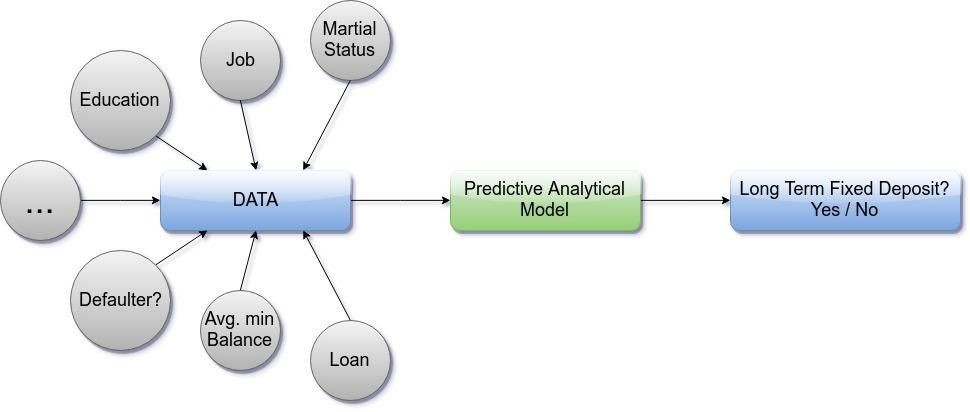


Figure 1: Overview of the problem

## Outline

The remainder of the report is organized in the following format. Chapter 2 summarizes the work that was performed previously by other researchers on the dataset. Chapter 3 covers the background information, which is the foundation knowledge on which the report is based. This includes the dataset analyzed in the report and the methods used for elucidating it. It explains the details of the three algorithms utilized for analysis. In addition, it covers the evaluation metrics used to compare the results.. Chapter 4 focuses on data preprocessing steps used to prepare the data as well as feature engineering. Chapter 5 discusses the implementation process. Chapter 6 presents the results, the methods used for evaluation as well as the comparison of various methods employed. Chapter 7 concludes the report work and proposes further research ideas.

**CHAPTER 2**

# LITERATURE SURVEY

## Introduction

This chapter reviews the research studies that have been carried out by various scholars about the same problem.

## Review of Literature

The problem described in this report has been the centerpiece of many Data Mining and Machine learning research works in the past decade or so.

Lot of work has been carried out to predict heart disease using UCI Machine Learning dataset. Different levels of accuracy have been attained using various data mining and Machine learning techniques which are explained as follows.

Avinash Golande and et.al.;studies various different ML algorithms that can be used for classification of heart disease. Research was carried out to study Decision Tree, KNN and K-Means algorithms that can be used for classification and their accuracy were compared. This research concludes that accuracy obtained by Decision Tree was highest further it was inferred that it can be made efficient by combination of different techniques and parameter tuning[1].

Fahd Saleh Alotaibi has designed a ML model comparing five different algorithms . Rapid Miner tool was used

which resulted in higher accuracy compared to Matlab and

Weka tool. In this research the accuracy of Decision

Tree,Logistic Regression, Random forest, Naive Bayes and SVM classification algorithms were compared. Decision

tree algorithm had the highest accuracy[3].

Theresa Princy. R,etal, executed a survey including different classification algorithm used for predicting heart disease. The classification techniques used were Naive Bayes, KNN (K-Nearest Neighbour), Decision tree, Neural network and accuracy of the classifiers was analyzed for different number of attributes [3].

Nagaraj M Lutimath, et al., has performed the heart disease

prediction using Naive bayes classification and SVM

(Support Vector Machine). The performance measures used

in analysis are Mean Absolute Error, Sum of Squared Error

and Root Mean Squared Error, it is established that SVM was

emerged as superior algorithm in terms of accuracy over

Naive Bayes [4].

The main idea behind the proposed system after reviewing

the above papers was to create a heart disease prediction

system based on the inputs. We analysed the classification algorithms namely Decision Tree, Random Forest, Logistic Regression,XGBOOST

K Nearest Neighbor and Support Machine Vector based on their Accuracy, Precision, Recall and f1 scores and identified the best classification algorithm which can be used in the heart disease prediction.

**Chapter 3**

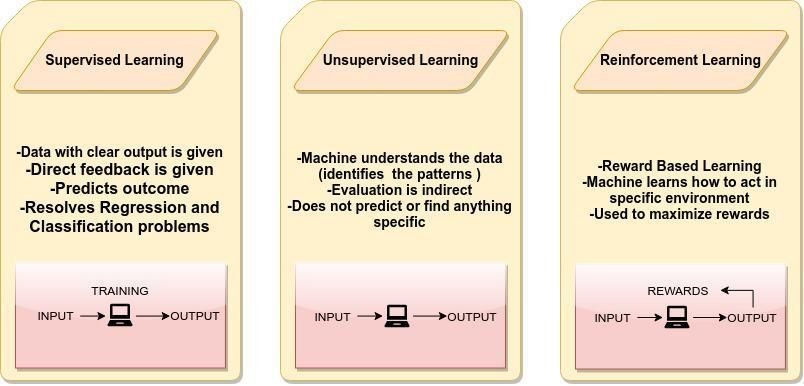
Background

## 3.1 Introduction

This chapter discusses the foundational knowledge on which this report is based. It covers the details of various concepts about Machine learning most notably the fundamental information on the algorithms used for analysis later in the report. It also discusses briefly, the data analytics techniques/processes followed in the report.

## 3.2. Machine learning

Machine learning is a subfield of computer science that empowers computers to act, learn and make decisions like humans, by feeding them data and information in the form of observations and or real-world interactions without the need of explicit programming them. Machine learning systems automatically learn from the inputs that are fed to them. This is a better alternative to manually constructing them because it saves us a lot of time and resource. [1] Such an approach overcomes the shortcomings of traditional programming methods which are often, very static. Various fields wherein algorithms are explicitly programmed are infeasible in areas such as email filtering, intruder detection, optical character recognition (OCR), computer vision, recommending new products to the users or customer segmentation etc, here machine learning algorithms perform significantly better.

Figure 2: Types of Machine learning techniques

## Data Analytics

Data analytics is a widely popular and emerging field that can be defined as the “the science of transforming data into useful insights for better decision making”. Data analytics employs scientific methods as well as advanced use of information technology techniques that support processing of highly sophisticated data and its subsequent analyses.Technically, the following steps can define a data analytics process : Data sourcing (obtainment of data), Data cleansing (detection & correction of incorrect/inaccurate data), and Data Analysis, Data modelling & visualisation and interpretation of data. [2]

In Data analytics, machine learning can be employed to formulate complex models and algorithms that help in making prediction; when this is put into commercial use, this is known as predictive analytics. The core of predictive analytics relies on capturing relationships between explanatory variables and the predicted variables from past occurrences, and exploiting them to predict the unknown outcome.This helps us in producing authentic results, and also assist in revealing hidden insights by grasping the trends and relationships in the data.

## Algorithms

The project makes use of various classification algorithms for predictions and comparisons. All of these algorithms provide a different method to allow model generation for classification of future data instances.

### Logistic regression

Logistic regression is a classification technique that analyzes the relationship between the various attributes. The class may be continuous or categorical but predictions are made on a binary class. The data is first split into a positive and negative class and logistic regression is run. The goal of logistic regression is

to find the best fitting model that will describe the relationship between the inputs and the class. The log odds of the outcome is modeled as a linear combination of the predictor variables.A prediction is made of the probability of the response based on several predictor variables that are independent. Logistic regression generates coefficients as well as standard errors and significance levels of a formula to predict a logic transformation. Instead of the selecting parameters that minimize the sum of squared errors as performed in ordinary regression, logistic regression estimation chooses parameters that maximize the likelihood of observing the sample values.[3]

### Support Vector Machine

SVM is a classification technique based on the concept of decision planes that define decision boundaries. It is a supervised learning algorithm that aims to map the data into space and divide it with a maximized clear boundary. A training dataset identifies the decision boundaries and classifies each bounded area to a specific target value. New instances or records that fall into one of the classification bounded areas will then be categorized as the target value specified for that bounded area. Therefore, all new data points are predicted to belong to one of the divided sides. During training when boundaries are being identified there may be several decision boundaries that can be made to separate two different spaces that is expected to perform equally well on unseen data. In such instances, the decision boundaries with large margins are selected as they tend to have better generalization errors, than those with small margins. Classifiers that produce decision boundaries with small margins are more prone to model overfitting and tend to generalize poorly on unseen data. Therefore, SVM is an optimization algorithm which selects the boundary with the maximum margin. It does not use a greedy-based strategy, which typically finds the local optimal solution, but rather finds the global optimal solution. Depending upon the data, these boundaries may be linear or nonlinear. Non-linear SVM is performed by the use of kernel tricks, which essentially enable the mapping of the inputs into a multi-dimensional feature space. SVM can be applied to categorical data by attributing each categorical value to a numerical value. [4] The LibSVM library

enables SVM classification, regression as well as distribution estimation. It also supports multi-class classifications. The library provides several kernels for use including linear, polynomial, radial basis function and sigmoid. [5]

### kNN (k- Nearest Neighbors)

kNN (k- Nearest Neighbors) is a machine learning algorithm that can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing kNN modeling. [6]

### Decision Tree

Decision tree is a type of supervised machine learning algorithm, mostly used for predicting outcomes in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.[7]

It uses a [tree-like](https://en.wikipedia.org/wiki/Tree_(graph_theory)) [graph](https://en.wikipedia.org/wiki/Diagram) or [model](https://en.wikipedia.org/wiki/Causal_model) of decisions and their possible consequences, including [chance](https://en.wikipedia.org/wiki/Probability) event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.Types of decision tree is based on the type of target variable we have. It can be of two types:

* + - 1. Categorical Variable Decision Tre**e:** Decision Tree which has categorical target variable then it called as categorical variable decision tree. Example:- In

above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. YES or NO.

* + - 1. Continuous Variable Decision Tree**:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

### Random Forest

Random forest is a class of ensemble methods that generates multiple decision trees from the training set. Ensemble methods are techniques that improve classification accuracy by aggregating the predictions of multiple classifiers. An ensemble method creates a set of base classifiers using training data. It then performs classification by taking a vote on the predictions that are made by each base classifier. For an ensemble method classifier to outperform a single classifier, two conditions should be met. The base classifiers should all be independent of each other and the base classifiers should make predictions better than random guessing. Random forest combines predictions from many different decision trees with each tree constructed using values of an independent set of random vectors. First, the original training data is used and randomization is applied. Randomization in random forest helps to reduce the correlation among the decision trees so that the generalization error can be improved. For example, a set of random vectors may be created, where each will be independently used to create a decision tree. The second step is to use the randomized data to build multiple decision trees. Finally a combination of these decision trees yields the final predictions. [4].

**3.4.6 XGBOOST:**

XGBOOST is an algorithm that has recently been dominating applied Machine learning. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the [Gradient Boosting](https://en.wikipedia.org/wiki/Gradient_boosting) framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

### 3.4.7 Voting Classifier

The main concept behind the Voting Classifier is to merge various machine learning classifiers and use a majority vote or the average predicted probabilities (soft vote) to predict the labels of a class. Such a classifier can be useful if we have a set of equally well performing model in order to balance out their individual weaknesses.

In majority voting (hard vote), the predicted class label for a particular sample is the class label that represents the majority (mode) of the class labels predicted by each individual classifier.

In contrast to majority voting (hard voting), soft voting returns the class label as argmax of the sum of predicted probabilities.Specific weights can be assigned to each classifier via the weights parameter. When weights are provided, the predicted class probabilities for each classifier are collected, multiplied by the classifier weight, and averaged. The final class label is then derived from the class label with the highest average probability.[9]

## Evaluation Metrics

The results of predictive models can be viewed in various forms such as by using confusion matrix, root mean squared error, AUC-ROC etc. A confusion matrix is a table that displays the number of instances that are correctly and incorrectly classified in terms of each category within the attribute that is the target class. The positive class is with respect to the current category and the negative class includes 13 all categories other than the current. The confusion matrix displays the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values for a given attribute. TP is the number of values predicted to be positive by the algorithm and was actually positive in the dataset. TN represents the number of values that are expected to not belong to the positive class and actually do not belong to it. FP depicts the number of instances misclassified as belonging to the positive class thus is actually part of the negative class. FN shows the number of instances classified as the negative class but should belong to the positive class. Figure 3 below shows a confusion matrix where the columns represent the prediction and the rows are the actual classification.

|  |  |  |
| --- | --- | --- |
|  | PREDICTED NEGATIVE | PREDICTED POSITIVE |
| ACTUAL NEGATIVE | **TN** | **FP** |
| ACTUAL POSITIVE | **FN** | **TP** |

Figure 3: Confusion Matrix, Columns Represent Prediction and Rows Represent Actual Classification

A common evaluation metric for algorithms is classification accuracy, which is simply referred to as accuracy. Accuracy can be derived from the TP, TN, FP and FN values of a confusion matrix. The equation for accuracy, shown below , identifies the ratio of all values that were correctly classified based on both the positive and negative class over the total number of instances examined. Since the classification accuracy includes values from both the positive class as well as the negative class, the value is consistent for an attribute regardless of the category from which it is extracted.



Accuracy exhibits a phenomenon known as the accuracy paradox. The accuracy paradox states that “predictive models with a given level of accuracy may have greater predictive power than models with higher accuracy” [11]. A useless model, one that predicts only the positive class or only the negative class, can have higher accuracy than a model with some predictive power. Predictive power is the power to make a good prediction. For example, if a model only predicts one class, it has extremely low predictive power. This can be illustrated by the following scenario. Consider the confusion matrices in Figure 4 below. Examining the matrix on the top, the accuracy of the model is accuracy = (100 + 10)/(100 + 50 + 5 + 10) = 66.7%. Now consider the confusion matrix on the bottom which always predicts the negative class.

The accuracy of this matrix is accuracy = (150 + 0)/(150 + 0 + 15 + 0) = 90.9% which is 24.2% higher than from the confusion matrix with more predictive power. Thus, even though this has higher accuracy it is useless as a predictive model since it always predicts the same class. As a general rule, “when TP < FP, then accuracy will always increase when we change a classification rule to always output ‘negative’ category. Conversely, when TN < FN, the same will happen when we change our rule to always output ‘positive’. [10]

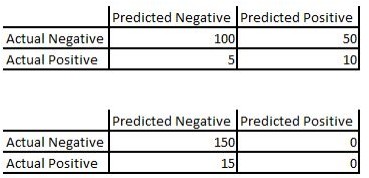
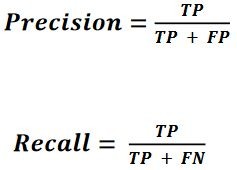


Figure 4: Confusion Matrix of a Model with Some Predictive Power (Top) and a Confusion Matrix of a Model with Zero Predictive Power (Bottom) as Items Are Always Classified as Part of the Negative Class.

Thus, all models are not suitable to be evaluated using accuracy. Accuracy is more suited for datasets that contain balanced positive and negative classes. For imbalanced datasets, other metrics such as precision and recall are more desirable [12] .Precision represents the amount of results that are relevant while recall is a measure of the amount of relevant results returned. A value of 1 is the highest possible for both measures, while 0 is the lowest measure. Both these values are dependent on the category being analyzed within the target class.Precision and recall has been given in the equation below. The concepts or precision and recall are illustrated in Figure 5.



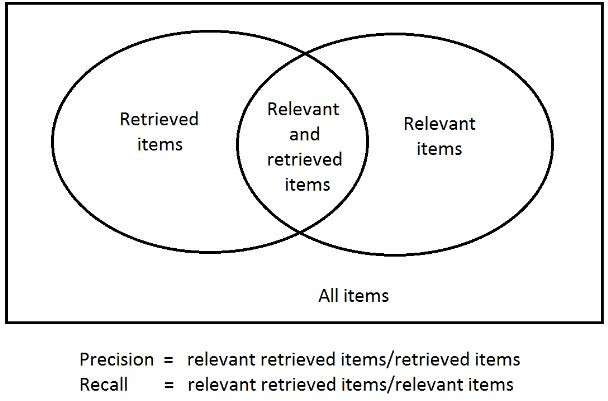
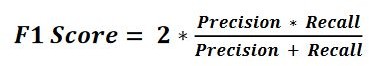


Figure 5: Precision and Recall

Precision says nothing about the data instances not correctly classified and recall says nothing about the data instances incorrectly labeled as the positive class. Thus both values are often examined as this information is more valuable. However it may be difficult to increase both values together. For example, if the TP of a minority class is increased the number of FP may also increase, which in turn reduces precision. [12] As a result, a single measure that is a combination of both measures is more ideal. This measure, known as the F1 score, is a harmonic mean of precision and recall where both precision and recall are weighted equally. The ideal classification algorithm will exhibit high precision, recall and F1 scores values. The equation for F1 score is shown below.



**CHAPTER 4**

# DATASET EXPLORATION

## Introduction

This chapter describes the information about various features and parameters present in the dataset. It also details the segmentation of features carried out for the easier understanding of the dataset. At the end, a comprehensive exploratory data analysis is performed using different data visualisation techniques for a better perspective about the dataset.

## Overview of Dataset

The data used in this project consists of a multivariate dataset from UCI machine learning repository. The dataset contains 303 instances with 76 attributes, but all published experiments refer to using subset 14 including target attribute .

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 303 |
| **Attribute Characteristics:** | Integer, Real | **Number of Attributes:** | 13 |
| **Associated Tasks:** | Classification | **Missing Values?** | N/A |

Table 2 : Snapshot about the Dataset

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
| 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 |

Table 3 : Snapshot of Raw Dataset

## Description of Features

A detailed description of the dataset attributes is presented below in Table 2. The 'Attribute' column contains the name of the feature followed by its 'Description'. The 'type' column tells us whether the data is categorical (CAT) or continous (CON) followed by its category which we have stated above. The last column tells us about the possible values that each feature can have.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Attribute | Description | Type | Values |
| 1 | age | age in years | CON | 1-100 |
| 2 | sex | male or female | CAT | 0,1 |
| 3 | cp | chestpain type | CAT | 0,1,2,3 |
| 4 | trestbps | resting blood pressure  (in mm Hg) | CON | 100-200 |
| 5 | chol | serum cholestoral in mg/dl | CON | 126-564 |
| 6 | fbs | fasting blood sugar > 120 mg/dl | CAT | 0,1 |
| 7 | restecg | resting electrocardiographic  results | CAT | 0,1,2 |
| 8 | thalach | maximum heart rate achieved | CON | 71-202 |
| 9 | exang | exercise induced angina | CAT | 0,1 |
| 10 | oldpeak | ST depression induced by  exercise relative to rest | CON | 0.0-6.2 |
| 11 | slope | slope of the peak exercise ST segment | CAT | 0.1.2 |
| 12 | ca | No. of major vessels colored by flourospy | CAT | 0,1,2,3,4 |
| 13 | thal | thalium stress result | CAT | 0,1,2,3 |
| 14 | target | Having heart disease or not | CAT | 0,1 |

Table 2: Description of features in the raw dataset

A few attributes are of the continuous or numerical type. These values will need to be discretized into a smaller number of categories.

## 4.4 Exploratory Data Analysis

Before preprocessing any form of data, it has to be understood at the first stage by using statistical techniques and visualization. In order to get a sense of data quality, we try to understand the major characteristics such as central tendency ( mean, median, standard deviation, mode etc). Apart from that we also make use of charts that aid us in understanding the data well.

### 4.4.1 Descriptive Statistics

For data, the tables column will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | count | mean | std | min | 25% | 50% | 75% | max |
| age | 303 | 54.36634 | 9.082101 | 29 | 47.5 | 55 | 61 | 77 |
| sex | 303 | 0.683168 | 0.466011 | 0 | 0 | 1 | 1 | 1 |
| cp | 303 | 0.966997 | 1.032052 | 0 | 0 | 1 | 2 | 3 |
| trestbps | 303 | 131.6238 | 17.53814 | 94 | 120 | 130 | 140 | 200 |
| chol | 303 | 246.264 | 51.83075 | 126 | 211 | 240 | 274.5 | 564 |
| fbs | 303 | 0.148515 | 0.356198 | 0 | 0 | 0 | 0 | 1 |
| restecg | 303 | 0.528053 | 0.52586 | 0 | 0 | 1 | 1 | 2 |
| thalach | 303 | 149.6469 | 22.90516 | 71 | 133.5 | 153 | 166 | 202 |
| exang | 303 | 0.326733 | 0.469794 | 0 | 0 | 0 | 1 | 1 |
| oldpeak | 303 | 1.039604 | 1.161075 | 0 | 0 | 0.8 | 1.6 | 6.2 |
| slope | 303 | 1.39934 | 0.616226 | 0 | 1 | 1 | 2 | 2 |
| ca | 303 | 0.729373 | 1.022606 | 0 | 0 | 0 | 1 | 4 |
| thal | 303 | 2.313531 | 0.612277 | 0 | 2 | 2 | 3 | 3 |
| target | 303 | 0.544554 | 0.498835 | 0 | 0 | 1 | 1 | 1 |

Table 3: Descriptive Summary of Dataset.

### 4.4.2 Exploratory Data Visualization

### 

Figure 1: From the above barplot we check the balance of the dataset. Bargraph visualisation

provides an Intuitive understanding that approximately patients have heart disease and approximately 140 People haven’t heart disease from the tottal 303 entries of the dataset.

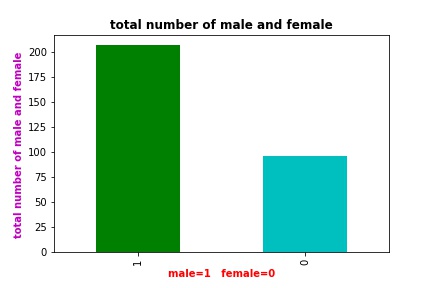


Figure2 : it provides us visulisation of total number of males and females in the dataset

As from above figure we have approximately 200 males and approximately 90 females

0 , 1 represents male and female respectively on the X-axis and Y-axis represents total

number of entries.

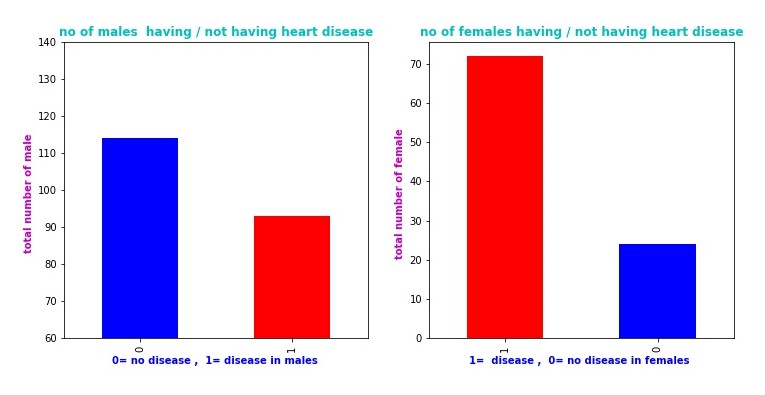


Figure 3 : Subplot1 provides us visulistaion of the of the total number of males having heart

disease among all the males in the dataset. Subplot2 provides us visulistaion of the of the

total number of females having heart disease among all the females in the dataset.

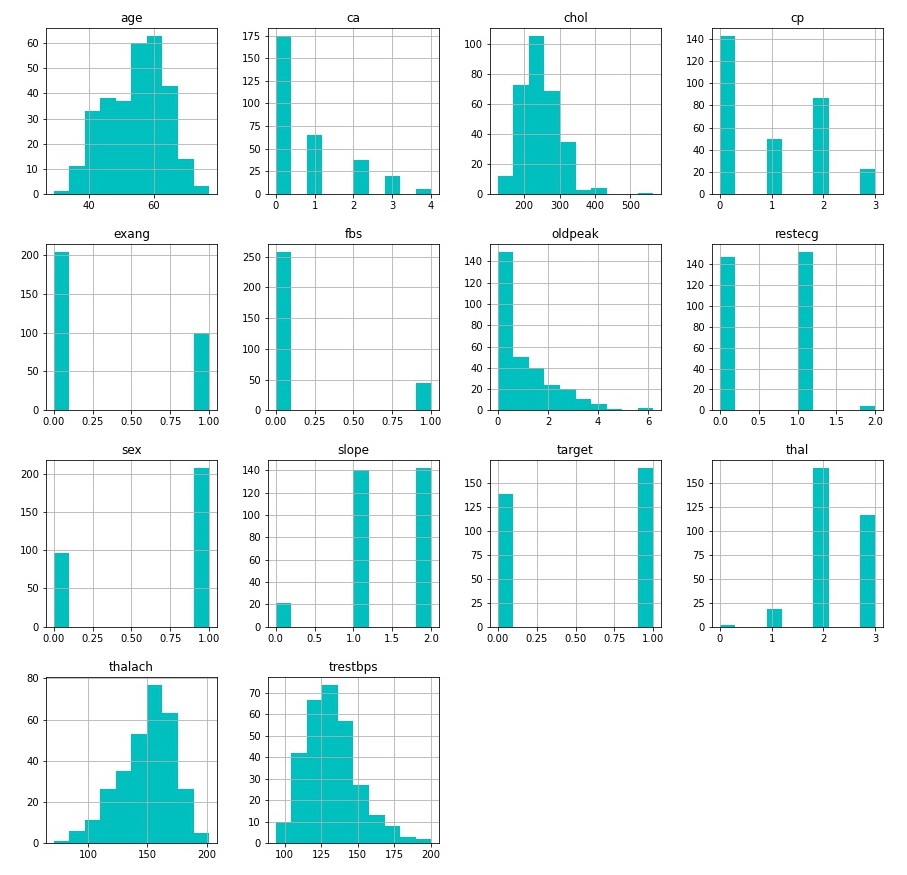


Figure 4 : Above subplots are the histogram plots , these shows us the number of times value of

each attribute has occurred , e,g attribute sex has two values i.e, 0 , 1 so from the sex subplot we

found that 0 has occurred 96 (0 represents females) times and 1 has occurred 207 times

(1 represents male).

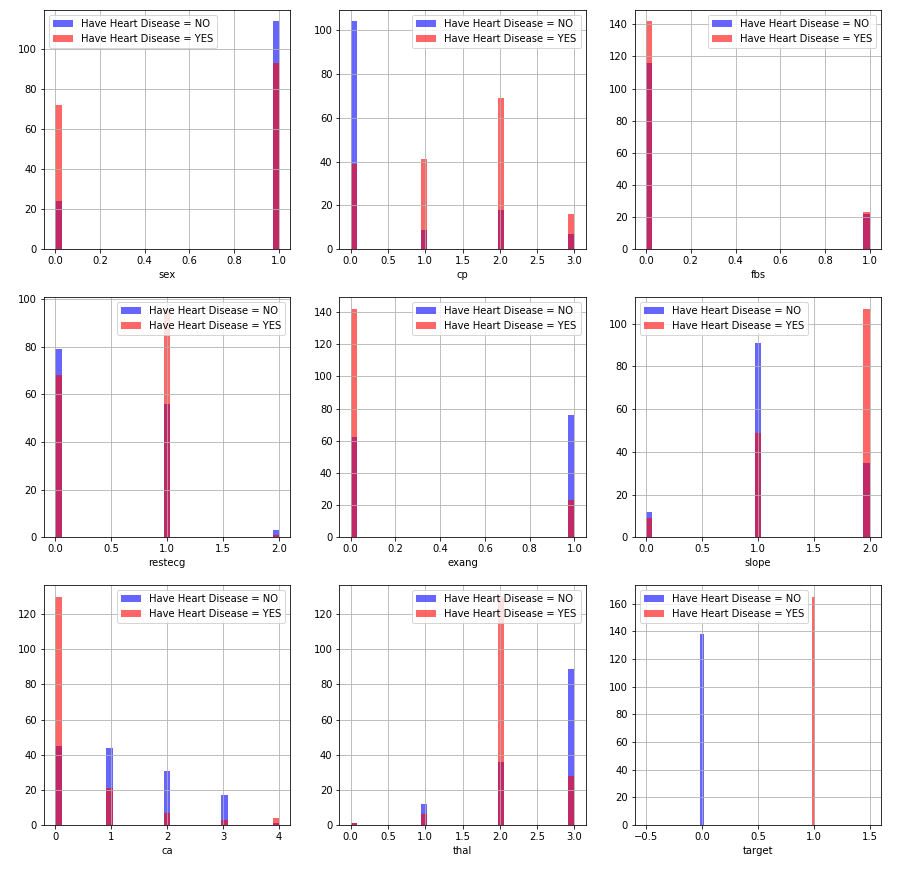


Figure 5 : Above subplots are genrated using histogram function in matplotlib and these are formed from

categorical attributes present in the dataset. Above Subplots visualizes that each attribute is plotted against

count of not having disease and having disease . Like in first subplot approximately 23, 110 females and

males respectively have not disease and 24 , 114 females and males respectively have heart disease and similarly same logic behind all subplots. Three colors in these subplots instead of two colors are because overlap of blue and red.

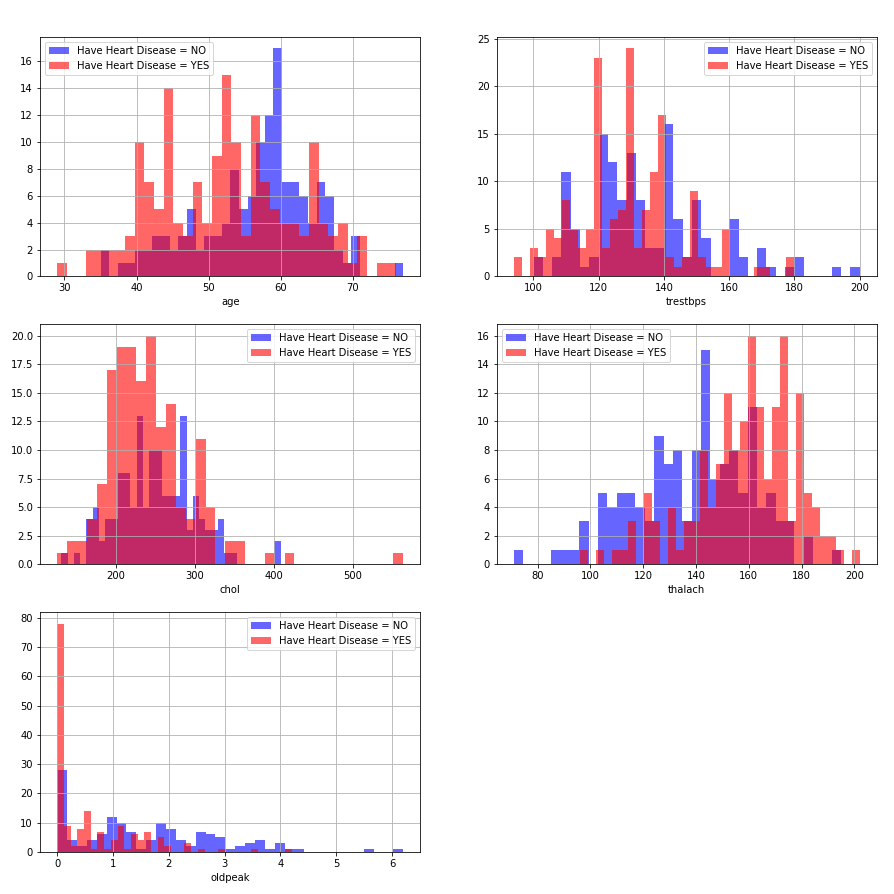


Figure 6: This is the plot of attributes having continuous values , it gives the similar insights as given

by Figure 5.

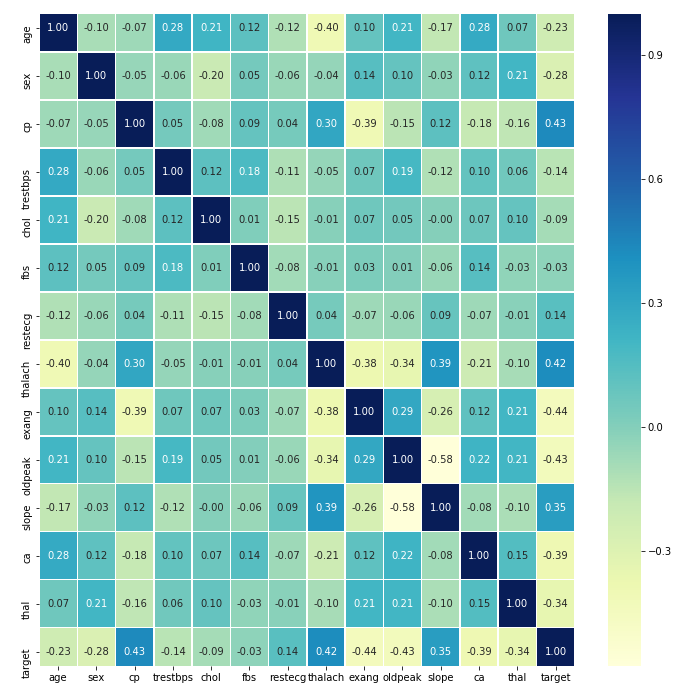


Figure 7: this figure is the heat map . First of all lets understand what a heat map is, According

to Wikipedia a heat map  is a data visualization technique that shows magnitude of a phenomenon

as color in two dimensions. In the above figure we visualize correlation between the attributes.

**Correlation** is a statistical technique that can show whether and how strongly pairs of variables

are related. For example correlation between same attributes like correlation between age

and age is 1 and similarly correlation between two different attributes is either positive or

negative ranging from -1 to +1. The main diagonal shows correlation =1 this means attributes are

fully related .

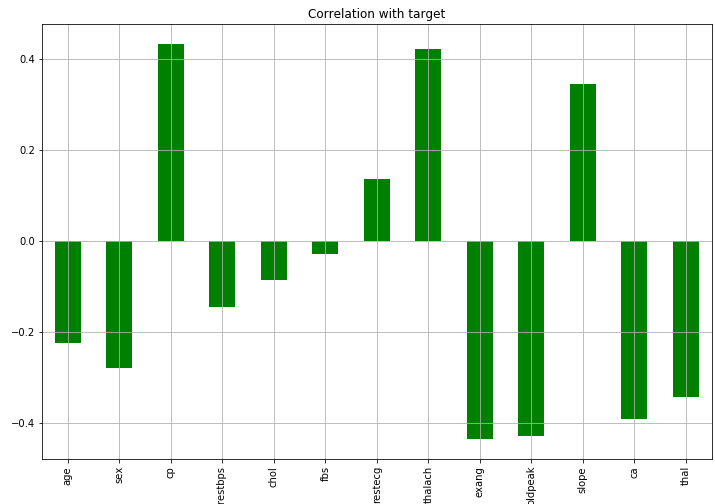


Figure 8 : It shows the correlation between the target and input attributes only **cp , thalach , restecg**

**slope** have positive correlation with **target** and **age, sex ,trestbps , chol , fbs, exang, oldpeak have**

**negative** correlation with **targe**t.

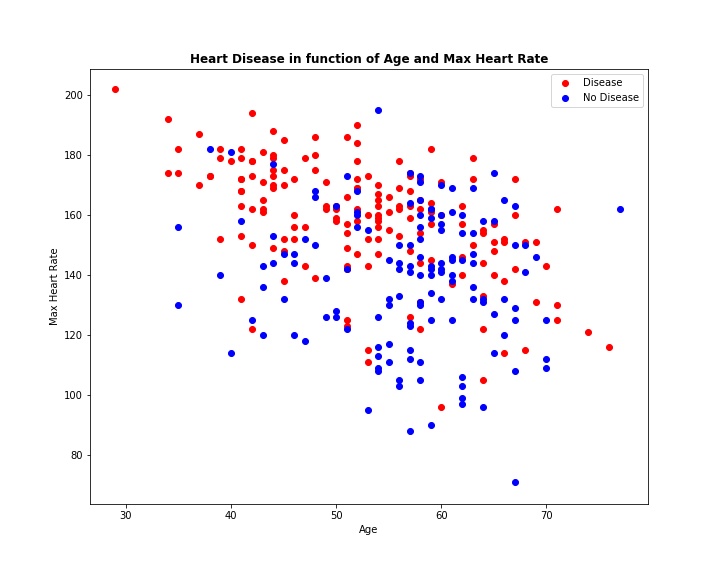


Figure 9: The above figure is scatter plot . A scatter plot is type of plot or mathematical diagram using

Cartesian coordinates to display values or typically two variables for a set of data.

**CHAPTER 5**

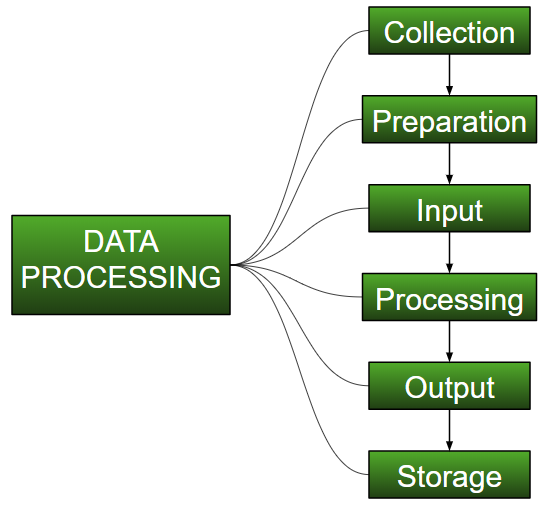
# IMPLEMENTATION

## Introduction

This chapter describes the data preprocessing methods employed in the project.

## Data Preprocessing Overview

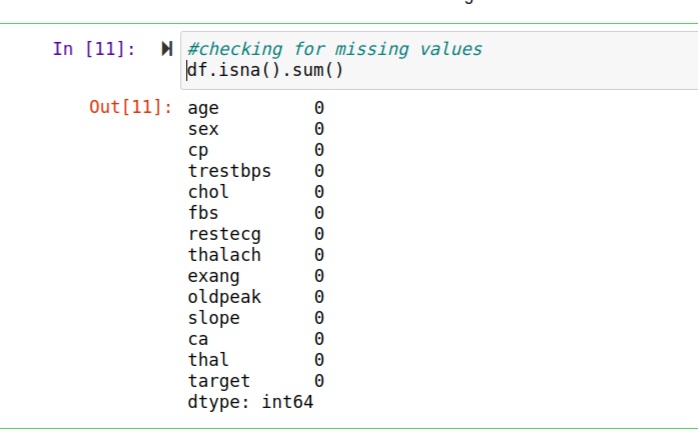
The steps involved in any Data science/Machine learning process generally include data collection, data preprocessing, model generation and evaluation. Data preprocessing is the important step in actual data analytics and aims at making sure that data is ready to be analysed. Data preprocessing modifies the dataset to improve quality and provide more meaningful inputs to the data model. Data preprocessing involves concepts such as Imputation , Standardization and one-Hot handlingand Multicollinearty.

Figure 32: Data Preprocessing

The original dataset was preprocessed to improve data quality. Preprocessing techniques of Standardization and One-hot encoding were applied on several attributes after analyzing the dataset. Since our dataset had no missing values there was no need of imputation.

**5.2.1 Imputation**

Imputation is simply the process of substituting the missing values of our dataset. We can do this by defining our own customised function or we can simply perform imputation by using the SimpleImputer class provided by sklearn. Since our dataset had no missing values there was no need to apply Imputation on our dataset.

Fig 33: total no of missing values in each featue.

From fig 33 it is evident that our dataset does not require Imutation.

**5.2.2 Standardization**

It is an integral preprocessing step. In Standardization, we transform our values such that the

they are scaled in some range.

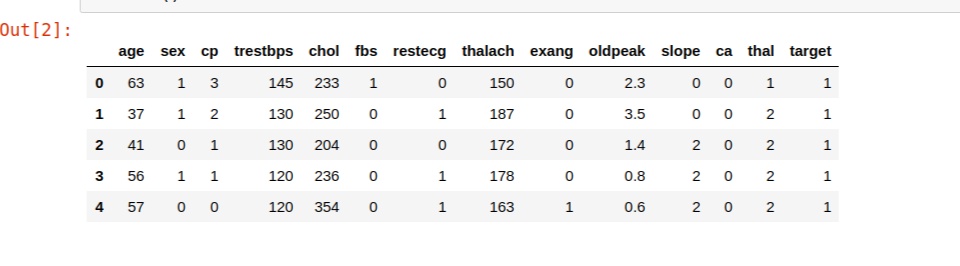
****

Fig 34: Dataset

Consider the above data frame, here we have 5 continuous values: age, tresbps, chol, thalach, and oldpeak. They are not on the same scale and since some are more likely to be greater than other features; therefore, our model will give more weightage to the features that have greater values , which is not the ideal scenario as other features are also an integral factor here. In order to avoid this issue, we perform Standardization.

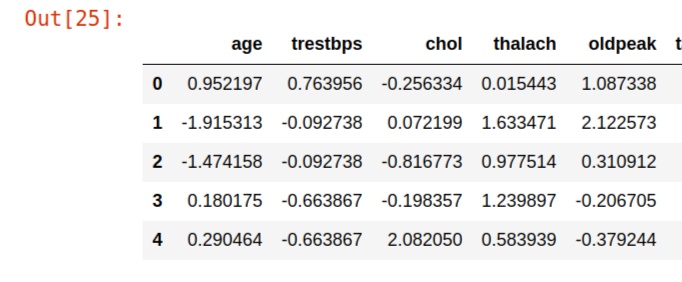


Fig: 35 Dataset after standardization.

### 5.2.3 One-Hot Encoding

So in One-Hot Encoding what we essentially do is that we create ’n’ columns where n is the number of unique values that the nominal variable can take.

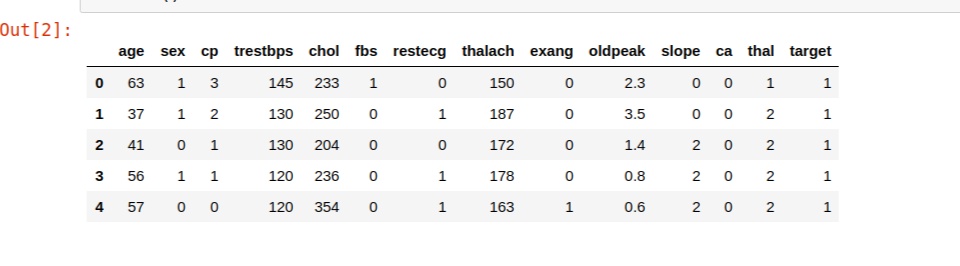


Fig 35: Dataset

Ex — Here if ca can take 0,1,2,3 then we will just create four new columns namely — ca\_0, ca\_1, ca\_2, ca\_3 and if ca\_0 is 1 then the values of other columns will be 0.

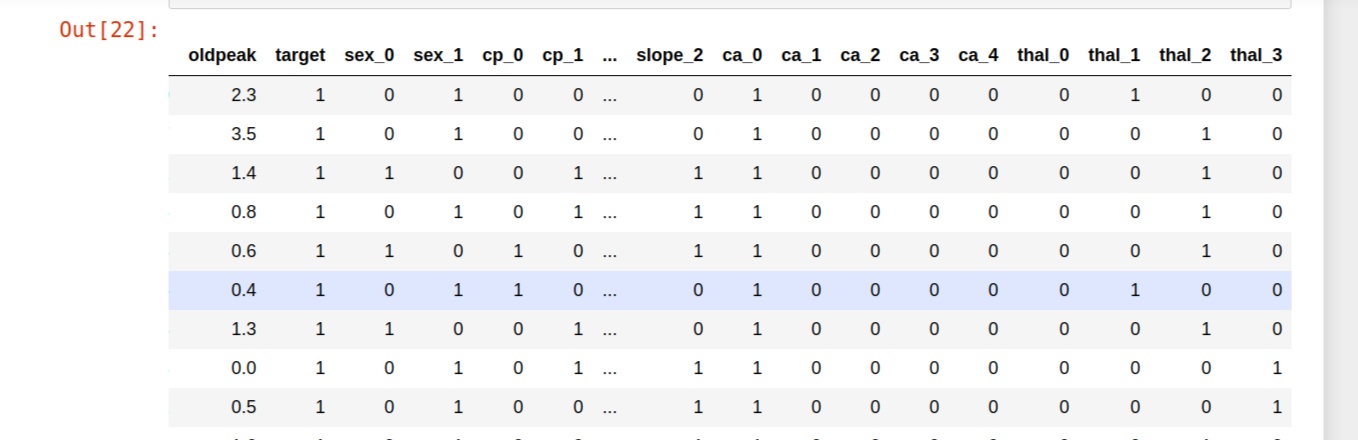


Fig 36: Dataset after one-hot encoding.

In this fig it is evident that out of the n columns, only one column can have value = 1 and the rest all will have value = 0.

## Establishing Feature Importance

By applying the *Select by Model* approach from the scikit-learn library in Python and choosing the Decision Tree Forest technique to apply the *feature\_importances* method we can establish how many features carry substantial information in the dataset and thus eliminate weak features. This will help in reducing the time complexity as well as some undesirable noise.

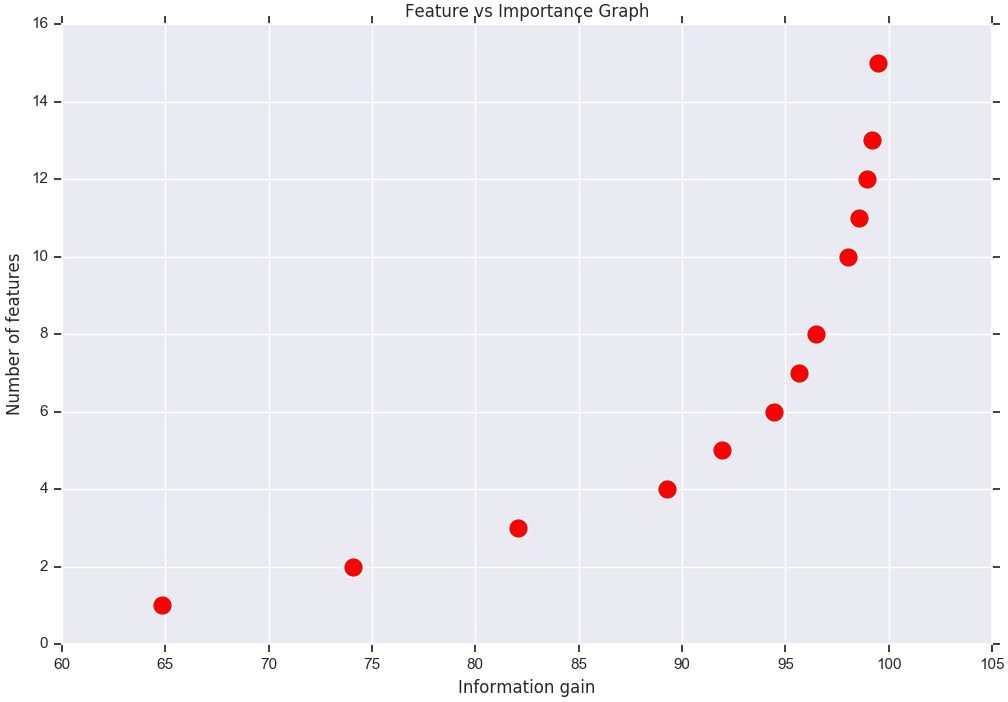


Figure 33:: Number of features versus information gain

The above graph describes the increase in information gain corresponding to the increase in the number of features. From the above graph it is inferred that among all the features in the dataset, there are 7 features that contribute 95.66 % of the total information which is a noteworthy result considering the amount of time that can be

saved by skipping the other features that do not contribute substantially to the information gain.

The detailed information about these 7 important features is further detailed in the following figure 34.

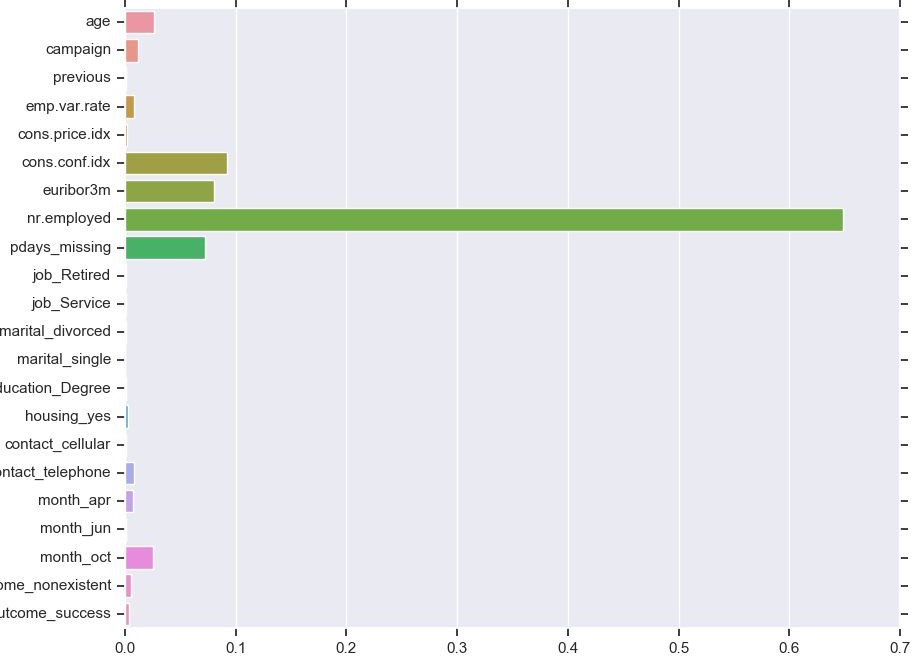


Figure 34: Contribution of features to output

From the above figure 34 we can observe the contribution of each feature to the total information output. We can thus zero-in and select only those features of importance that actually contribute substantially to our output.

**CHAPTER 6**

# RESULTS AND COMPARISON

## Introduction

This chapter presents the results and findings of the project while also providing some comprehensive comparisons and analyses.

## Classification Results

The following table presents a detailed comparison between different classification algorithms implemented in the project based on their performance when evaluated with a diverse group of evaluation parameters. In the precision column the least performance is exhibited by Random forest , Decision tree and xgboost algorithms while SVM perform the best in this criteria. It is evident from this fig that SVM outperformes every other Algorithm on precision,F1score and accuracy evaluation parameters while logistic regression does the same for recall score.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithms | Precision | Recall | F1 score | Accuracy |
| Logistic Regression | 85.91% | 90.43% | 88.14% | 86.81% |
| KNN | 88% | 88% | 88% | 86.81% |
| Decision Tree | 84% | 74% | 78.92% | 78.02% |
| Random Forest | 84% | 84% | 84% | 82.42% |
| SVM | 89% | 88% | 88.89% | 87.91 |
| XGBOOST | **84%** | **84%** | **84%** | **82.42%** |

Fig 38: Comparison of different algorithms without hyperparameter tuning.

The Fig 38 shows performance of various Algorithms on basis of there precision, recall, f1 and accuracy score without hyperparameter tuning. What can be easily deducted from the figure is that SVM is bettter Algorithm(without hyperparameter tuning ) for our dataset in comparison to other algorithms used in this project.

## 6.3 Impact of Hyperparameter Tuning

The following table describes the change in the results of various classification algorithms when they are applied with hyperparameter tuning. A positive change occurs in performance of all the algorithms and except SVM where its precision, f1 and accuracy score are reduced to 84.62%, 86.27% and 84.62% respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithms | Precision | Recall | F1 | Accuracy |
| Logistic Regression | 86.54% | 90% | 88.24% | 86.81% |
| KNN | 87.04% | 94% | 90.38% | 89.01% |
| Decision Tree | 87.50% | 84% | 85.71% | 84.62% |
| Random Forest | 84.31% | 86% | 85.15% | 83.52% |
| SVM | 84.62% | 88% | 86.27% | 84.62% |
| **XGBOOST** | **84.31%** | **86%** | **85.15%** | **83.52%** |

Fig 39: Comparison of different algorithms with hyperparameter tuning.

By observing the above figure it is evident that among all the classification algorithms the maximum change in performance is for KNN.

**6.4 Voting Classifier**

The following table describes the result of various classification algorithms when they are ensembled using Voting Classifier:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F1 | Accuracy |
| **Voting Classifier** | **85.19%** | **92%** | **88.46%** | **86.81%** |

Fig 40: Performance of Voting Classifier.

By observing above there figures its evident that KNN with hyperparameter tuning is the best algorithm for the dataset used in our project.

**CHAPTER 7**

# CONCLUSION AND FUTURE SCOPE

## Conclusion and Future Scope:

With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. The motivation for the study was to find the most efficient ML algorithm for detection of heart diseases. The data analysis works undertaken in the project provide meaningful insights regarding the raw as well as the engineered dataset related to the problem. This study compares the accuracy ,Precision, confusion matrics and f1 score of Decision Tree, Logistic Regression, Random Forest, K Nearest Neighbor, XGBOOST, Support Machine Vector with Hperparameter tuning and cross validation and at last ensembling all of the algorithms using Voting Classifier for predicting heart disease using UCI machine learning repository dataset. The result of this study indicates that the K Nearest Neighbor algorithm is the most efficient algorithm with accuracy, Precision, Recall and F1 score of 89.01%, 83.33%, 88.24%, and 85.71% respectively for prediction of heart disease. In future the work can be enhanced by developing a web application based on the K Nearest Neighbor algorithm as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting the heart disease effectively and efficiently.

# REFERENCES

## References

1.Avinash Golande, Pavan Kumar T, Heart Disease Prediction Using Effective Machine Learning Techniques, International Journal of Recent Technology and Engineering, Vol 8, pp.944-950,2019

2.T.Nagamani, S.Logeswari, B.Gomathy, Heart Disease Prediction using Data Mining with Mapreduce Algorithm, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-3, January 2019.

3. Fahd Saleh Alotaibi, Implementation of Machine Learning Model to Predict Heart Failure Disease, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 6, 2019.

4. Anjan Nikhil Repaka, Sai Deepak Ravikanti, Ramya G Franklin, Design And Implementation Heart Disease Prediction Using Naives Bayesian, International Conference on Trends in Electronics and Information(ICOEI 2019).

5. Nagaraj M Lutimath,Chethan C,Basavaraj S Pol.,Prediction Of Heart Disease using Machine Learning, International journal Of Recent Technology and Engineering,8,(2S10), pp 474-477, 2019.

6. Jafar Alzubi, Anand Nayyar, Akshi Kumar. "Machine Learning from Theory to Algorithms: An Overview", Journal of Physics: Conference Series, 2018.

7. Theresa Princy R,J. Thomas,’Human heart Disease Prediction System

using Data Mining Techniques’, International Conference on Circuit

Power and Computing Technologies,Bangalore,2016

8. Nagaraj M Lutimath,Chethan C,Basavaraj S Pol.,’Prediction Of Heart

Disease using Machine Learning’, International journal Of Recent

Technology and Engineering,8,(2S10), pp 474-477, 2019.

9. Fajr Ibrahem Alarsan., and Mamoon Younes ‘Analysis and

classification of heart diseases using heartbeat features and machine

learning algorithms’,Journal Of Big Data,2019;6:81.

10. Sonam Nikhar, A.M. Karandikar “Prediction of Heart Disease Us-

ing Machine Learning Algorithms” in International Journal of Ad-

vanced Engineering, Management and Science (IJAEMS) June-

2016 vol-2

11.Deeanna Kelley “Heart Disease: Causes, Prevention, and Current

Research” in JCCC Honors Journal.

12.Internet source [Online].Available (Accessed on May 1 2020):

http://acadpubl.eu/ap.

# APPENDIX

## Decison Tree Classifier:

from sklearn.tree import DecisionTreeClassifier

params = {“criterion”:(“gini”, “entropy”),

“splitter”:(“best” , “random”),

“max\_depth”:(list(range(1,20))),

“min\_samples\_split”:[2,3,4],

“min\_samples\_leaf”:list(range(1,20))

}

tree\_tuned=DecisionTreeClassifier(random\_state=42)

grid\_search\_cv = GridSearchCV(tree\_tuned, params, scoring= “accuracy”,n\_jobs= -1,

verbose = 1, cv=3, iid=True)

grid\_search\_cv.fit(X\_train,y\_train)

grid\_search\_cv.best\_estimator\_

tree\_tuned=DecisionTreeClassifier(criterion=“gini”

max\_dept=3,

min\_samples\_leaf=2,

min\_samples\_split=2,

splitter= ‘random’)

tree\_tuned.fit(X\_train, y\_train)

print\_score(tree\_tuned, X\_train, y\_train, X\_test, y\_test,train=True)

print\_score(tree\_tuned, X\_train, y\_train, X\_test, y\_test,train=False)

## Logistic Regression:

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import GridSearchCV

params = {“C”: np.logspace(-4,4,20), “solver”: [“liblinear]}

log\_reg\_tuned =LogisticRegression()

grid\_search\_cv = GridSearchCV(log\_reg\_tuned, params, scoring= “accuracy”,

n\_jobs= -1, verbose = 1, cv=5, iid= True)

grid\_search\_cv.fit(X\_train,y\_train)

grid\_search\_cv.best\_estimator\_

**Random Forest Classifier:**

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import RandomizedSearchCv

n\_estimators = [int(x) for x in np.linspace(start=200,stop=2000,num=10)]

max\_features = [‘auto’ , ‘sqrt’]

max\_depth = [int(x) for x in np.linspace(10,110, num=11)]

max\_depth.append(None)

min\_samples\_split = [2, 5, 10]

min\_samples\_leaf = [1, 2, 4]

bootstrap = [True, False]

random\_grid = {‘ n\_estimators’ : n\_estimators, ‘max\_features’:max\_features,

‘max\_depth’: max\_depth, ‘min\_samples\_split’: min\_samples\_split,

‘min\_samples\_leaf’: min\_samples\_leaf, ‘bootstrap’: bootstrap}

random\_forest\_tuned = RandomForestClassifier(random\_state=42)

rf\_random = RandomizedSearchCV(estimator=rand\_forest\_tuned,param\_distributions=ra ndom\_grid,

n\_iter = 100, cv=3,verbose =2, random\_s tate=42, n\_job s=-1)

rf\_random.fit(X\_train,y\_train)

rf\_random.best\_estimator\_

random\_forest\_tuned = RandomForestClassifier(bootstrap= True,

max\_dpth=70,

max\_features= ‘auto’

min\_samples\_leaf=4,

min\_samp les\_split=10,

n\_estim ators=400)

random\_forest\_tuned.fit(X\_train, y\_train)

print\_score(rand\_forest\_tuned, X\_train, y\_train, X\_test, y\_test, train=True)

print\_score(rand\_forest\_tuned, X\_train, y\_train, X\_test, y\_test, train=False)

**K Nearest-Neighbours:**

from sklearn.neighbors import KneighborsClassifier

train\_score = []

test\_score = []

neighbors = range(1,21)

for k in neighbors:

model\_tuned = KneighborsClassifier(n\_neighbors=k)

model\_tuned.fit(X\_train,y\_train)

train\_score.append(accuracy\_score(y\_train,model\_tuned. predict(x\_train)))

test\_score.append(accuracy\_score(y\_test, model\_tuned.predict(X\_test)))

plt.figure(figsize=(12,8))

plt.plot(neighbors,train\_score,label= ‘Train score’)

plt.plot(neighbors,test\_score,label= ‘Test score’)

plt.xticks(np.arange(1,21,1))

plt.xlabel(‘Numbet of neighbors’)

plt.ylabel(‘Model score’)

plt.legend()

print(f “Maximum KNN score on the test data: {max(test\_score)\*100: .2f}%”}

knn\_classifier = KneighborsClassifier(n\_neighbors=19)

knn\_classifier.fit(X\_train,y\_train)

print\_score(knn\_classifier, X\_train, y\_train, X\_test,y\_test,train=True)

print\_score(knn\_classifier, X\_train, y\_train, X\_test,y\_test,train=False)

**Support Vector Machine:**

from sklearn.svm import SVC

svm\_model =SVC(kernel= ‘rbf’ ,gamma=0.1,C= 1.0)

params = { “C”: (0.1, 0.5, 1, 2, 5, 10, 20),

“gamma”: (0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1),

“kernel”: (‘linear’, ‘poly’, ‘rbf’)}

svm\_grid = GridSearchCV(svm\_model, params, n\_jobs= -1, cv=5, verbose=1,scoring=“accuracy”)

svm\_grid.fit(X\_train,y\_train)

svm\_grid.best\_estimator\_

svm\_model = SVC( C=5, gamma= 0.01, kernel= ‘rbf’)

svm\_model.fit(X\_train,y\_train)

print\_score(svm\_model, X\_train, y\_train, X\_test,y\_test,train=True)

print\_score(svm\_model, X\_train, y\_train, X\_test,y\_test,train=False)

**XG XGBOOST:**

fr from xgboost import XGBClassifier

n\_ n\_estimators = [100, 500, 900, 1100, 1500]

ma max\_depth = [2, 3, 5, 10, 15]

bo booster = ['gbtree', 'gblinear']

ba base\_score = [0.25, 0.5, 0.75, 0.99]

le learning\_rate = [0.05, 0.1, 0.15, 0.20]

mi min\_child\_weight = [1, 2, 3, 4]

h hyperparameter\_grid = {'n\_estimators': n\_estimators, 'max\_depth’:

max\_depth,

'learning\_rate' : learning\_rate,'min\_child\_weight’

min\_child\_weight,

'booster' : booster, 'base\_score' : base\_score

}

x xgb\_model = XGBClassifier()

xgb\_cv = RandomizedSearchCV(estimator=xgb\_model,

param\_distributions=hyperparameter\_grid,

cv=5, n\_iter=650, scoring = 'accuracy',n\_jobs =-1, iid=True,

verbose=1, return\_train\_score = True, random\_state=42)

xgb\_cv.fit(X\_train, y\_train)

xgb\_cv.best\_estimator\_

xgb\_best\_tuned = XGBClassifier(base\_score=0.25,

booster='gbtree',

learning\_rate=0.05,

max\_depth=5,

min\_child\_weight=2,

n\_estimators=100)

xgb\_best\_tuned.fit(X\_train, y\_train)

print\_score(xgb\_best\_tuned, X\_train, y\_train, X\_test, y\_test,

train=True)

print\_score(xgb\_best\_tuned, X\_train, y\_train, X\_test, y\_test,

train=False)

**Vo Voting Classifier:**

vo voting\_clf = VotingClassifier( estimators=[

('lr',log\_reg\_tuned ),

('knn',knn\_classifier ),

('dt',tree\_tuned),

('rf',rand\_forest\_tuned),

('xgb',xgb\_best\_tuned),

],

voting ='hard')

v voting\_clf.fit(X\_train,y\_train)

print\_score(voting\_clf, X\_train, y\_train, X\_test, y\_test, train=True)

print\_score(voting\_clf, X\_train, y\_train, X\_test, y\_test,train=False)

**Reading dataset:**

df = pd.read\_csv(r"location/heart.csv")

**Description of data:**

df.describe()

**Seperating categorical and continuous values:**

categorical\_val = []

continous\_val = []

for column in df.columns:

print('==============================')

print(f"{column} : {df[column].unique()}")

if len(df[column].unique()) <= 10:

categorical\_val.append(column)

else:

continous\_val.append(column)

**Co-relation between each feature:**

numeric\_columns=['trestbps','chol','age','oldpeak','thalach','cp','fbs','sex','restecg','ca','exang','slope','thal','target']

sns.heatmap(df[numeric\_columns].corr(),annot=True,cmap='terrain',linewidth=0.1)

fig= plt.gcf()

fig.set\_size\_inches(13,7)

plt.show()

**Co-relation of features with Target:**

df.drop('target', axis=1).corrwith(df.target).plot(kind='bar', grid=True, figsize=(12, 8),

title="Correlation with target")

### Creating a dummy columns for categorical variables:

### dataset = pd.get\_dummies(df, columns = categorical\_val)

### Scaling features:

### s\_sc = StandardScaler()

### col\_to\_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']

### dataset[col\_to\_scale] = s\_sc.fit\_transform(dataset[col\_to\_scale])

### Splitting dataset into train and test set:

### from sklearn.model\_selection import train\_test\_split

### X = dataset.drop('target', axis=1)

### y = dataset.target

### X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=42)

**Performance of different Algorithms:**

from sklearn.metrics import accuracy\_score, confusion\_matrix, precision\_score, recall\_score, f1\_score

train\_score\_1={}

f1\_train1={}

precision\_train1={}

recall\_train1={}

test\_score\_1={}

f1\_test1={}

precision\_test1={}

recall\_test1={}

def print\_score(algo\_name,clf, X\_train, y\_train, X\_test, y\_test, train=True):

if train:

pred = clf.predict(X\_train)

print("Train Result:\n================================================")

print(f"Accuracy Score: {accuracy\_score(y\_train, pred) \* 100:.2f}%")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("Classification Report:", end='')

print(f"\tPrecision Score: {precision\_score(y\_train, pred) \* 100:.2f}%")

print(f"\t\t\tRecall Score: {recall\_score(y\_train, pred) \* 100:.2f}%")

print(f"\t\t\tF1 score: {f1\_score(y\_train, pred) \* 100:.2f}%")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(f"Confusion Matrix: \n {confusion\_matrix(y\_train, pred)}\n")

train\_score\_1["Training score "+algo\_name]=accuracy\_score(y\_train, pred) \* 100

f1\_train1[algo\_name+" training f1 score "]=f1\_score(y\_train, pred) \* 100

precision\_train1[algo\_name+" training precision score"]=precision\_score(y\_train, pred) \* 100

recall\_train1[algo\_name+" training recall score"]= recall\_score(y\_train, pred) \* 100

elif train==False:

pred = clf.predict(X\_test)

print("Test Result:\n================================================")

print(f"Accuracy Score: {accuracy\_score(y\_test, pred) \* 100:.2f}%")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print("Classification Report:", end='')

print(f"\tPrecision Score: {precision\_score(y\_test, pred) \* 100:.2f}%")

print(f"\t\t\tRecall Score: {recall\_score(y\_test, pred) \* 100:.2f}%")

print(f"\t\t\tF1 score: {f1\_score(y\_test, pred) \* 100:.2f}%")

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print(f"Confusion Matrix: \n {confusion\_matrix(y\_test, pred)}\n")

test\_score\_1["Testing score"+algo\_name]=accuracy\_score(y\_test, pred) \* 100

f1\_test1[algo\_name+" testing f1 score"]=f1\_score(y\_test, pred) \* 100

precision\_test1[algo\_name+" testing precision score"]=precision\_score(y\_test, pred) \* 100

recall\_test1[algo\_name+" testing recall score"]= recall\_score(y\_test, pred) \* 100