**PREDICTIVE ANALYTICS FOR CUSTOMER TARGETING:**

**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.

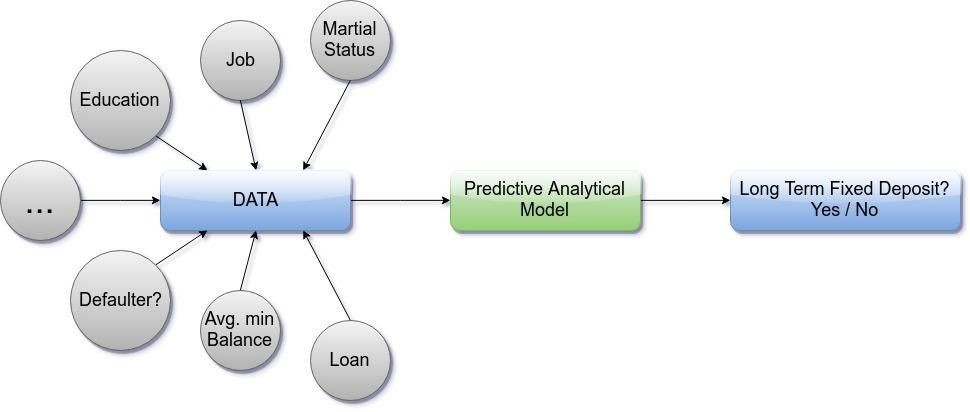


Figure 1: Overview of the problem

## 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. Because of the pertinent nature of the problem in understanding skewed class problems and its importance in designing complex decision support systems many researchers have approached the problem with diverse methods, techniques and have presented tremendous results and prediction models for the same.Moro, Cortez and Rita [13] used this bank dataset in addition to an external dataset to determine the best set of features and analyze different data mining models on the term deposit subscription class. Research was conducted by first combining the dataset with statistical data from a website belonging to the central bank of the Portuguese Republic. This external dataset allowed the inclusion of bank client information, product information as well as data related to social and economic information. With the combination of the two datasets, a total of 150 features were created. Feature selection was performed on different sets of features and compared by two metrics including area of the receiver operating characteristic curve (AUC) and area of the LIFT cumulative curve (ALIFT). The feature set of 22 features was used for further analysis to compare four algorithms. Logistic regression, decision trees, SVMs and neural networks were applied on the reduced set of features and the results showed 18 that neural networks had the best value with an AUC value of 0.8 and ALIFT value of 0.7.

The term deposit subscription feature of this dataset was also analyzed using a combination of business intelligence (BI) and data mining techniques. According to Moro, Laureano and Cortez [14] “BI is an umbrella term that includes architectures, tools, databases, applications and methodologies with the goal of using data to support decisions of business managers”. The Cross-Industry Standard Process for Data Mining (CRISP-DM) model was used. This methodology defines the process of

generating a model that can be used for predicting in real life. It has six phases which include business understanding, data understanding, data preparation, modeling, evaluation and deployment. The business understanding phase is used to define a business goal which needs to be achieved by generating a predictive model. The data understanding, data preparation, modeling and evaluation phases are similar to the data collection and preprocessing, model creation, and analysis phases followed in a typical data mining process. The last phase of this step is deployment of the model in the real world. Based on the application of the CRISP-DM methodology, SVM displayed the highest predictive power as compared to naïve bayes and decision trees when measured using AUC and ALIFT.

Another paper by Elsalamony [15] used the dataset with the goal of determining influencing attributes on the term deposit subscription attribute. The algorithms used were multilayer perception neural network (MLPNN), Bayesian Networks, Logistic Regression, and C5.0. The metrics used for analysis included classification accuracy, sensitivity, and specificity. The results showed that the duration of the last conversation was the most influencing factor on success of the client’s subscription to the term deposit, Logistic Regression, and MLPNN. According to Bayesian Networks the most influencing attribute was the client’s age.

**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.

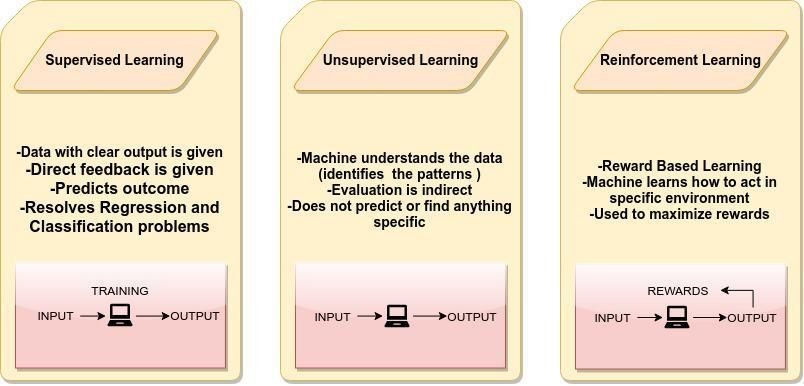


Figure 2: Types of Machine learning techniques

## 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].

### Extra Tree

The Extra-Tree method (standing for extremely randomized trees) was proposed with the main objective of further randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree. With respect to random forests, the method drops the idea of using bootstrap copies of the learning sample, and instead of trying to find an optimal cut-point for each one of the K randomly chosen features at

each node, it selects a cut-point at random. This idea is rather productive in the context of many problems characterized by a large number of numerical features varying more or less continuously: it leads often to increased accuracy thanks to its smoothing and at the same time significantly reduces computational burdens linked to the determination of optimal cut-points in standard trees and in random forests. From a statistical point of view, dropping the bootstrapping idea leads to an advantage in terms of bias, whereas the cut-point randomization has often an excellent variance reduction effect. This method has yielded state-of-the-art results in several high-dimensional complex problems. From a functional point of view, the Extra-Tree method produces piece-wise multilinear approximations, rather than the piecewise constant ones of random forests. [8]

### AdaBoost

AdaBoost, short for Adaptive [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm)), is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [meta-algorithm](https://en.wikipedia.org/wiki/Meta-algorithm) . It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and [outliers](https://en.wikipedia.org/wiki/Outlier). In some problems it can be less susceptible to the [overfitting](https://en.wikipedia.org/wiki/Overfitting_(machine_learning)) problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction

### 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]

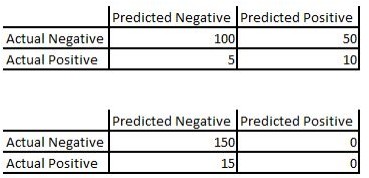
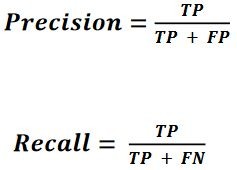


Figure 4: Confusion Matrix of a Model with Some Predictive Power (Top) and a Confusion Matrix of a Model with Zero Predictive Power (Bottom) as Items Are Always Classified as Part of the Negative Class.

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.



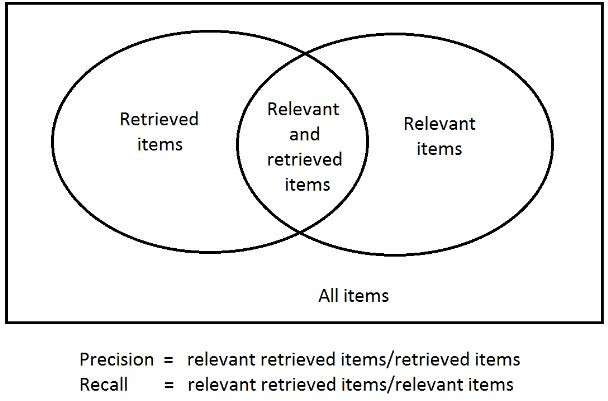
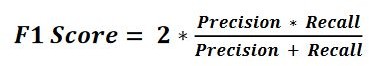


Figure 5: Precision and Recall

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:** | 233 |
| **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.

## 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.

### 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.

### Exploratory Data Visualizations

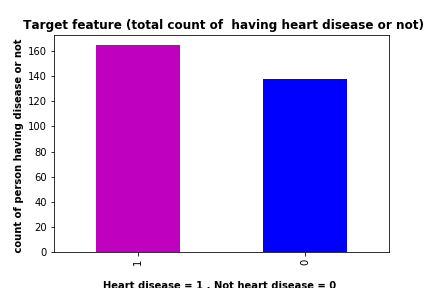


Figure 1: From the above barplot we check the balance of the dataset. Bargraph visualisation

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

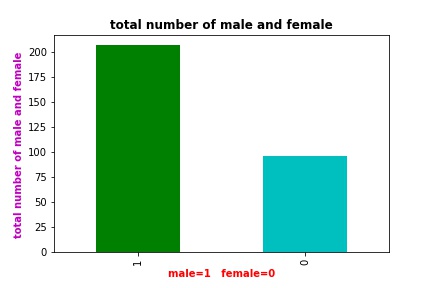


Figure2 : it provides us visulisation of total number of males and females in the dataset

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.

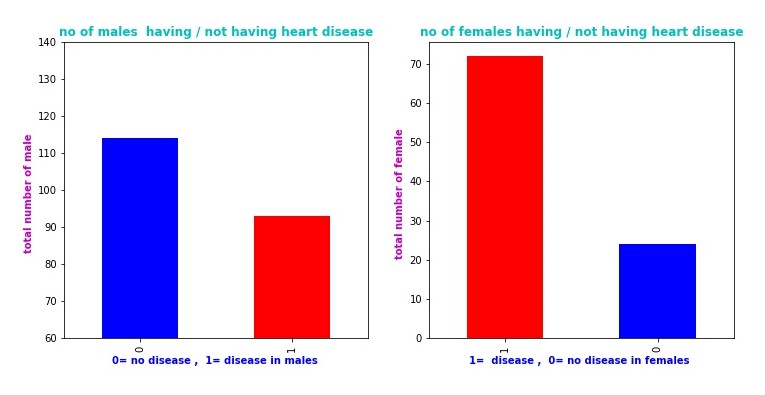


Figure 3 : Subplot1 provides us visulistaion of the of the total number of males having heart

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.

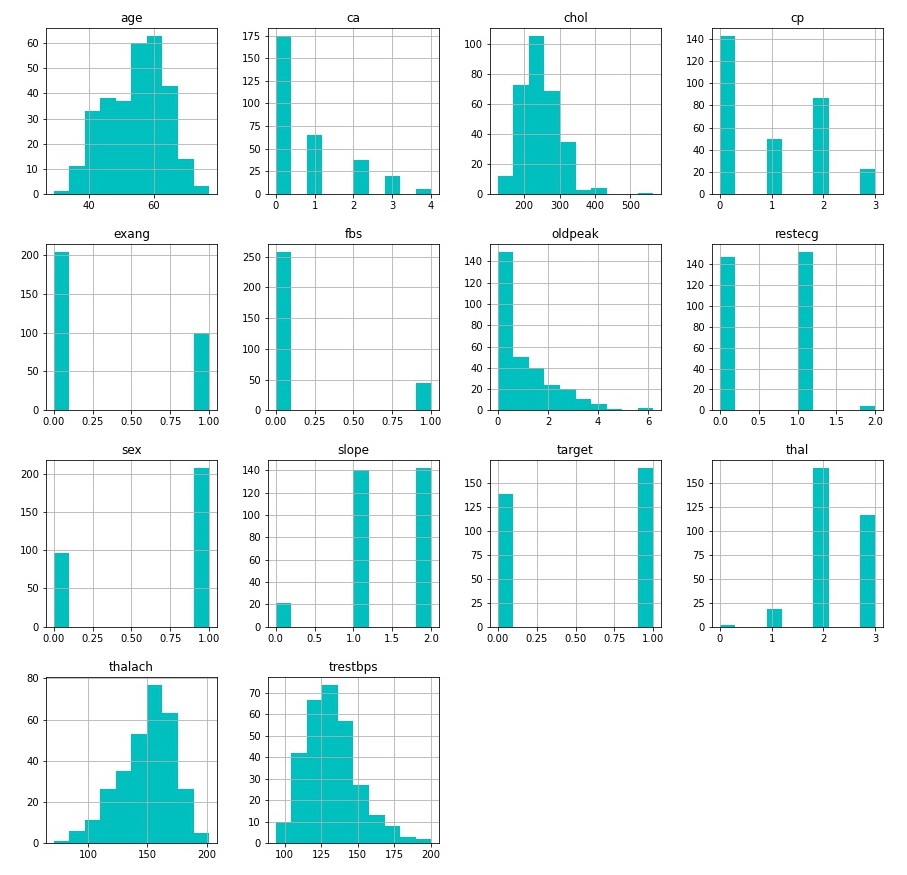


Figure 4 : Above subplots are the histogram plots , these shows us the number of times value of

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).

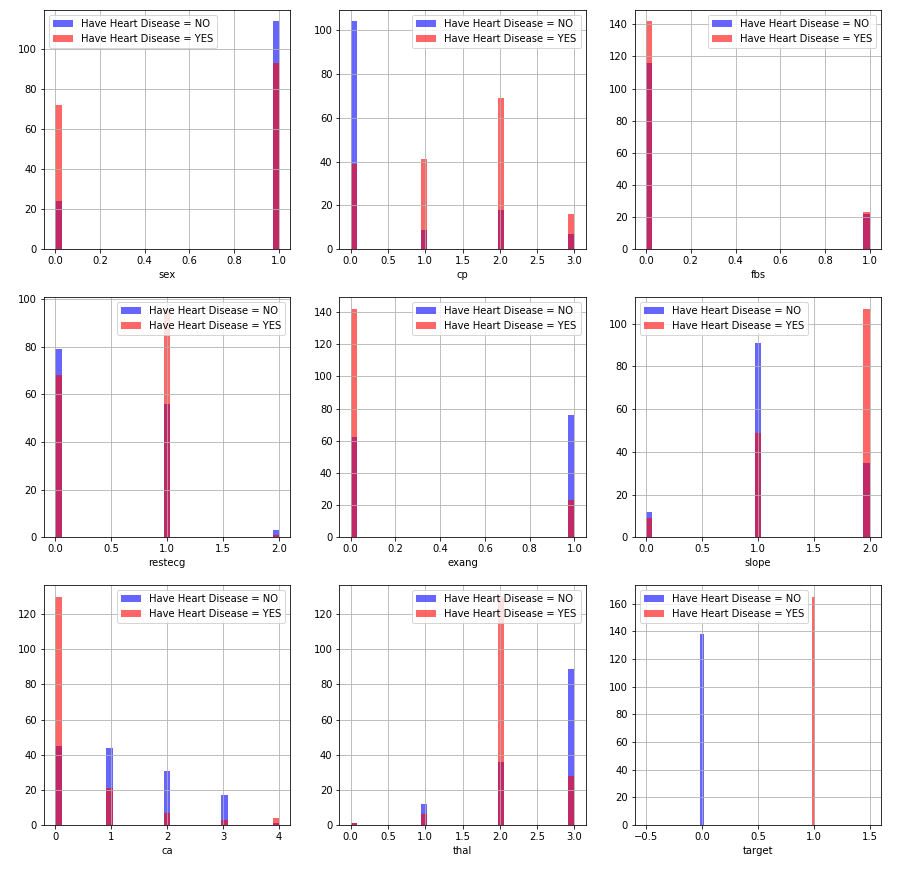
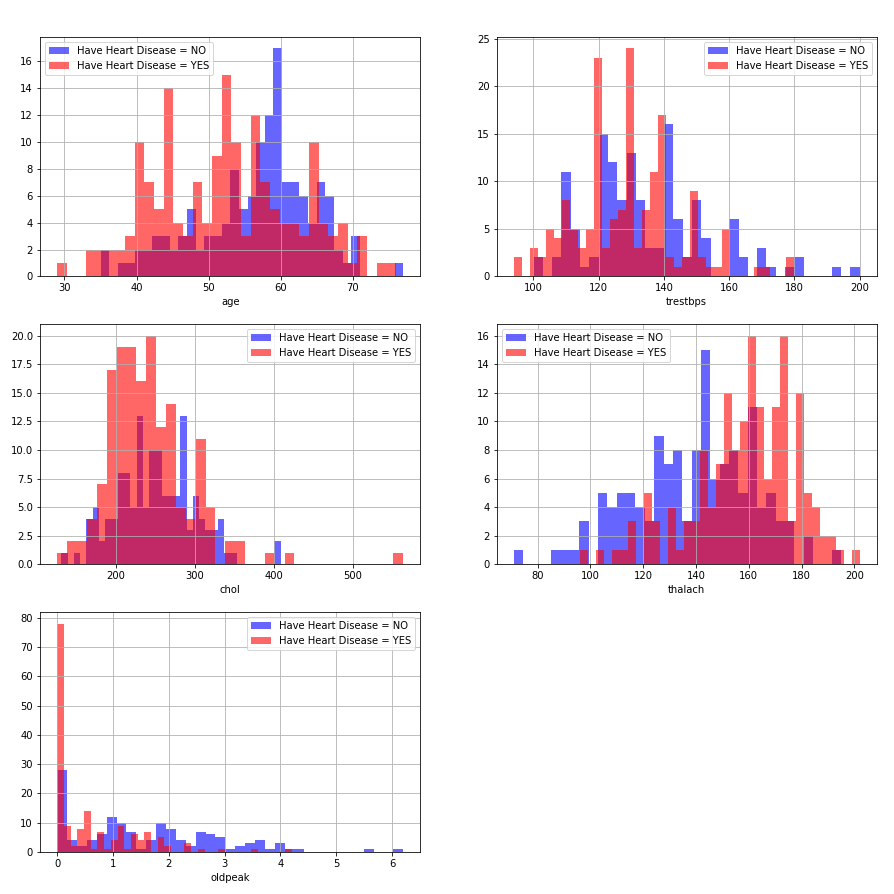
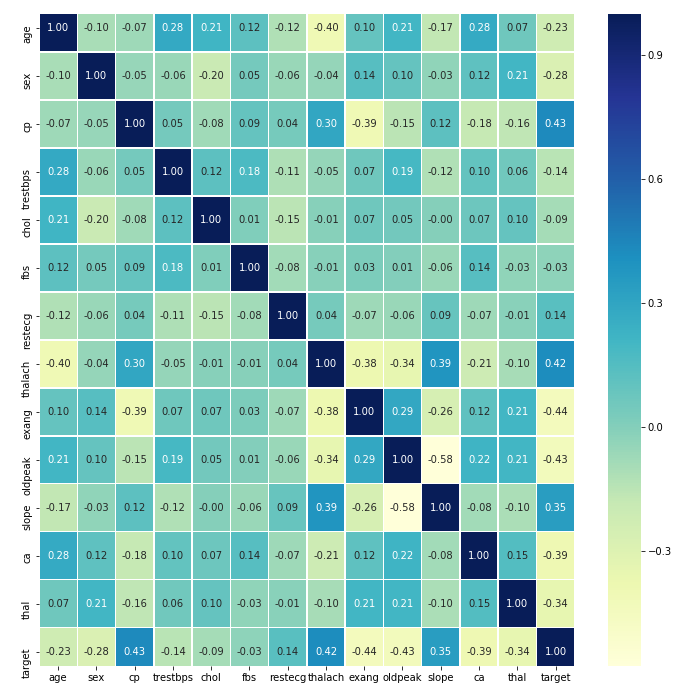
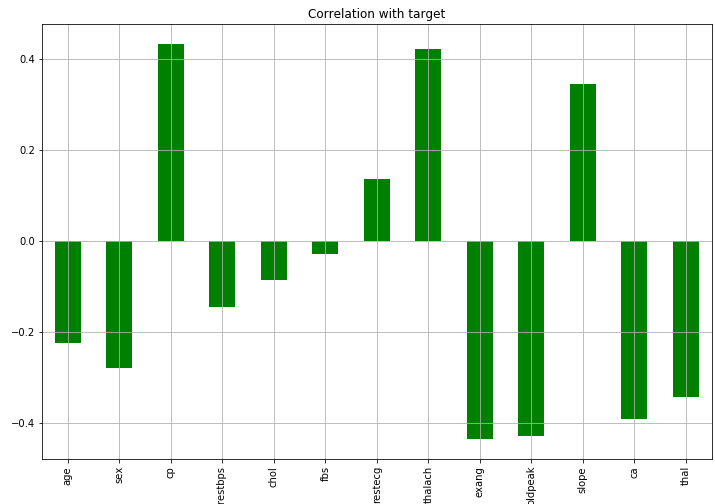
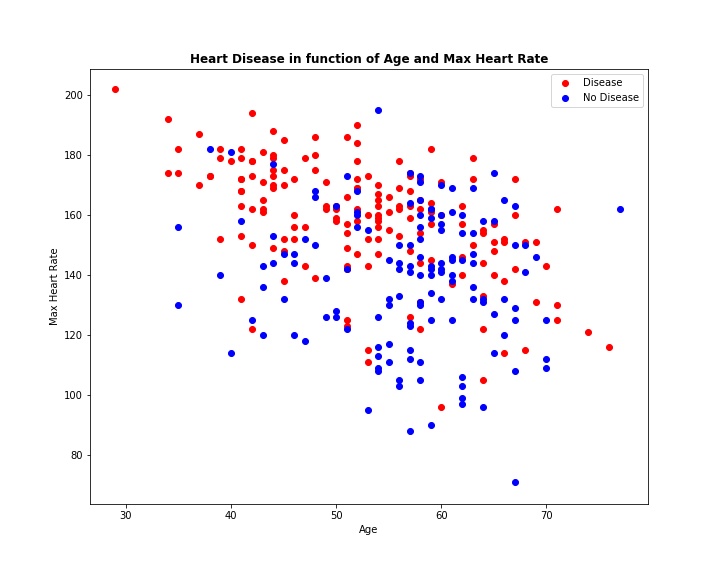


Figure 5 : it shows



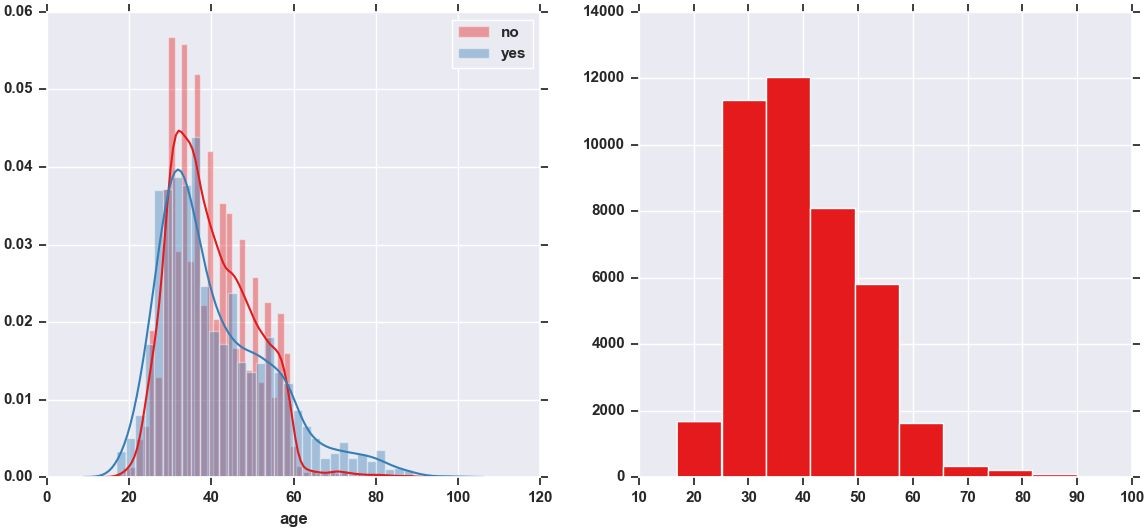






success of current marketing campaigns as well which in this case translates to buying of long term deposits.

Figures 18 & 19, present information about ‘pdays’ or the number of days after which the client was last contacted from the previous campaign. The values for this feature in the raw dataset either range from 0-27 or are taken as 999 (for clients who were not previously contacted). The extensive occurrence of 999 values in this figure again proves that the information about previous campaigns is not provided/ recorded properly by the bank.



Figures 20 & 21: Probability density function for age & its histogram.

From the information provided by the figures 20 & 21 in terms of the trend in age values vis-a-vis results (yes/no) we can easily infer that a huge chunk of clients who were contacted by the telemarketers were in the ages of 30-45. Another interesting observation from the distribution plot suggests that there is not much difference in the average and median ages of clients having a yes outcome or a no outcome, with mean values 39.9 for ‘no’ and 40.9 for ‘yes’ and median values of 38.0 for ‘no’ and 37.0 for ‘yes’ respectively. It is interesting to note that even within the most likely age group (30-45) that can buy the product there cannot be any estimation of the likelihood of the telemarketers succeeding.

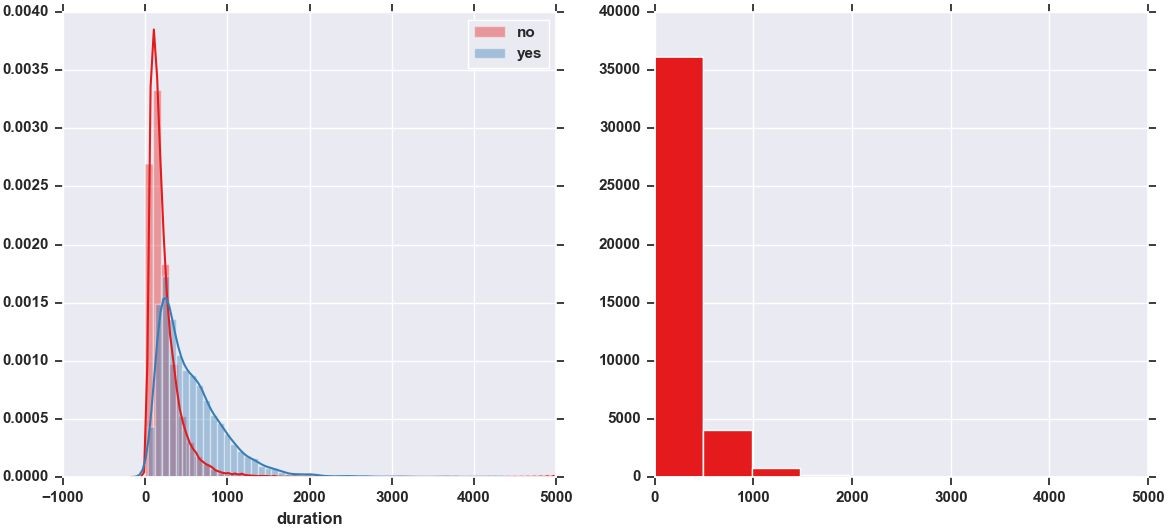


Figure 22 & 23: Probability density function of duration (last contact duration in seconds) & its histogram.

It is quite evident from the figures 22 & 23 that most of the calls made by the telemarketers last for about 500 seconds or about 8.3 minutes. We can also infer from these graphs that most of the results (yes and no both) are elicited in the initial phase of the conversation itself, pointing towards a sentiment that most of the clients either have a pre-meditated reply or are drawn towards the purchase in the introductory part of the conversation itself.

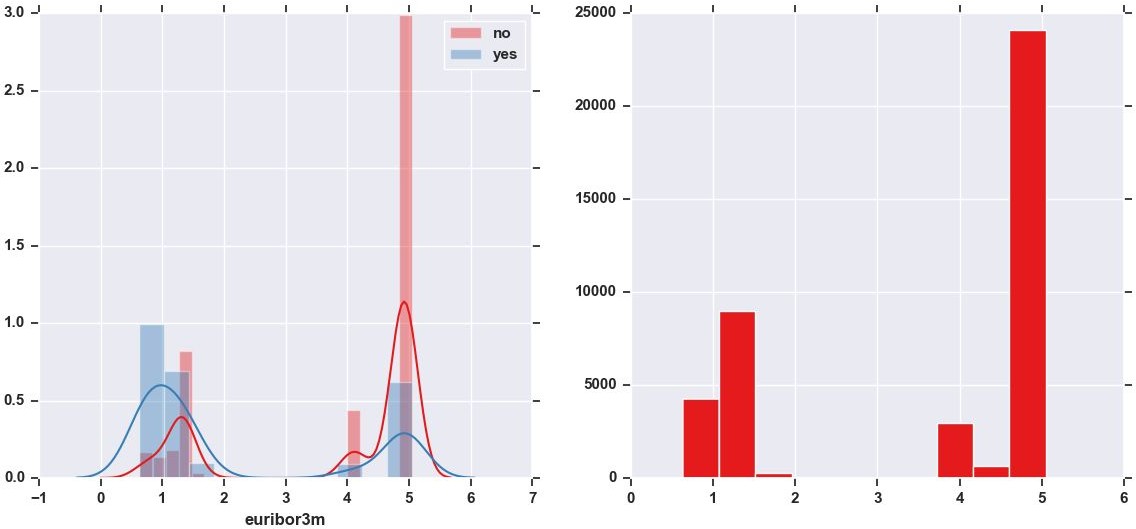


Figure 24 & 25 : Probability density function of Euribor rate & its histogram

From the figures 24 & 25, we can observe the records of the clients buying a long-term deposit for a particular Euribor rate. Euribor is a daily reference rate, published by the European Money Markets Institute, based on the averaged interest rates at which Eurozone banks offer to lend unsecured funds to other banks in the euro wholesale money market (or interbank market). From the visualization we also observe that the data near the euribor = 3 does not exist, thus raising a potent question about if there exists data about clients buying long term deposits in this period. What we also infer from the graph is that the proportion of yes is maximum at euribor around 1 than at a higher euribor rate.

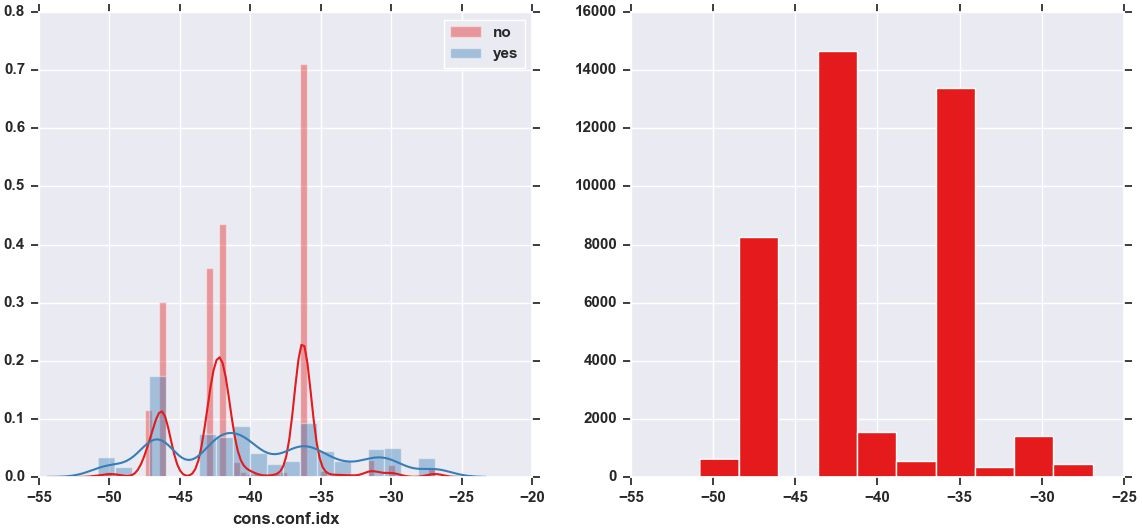


Figure 26 & 27: Probability density function of Consumer Confidence Index & its histogram.

Figures 26 & 27 present information about the Consumer Confidence Index – Which is an indicator of degree of optimism on the state of the economy, expressed through consumer activity of savings and spending. The graph indicates that the consumer confidence index noted in the dataset is distributed normally around three values in range –50 to –35. What can also be noted that the proportion of yes to no is highest at index values between –47 to –45.

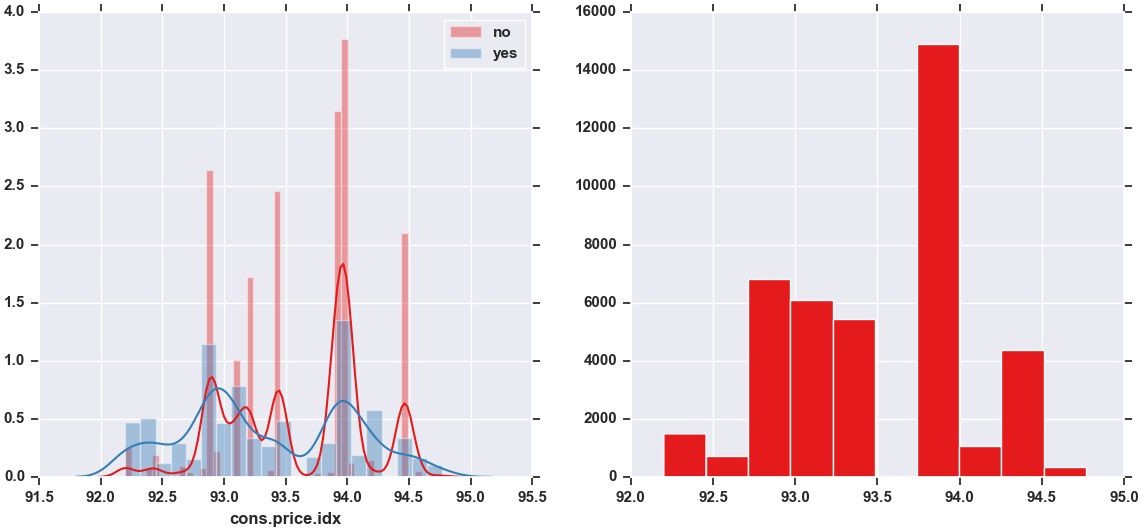
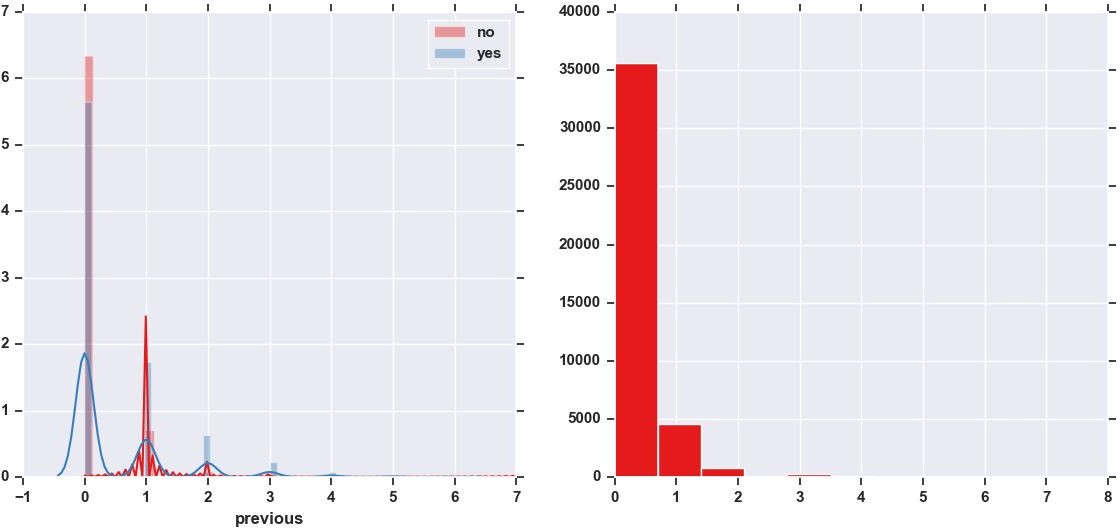


Figure 28 & 29: Probability density function of consumer price index & its histogram.

The figures 28 & 29 give information about Consumer Price Index - A statistical estimate constructed using the prices of a sample of representative items whose prices are collected periodically. A valuable inference derived from these figures points toward an important insight that the proportion of clients who bought a long term deposit to clients who did not was highest when consumer price index stayed between the ranges of 92.201 92.787.



Figures 30 & 31: Probability density for Previous (number of contacts performed to this client before this campaign) and its histogram.

In the figures 30 and 31, we can observe that the maximum proportion of clients who bought the long term deposits were never contacted at all followed by those who were

contacted once, and subsequently decreasing as the number of contacts goes up. This indicates that the less contacted clients are more likely to buy the product than those who were contacted more.

**CHAPTER 5**

# IMPLEMENTATION

## Introduction

This chapter describes the data preprocessing methods employed in the project and also covers the various feature engineering methods undertaken in the project. It also discusses the imputation analysis carried out 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 techniques such as aggregation, sampling, discretization, variable transformation and dimensionality reduction through feature subset selection and feature creation.

* + - Aggregation is the combination of data objects, which are the actual data

instances, into a single data instance.

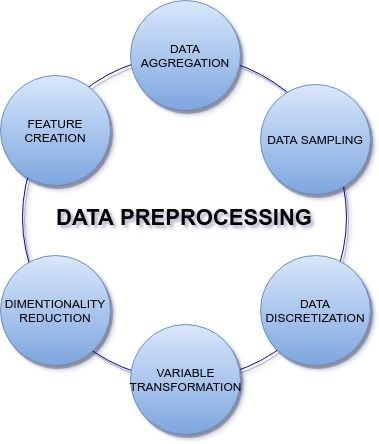
* + - Sampling means using a representative subset of the dataset most often to avoid the time and expense of utilizing the full dataset.
    - Discretization and categorization involves reducing the number of categories associated with a categorical attribute and generating categories for continuous attributes. This is especially useful for algorithms which require only categorical attributes.
    - Variable transformation is a transformation applied to each value of an attribute. An example of this is to take the absolute value of the values when only the magnitude is needed.
    - Dimensionality reduction is a technique applied on a dataset with a large number of attributes in order to remove irrelevant features that do not aid in pattern identifying within the dataset. Feature subset selection achieves dimensionality reduction by utilizing only a subset of the features available in the dataset.
    - Feature creation involves creating a completely new set of attributes from the current attributes.

Figure 32: Data Preprocessing

The original dataset was preprocessed to improve data quality. Preprocessing techniques of discretization and categorization were applied on several attributes after examining results from the initial iteration where models were generated using the original dataset. In addition, several attributes which were continuous were converted to categorical (pdays). Also, attributes with a very large number of categories were combined into a single attribute(education, job). The sampling technique was not applied in this dataset since the data was of manageable size. Thus, the results are representative of the full dataset. Dimensionality reduction was also not applied since the number of attributes was significantly smaller than the number of data instances. Apart from that unknown some values were imputed using cross-tabulation methods while some attributes containing unknown values were imputed by the most frequently occurring .

### Data Aggregation

The Feature engineering of the attributes education and job has been done in our dataset. In table 4 various categories have been clubbed together as per their level.

|  |  |
| --- | --- |
| Feature : Education | |
| Before Feature Engineering | After Feature Engineering |
| illiterate | Basic |
| basic.4y |
| basic.6y |
| basic.9y | Mid |
| high.school |
| professional.course | Degree |
| university.degree |

Table 4: Feature engineering of education attribute

|  |  |
| --- | --- |
| Feature : Job | |
| Before Feature Engineering | After Feature Engineering |
| housemaid | Unskilled |
| blue-collar |
| admin | Service |
| services |
| technician |
| self-employed | Professional |
| entrepreneur |
| management |
| student | student |

|  |  |
| --- | --- |
| unemployed | retired |
| retired | unemployed |

Table 5: Feature engineering of job attribute

In the feature engineering of attribute Job, various categories have been clubbed together to make a substantial category as per the data analytics report in Chapter 4.

### Data Discretization

|  |  |  |  |
| --- | --- | --- | --- |
| **poutcome** | failure | nonexistent | success |
| **pdays** |  |  |  |
| 0 | NaN | NaN | 15 |
| 1 | NaN | NaN | 26 |
| 2 | NaN | NaN | 61 |
| 3 | 4 | NaN | 435 |
| 4 | 2 | NaN | 116 |
| 5 | 4 | NaN | 42 |
| 6 | 25 | NaN | 387 |
| 7 | 15 | NaN | 45 |
| 8 | 6 | NaN | 12 |
| 9 | 24 | NaN | 40 |
| 10 | 7 | NaN | 45 |
| 11 | 3 | NaN | 25 |
| 12 | 13 | NaN | 45 |
| 13 | 8 | NaN | 28 |
| 14 | 5 | NaN | 15 |
| 15 | 9 | NaN | 15 |
| 16 | 2 | NaN | 9 |
| 17 | 5 | NaN | 3 |
| 18 | 5 | NaN | 2 |
| 19 | 1 | NaN | 2 |
| 20 | 1 | NaN | NaN |
| 21 | 2 | NaN | NaN |
| 22 | NaN | NaN | 3 |
| 25 | 1 | NaN | NaN |
| 26 | NaN | NaN | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 27 | NaN | NaN | 1 |
| 999 | 4110 | 35563 | NaN |

Table 6: Cross-tabulation of pdays and poutcome

It is evident from the above table (6) that a large chunk of values from pdays are missing. And we can see, '999' occurs nearly '35K' times when the poutcome is 'non-existent', which implies that it is because of the reason that customer was never contacted before. We will replace the numerical variable 'pdays' with the categorical variable. The categories will be as follows: p\_days\_missing, pdays\_less\_5, pdays\_bet\_5\_15, and pdays\_greater\_15 as shown in the table 7 below

|  |  |
| --- | --- |
| Feature : ‘pdays’ | |
| Range of values | New categorical variable name |
| =999 | pdays\_missing |
| <5 | pdays\_less\_5 |
| >15 | pdays\_greater\_15 |
| >=5 & <=15 | pdays\_bet\_5\_15 |

Table 7: Discretization of pdays attribute

We will drop the 'duration' variable as the value of that will only be known at the end of the call. Therefore, at that time we will also know the outcome of the call. The 'duration' variable will lead to leakage in the data and the prediction model will not be realistic. Also 'day\_of\_week' variable is not giving any significant information so we will drop both the features. One hot encoding and has been used to convert the categorical variable into numeric variables so that they can be used in a Machine Learning algorithm.

### Imputation Analysis

To infer the missing values in 'job' and 'education', we make use of the cross-tabulation between 'job' and 'education'. Our hypothesis here is that 'job' is influenced by the 'education' of a person. Hence, we can infer 'job' based on the education of the person. Moreover, since we are just filling the missing values, we are not much concerned about the causal inference. We, therefore, can use the job to predict the education as shown in the table 8 below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| education | Basic | Degree | Mid | unknown |
| job |  |  |  |  |
| Professional | 516 | 3830 | 1246 | 209 |
| Retired | 675 | 526 | 421 | 98 |
| Service | 732 | 11636 | 8155 | 611 |
| Student | 39 | 213 | 456 | 167 |
| Unemployed | 146 | 404 | 445 | 19 |
| Unskilled | 4304 | 745 | 4769 | 496 |
| unknown | 74 | 57 | 68 | 131 |

Table 8: Cross tabulation of education and job

We have distributed the unknown values to that category of attribute where the count is high. After distribution we will get the resulting table 9 as shown below

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| job | Professional | Retired | Service | Student | Unemployed | Unskilled | unknown |
| education |  |  |  |  |  |  |  |
| Basic | 516 | 733 | 732 | 39 | 146 | 4557 | 0 |
| Degree | 3969 | 566 | 12086 | 293 | 413 | 745 | 0 |
| Mid | 1338 | 421 | 8419 | 543 | 455 | 5086 | 0 |
| unknown | 0 | 0 | 0 | 0 | 0 | 0 | 131 |

Table 9: Cross-tabulation result after distribution

It is evident from table 9, that unknown value count we have now is only 131 now which will be distributed to the most frequently occurring category in the attribute. Similarly we will impute unknown values in housing and loan by using cross-tabulation between 'house' and 'job' and between 'loan' and 'job.' Here, our

hypothesis is that housing loan status (Yes or No) should be in the proportion of each job category.

## 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.

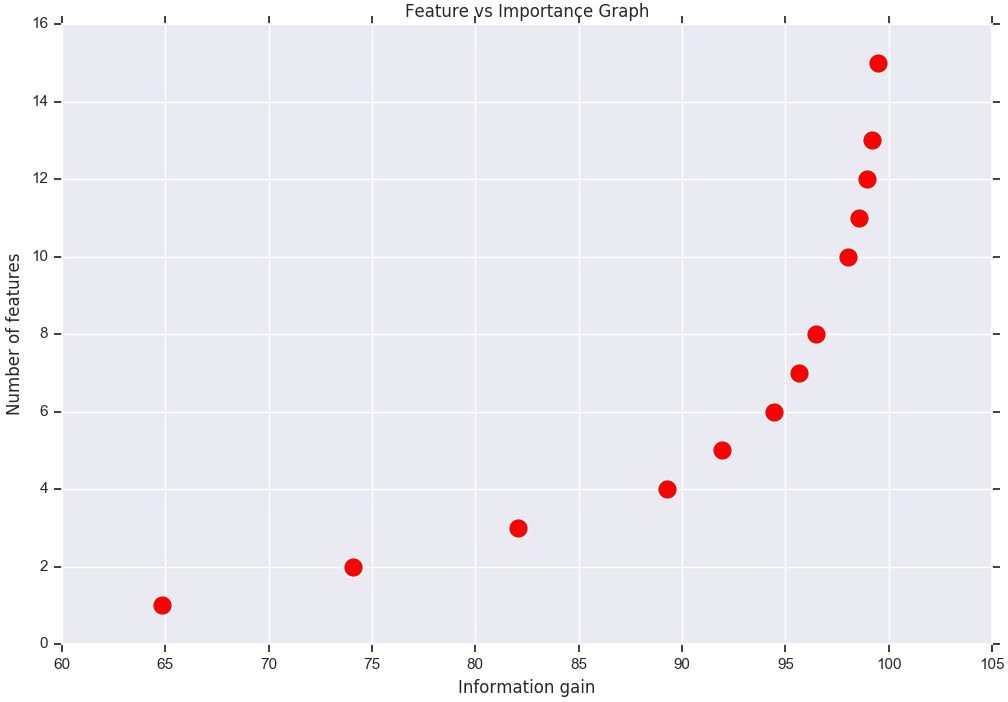


Figure 33:: Number of features versus information gain

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.

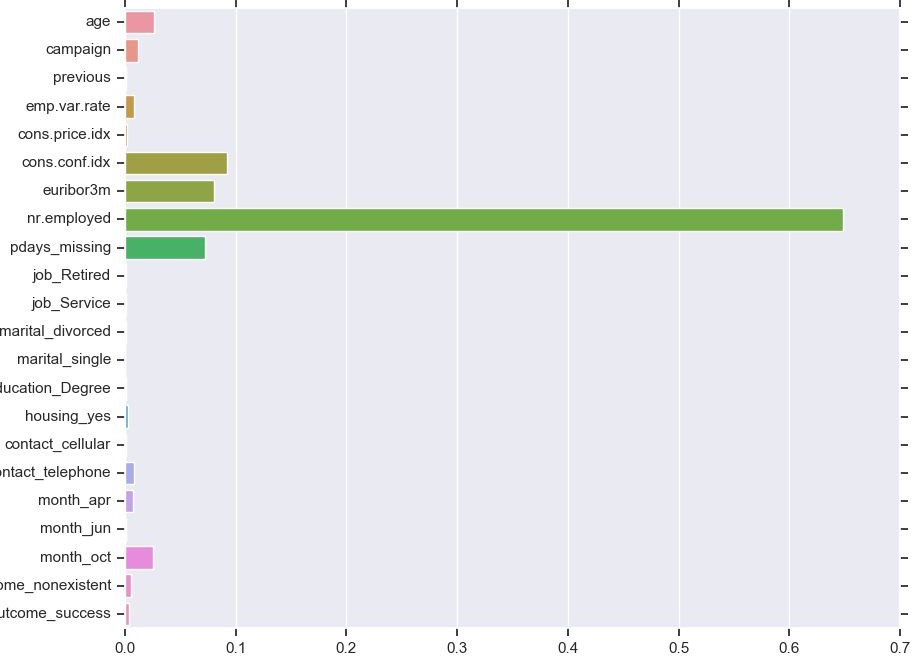


Figure 34: Contribution of features to output

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. The accuracy column in the table does not exhibit any major difference and remains somewhat uniform for all classification algorithms. In the cross validation accuracy column the least performance is exhibited by Extra tree and Decision tree algorithms while Random forest & Logistic regression perform the best in this criteria. Another inference that seems important from the table is *execution time*. It is highest for Linear SVM and lowest for Decision tree and Random forest algorithms.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Class | Precision | Recall | F1 score | Accuracy | Cross validation accuracy | Execution time |
| Decision tree | ‘no’ | 0.92 | 0.98 | 0.95 | 90.48% | 72.83% | 2.7898 s |
| ‘yes’ | 0.65 | 0.24 | 0.35 |
| total | 0.89 | 0.90 | 0.88 |
| Logistic regression | ‘no’ | 0.92 | 0.98 | 0.95 | 90.42% | 84.40% | 12.651 s |
| ‘yes’ | 0.64 | 0.24 | 0.35 |
| total | 0.89 | 0.90 | 0.88 |
| Random forest | ‘no’ | 0.91 | 0.99 | 0.95 | 90.43% | 83.06% | 3.6097 s |
| ‘yes’ | 0.68 | 0.19 | 0.30 |
| total | 0.89 | 0.90 | 0.88 |
| Ada boost | ‘no’ | 0.91 | 0.99 | 0.95 | 90.48% | 80.60% | 18.008 s |
| ‘yes’ | 0.67 | 0.22 | 0.33 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | total | 0.89 | 0.90 | 0.88 |  |  |  |
| Linear SVM | ‘no’ | 0.91 | 0.99 | 0.95 | 90.43% | 80.87% | 220.99 s |
| ‘yes’ | 0.68 | 0.19 | 0.30 |
| total | 0.89 | 0.90 | 0.88 |
| K-Nearest Neighbou r | ‘no’ | 0.91 | 0.99 | 0.95 | 90.48% | 82.93% | 20.472 s |
| ‘yes’ | 0.67 | 0.22 | 0.33 |
| total | 0.89 | 0.90 | 0.88 |
| Extra Tree | ‘no’ | 0.91 | 0.99 | 0.95 | 90.48% | 72.34% | 10.459 s |
| ‘yes’ | 0.67 | 0.22 | 0.33 |
| total | 0.89 | 0.90 | 0.88 |
| Voting Classifier | ‘no’ | 0.92 | 0.98 | 0.95 | 90.53% | 74.10% | 33.621 s |
| ‘yes’ | 0.65 | 0.24 | 0.36 |
| total | 0.89 | 0.91 | 0.89 |

Table10: Comprehensive analysis - I

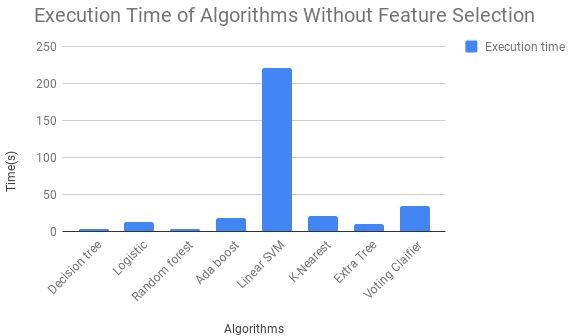


Figure 35: Execution time of algorithms without feature selection

The figure 35 details the execution time of various algorithms without applying feature selection on the data. What can be easily deduced from the figure is that, Linear SVM has the highest execution time among all the algorithms. The other algorithms all fall in similar range of less than 50 (seconds) among which Decision tree and Random forest algorithms take the least time.

## Impact of Feature Selection

The following table describes the change in the results of various classification algorithms when only selected features are fed to them. A positive change occurs in the execution times of all the algorithms and especially in Linear SVM where there is a substantial decrease of a whopping 8 seconds.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Class | Precision | Recall | F1 score | Accuracy | Cross validation accuracy | Execution time |
| Decision tree | ‘no’ | 0.91 | 0.98 | 0.95 | 90.13% | 72.83% | 2.5228 s |
| ‘yes’ | 0.61 | 0.22 | 0.32 |
| total | 0.88 | 0.90 | 0.88 |
| Logistic regression | ‘no’ | 0.91 | 0.99 | 0.95 | 90.43% | 84.40% | 11.346 s |
| ‘yes’ | 0.68 | 0.20 | 0.31 |
| total | 0.89 | 0.90 | 0.88 |
| Random forest | ‘no’ | 0.91 | 0.99 | 0.95 | 90.42% | 83.35% | 3.3314 s |
| ‘yes’ | 0.68 | 0.20 | 0.30 |
| total | 0.89 | 0.90 | 0.88 |
| Ada boost | ‘no’ | 0.91 | 0.99 | 0.95 | 90.39% | 80.79% | 16.596 s |
| ‘yes’ | 0.65 | 0.22 | 0.33 |
| total | 0.89 | 0.90 | 0.88 |
| Linear SVM | ‘no’ | 0.91 | 0.99 | 0.95 |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ‘yes’ | 0.68 | 0.19 | 0.30 | 90.43% | 78.55% | 212.52 s |
| total | 0.89 | 0.90 | 0.88 |
| K-Nearest Neighbou**r** | ‘no’ | 0.91 | 0.99 | 0.95 | 90.43% | 82.93% | 17.127 s |
| ‘yes’ | 0.68 | 0.19 | 0.30 |
| total | 0.89 | 0.90 | 0.88 |
| Extra Tree | ‘no’ | 0.91 | 0.99 | 0.95 | 90.43% | 71.88% | 10.063 s |
| ‘yes’ | 0.68 | 0.19 | 0.30 |
| total | 0.89 | 0.90 | 0.88 |
| Voting Classifier | ‘no’ | 0.91 | 0.99 | 0.95 | 90.31% | 74.86% | 31.644 s |
| ‘yes’ | 0.64 | 0.21 | 0.31 |
| total | 0.88 | 0.90 | 0.88 |

Table 11: Comprehensive analysis - II

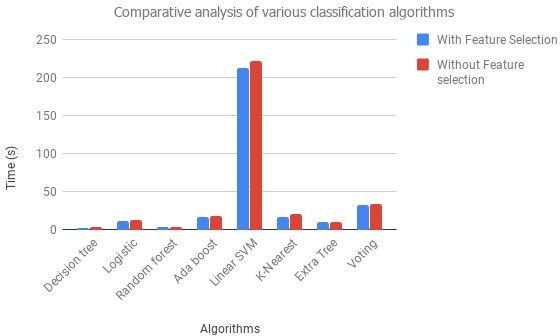


Figure 36: Execution time of various algorithms with and without feature selection

By observing the above figure it is evident that among all the classification algorithms the maximum change in execution time occurs in Linear SVM this indicates that for higher datasets this change (here, around 10 seconds) can spawn remarkable optimisation in the execution time of the algorithm.

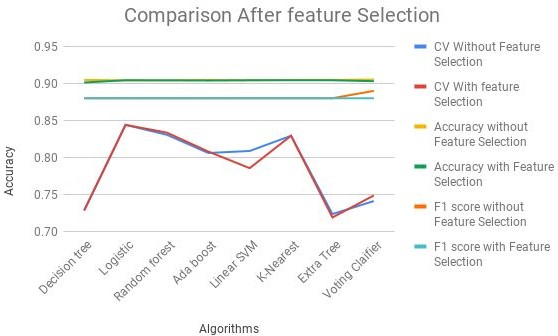


Figure 37: Comparison of algorithms vs various performance parameters

The illustrated figure 37 shows extensively that even after performing feature selection on the data, most of the accuracy and performance parameters remain almost the same. This is remarkable considering the amount of time and computational resources that can be saved by feeding algorithms less features for processing without having to care for any drop in the performance.

**CHAPTER 7**

# CONCLUSION AND FUTURE SCOPE

## Conclusion and Future Scope

The project fulfils its task of providing solution to the telemarketing classification problem and in doing so, describes a comprehensive yet structured approach for tackling those classification problems that have highly skewed datasets. The data analysis works undertaken in the project provide meaningful insights regarding the raw as well as the engineered dataset related to the problem. The project also delineates the approach for establishing importance of features that have the highest influence on whether a client purchases a term deposit or not. Through this project it was discovered that there are 14 variables that actually impact the decision of clients to purchase term deposits out of which 7 are considered as strong. We also discovered that macroeconomic indicators, have the most considerable effect on the output variable, this insight can be especially used by banks and other institutions for developing better marketing strategies. Further research can be performed by combining this dataset with another dataset to find more robust results which can lead to exploration of better predicting models.. Use of a bigger dataset would also be very interesting to see and if similar conclusions can be made when a larger dataset is used. Usage and predictability of other attributes in the dataset or usage of different metrics and algorithms can also be analyzed for a better and detailed understanding of the problem.

# REFERENCES

## References

1. M Ali, FR Lone and A Hussain, Machine Learning: Concept and Applications in Smart City Projects in India, IJSRCSEIT, ISSN : 2456-3307, 2017.
2. M Ali, A Hussain and FR Lone, Data Analytics for Formulation of Effective Public Policy, IJSRCSEIT, ISSN : 2456-3307, 2018.
3. "Logistic regression," 14 August 2015. [Online]. Available: <https://www.medcalc.org/manual/logistic_regression.php>.
4. P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Pearson Education, Inc., 2006.
5. C.-C. Chang and C.-J. Lin, "LIBSVM -- A Library for Support Vector Machine," [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
6. “kNN (k- Nearest Neighbors)” 9 September 2017. [Online] Available: [https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)

[/](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/).

1. “ Tree Based Modelling”, 12 April 2016. [Online] Available : [https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling](https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/)

[-scratch-in-python/](https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/)

1. P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees.Machine Learning, 36 (1):3–42, 2006.
2. Ensemble methods: <http://scikit-learn.org/stable/modules/ensemble.html#forest>
3. Alan, "WHY ACCURACY ALONE IS A BAD MEASURE FOR CLASSIFICATION TASKS, AND WHAT WE CAN DO ABOUT IT," 25 March 2013. [Online]. Available: [http://blog.tryolabs.com/2013/03/25/why-accuracy-alonebad-measure-classification-ta](http://blog.tryolabs.com/2013/03/25/why-accuracy-alonebad-measure-classification-tasks-and-what-we-can-do-about-it/) [sks-and-what-we-can-do-about-it/](http://blog.tryolabs.com/2013/03/25/why-accuracy-alonebad-measure-classification-tasks-and-what-we-can-do-about-it/).
4. Valverde-Albacete FJ, Peláez-Moreno C (2014) 100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox. PLoS ONE 9(1): e84217. doi:10.1371/journal.pone.0084217.
5. N. V. Chawla, "C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure," Washington DC, 2003.
6. S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014.
7. S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.),

Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS. [bank.zip].

1. H. A. Elsalamony, "Bank Direct Marketing Analysis of Data Mining," International Journal of Computer Applications, vol. 85, no. 7, pp. 12-22, January 2014.

# APPENDIX

## Decison Tree Classifier:

from sklearn.tree import DecisionTreeClassifier start\_time = time.time()

clf\_ftr = DecisionTreeClassifier(criterion='entropy',min\_samples\_split = 9,max\_depth=6,random\_state=23)

clf\_ftr = clf\_ftr.fit(d\_train, l\_train) predicted = clf\_ftr.predict(d\_test)

scores = cross\_val\_score(clf\_ftr, features, target, cv=skf)

print("Decision Tree Classifier Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test)))

print("Cross validation score: {0:.2%} )".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## Logistic Regression:

from sklearn.linear\_model import LogisticRegression start\_time = time.time()

lrc = LogisticRegression() lrc.fit(d\_train, l\_train) predicted = lrc.predict(d\_test)

scores = cross\_val\_score(lrc, features, target, cv=skf)

print("Logistic Regression Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test)))

print("Cross validation score: {0:.2%} )".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## Random Forest Classifier:

from sklearn.ensemble import RandomForestClassifier start\_time = time.time()

rfc = RandomForestClassifier(n\_estimators=10, max\_depth=4, min\_samples\_split=9) rfc.fit(d\_train, l\_train)

predicted = rfc.predict(d\_test)

scores = cross\_val\_score(rfc, features, target, cv=skf)

print("Random Forest Classifier: {0:.2%}".format(accuracy\_score(predicted, l\_test))) print("Cross validation score: {0:.2%} ".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## AdaBoost Classifier:

from sklearn.ensemble import AdaBoostClassifier start\_time = time.time()

ad = AdaBoostClassifier(base\_estimator=rfc, n\_estimators=5) ad.fit(d\_train, l\_train)

predicted = ad.predict(d\_test)

scores = cross\_val\_score(ad, features, target, cv=skf)

print("Ada Boost Classifier Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test)))

print("Cross validation score: {0:.2%} ".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## Support Vector Machine:

from sklearn.svm import LinearSVC start\_time = time.time()

sclf = LinearSVC() sclf.fit(d\_train, l\_train) predicted = sclf.predict(d\_test)

scores = cross\_val\_score(sclf, features, target, cv=skf)

print("Support Vector Machine: {0:.2%}".format(accuracy\_score(predicted, l\_test)))

print("Cross validation score: {0:.2%} (+/- {1:.2%})".format(np.mean(scores), np.std(scores)\*2))

print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## K Nearest-Neighbours:

from sklearn.neighbors import KNeighborsClassifier start\_time = time.time()

k\_clf = KNeighborsClassifier() k\_clf.fit(d\_train, l\_train) prediction = k\_clf.predict(d\_test)

scores = cross\_val\_score(k\_clf,features, target, cv=skf)

print("KNN Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test))) print("Cross validation score: {0:.2%} ".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## Extra Tree Classifier:

from sklearn.ensemble import ExtraTreesClassifier start\_time = time.time()

ex\_clf = ExtraTreesClassifier() ex\_clf.fit(d\_train, l\_train) prediction = ex\_clf.predict(d\_test)

scores = cross\_val\_score(ex\_clf,features, target, cv=skf)

print("Extra tree Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test))) print("Cross validation score: {0:.2%} ".format(np.mean(scores), np.std(scores)\*2)) print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## Voting Classifier:

from sklearn.ensemble import VotingClassifier start\_time = time.time()

vclf1 = VotingClassifier(estimators=[('rfc\_1', ad), ('log\_reg', lrc), ('des\_tree', clf\_ftr)], voting='soft')

vclf1.fit(d\_train, l\_train) predicted = vclf1.predict(d\_test)

scores = cross\_val\_score(vclf1,features, target, cv=skf)

print("Voting Classifier Accuracy: {0:.2%}".format(accuracy\_score(predicted, l\_test)))

print("Cross validation score: {0:.2%} (+/- {1:.2%})".format(np.mean(scores), np.std(scores)\*2))

print("Execution time: {0:.5} seconds \n".format(time.time()-start\_time)) print(classification\_report(l\_test,predicted))

## For New Data:

def convertInputFeatures(df): #Grouping Education variable

df\_basic\_ed = df[(df.education == 'basic.4y') | (df.education == 'basic.6y') | (df.education == 'illiterate')]

df\_mid\_ed = df[(df.education == 'basic.9y') | (df.education == 'high.school')] df\_degree\_ed = df[(df.education == 'professional.course') | (df.education ==

'university.degree')]

df.loc[df\_basic\_ed.index, 'education'] = 'Basic' df.loc[df\_mid\_ed.index, 'education'] = 'Mid' df.loc[df\_degree\_ed.index, 'education'] = 'Degree'

df\_unskilled = df[(df.job == 'blue-collar') | (df.job == 'housemaid')]

df\_service = df[(df.job == 'admin.') | (df.job == 'services') | (df.job == 'technician')] df\_professional = df[(df.job == 'entrepreneur') | (df.job == 'self-employed') | (df.job

== 'management')]

df\_student = df[(df.job == 'student')] df\_retired = df[(df.job == 'retired')] df\_unemployed = df[(df.job == 'unemployed')]

df.loc[df\_unskilled.index, 'job'] = 'Unskilled' df.loc[df\_service.index, 'job'] = 'Service' df.loc[df\_professional.index, 'job'] = 'Professional' df.loc[df\_student.index, 'job'] = 'Student' df.loc[df\_retired.index, 'job'] = 'Retired' df.loc[df\_unemployed.index, 'job'] = 'Unemployed' #Removing insignificant variables df.drop(['day\_of\_week'],axis=1,inplace=True) df.drop('duration',axis=1,inplace=True df\_loan\_unknown = df[(df.loan == 'unknown')] df.loc[df\_loan\_unknown.index, 'loan'] = 'no' df\_housing\_unknown = df[(df.housing == 'unknown')] df.loc[df\_housing\_unknown.index, 'housing'] = 'yes'

return df

### Removing insignificant variables

df.drop(['day\_of\_week'],axis=1,inplace=True) df.drop('duration',axis=1,inplace=True)

### Creating a dummy variable:

dfnew\_val = [[40,"services","married","basic.9y","no","unknown","unknown","telephone","jun","f

ri",17,3,999,1,"nonexistent",1.4,94.465,-41.8,4.959,5228.1]]

df\_new = pd.DataFrame(new\_val)

df\_new.columns = ["age","job","marital","education","default","housing","loan","contact","month","day

\_of\_week","duration","campaign","pdays","previous","poutcome","emp.var.rate","co ns.price.idx","cons.conf.idx","euribor3m","nr.employed"]

### Conerting categorical values to numerical:

df\_new = pd.get\_dummies(df\_new)

### Predicting using classifier:

rfc.predict(np.reshape(df\_new[imp\_cols].loc[0, :].values.tolist(),(1,-1)))

### Testing:

With instance: Feature set age job marital education default housing loan contact month day\_of\_week duration campaign pdays previous poutcome emp.var.rate cons.price.idx cons.conf.idx euribor3m nr.employed 0 40 services married basic.9y no unknown unknown telephone jun fri 17 3 999 1 nonexistent 0 0 0 0 0