A PROJECT REPORT
ON

Statistical modelling for stock market prices of bank of America Corporation

Submitted to

Osmania University
in partial fulfilment of the requirements for the award of

MASTER OF SCIENCE
IN
STATISTICS

DEPARTMENT OF STATISTICS
UNIVERSITY COLLEGE OF SCIENCE
OSMANIA UNIVERSITY
HYDERABAD-INDIA

By

R. KRISHNA SAI	**1007-17-508-038**
B. CHARANLAL	**1007-17-508-040**
N. SAI KISHORE	**1007-17-508-036**
G. HATHIRAM	**1007-17-508-035**
G.M SARITHA	**1007-17-508-041**
P. BHARATHI	**1007-17-508-039**

Under the Supervision of

T.SANDHYA

2018

A PROJECT REPORT
ON

Statistical modelling for stock market prices of bank of America Corporation

Submitted to
Osmania University in
partial fulfilment of the
Requirements for the award of Master
of Science in Statistics

DEPARTMENT OF STATISTICS
UNIVERSITY COLLEGE OF SCIENCE
OSMANIA UNIVERSITY
HYDERABAD – INDIA

By

R. KRISHNA SAI	**1007-17-508-038**
B. CHARANLAL	**1007-17-508-040**
N. SAI KISHORE	**1007-17-508-036**
G. HATHIRAM	**1007-17-508-035**
G.M SARITHA	**1007-17-508-041**
P. BHARATHI	**1007-17-508-039**

Under the Supervision of

T. SANDHYA

2018

DECLARATION

The research presented in this project has been carried out in the **Department of Statistics, Osmania University, Hyderabad.** The work is original has not been submitted so far, in part or full, for any other degree of diploma of any university.

Mr. R.KRISHNA SAI

Mr. B.CHARANLAL

Mr. N.SAI KISHORE

Mr.G. HATHIRAM

Mrs.G.M SARITHA

Mrs.P BHARATHI

Department of Statistics

Osmania University

Hyderabad – 500 007, T.S.

INDIA

CERTIFICATE

This is to certify that

Mr. R.KRISHNA SAI	**1007-17-508-038**
Mr. B.CHARANLAL	**1007-17-508-040**
Mr. N.SAI KISHORE	**1007-17-508-036**
Mr. G.HATHIRAM	**1007-17-508-035**
Mrs. G.M.SARITHA	**1007-17-508-041**
Mrs. P.BHARATHI	**1007-17-508-039**

Have submitted the project titled **"PREDICTING WHETHER A WOMAN HAS DIABETES OR NOT"** in partial fulfilment for the degree of Master of Science in Statistics.

Head

Department of Statistics

Internal Examiner External Examiner

ACKNOWLEDGEMENTS

I deem it a great pleasure to express my deep sense of gratitude and indebtedness to my research supervisor **T.SANDHYA,** Statistics department, University College of Science, Osmania University for his valuable guidance, and enlightening discussions throughout the progress of my project work.

I also express my sincere and heartfelt thanks to **PROF.C.JAYALAKSHMI ,**Head of Department, Department of Statistics, Osmania University for providing the necessary support and facilities in the department for completion of this work successfully.

It is indeed with great pleasure I record my thanks to **Dr . G . JAYASREE ,** Chairperson, Board of Studies , Department of Statistics , Osmania University for having provided with all the facilities to carry out our work.

I thank **Dr.N.Ch.BHATRACHARYULU , Dr.K.VANI, Dr.S.A.JYOTHI RANI, Dr.G.SIRISHA, Mrs.J.L.PADMA SHREE ,** for their encouragement and constant help during the research.

I would like to express my deepest gratitude to **Dr. VENUGOPALA RAO, BALA KARTHIK** for their advice, guidance and involvement at various stages of this work , I would also like to thank them for their understanding and constant encouragement throughout this project.

I thank all Non-Teaching members of the Department of Statistics, who helped me during my Thesis work.

I am thankful to the Osmania University for permitting me to carry out this work

CONTENTS

Chapter 1

Introduction

Introduction

PREDICTING WHETHER A WOMAN HAS DIABETES OR NOT Data:

- To predict whether a woman has Diabetes or not. To base on given information

Data description:

- Fromthe available data we have 768 instances and 8 attributes they are:

1. Number of times pregnancies (preg)
2. Plasma glucose concentration a 2 hrs in an oral glucose tolerance test (plas)
3. Diastolic blood pressure in mm Hg (pres)
4. Triceps skin fold thickness in mm (skin)
5. 2 hr serum insulin in mu u/ml (insu)
6. Body mass index measured as weight in kg (height in m)^2 (mass)
7. Diabetes pedigree function (pedi)
8. Age in yrs (age)

Label of data:

From the data the outcome variable(which isdependent) gives whether a woman has diabetes or not, indicates as 0 and 1. Here 0 represents a woman has no diabetes and 1 represents a woman has diabetes

SOURCE OF DATA:

The data is extracted from(http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes) female patients of at least 21 yrs old of Pima Indianheritage.

Review of the Chapters:

Chapter 2 gives the brief introduction about machine learning techniques like need of ML today, types of ML Algorithms and various models in each algorithm and what technique to use when and how to validate, Tune the ML algorithms and how to measure the performance of the ML model

Section 3 describes the various results obtained for the problem. This section contains all the outputs generated through the ML algorithms applied on the data as well as validation and performance matrices.

Section 4 describes the summary and conclusions followed by Bibliography.

Chapter 2

Review of Machine learning process

Review of Machine Learning Process

2.0 Need of Machine Learning

In this age of modern technology, there is one resource that we have in abundance: a large amount of structured and unstructured data. In the second half of the twentieth century, machine learning evolved as a subfield of artificial intelligence that involved the development of self-learning algorithms to gain knowledge from that data in order to make predictions. Instead of requiring humans to manually derive rules and build models from analysing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve the performance of predictive models, and make data-driven decisions. Not only is machine learning becoming increasingly important in computer science research but it also plays an ever greater role in our everyday life.

2.1 Machine Learning Process

The CRISP-DM (Cross-Industry Standard Process for Data Mining) Process was designed specifically for the data mining. However, it is flexible and thorough enough that it can be applied to any analytical project whether it is predictive analytics, data science, or Machine learning. The Process has the following six phases

oBusiness Understanding **oData Understanding** **oData preparation**
oModelling **oEvaluation** **oDeployment**

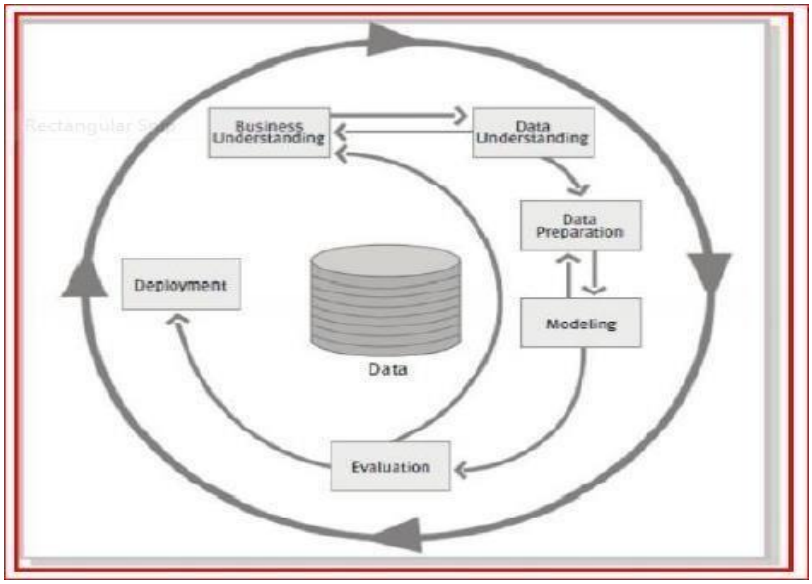


Fig 2.1 crisp- dm diagram

And, each phase has different steps covering important tasks which are mentioned below:

2.1.1) Business Understanding

It is very important step of the process in achieving the success. The purpose of this step is to identify the requirements of the business so that you can translate them into analytical objectives. It has the following tasks: 1) Identify the Business objective

2) Assess the situation

3) Determine the Analytical goals

4) Produce a project plan

2.1.2) Data Understanding

After enduring the all-important pain of the first step, you can now get your hands on the data. The task in this process consist the following

1) Collect the data

2) Describe the data

3) Explore the data

4) Verify the data Quality

2.1.3) Data Preparation

This step is relatively self-explanatory and in this step the goal is to get the data ready to input in the algorithms. This includes merging, feature engineering, and transformations. If imputation for missing values / outliers is needed then, it happens in this step. The key five tasks under this step are as follows: 1) Select the data

2) Clean the data

3) Construct the data

4) Integrate the data

5) Format the data

2.1.4) Modelling

Oddly, this process step includes the consideration that you already thought of and prepared for. In this, one will need at least a modicum of an idea about how they will be modelling. Remember, that this is flexible, iterative process and some strict linear flow chart such as an aircrew checklist.

Below are the tasks in this step:

1) Select a modelling technique

2) Generate a test design

3) Build a model

4) Assess a Model

Both cross validation of the model (using tran/test or K fold validation) and model assessment which involves comparing the models with the chosen criterion (RMSE, Accuracy, ROC) will be performed under this phase.

2.1.5) Evaluation

In the evaluation process, the main goal is to confirm that the work that has been done and the model selected at this point meets the business objective. Ask yourself and others, have we achieved the definition of success? And, here are the tasks in this step: 1) Evaluate the results

2) Review the process

3) Determine the next steps

2.1.6) Deployment

If everything is done according to the plan up to this point, it might come down to flipping a switch and your model goes live. Here are the tasks in this step: 1) Deploying the plan

2) Monitoring and maintenance of the plan

3) Producing the final report

2.2 Types of Machine Learning

Broadly, the Machine Learning Algorithms are classified into 3 types.

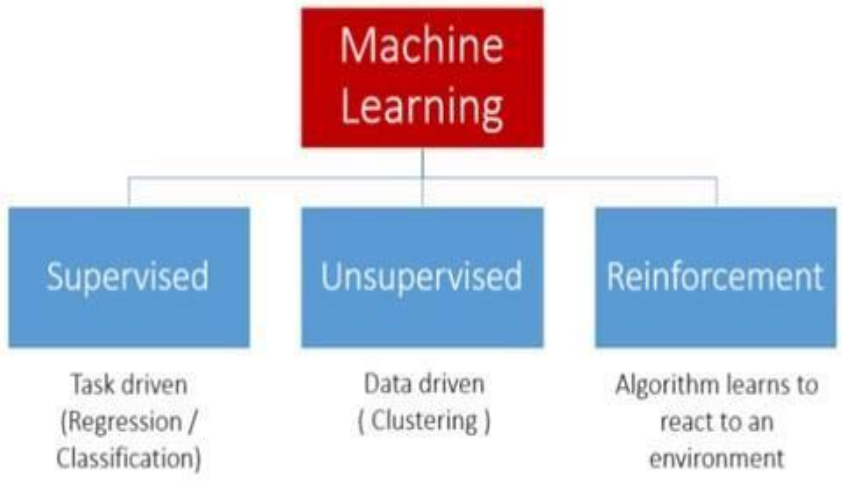


Fig 2.2 Types of machine Learning

2.2.1) Supervised Learning

This algorithm consists of a target / outcome / dependent variable which is to be predicted from a given set of predictors / independent variables. Using these set of variables, we generate a function that maps inputs to desired output. The training process continues until the model achieves a desired level of accuracy on the training data.

The process of Supervised Learning model is illustrated in the below picture:

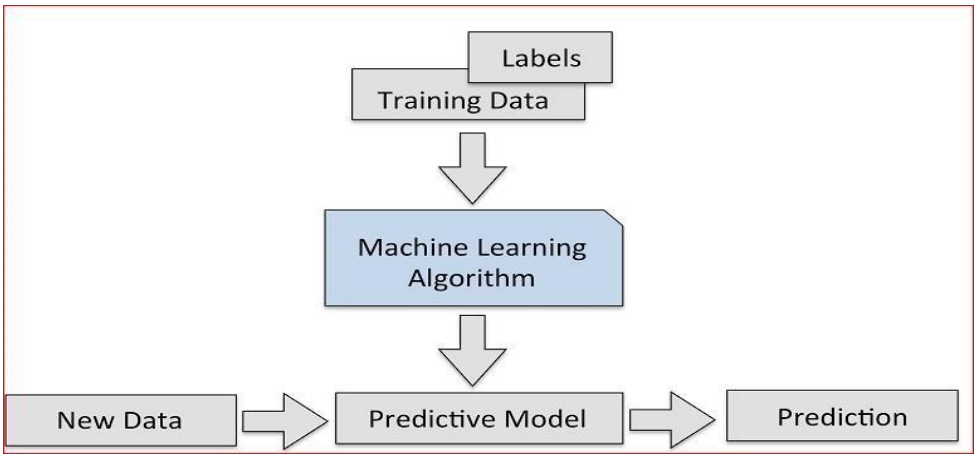


Fig 2.2.1 Supervised Learning

Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression,…etc

X Classification

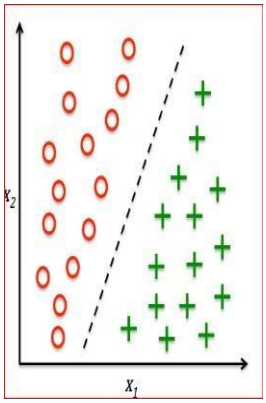


Fig 2.2.1 classification

Regression

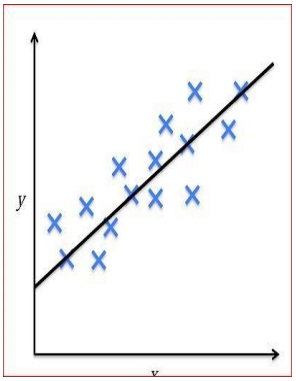


Fig 2.2.1 Regression

2.2.2) Unsupervised Learning

In this algorithm, we will not have any target or outcome variable to predict / estimate. It is used for clustering population into different groups, which is widely used for segmenting customers in different groups for specific intervention. (More of Exploratory Analysis)

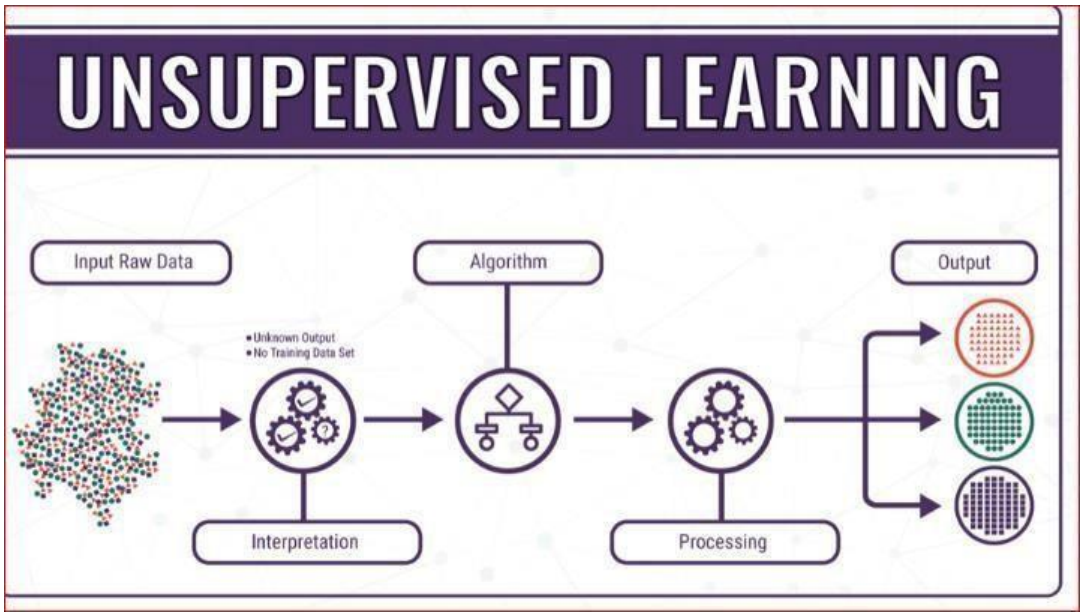


Fig 2.2.2 Unsupervised Learning

Examples of Unsupervised Learning: Data reduction techniques, Cluster Analysis, Market Basket Analysis,…etc

Cluster Analysis

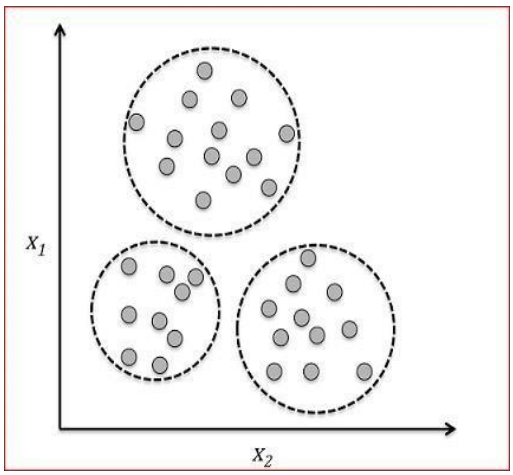


Fig 2.2.2 cluster Analysis

Data Reduction Techniques

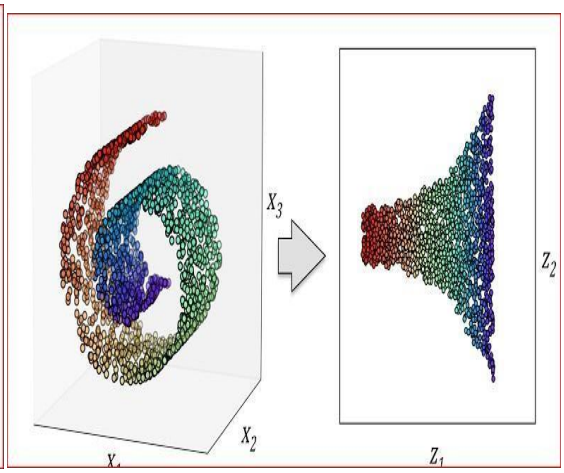


Fig 2.2.2 reduction Techiniqies 2.2.3)

Reinforcement Learning

Using this algorithm, the machine is trained to make specific decisions.

It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions.

The process of reinforcement learning is illustrated in the below picture:

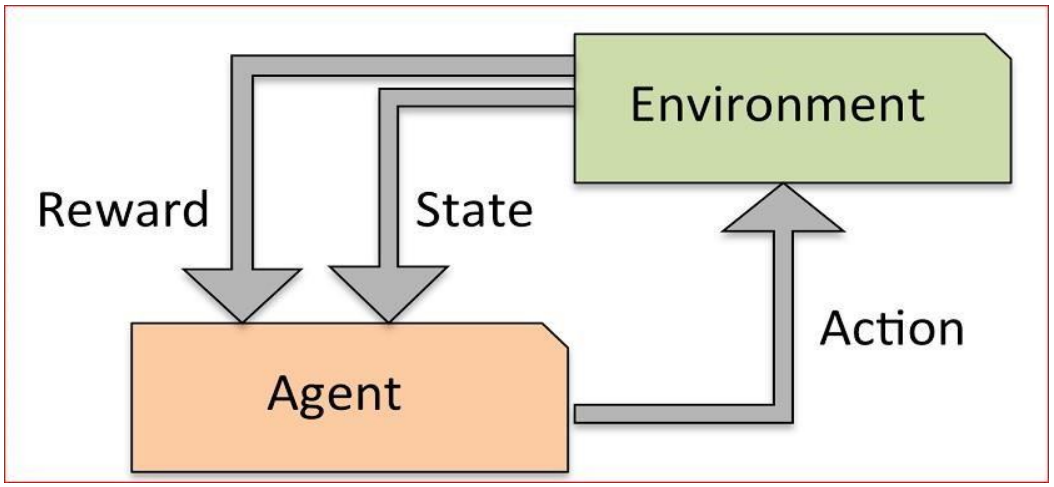


Fig 2.2.3 Reinforcement Learning

Examples of Reinforcement Learning: Markov Decision Process, Self-driving cars,...etc

2.3 Choosing the algorithm

Choosing the right algorithm will depend on the type of the problem we are solving and also Depends on the scale of the dependent variable. In case of continuous target variable, we will Use regression algorithms and in case of categorical target, we will use classification algorithms And for the model which doesn't have target variable, we will use either cluster analysis / data Reduction techniques.

Below picture describes the process of choosing the right algorithm:

2.3 Choosing the algorithm

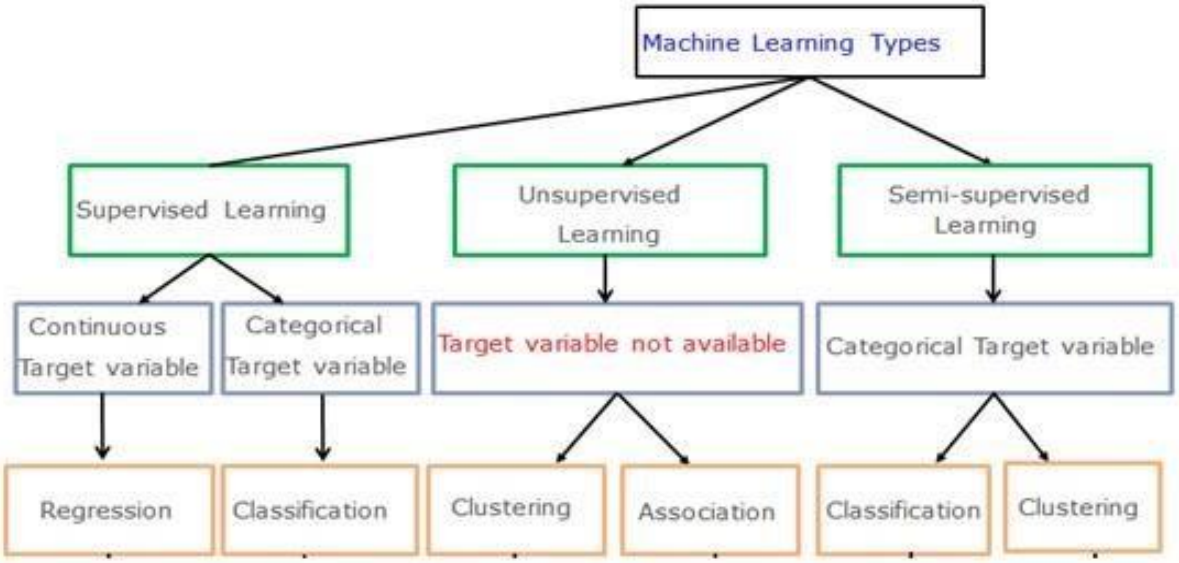


Fig 2.3 choosing the algorithm

2.3.1) Types of Regression Algorithms

There are many Regression algorithms in machine learning, which will be used in different regression applications. Some of the main regression algorithms are as follows:

a) **Simple Linear Regression:-**In simple linear regression, we predict scores on one variable from the data of second variable. The variable we are forecasting is called the criterion variable and referred to as Y. The variable we are basing our predictions on is called the predictor variable and denoted as X.

b) **Multiple Linear Regression:-**Multiple linear regression is one of the algorithms of regression technique, and is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regressionsis used to explain the relationship between one dependent variable with two or more independent variables. The independent variables can be either continuous or categorical.

c) **Polynomial Regression:-**Polynomial regression is another form of regression in which the maximum power of the independent variable is more than 1. In this regression technique, the best fit line is not a straight line instead it is in the form of a curve.

d) **Support Vector Machines:-**Support Vector Machines can be applied to regression problems as well as Classification. It contains all the features that characterises maximum margin algorithm. Linear learning machine maps a non-linear function into high dimensional kernelinduced feature space. The system capacity will be controlled by parameters that do not depend on the dimensionality of feature space.

e) **Decision Tree Regression:-**Decision tree builds regression models in the form of a tree structure. It breaks down the data into smaller subsets and while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

f) **Random Forest Regression:-**Random Forest is also one of the algorithms used in regression technique. It is very a flexible, easy to use machine learning algorithm that produces, even without hyper -parameter tuning, a great result most of the time. It is also one of the most widely used algorithms because of its simplicity and the fact that it can used for both regression and classification tasks. The forest it builds is an ensemble of Decision Trees, most of the time trained with the "bagging" method.

Other than these we have regularized regression models like **Ridge, LASSO** and **Elastic Net regression** which is used to select the key parameters and these is also **Bayesian regression** which works with the Bayes theorem.

2.3.2) Types of Classification Algorithms

There are many Classification algorithms in machine Learning, which can be used for different classification applications. Some of the main classification algorithms are as follows:

a) **Logistic Regression/Classification**:-Logistic regression falls under the category of supervised learning; it measures the relationship between the dependent variable which is categorical with one or more than one independent variables by estimating probabilities using a logistic/sigmoid function. Logistic regression can generally be used when the dependent variable is Binary or Dichotomous. It means that the dependent variable can take only two possible values like "Yes or No", "Living or dead".

b) **K -Nearest Neighbours:-**k-NN algorithm is one of the most straightforward algorithms in classification, and it is one of the most used ML algorithms. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. It can also use for regression — output is the value of the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbours.

b) **Naive Bayes:-**Naive Bayes is a type of Classification technique based on Bayes' theorem, with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a Particular feature in a class is unrelated to the presence of any other function. Naive Bayes model is accessible to build and particularly useful for extensive datasets.

d) Decision Tree Classification:-Decision tree builds classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The first decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

e) Support Vector Machines:-A Support Vector Machine is a type of Classifier, in which a discriminative classifier is formally defined by a separating hyper plane. The algorithm outputs an optimal hyper plane which categorises new examples. In two dimensional space, this hyper plane is a line dividing a plane in two parts where in each class lay in either side.

f) Random Forest Classification:-Random Forest is a supervised learning algorithm. It creates a forest and makes it somehow random. The forest it builds is an ensemble of Decision Trees, most of the times the decision tree algorithm trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result. And Random Forest is also very powerful to find the variable importance in classification/ Regression problems.

2.3.3) Types of Unsupervised Learning

Clustering is the type of unsupervised learning in which an unlabelled data is used to draw inferences. It is the process of grouping similar entities together. The goal of this unsupervised machine learning technique is to find similarities in the data points and group similar data points together and also to figure out which cluster should a new data point belong to.

Types of Clustering Algorithms:-There are many Clustering algorithms in machine learning, which can be used for different clustering applications. Some of the main clustering algorithms are as follows:

a) Hierarchical Clustering:-Hierarchical clustering is one of the algorithms of clustering technique, in which similar data is grouped in a cluster. It is an algorithm

b) that builds the hierarchy of clusters. This algorithm starts with all the data points assigned to a bunch of their own. Then, two nearest groups are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

c) It starts by assigning each data point to its bunch. Finds the closest pair using Euclidean distance and merges them into one cluster. This process is continued until all data points are clustered into a single cluster.

d) b) K -Means Clustering:-K-Means clustering is one of the algorithms of clustering technique, in which similar data is grouped into a cluster. K-means is an iterative algorithm that aims to find local maxima in each iteration. It starts with K as the input which is the desired number of clusters. Input k centroids in random locations in your space. Now, with the use of the Euclidean distance method, calculates the distance

between data points and centroids, and assign data point to the cluster which is close to its centroid. Re calculate the cluster centroids as a mean of data points attached to it. Repeat until no further changes occur.

e) **Types of Dimensionality Reduction Algorithms:-**There are many dimensionality reduction algorithms in machine learning, which are applied for different dimensionality reduction applications. One of the main dimensionality reduction techniques is Principal Component Analysis (PCA) / Factor Analysis.

f) **Principal Component Analysis (Factor Analysis):-** Principal Component Analysis is one of the algorithms of Dimensionality reduction. In this technique, it transforms data into a new set of variables from input variables, which are the linear combination of real variables. These Specific new set of variables are known as principal components. As a result of the transformation, the first primary component will have the most significant possible variance, and each following component in has the highest possible variance under the constraint that it is orthogonal to the above components. Keeping only the best m < n components, reduces the data dimensionality while retaining most of the data information.

2.4 Choosing and comparing models through Pipelines

When you work on machine learning project, you often end up with multiple good models to choose from. Each model will have different performance characteristics. Using resampling methods like k-fold cross validation; you can get an estimate of how accurate each model may be on unseen data. You need to be able to use these estimates to choose one or two best models from the suite of models that you have created.

2.4.1) Model Validation

When you are building a predictive model, you need to evaluate the capability or generalization power of the model on unseen data. This is typically done by estimating accuracy using data that was not used to train the model, often referred as cross validation.

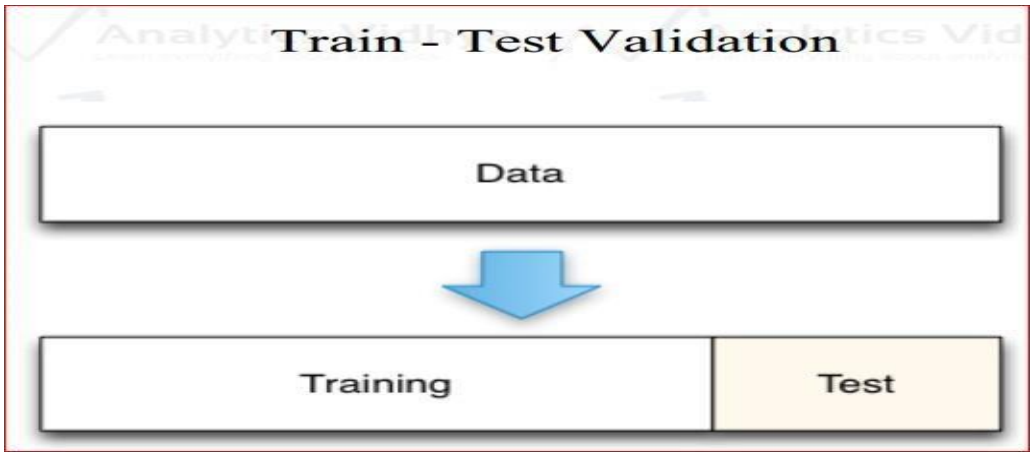


Fig 2.4.1 Model Validation

A few common methods used for Cross Validation:

1) The Validation set Approach (Holdout Cross validation)

In this approach, we reserve large portion of dataset for training and rest remaining portion of the data for model validation. Ideally people will use 70-30 or 80-20 percentages for training and validation purpose respectively.
A major disadvantage of this approach is that, since we are training a model on a randomly chosen portion of the dataset, there is a huge possibility that we might miss-out on some interesting information about the data which, will lead to a higher bias.

2) K-fold crossvalidation:

As there is never enough data to train your model, removing a part of it for validation may lead to a problem of under fitting. By reducing the training data, we risk losing important patterns/ trends in data set, which in turn increases error induced by bias. So, what we require is a method that provides ample data for training the model and also leaves ample data for validation. K Fold cross validation does exactly that.
In K Fold cross validation, the data is divided into k subsets. Now the holdout method is repeated k times, such that each time, one of the k subsets is used as the test set/ validation set and the other k-1 subsets are put together to form a training set. The error estimation is averaged over all k trials to get total effectiveness of our model. As can be seen, every data point gets to be in a validation set exactly once, and gets to be in a training set k-1 times. This significantly reduces the bias as we are using most of the data for fitting, and also significantly reduces variance as most of the data is also being used in validation set. Interchanging the training and test sets also adds to the effectiveness of this method. As a general rule and empirical evidence, K = 5 or 10 is preferred, but nothing's fixed and it can take any value.

Below are the steps for it:

• Randomly split your entire dataset into k"folds" for each k-fold in your dataset, build your model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for k_{th} fold.

- Record the error you see on each of the predictions.
- Repeat this until each of the k-folds has served as the test set.
- The average of your k recorded errors is called the cross-validation error and will serve as your performance metric for the model.

Below is the visualization of a k-fold validation when k=5.

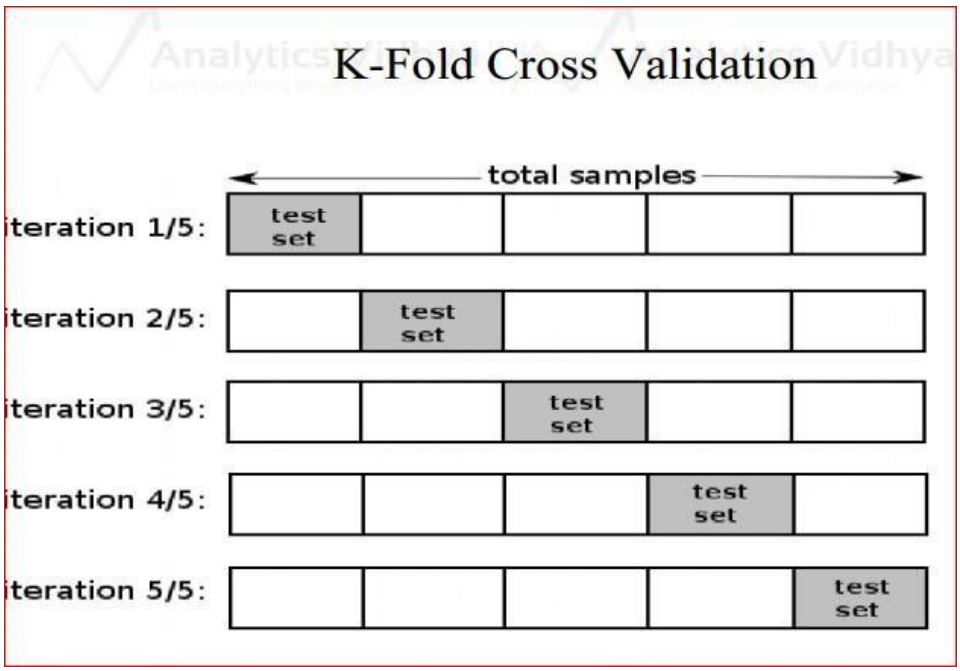


Fig 2.4.1K-fold cross validation

How to choose K:

- Smaller dataset: 10-fold cross validation is better
- Moderate dataset: 5 or 6 fold cross validation works mostly
- Big dataset: Train – Val split for validation

Other than this, we have Leave one out cross validation (LOOCV), in which each record will be left over from the training and then, the same will be used for testing purpose. This process will be repeated across all the respondents.

2.5 Model Diagnosis with over fitting and under fitting

2.5.1) Bias and Variance

A fundamental problem with supervised learning is the bias variance trade-off. Ideally, a model should have two key characteristics

1) Sensitive enough to accurately capture the key patterns in the training dataset.

2) Generalized enough to work well on any unseen dataset.

Unfortunately, while trying to achieve the above-mentioned first point, there is an ample chance of over-fitting to noisy or unrepresentative training data points leading to a failure of generalizing the model. On the other hand, trying to generalize a model may result in failing to capture important regularities.
If model accuracy is low on a training dataset as well as test dataset, the model is said to be under-fitting or that the model has high bias. The **Bias** refers to the simplifying assumptions made by the algorithm to make the problem easier to solve. To solve an under-fitting issue or to reduce bias, try including more meaningful features and try to increase the model complexity by trying higher-order interactions
The **Variance** refers to sensitivity of a model changes to the training data. A model is giving high accuracy on a training dataset, however on a test dataset the accuracy drops drastically then, the model is said to be over-fitting or a model that has high variance.
To solve the over-fitting issue Try to reduce the number of features, that is, keep only the meaningful features or try regularization methods that will keep all the features. Ideal model will be the trade-off between Under fitting and over fitting like mentioned in the below picture.

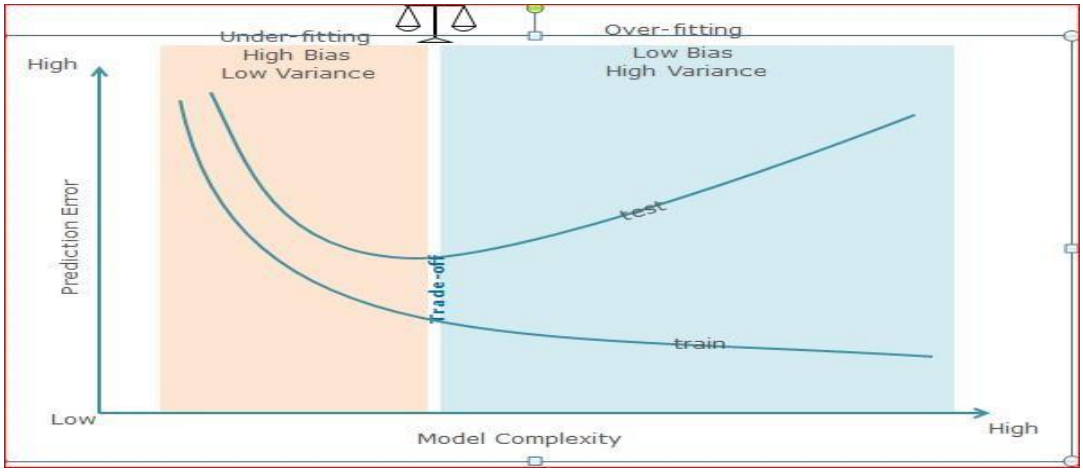


Fig 2.5.1 Bias and Variance

And, the Hyper parameters will be tuned in the below mentioned ways to reach the optimal solution:

1) Grid Search
2) Random Search
3) Manual Tuning

2.5.2) Model Performance Matrix

Model evaluation is an integral part of the model development. Based on model evaluation and subsequent comparisons, we can take a call whether to continue our efforts in model enhancement or cease them and select the final model that should be used / deployed.

1. Evaluating Classification Models

Confusion Matrix

Confusion matrix is one of the most popular ways to evaluate a classification model. A confusion matrix can be created for a binary classification as well as a multi-class classification model.

A confusion matrix is created by comparing the predicted class label of a data point with its actual class label. This comparison is repeated for the whole dataset and the results of this comparison are compiled in a matrix or tabular format

Table 2.5.2 Confusion Matrix

		Predicted classed		
		Positive (C_0)	Negative (C_1)	
Actual class	Positive (C_0)	a = number of correctly Classified c0 cases	c = number of c_0 cases Incorrectly classified as c_1	Precision = a/(a + c)
	Negative (C_0)	b = number of c_1 cases Incorrectly classified as c_0	d = number of correctly classified c_1 cases	
		Sensitivity (Recall) = a/(a+b)	Specificity = d/c+d	Accuracy = (a+b)(a+b+c+d)

Specificity : The ratio of actual negative cases that are identified correctly.

Table 5-3 shows an example confusion matrix.

Table 5-3. Example of classifications Accuracy measurement

		Predicted classed		
		Positive (C_0)	Negative (C_1)	
Actual class	Positive (C_0)	80	30	Precision = 70/110=0.63
	Negative (C_1)	40	90	
		Recall=80/120=0.67	Specificity = 90/240=0.75	Accuracy = 80+90/240=0.71

And, below are the various measures that will be used to assess the performance of the model based on the requirement of the problem and as well as data.

Metric	Description	Formula
Accuracy	What% of predictions were Correct?	(TP + TN)/(TP + TN + EP + FN)
Misclassification rate	What % of prediction is wrong?	(FP + FN)/(TP + TN + FP + FN)
True positive rate OR Sensitivity or recall (completeness)	What % of positive cases did Model catch?	TP/(FN + TP)
False positive Rate	What % 'NO' were predicted as 'Yes'?	FP/FP+TN)
Specificity	What % 'NO' were predicted as 'NO'?	TN/(TN + FP)
Precision(exactness)	What % of positive predictions Were correct?	TP(TP + FP)
FI score	Weighted average of precision And recall	2*((precision*recall)/ (precision + recall))

2. Regression Model Evaluation

A regression line predicts the y values for a given x value. Note that the values are around the average. The prediction error (called as root-mean-square error or RSME) is given by the following formula:

$$RMSE = \sqrt{\frac{\sum_{k=0}^{n} \left(\hat{Y}_k - y_k \right)^2}{n}}$$

And, the regression will also assessed by R square (Co efficient of determination).

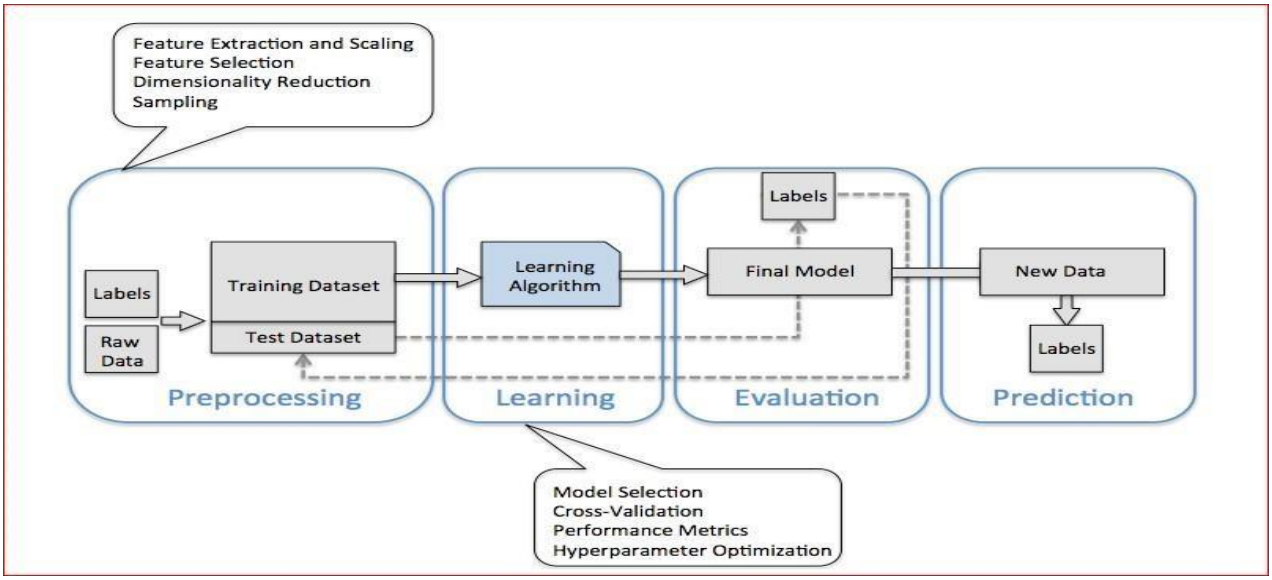
3. Evaluating Unsupervised Models

The Unsupervised algorithms will be assessed by the profile of the factors/ clusters which were derived through the models.

2.6 Overall Process of Machine Learning

To put overall process together, below is the picture that describes the road map for building ML Systems

Table 2.6 Overall Process of Machine Learning



CHAPTER 3

Multivariate Analysis - At Work

Multivariate Analysis - At Work

3.1 An Approach to the Problem:

In order to carry out the analysis, we have extracted 768 recordsfrom the female patients of at least 21 years oldand the information of the same is mentioned inChapter 1.

In this Chapter, we are going to discuss about the results of different Machine Learning methods used in order to obtain the solution for the problem mentioned in Chapter1.

As mentioned in Chapter 2, the first step of a ML Algorithm is Data cleaning and preparing data for the modelling. As a first step, we have to check whether the data was read properly and all the scale types are as per the data.

Structure of the data:

In this we are converting categorical variables into factor variables

```
'data.frame': 768 obs. of  9 variables:
 $ preg_times   : int  6 1 8 1 0 5 3 10 2 8 ...
 $ glucose_test : int  148 85 183 89 137 116 78 115 197 125 ...
 $ blood_press  : int  72 66 64 66 40 74 50 0 70 96 ...
 $ tsk_thickness: int  35 29 0 23 35 0 32 0 45 0 ...
 $ serum        : int  0 0 0 94 168 0 88 0 543 0 ...
 $ bm_index     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
 $ pedigree_fun : num  0.627 0.351 0.672 0.167 2.288 ...
 $ age          : int  50 31 32 21 33 30 26 29 53 54 ...
 $ dep          : Factor w/ 2 levels "0","1": 2 1 2 1 2 1 2 1 2 2 ...
```

3.1 Output**Structure of the data**

From the above dependent variable converted in to factor

Understanding data using Descriptive Statistics:

```
> summary(dataset)
   preg_times      glucose_test     blood_press     tsk_thickness       serum          bm_index
 Min.   : 0.000   Min.   : 44.00   Min.   : 24.00   Min.   : 7.00   Min.   :  0.00   Min.   :18.20
 1st Qu.: 1.000   1st Qu.: 99.75   1st Qu.: 64.00   1st Qu.:25.00   1st Qu.:  0.00   1st Qu.:27.50
 Median : 3.000   Median :117.00   Median : 72.20   Median :29.15   Median : 30.50   Median :32.40
 Mean   : 3.742   Mean   :121.69   Mean   : 72.41   Mean   :29.15   Mean   : 72.46   Mean   :32.46
 3rd Qu.: 6.000   3rd Qu.:140.25   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.25   3rd Qu.:36.60
 Max.   :10.000   Max.   :199.00   Max.   :122.00   Max.   :99.00   Max.   :293.00   Max.   :67.10
  pedigree_fun         age        dep
 Min.   :0.0780   Min.   :21.00   0:500
 1st Qu.:0.2437   1st Qu.:24.00   1:268
 Median :0.3725   Median :29.00
 Mean   :0.4719   Mean   :33.24
 3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :2.4200   Max.   :81.00
```

3.1 Output Descriptive Statistics

Form the above we can understand the measures of location (spread) mean, median, minimum, maximum etc. And proportion of the data .

Checking for missing Values:

Then, check if there are any missing values in the data

```
 preg_times  glucose_test   blood_press tsk_thickness         serum      bm_index  pedigree_fun
          0             0             0             0             0             0             0
        age           dep
          0             0
```

3.1 outputChecking for missing Values

The missing values for the continuous variables will be imputed using Mean / Median value of the valid records and the categorical variables will be imputed using Mode value.

Form the above table we observe there are no missing values in the dataSo there is no need of imputing.

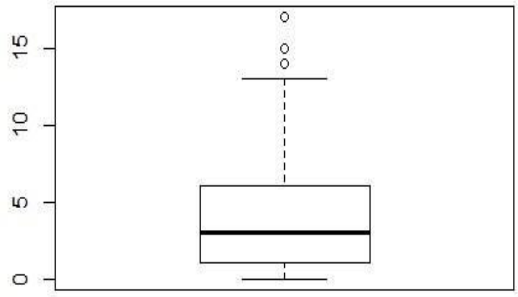
Checking for Outliers:

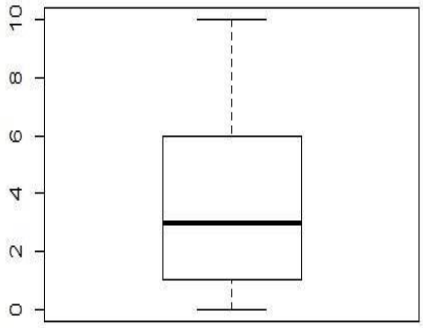
We used Box-plots to check for Outliers in each of the continuous variables.

Boxplot for preg_times:

With outliers

without outliers



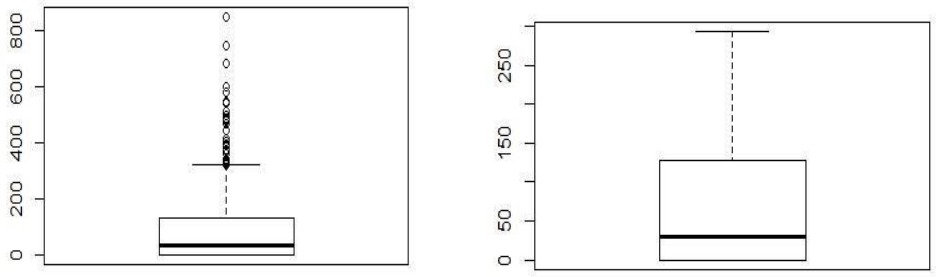


3.1. With outliers **3.1.without outliers**

Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile will be imputed using 5th percentile value.

Boxplot for serum

With outliers **without outliers**



3.1. with outliers **3.1.without outliers**

Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile will be imputed using 5th percentile value.

Understanding data visually:

Also, look at the data visually to understand the relationships between and within the variables

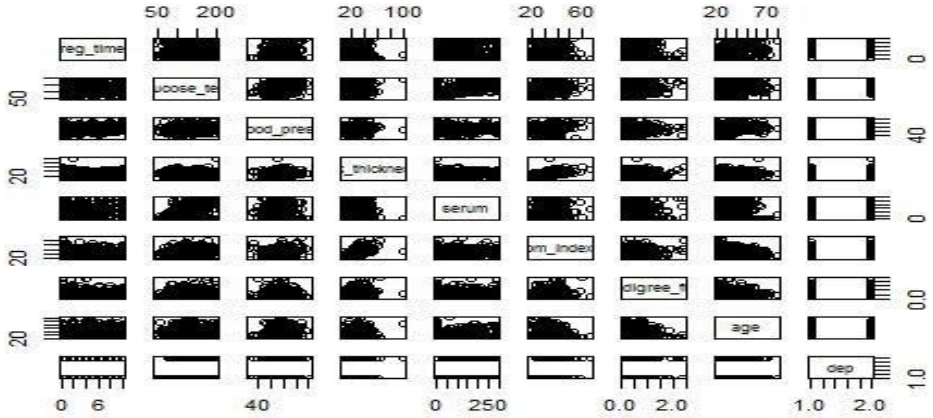


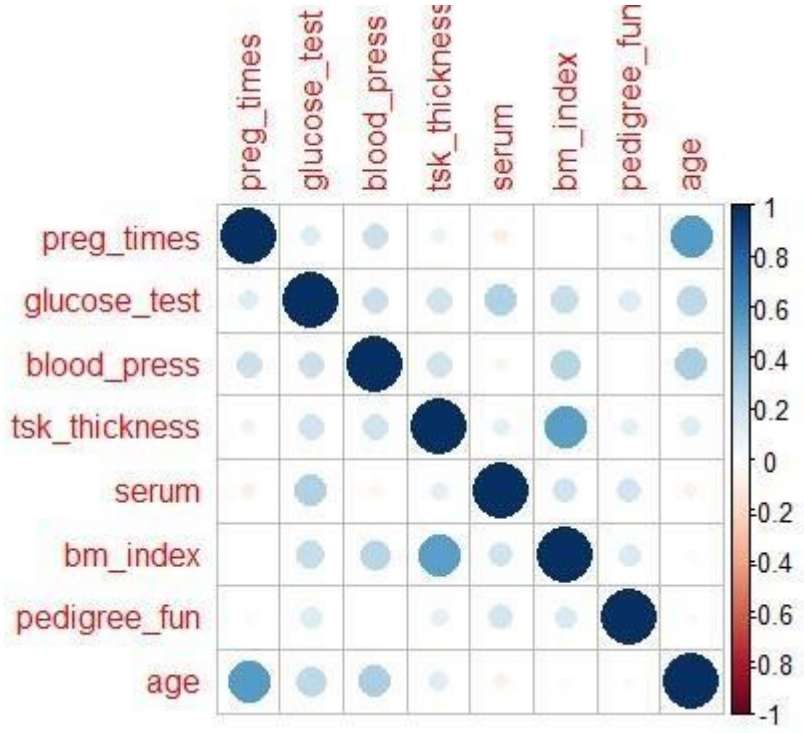
Fig 3.1 data visuall

Preg_timesvsdep:

From the plot, we can understand that there is no relationship between preg_times and dep.

Understanding relationships between variables:

For the continuous variables, we will look at the Correlation plots between variables to understand the relationships between variables.



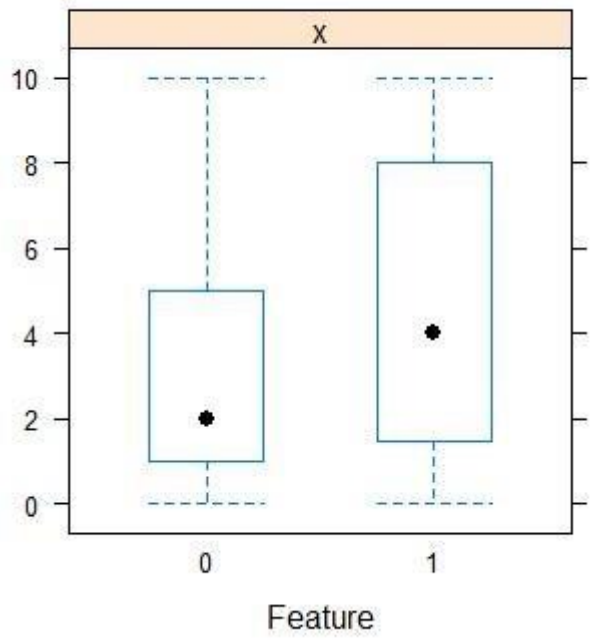
3.1 relationships between variables

Here, the circle size refers to the strength of the relation and colour refers to the direction of the relationship.

From the plot, we can see that preg_times and age,bm_index and tsk_thickness are highly positively correlated.

For the continuous vs categorical variable, we will look t Feature plots to understand the relationships

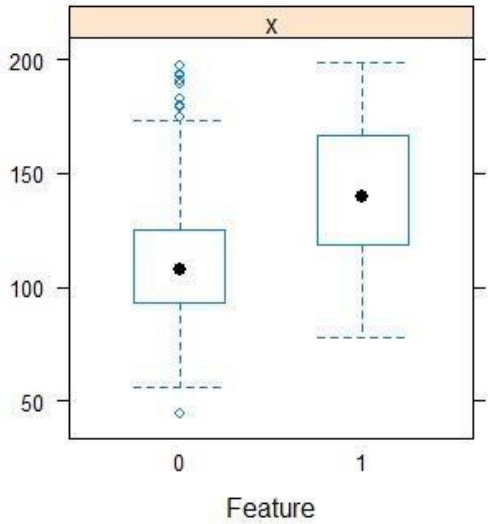
Preg_timesvsdep:



3.1.Feature plot

From the plot, we can understand that there is no relatioship between preg_timesanddep

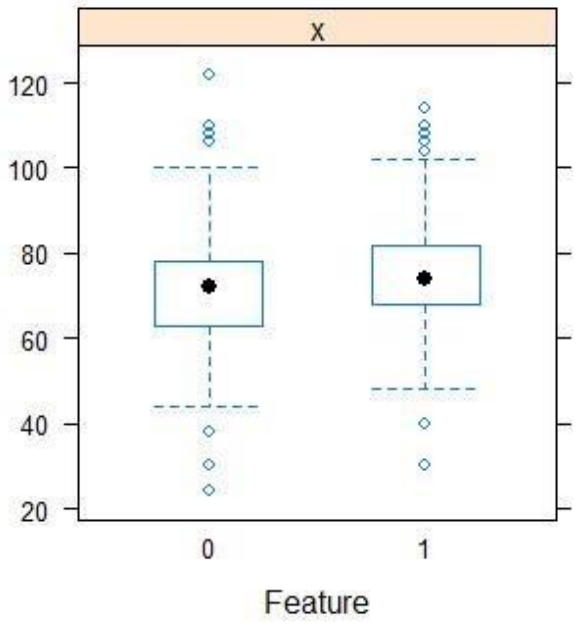
Glucose_test vs dep:



3.1.Feature plot

From the plot, we can understand that there is no relationship between glucose_testand dep.

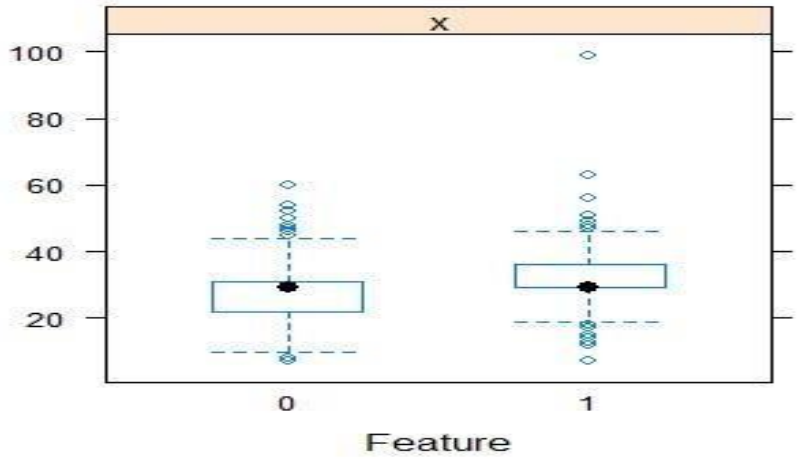
Blood_press vs dep:



3.1.Feature plot

From the plot, we can understand that there is no relationship between blood_pressand dep.

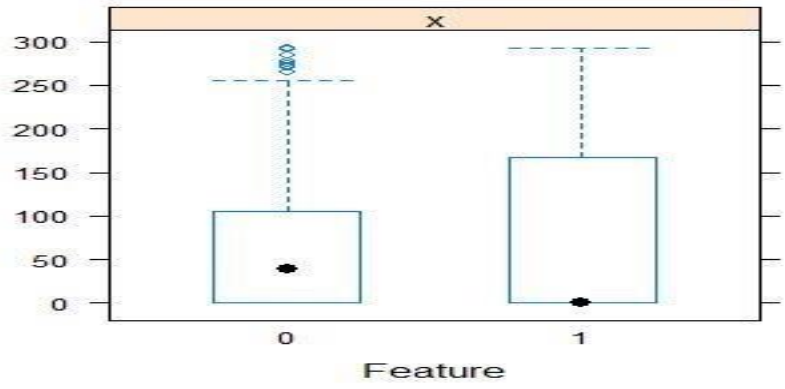
Tsk_thickness vs dep:



3.1.Feature plot

From the plot, we can understand that there is no relationship between Tsk_thicknessand dep.

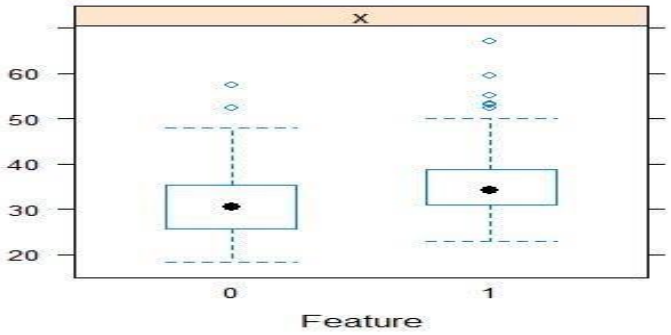
Serum vsdep:



3.1.Feature plot

From the plot, we can understand that there is no relationship between Serum &dep

BM_index vs dep:

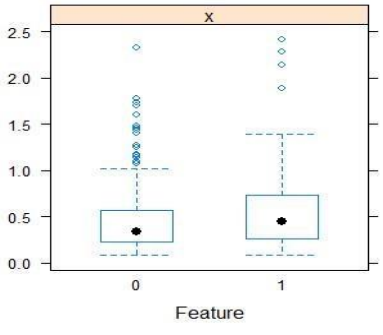


3.1.Feature plot

From the plot, we can understand that there is no relationship between BM_indexand dep.

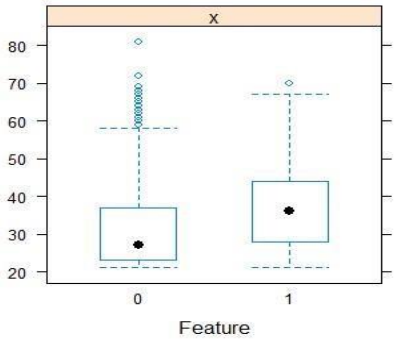
Pedigree_fun vs dep:

3.1.Feature plot



From the plot, we can understand that there is no relationship between Pedigree_funand dep.

Age vsdep:



3.1.Feature plot

From the plot, we can understand that there is no relationship between Age and dep.

Standardizing :

As we have different types of units ideaaly we need to do standardization …

Cross validation :

In order to carry out the anlaysis we have to split the data into training and test with 70% and 30%

```
> dim(scale_training)
[1] 768   9
> dim(training)
[1] 691   9
> dim(test)
[1] 77  9
```

3.1.Output of cross validation

From training we have 691 observations and 9 variables and by testing we have 77 observations and 9 variables

Running Pipeline using k-fold validation:

To predict the data we fit appropriate algorithms as follows .

Decision Trees:

The below information gives output of data by running decision trees

```
691 samples
  8 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 622, 622, 621, 621, 622, 623, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.01224490  0.7396062  0.4193355
  0.02346939  0.7414891  0.4020547
  0.29795918  0.6797120  0.1561668

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.02346939.
```

3.1.Output of Decision Tree

Logistic Regression:

The below information gives output of data by running logistic regression

```
Generalized Linear Model

691 samples
  8 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 622, 622, 621, 621, 622, 623, ...
Resampling results:

  Accuracy   Kappa
  0.7707855  0.4743456
```

3.1.Output ofrelationships between variables

Support vector machine:

The below information gives output of data by running support vector machine

```
No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 622, 622, 621, 621, 622, 623, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.7650503  0.4526644
  0.50  0.7645331  0.4527643
  1.00  0.7612207  0.4472905

Tuning parameter 'sigma' was held constant at a value of 0.1064177
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.1064177 and C = 0.25.
```

3.1.Output of relationships between variables

K- nearest neighbourhood:

The below information gives output of data by running KNN

```
691 samples
  8 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 622, 623, 622, 621, 622, 621, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.7351080  0.3975934
  7  0.7486792  0.4270666
  9  0.7529294  0.4373998
```

3.1.Output of K- nearest neighbourhood

RANDOM FOREST:

The below information gives output of data by running random forest

```
691 samples
  8 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 622, 623, 622, 621, 622, 621, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.7501555  0.4380910
  5     0.7409972  0.4208768
  8     0.7356757  0.4086970
```

3.1.Output RANDOM FOREST

Collect resample:

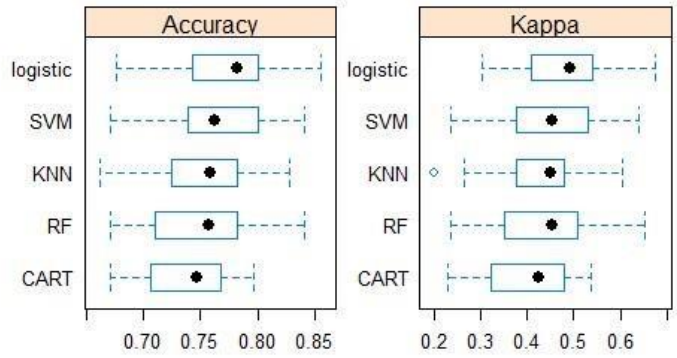
From below we can choose the best model among all the models from our pipeline which consist of CART, logistic,SVM , KNN ,RF

```
Models: CART, logistic, SVM, KNN, RF
Number of resamples: 30

Accuracy
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART     0.6714286 0.7069480 0.7464286 0.7414891 0.7681159 0.7971014    0
logistic 0.6764706 0.7446429 0.7826087 0.7707855 0.7992754 0.8550725    0
SVM      0.6714286 0.7391304 0.7626294 0.7650503 0.7992754 0.8405797    0
KNN      0.6617647 0.7246377 0.7591645 0.7529294 0.7798137 0.8285714    0
RF       0.6714286 0.7137681 0.7571429 0.7501555 0.7818095 0.8405797    0

Kappa
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
CART     0.2289417 0.3217956 0.4246095 0.4020547 0.4767545 0.5374449    0
logistic 0.3034826 0.4126999 0.4910962 0.4743456 0.5350666 0.6751412    0
SVM      0.2369668 0.3816618 0.4540343 0.4526644 0.5304054 0.6394299    0
KNN      0.2029131 0.3810714 0.4522024 0.4373998 0.4788954 0.6056338    0
RF       0.2379518 0.3536892 0.4553194 0.4380910 0.5068276 0.6519945    0
```

3.1.Output Collect resample

From the above table random forest is the best algorithm

Box plot in accuracy and kappa:



Tunning random forest:

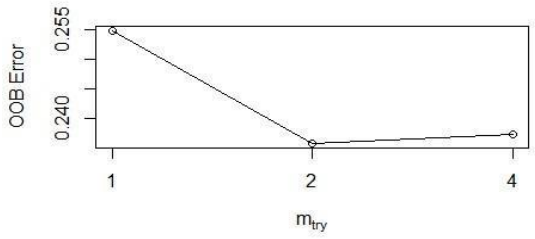
In order to find the key variable, we have applied random forest and tuned the model and find the variable importance.

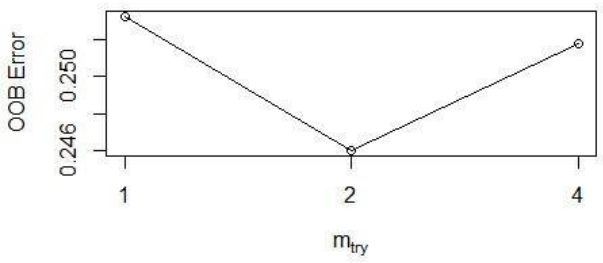
```
> print(bestmtry)
      mtry  OOBError
1.OOB    1 0.2532562
2.OOB    2 0.2460203
4.OOB    4 0.2518090
```

3.1.Output of Tunning random forest

Finding the key variables using random forest:

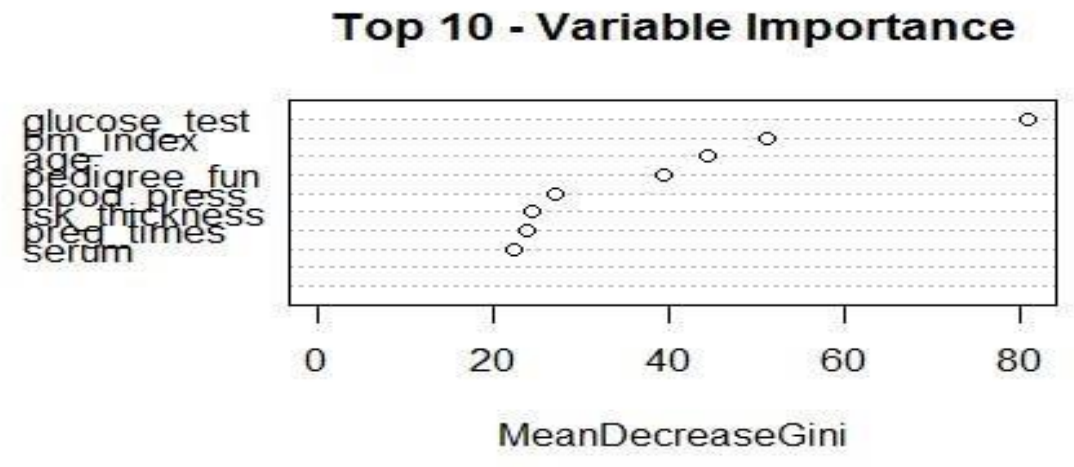
In order to find the key variable we run the RF using grid search and obtain key variable and then used the logistic regression with these key variables and obtained the outputs





Final model:

From the key variables i.e ,(preg_times , glucose_test , blood_press , tsk_thickness , Serum ,bm_index , pedigree_fun , age) we run the model as given below.



3.1.Final model

Now we build the final model with this significant variables and tested accuracy for both train and test data.

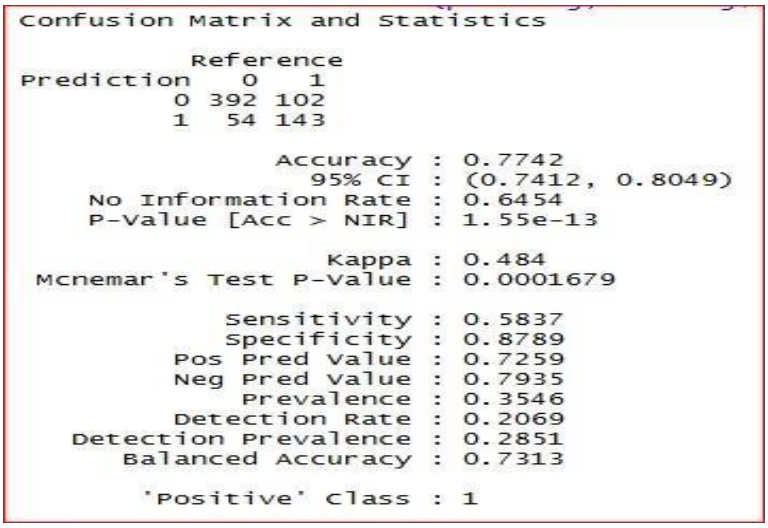
Final model With Logistic Regression:

```
Generalized Linear Model

691 samples
  4 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 621, 621, 622, 622, 621, 622, ...
Resampling results:

  Accuracy   Kappa
  0.7680323  0.469722
```

3.1.Output of Final model With Logistic Regression

From above data we have generalized linear model by using logistic regression .we got the accuracy of 77 %.

```
Confusion Matrix and Statistics (Training):
```

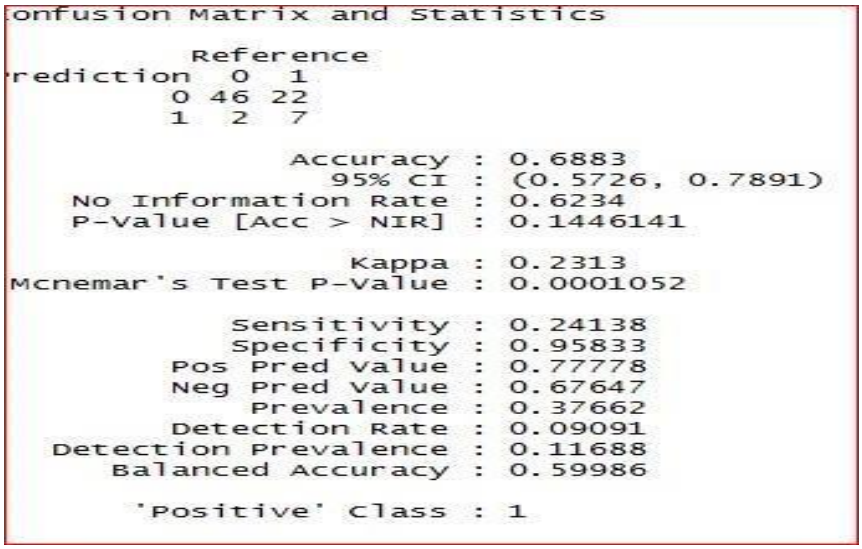


```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 392 102
         1  54 143

               Accuracy : 0.7742
                 95% CI : (0.7412, 0.8049)
    No Information Rate : 0.6454
    P-Value [Acc > NIR] : 1.55e-13

                  Kappa : 0.484
 Mcnemar's Test P-Value : 0.0001679

            Sensitivity : 0.5837
            Specificity : 0.8789
         Pos Pred Value : 0.7259
         Neg Pred Value : 0.7935
             Prevalence : 0.3546
         Detection Rate : 0.2069
   Detection Prevalence : 0.2851
      Balanced Accuracy : 0.7313

       'Positive' Class : 1
```

3.1.Output of Confusion Matrix and Statistics(Training)

By training the data we got accuracy of 77 %

Confusion Matrix and Statistics **(On testing) :**



```
onfusion Matrix and Statistics

          Reference
rediction  0  1
         0 46 22
         1  2  7

               Accuracy : 0.6883
                 95% CI : (0.5726, 0.7891)
    No Information Rate : 0.6234
    P-Value [Acc > NIR] : 0.1446141

                  Kappa : 0.2313
Mcnemar's Test P-Value : 0.0001052

            Sensitivity : 0.24138
            Specificity : 0.95833
         Pos Pred Value : 0.77778
         Neg Pred Value : 0.67647
             Prevalence : 0.37662
         Detection Rate : 0.09091
   Detection Prevalence : 0.11688
      Balanced Accuracy : 0.59986

       'Positive' Class : 1
```

3.1.Output of Confusion Matrix and Statistics **(On testing)**

By testing the data we got accuracy of 69 %

Chapter 4

SUMMARY

SUMMARY:

For the given data we have applied the pipe line of techniques and finalised the best model. We also identified the best model and the key variable using random forest and applied the linear model by using model we got the train and test data.

ACCURACY OF TRAIN DATA IS 77 %

ACCURACY OF TEST DATA IS 69 %

Hence by training and testing the data we got almost similar accuracy. So, we can conclude that our obtained model is generalized linear model

Chapter 5
R-CODE
DATA SET

R - Code

 # Setting working directory getwd()

setwd("D:/batch10/project 10th batch") str(dataset)

```
# Laoding required libraries to perform modeling
library(mice) library(randomForest) library(ggplot2)
library(glmnet)
########################################################################
# reading data from Folder and checking for the data types dataset<-
read.csv(file.choose(), header = T) str(dataset) summary(dataset)
############################################### # we can also set/change the
variable names using colnames colnames(dataset) <- c("preg_times",
"glucose_test", "blood_press", "tsk_thickness",
                "serum", "bm_index", "pedigree_fun", "age", "dep")
dataset$dep<- as.factor(dataset$dep) str(dataset) dim(dataset) head(dataset)
tail(dataset)
#Checking for Missing value print(all(!is.na(dataset))) #Missing
value  Proportion for all the variables sapply(dataset,
function(df) {
  (sum(is.na(df)==TRUE)/ length(df))*100;})
##############################################
# As Missing value is coded 0 we have mention 0 as missing value  cols_change<-
colnames(dataset)[!colnames(dataset) %in% c("preg_times","serum", "dep")] bool_data<-
dataset[cols_change] == 0 dataset[cols_change][bool_data] <- NA print(all(!is.na(dataset)))
#Missing value  Proportion for all the variables sapply(dataset, function(df)
{
  (sum(is.na(df)==TRUE)/ length(df))*100;})

# Missing values impute for small proportion of missing values library(Hmisc)
dataset$glucose_test[is.na(dataset$glucose_test)]<-mean(dataset$glucose_test,na.rm=T)
dataset$blood_press[is.na(dataset$blood_press)]<-mean(dataset$blood_press,na.rm=T)
dataset$tsk_thickness[is.na(dataset$tsk_thickness)]<-mean(dataset$tsk_thickness,na.rm=T)
```

```
dataset$bm_index[is.na(dataset$bm_index)]<-mean(dataset$bm_index,na.rm=T)
sum(is.na(dataset)) dataset$glucose_test=as.numeric(dataset$glucose_test)
dataset$blood_press=as.numeric(dataset$blood_press)
dataset$tsk_thickness=as.numeric(dataset$tsk_thickness)
dataset$bm_index=as.numeric(dataset$bm_index) str(dataset)
write.csv(dataset,"afterimp.csv")
boxplot(dataset$preg_times)  boxplot(dataset$serum)
dataset$preg_times[dataset$preg_times>quantile(dataset$preg_times, 0.95)] <-
quantile(dataset$preg_times, 0.95)
dataset$preg_times[dataset$preg_times<quantile(dataset$preg_times, 0.05)] <-
quantile(dataset$preg_times, 0.05) dataset$serum[dataset$serum>quantile(dataset$serum,
0.95)] <- quantile(dataset$serum, 0.95) dataset$serum[dataset$serum<quantile(dataset$serum,
0.05)] <- quantile(dataset$serum, 0.05) boxplot(preg_times ~ dep, data=dataset,
main="preg_timesvsdep ",  ylab="preg_times") pairs(dataset) pre1=dataset[c(1:8)] library(corrplot)
pre1.cor          =          cor(pre1)
corrplot(pre1.cor,          method="circle")
################################################
```

 # Continuous vs categories

```
library(caret) x <- dataset[,1:8] y
<- dataset[,9] featurePlot(x=x, y=y,
plot="box")
############################
############################
###############
```

 # Normalizing input data scale_training<-

```
as.data.frame(scale(dataset[, -9],   center = TRUE, scale
= TRUE)) scale_training$dep<- dataset[,
```

"dep"] names(scale_training)

table(scale_training$dep)

write.csv(scale_training,"scaledata.csv")

library(caret) x <- scale_training[,1:8] y <-

scale_training[,9] featurePlot(x=x, y=y, plot="box")

 # Splting data train_rows<- sample(1:nrow(scale_training),

size=0.9*nrow(scale_training)) train_rows training<- scale_training[train_rows,

] test<- scale_training[-train_rows,] dim(scale_training) dim(training) dim(test)

head(test) **# Model Pipleline** trainControl<-

trainControl(method="repeatedcv", number=10, repeats=3)

 # CART #DT set.seed(100) fit.cart<- train(dep~.,

data=training, method="rpart",

trControl=trainControl) summary(fit.cart) **#Logistic**

Regression set.seed(100) fit.glm<-train(dep~.,

data=training, method="glm",

trControl=trainControl) summary(fit.glm)

print(fit.glm)

#SVm set.seed(100) fit.svm<- train(dep~.,

data=training, method="svmRadial",

trControl=trainControl) print(fit.svm)

 # KNN set.seed(7) fit.knn<- train(dep~., data=training, method="knn", trControl=trainControl)

print(fit.knn)

 # Random Forest set.seed(7) fit.rf<- train(dep~., data=training, method="rf",

trControl=trainControl)

#collect resamples results<- resamples(list(CART=fit.cart, logistic=fit.glm, SVM=fit.svm, KNN=fit.knn, RF=fit.rf)) summary(results) scales<- list(x=list(relation="free"), y=list(relation="free")) bwplot(results, scales=scales)

 # Tuning random forest bestmtry<- tuneRF(training[, c(9)],training$dep, ntreeTry=300, stepFactor=2,improve=0.01, trace=TRUE, plot=TRUE, dobest=FALSE) print(bestmtry) **# Final model** rf_model3 <- randomForest(dep ~ ., data =training, ntree=300, mtry=2) varImpPlot(rf_model3, sort = T, n.var = 10, main = "Top 10 - Variable Importance") importance(rf_model3)

#$Final model With Logistic Regression dim(training) dim(test) trainControl<- trainControl(method="repeatedcv", number=10, repeats=3) final.glm<-train (dep ~ glucose_test + bm_index + age + pedigree_fun, data=training, method="glm", trControl=trainControl) summary(final.glm) print (final.glm) predslog<- predict(final.glm, data=training, type = "raw") tabtrain<- table(Predicted = predslog, Actual = training$dep) caret::confusionMatrix(predslog,training$dep,positive="1")

#On testing dim(test) names(test) p2 <- predict(final.glm,newdata=test,type="raw") tabtest<- table(Predicted = p2, Actual = test$dep) caret::confusionMatrix(p2,test$dep,positive="1

DATA SET

Data :

S.NO	pregnant	glucose	pressure	triceps	Insulin	mass	pedigree	Age	diabetes
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31	0.248	26	1
8	10	115	0	0	0	35.3	0.134	29	0
9	2	197	70	45	543	30.5	0.158	53	1
10	8	125	96	0	0	0	0.232	54	1
11	4	110	92	0	0	37.6	0.191	30	0
12	10	168	74	0	0	38	0.537	34	1
13	10	139	80	0	0	27.1	1.441	57	0
14	1	189	60	23	846	30.1	0.398	59	1
15	5	166	72	19	175	25.8	0.587	51	1
16	7	100	0	0	0	30	0.484	32	1
17	0	118	84	47	230	45.8	0.551	31	1
18	7	107	74	0	0	29.6	0.254	31	1
19	1	103	30	38	83	43.3	0.183	33	0
20	1	115	70	30	96	34.6	0.529	32	1
21	3	126	88	41	235	39.3	0.704	27	0
22	8	99	84	0	0	35.4	0.388	50	0

23	7	196	90	0	0	39.8	0.451	41	1
24	9	119	80	35	0	29	0.263	29	1
25	11	143	94	33	146	36.6	0.254	51	1
26	10	125	70	26	115	31.1	0.205	41	1
27	7	147	76	0	0	39.4	0.257	43	1
28	1	97	66	15	140	23.2	0.487	22	0
29	13	145	82	19	110	22.2	0.245	57	0
30	5	117	92	0	0	34.1	0.337	38	0
31	5	109	75	26	0	36	0.546	60	0
32	3	158	76	36	245	31.6	0.851	28	1
33	3	88	58	11	54	24.8	0.267	22	0
34	6	92	92	0	0	19.9	0.188	28	0

35	10	122	78	31	0	27.6	0.512	45	0
36	4	103	60	33	192	24	0.966	33	0
37	11	138	76	0	0	33.2	0.42	35	0
38	9	102	76	37	0	32.9	0.665	46	1
39	2	90	68	42	0	38.2	0.503	27	1
40	4	111	72	47	207	37.1	1.39	56	1
41	3	180	64	25	70	34	0.271	26	0
42	7	133	84	0	0	40.2	0.696	37	0
43	7	106	92	18	0	22.7	0.235	48	0
44	9	171	110	24	240	45.4	0.721	54	1
45	7	159	64	0	0	27.4	0.294	40	0
46	0	180	66	39	0	42	1.893	25	1
47	1	146	56	0	0	29.7	0.564	29	0

48	2	71	70	27	0	28	0.586	22	0
49	7	103	66	32	0	39.1	0.344	31	1
50	7	105	0	0	0	0	0.305	24	0
51	1	103	80	11	82	19.4	0.491	22	0
52	1	101	50	15	36	24.2	0.526	26	0
53	5	88	66	21	23	24.4	0.342	30	0
54	8	176	90	34	300	33.7	0.467	58	1
55	7	150	66	42	342	34.7	0.718	42	0
56	1	73	50	10	0	23	0.248	21	0
57	7	187	68	39	304	37.7	0.254	41	1
58	0	100	88	60	110	46.8	0.962	31	0
59	0	146	82	0	0	40.5	1.781	44	0
60	0	105	64	41	142	41.5	0.173	22	0
61	2	84	0	0	0	0	0.304	21	0
62	8	133	72	0	0	32.9	0.27	39	1
63	5	44	62	0	0	25	0.587	36	0
64	2	141	58	34	128	25.4	0.699	24	0
65	7	114	66	0	0	32.8	0.258	42	1
66	5	99	74	27	0	29	0.203	32	0
67	0	109	88	30	0	32.5	0.855	38	1
68	2	109	92	0	0	42.7	0.845	54	0
69	1	95	66	13	38	19.6	0.334	25	0
70	4	146	85	27	100	28.9	0.189	27	0
71	2	100	66	20	90	32.9	0.867	28	1
72	5	139	64	35	140	28.6	0.411	26	0
73	13	126	90	0	0	43.4	0.583	42	1

74	4	129	86	20	270	35.1	0.231	23	0
75	1	79	75	30	0	32	0.396	22	0
76	1	0	48	20	0	24.7	0.14	22	0
77	7	62	78	0	0	32.6	0.391	41	0
78	5	95	72	33	0	37.7	0.37	27	0

79	0	131	0	0	0	43.2	0.27	26	1
80	2	112	66	22	0	25	0.307	24	0
81	3	113	44	13	0	22.4	0.14	22	0
82	2	74	0	0	0	0	0.102	22	0
83	7	83	78	26	71	29.3	0.767	36	0
84	0	101	65	28	0	24.6	0.237	22	0
85	5	137	108	0	0	48.8	0.227	37	1
86	2	110	74	29	125	32.4	0.698	27	0
87	13	106	72	54	0	36.6	0.178	45	0
88	2	100	68	25	71	38.5	0.324	26	0
89	15	136	70	32	110	37.1	0.153	43	1
90	1	107	68	19	0	26.5	0.165	24	0
91	1	80	55	0	0	19.1	0.258	21	0
92	4	123	80	15	176	32	0.443	34	0
93	7	81	78	40	48	46.7	0.261	42	0
94	4	134	72	0	0	23.8	0.277	60	1
95	2	142	82	18	64	24.7	0.761	21	0
96	6	144	72	27	228	33.9	0.255	40	0
97	2	92	62	28	0	31.6	0.13	24	0
98	1	71	48	18	76	20.4	0.323	22	0

99	6	93	50	30	64	28.7	0.356	23	0
100	1	122	90	51	220	49.7	0.325	31	1
101	1	163	72	0	0	39	1.222	33	1
102	1	151	60	0	0	26.1	0.179	22	0
103	0	125	96	0	0	22.5	0.262	21	0
104	1	81	72	18	40	26.6	0.283	24	0
105	2	85	65	0	0	39.6	0.93	27	0
106	1	126	56	29	152	28.7	0.801	21	0
107	1	96	122	0	0	22.4	0.207	27	0
108	4	144	58	28	140	29.5	0.287	37	0
109	3	83	58	31	18	34.3	0.336	25	0
110	0	95	85	25	36	37.4	0.247	24	1
111	3	171	72	33	135	33.3	0.199	24	1
112	8	155	62	26	495	34	0.543	46	1
113	1	89	76	34	37	31.2	0.192	23	0
114	4	76	62	0	0	34	0.391	25	0
115	7	160	54	32	175	30.5	0.588	39	1
116	4	146	92	0	0	31.2	0.539	61	1
117	5	124	74	0	0	34	0.22	38	1
118	5	78	48	0	0	33.7	0.654	25	0
119	4	97	60	23	0	28.2	0.443	22	0
120	4	99	76	15	51	23.2	0.223	21	0
121	0	162	76	56	100	53.2	0.759	25	1
122	6	111	64	39	0	34.2	0.26	24	0
123	2	107	74	30	100	33.6	0.404	23	0

124	5	132	80	0	0	26.8	0.186	69	0
125	0	113	76	0	0	33.3	0.278	23	1
126	1	88	30	42	99	55	0.496	26	1
127	3	120	70	30	135	42.9	0.452	30	0
128	1	118	58	36	94	33.3	0.261	23	0
129	1	117	88	24	145	34.5	0.403	40	1
130	0	105	84	0	0	27.9	0.741	62	1
131	4	173	70	14	168	29.7	0.361	33	1
132	9	122	56	0	0	33.3	1.114	33	1
133	3	170	64	37	225	34.5	0.356	30	1
134	8	84	74	31	0	38.3	0.457	39	0
135	2	96	68	13	49	21.1	0.647	26	0
136	2	125	60	20	140	33.8	0.088	31	0
137	0	100	70	26	50	30.8	0.597	21	0
138	0	93	60	25	92	28.7	0.532	22	0
139	0	129	80	0	0	31.2	0.703	29	0
140	5	105	72	29	325	36.9	0.159	28	0
141	3	128	78	0	0	21.1	0.268	55	0
142	5	106	82	30	0	39.5	0.286	38	0
143	2	108	52	26	63	32.5	0.318	22	0
144	10	108	66	0	0	32.4	0.272	42	1
145	4	154	62	31	284	32.8	0.237	23	0
146	0	102	75	23	0	0	0.572	21	0
147	9	57	80	37	0	32.8	0.096	41	0
148	2	106	64	35	119	30.5	1.4	34	0

149	5	147	78	0	0	33.7	0.218	65	0
150	2	90	70	17	0	27.3	0.085	22	0
151	1	136	74	50	204	37.4	0.399	24	0
152	4	114	65	0	0	21.9	0.432	37	0
153	9	156	86	28	155	34.3	1.189	42	1
154	1	153	82	42	485	40.6	0.687	23	0
155	8	188	78	0	0	47.9	0.137	43	1
156	7	152	88	44	0	50	0.337	36	1
157	2	99	52	15	94	24.6	0.637	21	0
158	1	109	56	21	135	25.2	0.833	23	0
159	2	88	74	19	53	29	0.229	22	0
160	17	163	72	41	114	40.9	0.817	47	1
161	4	151	90	38	0	29.7	0.294	36	0
162	7	102	74	40	105	37.2	0.204	45	0
163	0	114	80	34	285	44.2	0.167	27	0
164	2	100	64	23	0	29.7	0.368	21	0
165	0	131	88	0	0	31.6	0.743	32	1
166	6	104	74	18	156	29.9	0.722	41	1
167	3	148	66	25	0	32.5	0.256	22	0
168	4	120	68	0	0	29.6	0.709	34	0
169	4	110	66	0	0	31.9	0.471	29	0
170	3	111	90	12	78	28.4	0.495	29	0
171	6	102	82	0	0	30.8	0.18	36	1
172	6	134	70	23	130	35.4	0.542	29	1
173	2	87	0	23	0	28.9	0.773	25	0

174	1	79	60	42	48	43.5	0.678	23	0
175	2	75	64	24	55	29.7	0.37	33	0
176	8	179	72	42	130	32.7	0.719	36	1
177	6	85	78	0	0	31.2	0.382	42	0
178	0	129	110	46	130	67.1	0.319	26	1
179	5	143	78	0	0	45	0.19	47	0
180	5	130	82	0	0	39.1	0.956	37	1
181	6	87	80	0	0	23.2	0.084	32	0
182	0	119	64	18	92	34.9	0.725	23	0
183	1	0	74	20	23	27.7	0.299	21	0
184	5	73	60	0	0	26.8	0.268	27	0
185	4	141	74	0	0	27.6	0.244	40	0
186	7	194	68	28	0	35.9	0.745	41	1
187	8	181	68	36	495	30.1	0.615	60	1
188	1	128	98	41	58	32	1.321	33	1
189	8	109	76	39	114	27.9	0.64	31	1
190	5	139	80	35	160	31.6	0.361	25	1
191	3	111	62	0	0	22.6	0.142	21	0
192	9	123	70	44	94	33.1	0.374	40	0
193	7	159	66	0	0	30.4	0.383	36	1
194	11	135	0	0	0	52.3	0.578	40	1
195	8	85	55	20	0	24.4	0.136	42	0
196	5	158	84	41	210	39.4	0.395	29	1
197	1	105	58	0	0	24.3	0.187	21	0
198	3	107	62	13	48	22.9	0.678	23	1
199	4	109	64	44	99	34.8	0.905	26	1

200	4	148	60	27	318	30.9	0.15	29	1
201	0	113	80	16	0	31	0.874	21	0
202	1	138	82	0	0	40.1	0.236	28	0
203	0	108	68	20	0	27.3	0.787	32	0
204	2	99	70	16	44	20.4	0.235	27	0
205	6	103	72	32	190	37.7	0.324	55	0
206	5	111	72	28	0	23.9	0.407	27	0
207	8	196	76	29	280	37.5	0.605	57	1
208	5	162	104	0	0	37.7	0.151	52	1
209	1	96	64	27	87	33.2	0.289	21	0
210	7	184	84	33	0	35.5	0.355	41	1

211	2	81	60	22	0	27.7	0.29	25	0
212	0	147	85	54	0	42.8	0.375	24	0
213	7	179	95	31	0	34.2	0.164	60	0
214	0	140	65	26	130	42.6	0.431	24	1
215	9	112	82	32	175	34.2	0.26	36	1
216	12	151	70	40	271	41.8	0.742	38	1
217	5	109	62	41	129	35.8	0.514	25	1
218	6	125	68	30	120	30	0.464	32	0
219	5	85	74	22	0	29	1.224	32	1
220	5	112	66	0	0	37.8	0.261	41	1
221	0	177	60	29	478	34.6	1.072	21	1
222	2	158	90	0	0	31.6	0.805	66	1
223	7	119	0	0	0	25.2	0.209	37	0
224	7	142	60	33	190	28.8	0.687	61	0

225	1	100	66	15	56	23.6	0.666	26	0
226	1	87	78	27	32	34.6	0.101	22	0
227	0	101	76	0	0	35.7	0.198	26	0
228	3	162	52	38	0	37.2	0.652	24	1
229	4	197	70	39	744	36.7	2.329	31	0
230	0	117	80	31	53	45.2	0.089	24	0
231	4	142	86	0	0	44	0.645	22	1
232	6	134	80	37	370	46.2	0.238	46	1
233	1	79	80	25	37	25.4	0.583	22	0
234	4	122	68	0	0	35	0.394	29	0
235	3	74	68	28	45	29.7	0.293	23	0
236	4	171	72	0	0	43.6	0.479	26	1
237	7	181	84	21	192	35.9	0.586	51	1
238	0	179	90	27	0	44.1	0.686	23	1
239	9	164	84	21	0	30.8	0.831	32	1
240	0	104	76	0	0	18.4	0.582	27	0
241	1	91	64	24	0	29.2	0.192	21	0
242	4	91	70	32	88	33.1	0.446	22	0
243	3	139	54	0	0	25.6	0.402	22	1
244	6	119	50	22	176	27.1	1.318	33	1
245	2	146	76	35	194	38.2	0.329	29	0
246	9	184	85	15	0	30	1.213	49	1
247	10	122	68	0	0	31.2	0.258	41	0
248	0	165	90	33	680	52.3	0.427	23	0
249	9	124	70	33	402	35.4	0.282	34	0
250	1	111	86	19	0	30.1	0.143	23	0

251	9	106	52	0	0	31.2	0.38	42	0
252	2	129	84	0	0	28	0.284	27	0
253	2	90	80	14	55	24.4	0.249	24	0
254	0	86	68	32	0	35.8	0.238	25	0

255	12	92	62	7	258	27.6	0.926	44	1
256	1	113	64	35	0	33.6	0.543	21	1
257	3	111	56	39	0	30.1	0.557	30	0
258	2	114	68	22	0	28.7	0.092	25	0
259	1	193	50	16	375	25.9	0.655	24	0
260	11	155	76	28	150	33.3	1.353	51	1
261	3	191	68	15	130	30.9	0.299	34	0

262	3	141	0	0	0	30	0.761	27	1
263	4	95	70	32	0	32.1	0.612	24	0
264	3	142	80	15	0	32.4	0.2	63	0
265	4	123	62	0	0	32	0.226	35	1
266	5	96	74	18	67	33.6	0.997	43	0
267	0	138	0	0	0	36.3	0.933	25	1
268	2	128	64	42	0	40	1.101	24	0
269	0	102	52	0	0	25.1	0.078	21	0
270	2	146	0	0	0	27.5	0.24	28	1
271	10	101	86	37	0	45.6	1.136	38	1
272	2	108	62	32	56	25.2	0.128	21	0
273	3	122	78	0	0	23	0.254	40	0
274	1	71	78	50	45	33.2	0.422	21	0
275	13	106	70	0	0	34.2	0.251	52	0

276	2	100	70	52	57	40.5	0.677	25	0
277	7	106	60	24	0	26.5	0.296	29	1
278	0	104	64	23	116	27.8	0.454	23	0
279	5	114	74	0	0	24.9	0.744	57	0
280	2	108	62	10	278	25.3	0.881	22	0
281	0	146	70	0	0	37.9	0.334	28	1
282	10	129	76	28	122	35.9	0.28	39	0
283	7	133	88	15	155	32.4	0.262	37	0
284	7	161	86	0	0	30.4	0.165	47	1
285	2	108	80	0	0	27	0.259	52	1
286	7	136	74	26	135	26	0.647	51	0
287	5	155	84	44	545	38.7	0.619	34	0
288	1	119	86	39	220	45.6	0.808	29	1
289	4	96	56	17	49	20.8	0.34	26	0
290	5	108	72	43	75	36.1	0.263	33	0
291	0	78	88	29	40	36.9	0.434	21	0
292	0	107	62	30	74	36.6	0.757	25	1
293	2	128	78	37	182	43.3	1.224	31	1
294	1	128	48	45	194	40.5	0.613	24	1
295	0	161	50	0	0	21.9	0.254	65	0
296	6	151	62	31	120	35.5	0.692	28	0
297	2	146	70	38	360	28	0.337	29	1
298	0	126	84	29	215	30.7	0.52	24	0
299	14	100	78	25	184	36.6	0.412	46	1
300	8	112	72	0	0	23.6	0.84	58	0

301	0	167	0	0	0	32.3	0.839	30	1
302	2	144	58	33	135	31.6	0.422	25	1
303	5	77	82	41	42	35.8	0.156	35	0
304	5	115	98	0	0	52.9	0.209	28	1
305	3	150	76	0	0	21	0.207	37	0
306	2	120	76	37	105	39.7	0.215	29	0
307	10	161	68	23	132	25.5	0.326	47	1
308	0	137	68	14	148	24.8	0.143	21	0
309	0	128	68	19	180	30.5	1.391	25	1
310	2	124	68	28	205	32.9	0.875	30	1
311	6	80	66	30	0	26.2	0.313	41	0
312	0	106	70	37	148	39.4	0.605	22	0
313	2	155	74	17	96	26.6	0.433	27	1
314	3	113	50	10	85	29.5	0.626	25	0
315	7	109	80	31	0	35.9	1.127	43	1
316	2	112	68	22	94	34.1	0.315	26	0
317	3	99	80	11	64	19.3	0.284	30	0
318	3	182	74	0	0	30.5	0.345	29	1
319	3	115	66	39	140	38.1	0.15	28	0
320	6	194	78	0	0	23.5	0.129	59	1
321	4	129	60	12	231	27.5	0.527	31	0
322	3	112	74	30	0	31.6	0.197	25	1
323	0	124	70	20	0	27.4	0.254	36	1
324	13	152	90	33	29	26.8	0.731	43	1
325	2	112	75	32	0	35.7	0.148	21	0
326	1	157	72	21	168	25.6	0.123	24	0

327	1	122	64	32	156	35.1	0.692	30	1
328	10	179	70	0	0	35.1	0.2	37	0
329	2	102	86	36	120	45.5	0.127	23	1
330	6	105	70	32	68	30.8	0.122	37	0
331	8	118	72	19	0	23.1	1.476	46	0
332	2	87	58	16	52	32.7	0.166	25	0
333	1	180	0	0	0	43.3	0.282	41	1
334	12	106	80	0	0	23.6	0.137	44	0
335	1	95	60	18	58	23.9	0.26	22	0
336	0	165	76	43	255	47.9	0.259	26	0
337	0	117	0	0	0	33.8	0.932	44	0
338	5	115	76	0	0	31.2	0.343	44	1
339	9	152	78	34	171	34.2	0.893	33	1
340	7	178	84	0	0	39.9	0.331	41	1
341	1	130	70	13	105	25.9	0.472	22	0
342	1	95	74	21	73	25.9	0.673	36	0

343	1	0	68	35	0	32	0.389	22	0
344	5	122	86	0	0	34.7	0.29	33	0
345	8	95	72	0	0	36.8	0.485	57	0
346	8	126	88	36	108	38.5	0.349	49	0
347	1	139	46	19	83	28.7	0.654	22	0
348	3	116	0	0	0	23.5	0.187	23	0
349	3	99	62	19	74	21.8	0.279	26	0
350	5	0	80	32	0	41	0.346	37	1
351	4	92	80	0	0	42.2	0.237	29	0

352	4	137	84	0	0	31.2	0.252	30	0
353	3	61	82	28	0	34.4	0.243	46	0
354	1	90	62	12	43	27.2	0.58	24	0
355	3	90	78	0	0	42.7	0.559	21	0
356	9	165	88	0	0	30.4	0.302	49	1
357	1	125	50	40	167	33.3	0.962	28	1
358	13	129	0	30	0	39.9	0.569	44	1
359	12	88	74	40	54	35.3	0.378	48	0
360	1	196	76	36	249	36.5	0.875	29	1
361	5	189	64	33	325	31.2	0.583	29	1
362	5	158	70	0	0	29.8	0.207	63	0
363	5	103	108	37	0	39.2	0.305	65	0
364	4	146	78	0	0	38.5	0.52	67	1
365	4	147	74	25	293	34.9	0.385	30	0
366	5	99	54	28	83	34	0.499	30	0
367	6	124	72	0	0	27.6	0.368	29	1
368	0	101	64	17	0	21	0.252	21	0
369	3	81	86	16	66	27.5	0.306	22	0
370	1	133	102	28	140	32.8	0.234	45	1
371	3	173	82	48	465	38.4	2.137	25	1
372	0	118	64	23	89	0	1.731	21	0
373	0	84	64	22	66	35.8	0.545	21	0
374	2	105	58	40	94	34.9	0.225	25	0
375	2	122	52	43	158	36.2	0.816	28	0
376	12	140	82	43	325	39.2	0.528	58	1
377	0	98	82	15	84	25.2	0.299	22	0

378	1	87	60	37	75	37.2	0.509	22	0
379	4	156	75	0	0	48.3	0.238	32	1
380	0	93	100	39	72	43.4	1.021	35	0
381	1	107	72	30	82	30.8	0.821	24	0
382	0	105	68	22	0	20	0.236	22	0
383	1	109	60	8	182	25.4	0.947	21	0
384	1	90	62	18	59	25.1	1.268	25	0
385	1	125	70	24	110	24.3	0.221	25	0
386	1	119	54	13	50	22.3	0.205	24	0

387	5	116	74	29	0	32.3	0.66	35	1
388	8	105	100	36	0	43.3	0.239	45	1
389	5	144	82	26	285	32	0.452	58	1
390	3	100	68	23	81	31.6	0.949	28	0
391	1	100	66	29	196	32	0.444	42	0
392	5	166	76	0	0	45.7	0.34	27	1
393	1	131	64	14	415	23.7	0.389	21	0
394	4	116	72	12	87	22.1	0.463	37	0
395	4	158	78	0	0	32.9	0.803	31	1
396	2	127	58	24	275	27.7	1.6	25	0
397	3	96	56	34	115	24.7	0.944	39	0
398	0	131	66	40	0	34.3	0.196	22	1
399	3	82	70	0	0	21.1	0.389	25	0
400	3	193	70	31	0	34.9	0.241	25	1
401	4	95	64	0	0	32	0.161	31	1
402	6	137	61	0	0	24.2	0.151	55	0

403	5	136	84	41	88	35	0.286	35	1
404	9	72	78	25	0	31.6	0.28	38	0
405	5	168	64	0	0	32.9	0.135	41	1
406	2	123	48	32	165	42.1	0.52	26	0
407	4	115	72	0	0	28.9	0.376	46	1
408	0	101	62	0	0	21.9	0.336	25	0
409	8	197	74	0	0	25.9	1.191	39	1
410	1	172	68	49	579	42.4	0.702	28	1
411	6	102	90	39	0	35.7	0.674	28	0
412	1	112	72	30	176	34.4	0.528	25	0
413	1	143	84	23	310	42.4	1.076	22	0
414	1	143	74	22	61	26.2	0.256	21	0
415	0	138	60	35	167	34.6	0.534	21	1
416	3	173	84	33	474	35.7	0.258	22	1
417	1	97	68	21	0	27.2	1.095	22	0
418	4	144	82	32	0	38.5	0.554	37	1
419	1	83	68	0	0	18.2	0.624	27	0
420	3	129	64	29	115	26.4	0.219	28	1
421	1	119	88	41	170	45.3	0.507	26	0
422	2	94	68	18	76	26	0.561	21	0
423	0	102	64	46	78	40.6	0.496	21	0
424	2	115	64	22	0	30.8	0.421	21	0
425	8	151	78	32	210	42.9	0.516	36	1
426	4	184	78	39	277	37	0.264	31	1
427	0	94	0	0	0	0	0.256	25	0
428	1	181	64	30	180	34.1	0.328	38	1

429	0	135	94	46	145	40.6	0.284	26	0
430	1	95	82	25	180	35	0.233	43	1
431	2	99	0	0	0	22.2	0.108	23	0
432	3	89	74	16	85	30.4	0.551	38	0
433	1	80	74	11	60	30	0.527	22	0
434	2	139	75	0	0	25.6	0.167	29	0
435	1	90	68	8	0	24.5	1.138	36	0
436	0	141	0	0	0	42.4	0.205	29	1
437	12	140	85	33	0	37.4	0.244	41	0
438	5	147	75	0	0	29.9	0.434	28	0
439	1	97	70	15	0	18.2	0.147	21	0
440	6	107	88	0	0	36.8	0.727	31	0
441	0	189	104	25	0	34.3	0.435	41	1
442	2	83	66	23	50	32.2	0.497	22	0
443	4	117	64	27	120	33.2	0.23	24	0
444	8	108	70	0	0	30.5	0.955	33	1
445	4	117	62	12	0	29.7	0.38	30	1
446	0	180	78	63	14	59.4	2.42	25	1
447	1	100	72	12	70	25.3	0.658	28	0
448	0	95	80	45	92	36.5	0.33	26	0
449	0	104	64	37	64	33.6	0.51	22	1
450	0	120	74	18	63	30.5	0.285	26	0
451	1	82	64	13	95	21.2	0.415	23	0
452	2	134	70	0	0	28.9	0.542	23	1
453	0	91	68	32	210	39.9	0.381	25	0

454	2	119	0	0	0	19.6	0.832	72	0
455	2	100	54	28	105	37.8	0.498	24	0
456	14	175	62	30	0	33.6	0.212	38	1
457	1	135	54	0	0	26.7	0.687	62	0
458	5	86	68	28	71	30.2	0.364	24	0
459	10	148	84	48	237	37.6	1.001	51	1
460	9	134	74	33	60	25.9	0.46	81	0
461	9	120	72	22	56	20.8	0.733	48	0
462	1	71	62	0	0	21.8	0.416	26	0
463	8	74	70	40	49	35.3	0.705	39	0
464	5	88	78	30	0	27.6	0.258	37	0
465	10	115	98	0	0	24	1.022	34	0
466	0	124	56	13	105	21.8	0.452	21	0
467	0	74	52	10	36	27.8	0.269	22	0
468	0	97	64	36	100	36.8	0.6	25	0
469	8	120	0	0	0	30	0.183	38	1
470	6	154	78	41	140	46.1	0.571	27	0
471	1	144	82	40	0	41.3	0.607	28	0
472	0	137	70	38	0	33.2	0.17	22	0
473	0	119	66	27	0	38.8	0.259	22	0
474	7	136	90	0	0	29.9	0.21	50	0
475	4	114	64	0	0	28.9	0.126	24	0
476	0	137	84	27	0	27.3	0.231	59	0
477	2	105	80	45	191	33.7	0.711	29	1
478	7	114	76	17	110	23.8	0.466	31	0

479	8	126	74	38	75	25.9	0.162	39	0
480	4	132	86	31	0	28	0.419	63	0
481	3	158	70	30	328	35.5	0.344	35	1
482	0	123	88	37	0	35.2	0.197	29	0
483	4	85	58	22	49	27.8	0.306	28	0
484	0	84	82	31	125	38.2	0.233	23	0
485	0	145	0	0	0	44.2	0.63	31	1
486	0	135	68	42	250	42.3	0.365	24	1
487	1	139	62	41	480	40.7	0.536	21	0
488	0	173	78	32	265	46.5	1.159	58	0
489	4	99	72	17	0	25.6	0.294	28	0
490	8	194	80	0	0	26.1	0.551	67	0
491	2	83	65	28	66	36.8	0.629	24	0
492	2	89	90	30	0	33.5	0.292	42	0
493	4	99	68	38	0	32.8	0.145	33	0
494	4	125	70	18	122	28.9	1.144	45	1
495	3	80	0	0	0	0	0.174	22	0
496	6	166	74	0	0	26.6	0.304	66	0
497	5	110	68	0	0	26	0.292	30	0
498	2	81	72	15	76	30.1	0.547	25	0
499	7	195	70	33	145	25.1	0.163	55	1
500	6	154	74	32	193	29.3	0.839	39	0
501	2	117	90	19	71	25.2	0.313	21	0
502	3	84	72	32	0	37.2	0.267	28	0
503	6	0	68	41	0	39	0.727	41	1
504	7	94	64	25	79	33.3	0.738	41	0

505	3	96	78	39	0	37.3	0.238	40	0
506	10	75	82	0	0	33.3	0.263	38	0
507	0	180	90	26	90	36.5	0.314	35	1
508	1	130	60	23	170	28.6	0.692	21	0
509	2	84	50	23	76	30.4	0.968	21	0
510	8	120	78	0	0	25	0.409	64	0
511	12	84	72	31	0	29.7	0.297	46	1
512	0	139	62	17	210	22.1	0.207	21	0
513	9	91	68	0	0	24.2	0.2	58	0
514	2	91	62	0	0	27.3	0.525	22	0
515	3	99	54	19	86	25.6	0.154	24	0
516	3	163	70	18	105	31.6	0.268	28	1
517	9	145	88	34	165	30.3	0.771	53	1
518	7	125	86	0	0	37.6	0.304	51	0

519	13	76	60	0	0	32.8	0.18	41	0
520	6	129	90	7	326	19.6	0.582	60	0
521	2	68	70	32	66	25	0.187	25	0
522	3	124	80	33	130	33.2	0.305	26	0
523	6	114	0	0	0	0	0.189	26	0
524	9	130	70	0	0	34.2	0.652	45	1
525	3	125	58	0	0	31.6	0.151	24	0
526	3	87	60	18	0	21.8	0.444	21	0
527	1	97	64	19	82	18.2	0.299	21	0
528	3	116	74	15	105	26.3	0.107	24	0
529	0	117	66	31	188	30.8	0.493	22	0

530	0	111	65	0	0	24.6	0.66	31	0
531	2	122	60	18	106	29.8	0.717	22	0
532	0	107	76	0	0	45.3	0.686	24	0
533	1	86	66	52	65	41.3	0.917	29	0
534	6	91	0	0	0	29.8	0.501	31	0
535	1	77	56	30	56	33.3	1.251	24	0
536	4	132	0	0	0	32.9	0.302	23	1
537	0	105	90	0	0	29.6	0.197	46	0
538	0	57	60	0	0	21.7	0.735	67	0
539	0	127	80	37	210	36.3	0.804	23	0
540	3	129	92	49	155	36.4	0.968	32	1
541	8	100	74	40	215	39.4	0.661	43	1
542	3	128	72	25	190	32.4	0.549	27	1
543	10	90	85	32	0	34.9	0.825	56	1
544	4	84	90	23	56	39.5	0.159	25	0
545	1	88	78	29	76	32	0.365	29	0
546	8	186	90	35	225	34.5	0.423	37	1
547	5	187	76	27	207	43.6	1.034	53	1
548	4	131	68	21	166	33.1	0.16	28	0
549	1	164	82	43	67	32.8	0.341	50	0
550	4	189	110	31	0	28.5	0.68	37	0
551	1	116	70	28	0	27.4	0.204	21	0
552	3	84	68	30	106	31.9	0.591	25	0
553	6	114	88	0	0	27.8	0.247	66	0
554	1	88	62	24	44	29.9	0.422	23	0
555	1	84	64	23	115	36.9	0.471	28	0

556	7	124	70	33	215	25.5	0.161	37	0
557	1	97	70	40	0	38.1	0.218	30	0
558	8	110	76	0	0	27.8	0.237	58	0
559	11	103	68	40	0	46.2	0.126	42	0
560	11	85	74	0	0	30.1	0.3	35	0
561	6	125	76	0	0	33.8	0.121	54	1
562	0	198	66	32	274	41.3	0.502	28	1

563	1	87	68	34	77	37.6	0.401	24	0
564	6	99	60	19	54	26.9	0.497	32	0
565	0	91	80	0	0	32.4	0.601	27	0
566	2	95	54	14	88	26.1	0.748	22	0
567	1	99	72	30	18	38.6	0.412	21	0

568	6	92	62	32	126	32	0.085	46	0
569	4	154	72	29	126	31.3	0.338	37	0
570	0	121	66	30	165	34.3	0.203	33	1
571	3	78	70	0	0	32.5	0.27	39	0
572	2	130	96	0	0	22.6	0.268	21	0
573	3	111	58	31	44	29.5	0.43	22	0
574	2	98	60	17	120	34.7	0.198	22	0
575	1	143	86	30	330	30.1	0.892	23	0
576	1	119	44	47	63	35.5	0.28	25	0
577	6	108	44	20	130	24	0.813	35	0
578	2	118	80	0	0	42.9	0.693	21	1
579	10	133	68	0	0	27	0.245	36	0
580	2	197	70	99	0	34.7	0.575	62	1

581	0	151	90	46	0	42.1	0.371	21	1
582	6	109	60	27	0	25	0.206	27	0
583	12	121	78	17	0	26.5	0.259	62	0
584	8	100	76	0	0	38.7	0.19	42	0
585	8	124	76	24	600	28.7	0.687	52	1
586	1	93	56	11	0	22.5	0.417	22	0
587	8	143	66	0	0	34.9	0.129	41	1
588	6	103	66	0	0	24.3	0.249	29	0
589	3	176	86	27	156	33.3	1.154	52	1
590	0	73	0	0	0	21.1	0.342	25	0
591	11	111	84	40	0	46.8	0.925	45	1
592	2	112	78	50	140	39.4	0.175	24	0
593	3	132	80	0	0	34.4	0.402	44	1
594	2	82	52	22	115	28.5	1.699	25	0
595	6	123	72	45	230	33.6	0.733	34	0
596	0	188	82	14	185	32	0.682	22	1
597	0	67	76	0	0	45.3	0.194	46	0
598	1	89	24	19	25	27.8	0.559	21	0
599	1	173	74	0	0	36.8	0.088	38	1
600	1	109	38	18	120	23.1	0.407	26	0
601	1	108	88	19	0	27.1	0.4	24	0
602	6	96	0	0	0	23.7	0.19	28	0
603	1	124	74	36	0	27.8	0.1	30	0
604	7	150	78	29	126	35.2	0.692	54	1
605	4	183	0	0	0	28.4	0.212	36	1
606	1	124	60	32	0	35.8	0.514	21	0

607	1	181	78	42	293	40	1.258	22	1
608	1	92	62	25	41	19.5	0.482	25	0
609	0	152	82	39	272	41.5	0.27	27	0
610	1	111	62	13	182	24	0.138	23	0
611	3	106	54	21	158	30.9	0.292	24	0
612	3	174	58	22	194	32.9	0.593	36	1
613	7	168	88	42	321	38.2	0.787	40	1
614	6	105	80	28	0	32.5	0.878	26	0
615	11	138	74	26	144	36.1	0.557	50	1
616	3	106	72	0	0	25.8	0.207	27	0
617	6	117	96	0	0	28.7	0.157	30	0
618	2	68	62	13	15	20.1	0.257	23	0
619	9	112	82	24	0	28.2	1.282	50	1
620	0	119	0	0	0	32.4	0.141	24	1
621	2	112	86	42	160	38.4	0.246	28	0
622	2	92	76	20	0	24.2	1.698	28	0
623	6	183	94	0	0	40.8	1.461	45	0
624	0	94	70	27	115	43.5	0.347	21	0
625	2	108	64	0	0	30.8	0.158	21	0
626	4	90	88	47	54	37.7	0.362	29	0
627	0	125	68	0	0	24.7	0.206	21	0
628	0	132	78	0	0	32.4	0.393	21	0
629	5	128	80	0	0	34.6	0.144	45	0
630	4	94	65	22	0	24.7	0.148	21	0
631	7	114	64	0	0	27.4	0.732	34	1

632	0	102	78	40	90	34.5	0.238	24	0
633	2	111	60	0	0	26.2	0.343	23	0
634	1	128	82	17	183	27.5	0.115	22	0
635	10	92	62	0	0	25.9	0.167	31	0
636	13	104	72	0	0	31.2	0.465	38	1
637	5	104	74	0	0	28.8	0.153	48	0
638	2	94	76	18	66	31.6	0.649	23	0
639	7	97	76	32	91	40.9	0.871	32	1
640	1	100	74	12	46	19.5	0.149	28	0
641	0	102	86	17	105	29.3	0.695	27	0
642	4	128	70	0	0	34.3	0.303	24	0
643	6	147	80	0	0	29.5	0.178	50	1
644	4	90	0	0	0	28	0.61	31	0
645	3	103	72	30	152	27.6	0.73	27	0
646	2	157	74	35	440	39.4	0.134	30	0
647	1	167	74	17	144	23.4	0.447	33	1
648	0	179	50	36	159	37.8	0.455	22	1
649	11	136	84	35	130	28.3	0.26	42	1
650	0	107	60	25	0	26.4	0.133	23	0

651	1	91	54	25	100	25.2	0.234	23	0
652	1	117	60	23	106	33.8	0.466	27	0
653	5	123	74	40	77	34.1	0.269	28	0
654	2	120	54	0	0	26.8	0.455	27	0
655	1	106	70	28	135	34.2	0.142	22	0
656	2	155	52	27	540	38.7	0.24	25	1

657	2	101	58	35	90	21.8	0.155	22	0
658	1	120	80	48	200	38.9	1.162	41	0
659	11	127	106	0	0	39	0.19	51	0
660	3	80	82	31	70	34.2	1.292	27	1
661	10	162	84	0	0	27.7	0.182	54	0
662	1	199	76	43	0	42.9	1.394	22	1
663	8	167	106	46	231	37.6	0.165	43	1
664	9	145	80	46	130	37.9	0.637	40	1
665	6	115	60	39	0	33.7	0.245	40	1
666	1	112	80	45	132	34.8	0.217	24	0
667	4	145	82	18	0	32.5	0.235	70	1
668	10	111	70	27	0	27.5	0.141	40	1
669	6	98	58	33	190	34	0.43	43	0
670	9	154	78	30	100	30.9	0.164	45	0
671	6	165	68	26	168	33.6	0.631	49	0
672	1	99	58	10	0	25.4	0.551	21	0
673	10	68	106	23	49	35.5	0.285	47	0
674	3	123	100	35	240	57.3	0.88	22	0
675	8	91	82	0	0	35.6	0.587	68	0
676	6	195	70	0	0	30.9	0.328	31	1
677	9	156	86	0	0	24.8	0.23	53	1
678	0	93	60	0	0	35.3	0.263	25	0
679	3	121	52	0	0	36	0.127	25	1
680	2	101	58	17	265	24.2	0.614	23	0
681	2	56	56	28	45	24.2	0.332	22	0
682	0	162	76	36	0	49.6	0.364	26	1

683	0	95	64	39	105	44.6	0.366	22	0
684	4	125	80	0	0	32.3	0.536	27	1
685	5	136	82	0	0	0	0.64	69	0
686	2	129	74	26	205	33.2	0.591	25	0
687	3	130	64	0	0	23.1	0.314	22	0
688	1	107	50	19	0	28.3	0.181	29	0
689	1	140	74	26	180	24.1	0.828	23	0
690	1	144	82	46	180	46.1	0.335	46	1
691	8	107	80	0	0	24.6	0.856	34	0
692	13	158	114	0	0	42.3	0.257	44	1
693	2	121	70	32	95	39.1	0.886	23	0
694	7	129	68	49	125	38.5	0.439	43	1

695	2	90	60	0	0	23.5	0.191	25	0
696	7	142	90	24	480	30.4	0.128	43	1
697	3	169	74	19	125	29.9	0.268	31	1
698	0	99	0	0	0	25	0.253	22	0
699	4	127	88	11	155	34.5	0.598	28	0
700	4	118	70	0	0	44.5	0.904	26	0
701	2	122	76	27	200	35.9	0.483	26	0
702	6	125	78	31	0	27.6	0.565	49	1
703	1	168	88	29	0	35	0.905	52	1

704	2	129	0	0	0	38.5	0.304	41	0
705	4	110	76	20	100	28.4	0.118	27	0
706	6	80	80	36	0	39.8	0.177	28	0
707	10	115	0	0	0	0	0.261	30	1
708	2	127	46	21	335	34.4	0.176	22	0

709	9	164	78	0	0	32.8	0.148	45	1
710	2	93	64	32	160	38	0.674	23	1
711	3	158	64	13	387	31.2	0.295	24	0
712	5	126	78	27	22	29.6	0.439	40	0
713	10	129	62	36	0	41.2	0.441	38	1
714	0	134	58	20	291	26.4	0.352	21	0
715	3	102	74	0	0	29.5	0.121	32	0
716	7	187	50	33	392	33.9	0.826	34	1
717	3	173	78	39	185	33.8	0.97	31	1
718	10	94	72	18	0	23.1	0.595	56	0
719	1	108	60	46	178	35.5	0.415	24	0
720	5	97	76	27	0	35.6	0.378	52	1
721	4	83	86	19	0	29.3	0.317	34	0
722	1	114	66	36	200	38.1	0.289	21	0
723	1	149	68	29	127	29.3	0.349	42	1
724	5	117	86	30	105	39.1	0.251	42	0
725	1	111	94	0	0	32.8	0.265	45	0
726	4	112	78	40	0	39.4	0.236	38	0
727	1	116	78	29	180	36.1	0.496	25	0
728	0	141	84	26	0	32.4	0.433	22	0
729	2	175	88	0	0	22.9	0.326	22	0
730	2	92	52	0	0	30.1	0.141	22	0
731	3	130	78	23	79	28.4	0.323	34	1
732	8	120	86	0	0	28.4	0.259	22	1
733	2	174	88	37	120	44.5	0.646	24	1
734	2	106	56	27	165	29	0.426	22	0
735	2	105	75	0	0	23.3	0.56	53	0
736	4	95	60	32	0	35.4	0.284	28	0
737	0	126	86	27	120	27.4	0.515	21	0

738	8	65	72	23	0	32	0.6	42	0
739	2	99	60	17	160	36.6	0.453	21	0
740	1	102	74	0	0	39.5	0.293	42	1
741	11	120	80	37	150	42.3	0.785	48	1
742	3	102	44	20	94	30.8	0.4	26	0
743	1	109	58	18	116	28.5	0.219	22	0
744	9	140	94	0	0	32.7	0.734	45	1
745	13	153	88	37	140	40.6	1.174	39	0
746	1s 2	100	84	33	105	30	0.488	46	0
747	1	147	94	41	0	49.3	0.358	27	1
748	1	81	74	41	57	46.3	1.096	32	0
749	3	187	70	22	200	36.4	0.408	36	1
750	6	162	62	0	0	24.3	0.178	50	1
751	4	136	70	0	0	31.2	1.182	22	1
752	1	121	78	39	74	39	0.261	28	0
753	3	108	62	24	0	26	0.223	25	0
754	0	181	88	44	510	43.3	0.222	26	1
755	8	154	78	32	0	32.4	0.443	45	1
756	1	128	88	39	110	36.5	1.057	37	1
757	7	137	90	41	0	32	0.391	39	0
758	0	123	72	0	0	36.3	0.258	52	1
759	1	106	76	0	0	37.5	0.197	26	0
760	6	190	92	0	0	35.5	0.278	66	1
761	2	88	58	26	16	28.4	0.766	22	0
762	9	170	74	31	0	44	0.403	43	1
763	9	89	62	0	0	22.5	0.142	33	0
764	10	101	76	48	180	32.9	0.171	63	0
765	2	122	70	27	0	36.8	0.34	27	0
766	5	121	72	23	112	26.2	0.245	30	0

767	1	126	60	0	0	30.1	0.349	47	1
768	1	93	70	31	0	30.4	0.315	23	0

Chapter 6

BIBLIOGRAPHY

1. Multivariate data analysis (Fifth Edition) --- Joseph F.Hair, RolphE.Anderson, Ronald l Tatham and William C.Black

2. Data Mining- Theories, Algorithms, and Examples – NoNG YE

3. A Practical Guide to Data Mining for Business and Industry -- Andrea Ahlemeyer-Stubbe, Shirley Coleman

4. Data Mining and Predictive Analytics – Daniel T. Larose, Chantal D.Lorse

5. machine_learning_mastery_with_r. – Jason Brownlee

6. master_machine_learning_algorithms -- Jason Brownlee

7. statistical_methods_for_machine_learning - Jason Brownlee

8. Machine Learning Using R -- KarthikRamasubramanian ,Abhishek Singh

9. Data Science for Business - Forster Provost & Tom Fawcett

10. Deep learning with Deep learning R by François Chollet