A PROJECT REPORT
ON

Predicting the students' knowledge status about the subject of Electrical DC Machines

Submitted to

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

MASTER OF SCIENCE
IN
STATISTICS

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

By

V.SUSHMITHA	**Roll No:1007-17-507-008**
A.SHIRISHA	**Roll No:1007-17-507-014**
S.BHOOMESHWAR	**Roll No:1007-17-507-019**
P.SOWMYA	**Roll No:1007-17-507-026**
K.ANJANI KRISHNA	**Roll No: 1007-17-507-018**
J.CHAITANYA	**Roll No:1007-17-507-036**
S.VINOD	**Roll No:1007-17-507-009**

Under the Supervision of

Dr. M. VENUGOPALA RAO

2018

A PROJECT REPORT
ON

Predicting the students' knowledge status about the subject of Electrical DC Machines

Submitted to
Osmania University
In partial fulfillment of the requirements for the award of
Master of Science in Statistics

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

By

V.SUSMITHA	Roll No.:1007-17-507-008
A. SHIRISHA	Roll No.:1007-17-507-014
S.BHOOMESHWAR	Roll No.:1007-17-507-019
P.SOWMYA	Roll No.:1007-17-507-026
K.ANJANIKRISHNA	Roll No.:1007-17-507-018
J.CHAITANYA	Roll No.:1007-17-507-036
S.VINOD	Roll No.:1007-17-507-009

Under the Supervision of
Dr. M. VENUGOPALA RAO

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.

V.SUSHMITHA

A.SHIRISHA

S.BHOOMESHWAR

P.SOWMYA

K.ANJANI KRISHNA

J.CHAITANYA

S.VINOD

Department of Statistics

Osmania University

Hyderabad – 500 007,
Telangana, INDIA.

CERTIFICATE

This is to certify that

Ms. V.SUSHMITHA	**Roll No:1007-17-507-008**
Ms. A.SHIRISHA	**Roll No:1007-17-507-014**
Mr.S.BHOOMESHWAR	**Roll No:1007-17-507-019**
Ms. P.SOWMYA	**Roll No:1007-17-507-026**
Mr. K.ANJANI KRISHNA	**Roll No:1007-17-507-018**
Ms. J.CHAITANYA	**Roll No:1007-17-507-036**
Mr.S.VINOD	**Roll No:1007-17-507-009**

Have submitted the project titled **"Predicting the students' knowledge status About the subject of Electrical DC Machines"** in partial fulfillment 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 **Dr. M. VENUGOPALA RAO**, 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, and 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 **T.SANDHYA, 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 and Scope of the Problem)

INTRODUCTION

1.1 Scope of the problem

This problem is related to predicting the students' knowledge status about the subject of Electrical DC Machines.

Objectives:
The key objectives are:

What are the attributes that strongly affect the student's knowledge level Developing an algorithm, by which we can classify the student's knowledge level based on attributes.

Given a set of characteristics for new users, predicting which category they fall into

1.2 Data Description:

To achieve the above objectives, we have considered data of 403 students with 5 different attributes and are predicting the "user knowledge level" (target value) i.e., '0' (low) or '1' (high) based on these 5 attributes. Description of each variable is as below:

1. STG (The degree of study time for goal object materials) (input value)
2. SCG (The degree of repetition number of user for goal object materials) (input value)
3. STR (The degree of study time of user for related objects with goal object) (input value)
4. LPR (The exam performance of user for related objects with goal object) (input value)
5. PEG (The exam performance of user for goal objects) (input variable)
6. UNS(The knowledge level of user) (target value)

1.3 Source: We have downloaded data from UCI repository.

Here, is the link to data set:
https://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling

1.4 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.

Chapter 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.

Chapter 4 describes the summary and conclusions followed by Bibliography.

Chapter 5 Appendix describes the Data and R code used in the analysis.

Chapter 2

Review of Machine Learning Process

REVIEW OF MACHINE LEARNING

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 analyzing 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 phase

Business Understanding

Data Understanding

Data preparation

Modeling

Evaluation

Deployment.

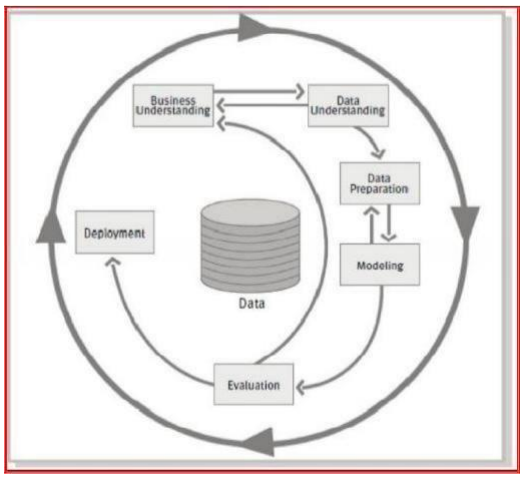


Fig 2.1 CRISP DM

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) Modeling

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 modeling. 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 modeling technique
2) Generate a test design
3) Build a model
4) Assess a Model

Both cross validation of the model (using train/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:

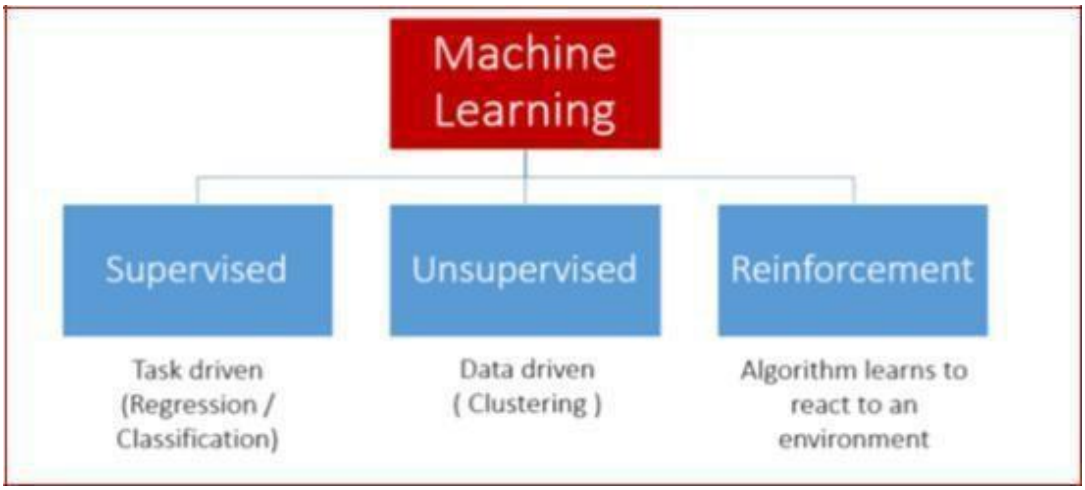


Fig 2.2 Types of machine learning.

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:

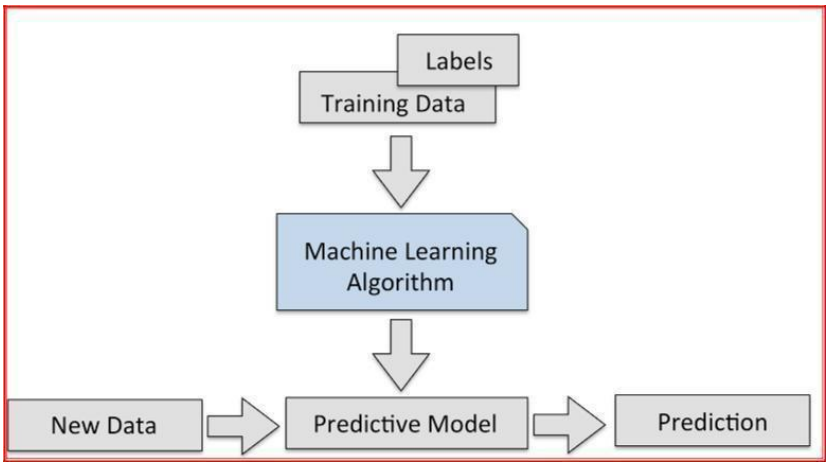


Fig 2.2.1 Supervised learning.

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

CLASSIFICATION **REGRESSION**

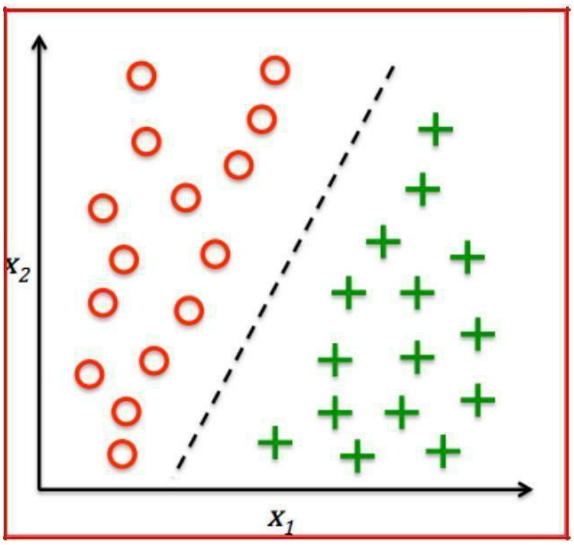


Fig 2.2.1(1) Classification.

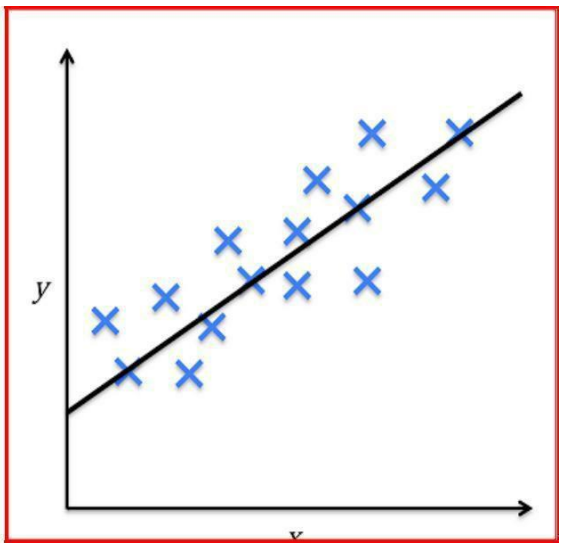


Fig 2.2.1(2) Regression.

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)

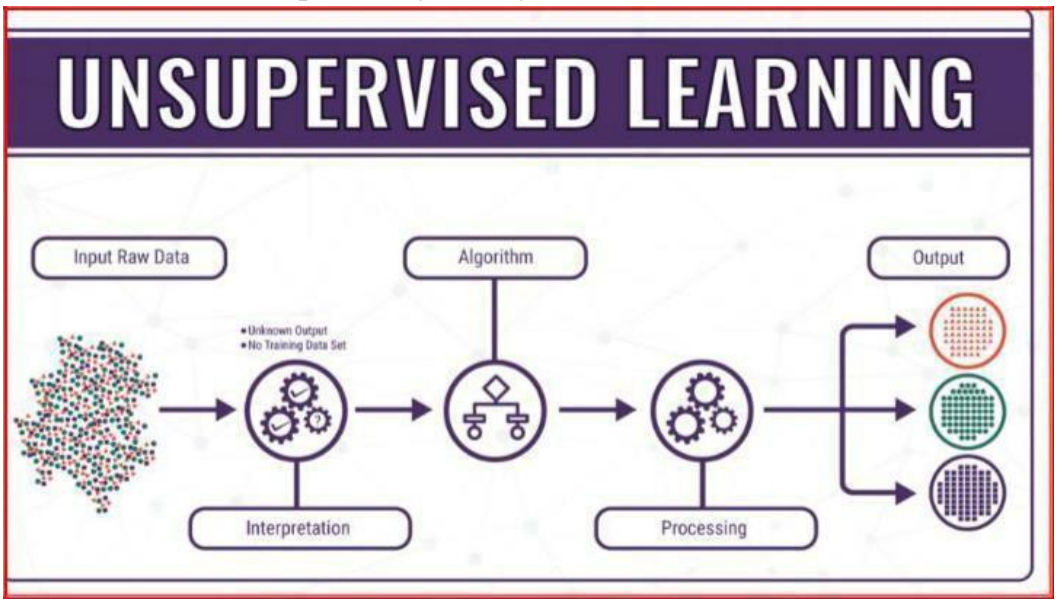


Fig 2.2.2 Unsupervised learning.

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

Cluster Analysis **Data Reduction Techniques**

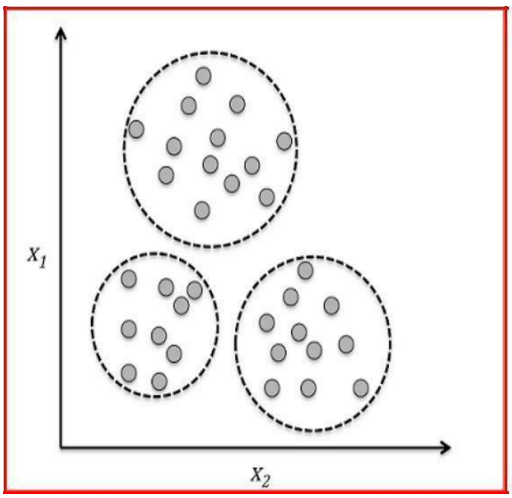


Fig 2.2.2(1) Cluster Analysis.

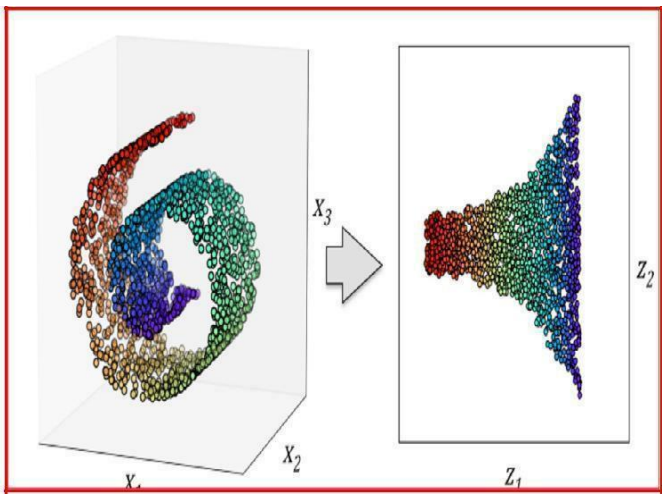


Fig 2.2.2(2) Data Reduction Techniques.

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:

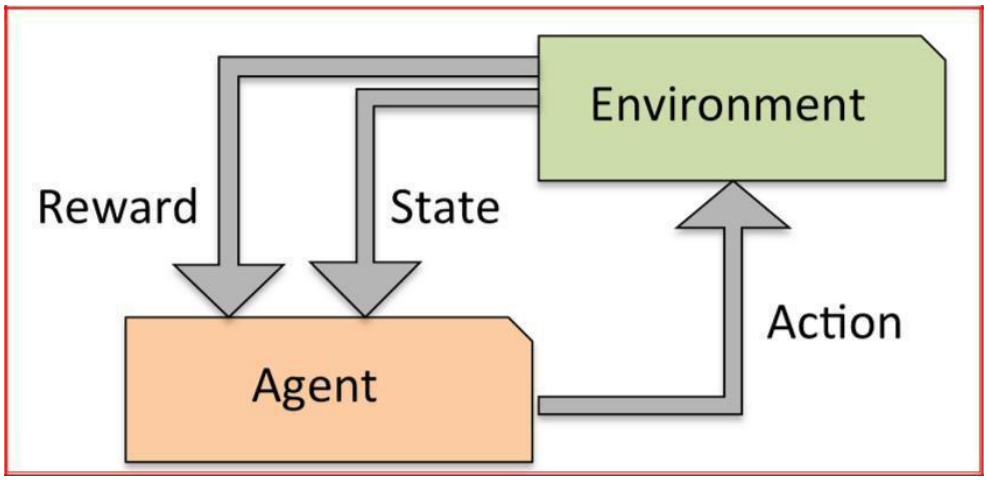


Fig 2.2.3 Reinforcement learning.

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

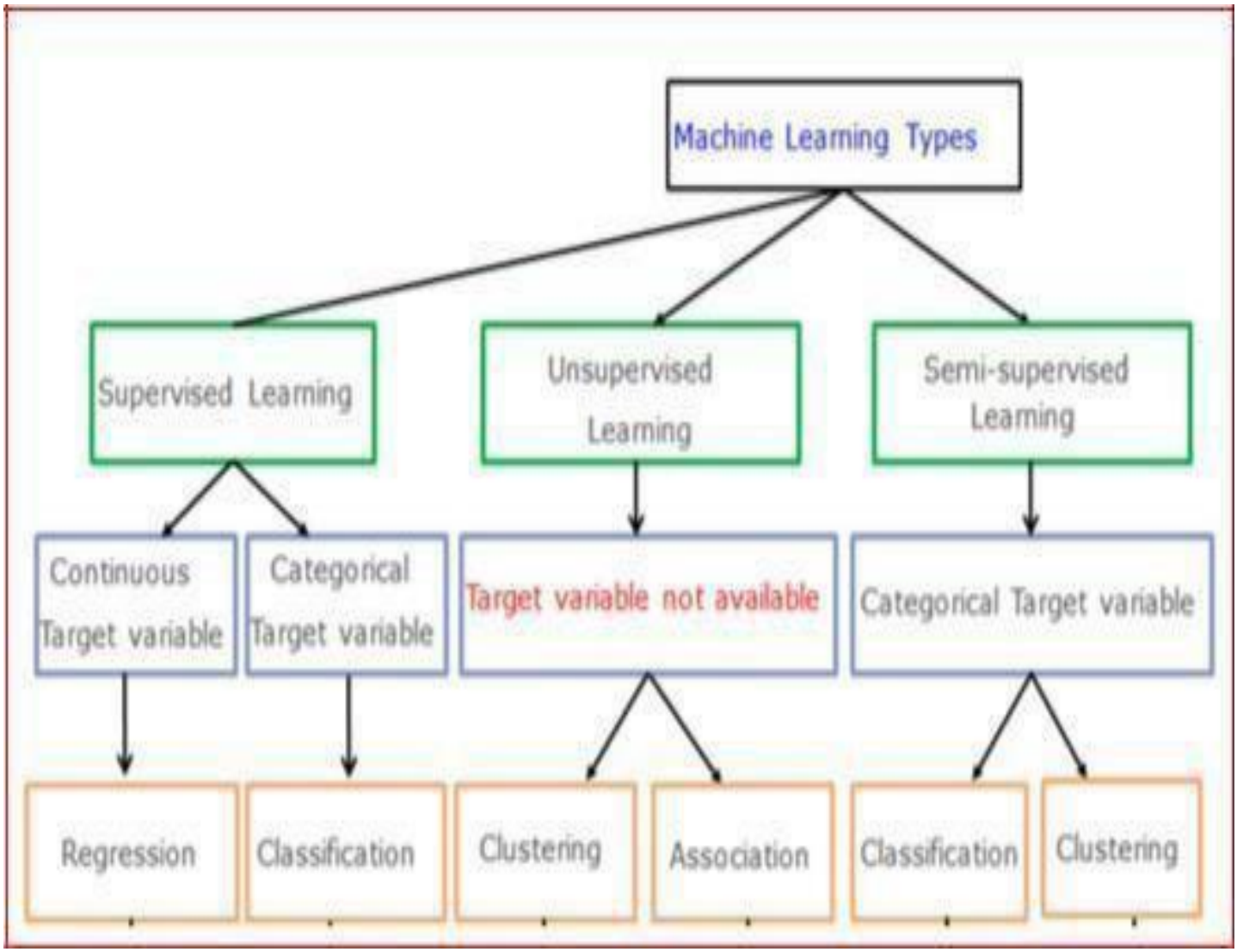


Fig 2.3 Algorithm.

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 techniques, and is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regressions are 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 characterizes maximum margin algorithm. Linear learning machine maps a non-linear function into high dimensional kernel-induced 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 regressions which are** 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 Neighbors:-**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 neighbors, with the object being assigned to the class most common among its k nearest neighbors. 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 neighbors.

c) 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 categorizes new examples. In two dimensional spaces, 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 unlabeled 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

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.

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.

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 centroid 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.

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.

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.

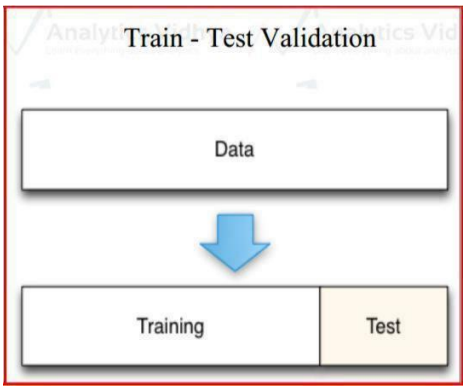


Fig 2.4.1 Model Validation.

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 cross validation

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.

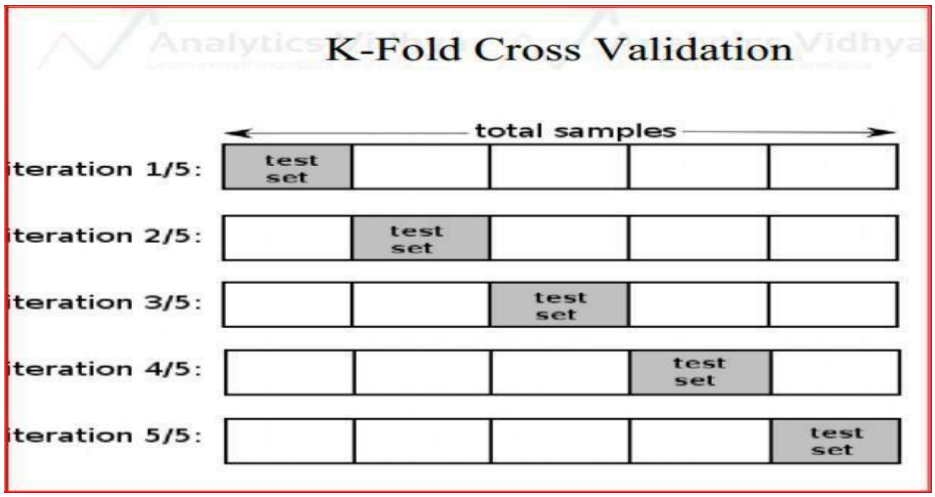


Fig 2.4.1(1) k-Fold cross validation

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

1. A fundamental problem with supervised learning is the bias variance trade-off. Ideally, a model should have two key characteristics
2. Sensitive enough to accurately capture the key patterns in the training dataset.
3. 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 interaction.

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.

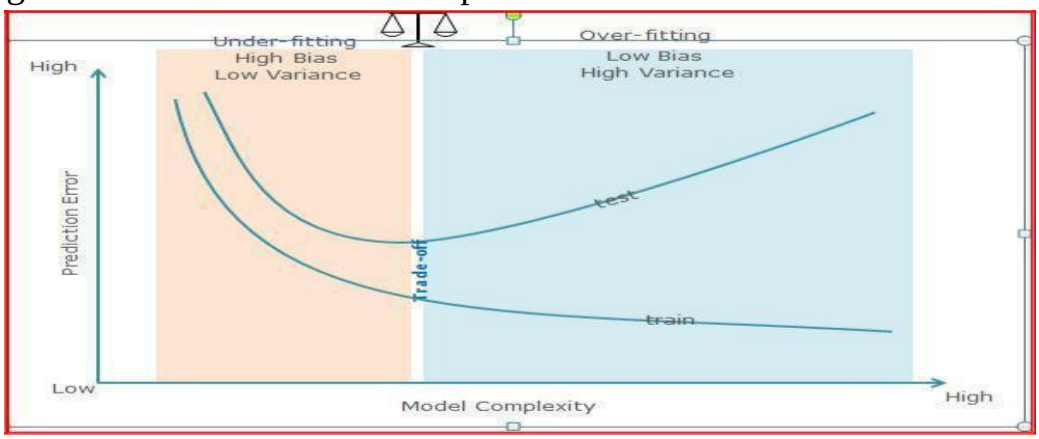


Fig 2.5.1 Bias and Variance

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				
Actual class		Positive (C_0)	Negative (C_1)	
	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.

shows an example confusion matrix.

. Example of classifications Accuracy measurement

Predicted classed				
Actual class		Positive (C_0)	Negative (C_1)	
	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.

Table 2.5.2(1) Various measures used to assess performance.

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

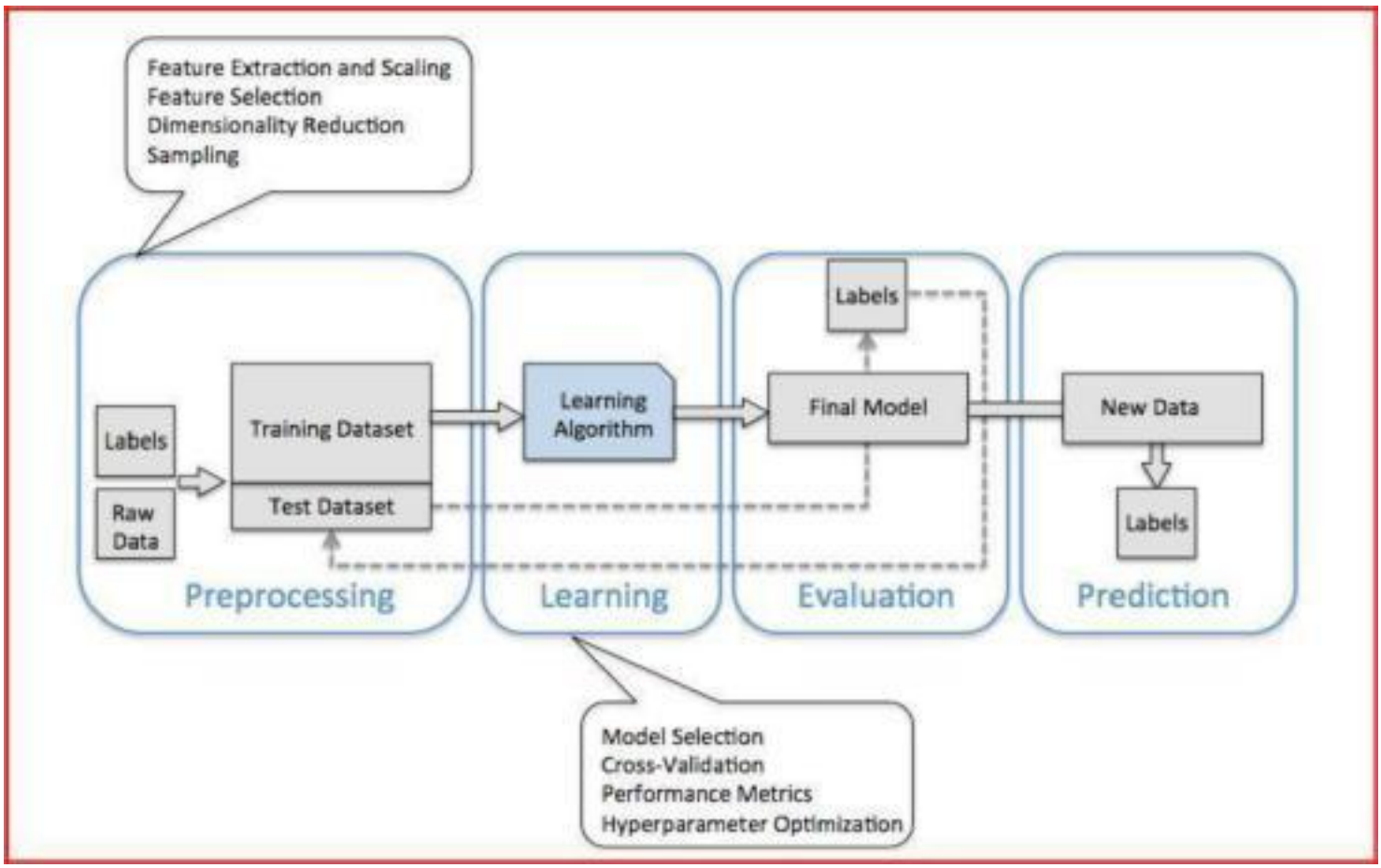


Fig 2.6 Process of machine learning.

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 258 records randomly from 403 records and the information of the same is mentioned in Chapter 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 Chapter 1.

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

Dimensions of data set

```
258   6
```

Here, there are 258 rows and 6 variables in the data.

Structure of the data file –gives the number of variables in the data and scales of the data

```
'data.frame':   258 obs. of  6 variables:
$ STG: num  0 0.08 0.06 0.1 0.08 0.09 0.1 0.15 0.2 0 ...
$ SCG: num  0 0.08 0.06 0.1 0.08 0.15 0.1 0.02 0.14 0 ...
$ STR: num  0 0.1 0.05 0.15 0.08 0.4 0.43 0.34 0.35 0.5 ...
$ LPR: num  0 0.24 0.25 0.65 0.98 0.1 0.29 0.4 0.72 0.2 ...
$ PEG: num  0 0.9 0.33 0.3 0.24 0.66 0.56 0.01 0.25 0.85 ...
$ UNS: int  0 1 0 1 0 1 1 0 0 1 ...
```

Output 3.1(1) Structure of data.

As we can see from the table above, the variables are read as integer type but, as UNS is of nominal in nature, it is necessary to convert this variable into factor. So, we converted UNS variable into factor type, as below:

```
'data.frame':   258 obs. of  6 variables:
$ STG: num  0 0.08 0.06 0.1 0.08 0.09 0.1 0.15 0.2 0 ...
$ SCG: num  0 0.08 0.06 0.1 0.08 0.15 0.1 0.02 0.14 0 ...
$ STR: num  0 0.1 0.05 0.15 0.08 0.4 0.43 0.34 0.35 0.5 ...
$ LPR: num  0 0.24 0.25 0.65 0.98 0.1 0.29 0.4 0.72 0.2 ...
$ PEG: num  0 0.9 0.33 0.3 0.24 0.66 0.56 0.01 0.25 0.85 ...
$ UNS: Factor w/ 2 levels "0","1": 1 2 1 2 1 2 2 1 1 2 ...
```

Output 3.1(2) Structure of data after changing integer into factor.

Understanding data using Descriptive Statistics:

To understand the data, we will first look at the summary of the data:

```
      STG              SCG              STR              LPR              PEG        
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
 1st Qu.:0.2407   1st Qu.:0.2100   1st Qu.:0.2913   1st Qu.:0.2500   1st Qu.:0.2500  
 Median :0.3270   Median :0.3025   Median :0.4900   Median :0.3300   Median :0.5000  
 Mean   :0.3711   Mean   :0.3557   Mean   :0.4680   Mean   :0.4327   Mean   :0.4585  
 3rd Qu.:0.4950   3rd Qu.:0.4975   3rd Qu.:0.6900   3rd Qu.:0.6475   3rd Qu.:0.6600  
 Max.   :0.9900   Max.   :0.9000   Max.   :0.9500   Max.   :0.9900   Max.   :0.9300  
      UNS        
 Min.   :0.0000  
 1st Qu.:0.0000  
 Median :1.0000  
 Mean   :0.5853  
 3rd Qu.:1.0000  
 Max.   :1.0000
```

Output 3.1(3) Data using Descriptive Statistics.

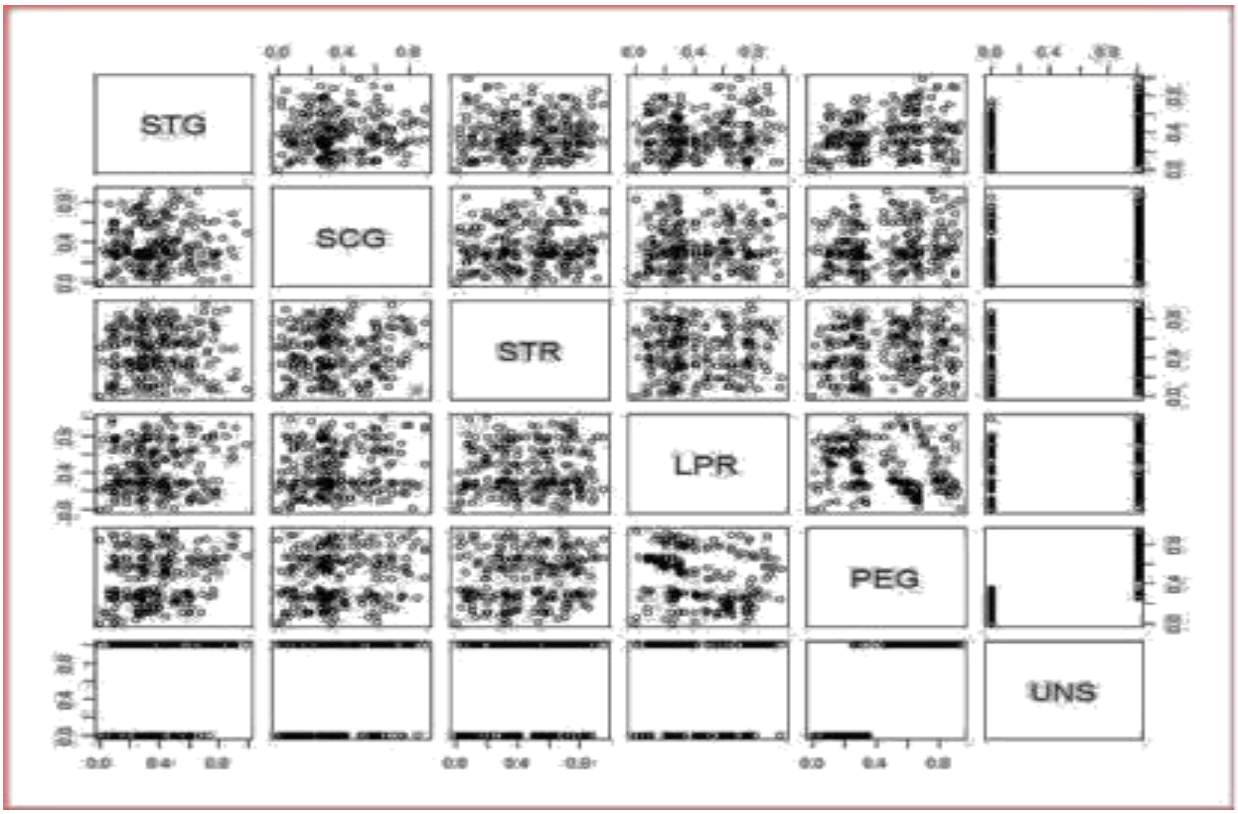
From the above table, we can see the counts and descriptive of each of the independent variables along with the dependent.

Understanding data visually:

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

To know the relation among all variables at a time we use pairs

To know the relation among all variables at a time we use pairs



Output 3.1(4) Pairs.

This plot describes the relations between all the variables.

Checking for missing values:

We also need to check if the data contains any missing values, which can be done as below

STG	SCG	STR	LPR	PEG	UNS
0	0	0	0	0	0

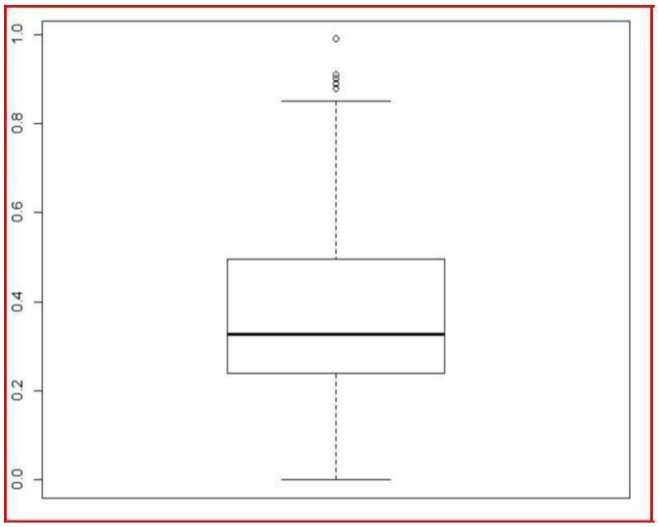
Output 3.1(5) Missing values.

As we can see from the table above, we do not have any missing values in our data so, we do not need to do any imputation

Identifying outliers:

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

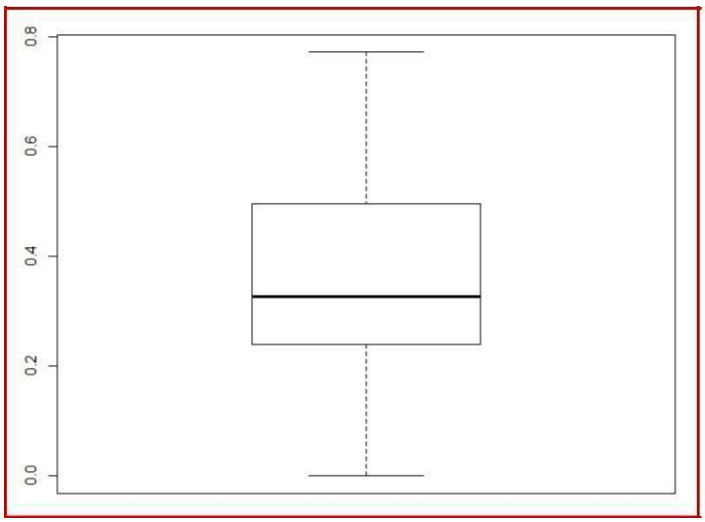
Box plot for STG:



Here there are outliers

Output 3.1(6) Boxplot.

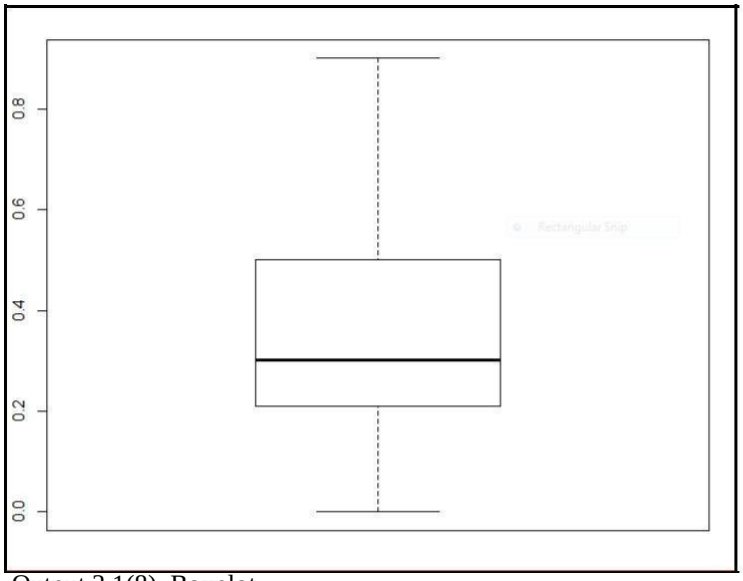
Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile will be imputed using 5th percentile value.



Box plot of STG after removing outliers , Here there are no more outliers

Output 3.1(7) Boxplot without outliers.

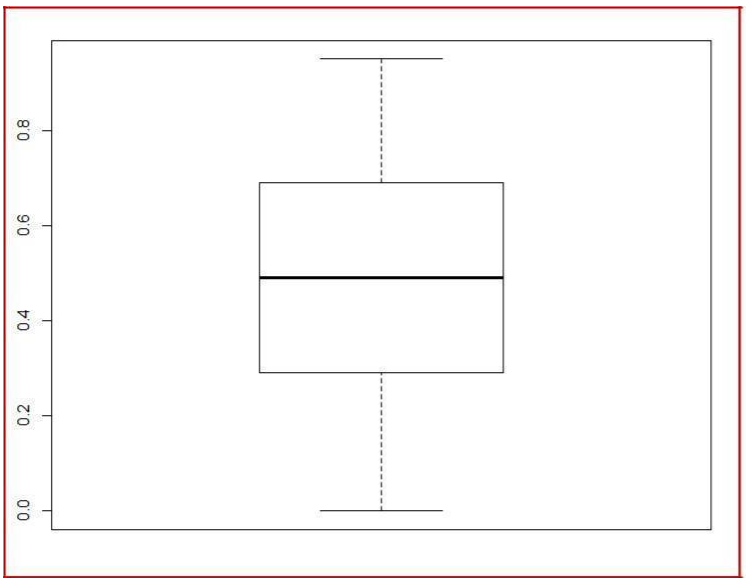
Boxplot for SCG:



Here there are no outliers

Output 3.1(8) Boxplot.

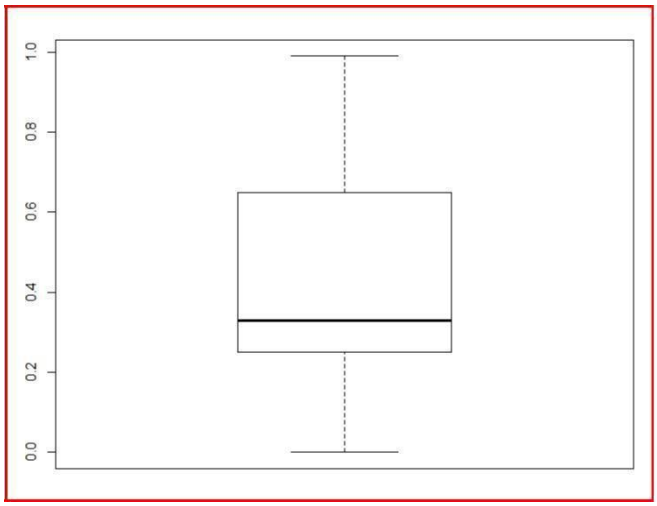
Boxplot for STR:



Here there are no outliers

Output 3.1(9) Boxplot.

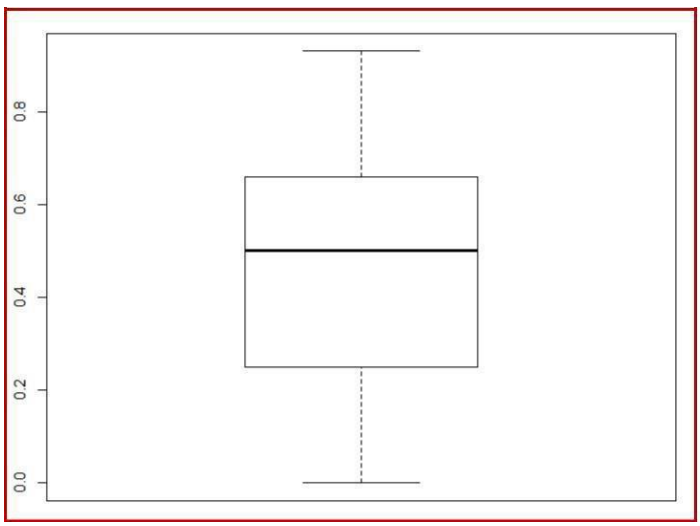
Boxplot for LPR:



Here there are no outliers

Output 3.1(10) Boxplot

Boxplot for PEG:

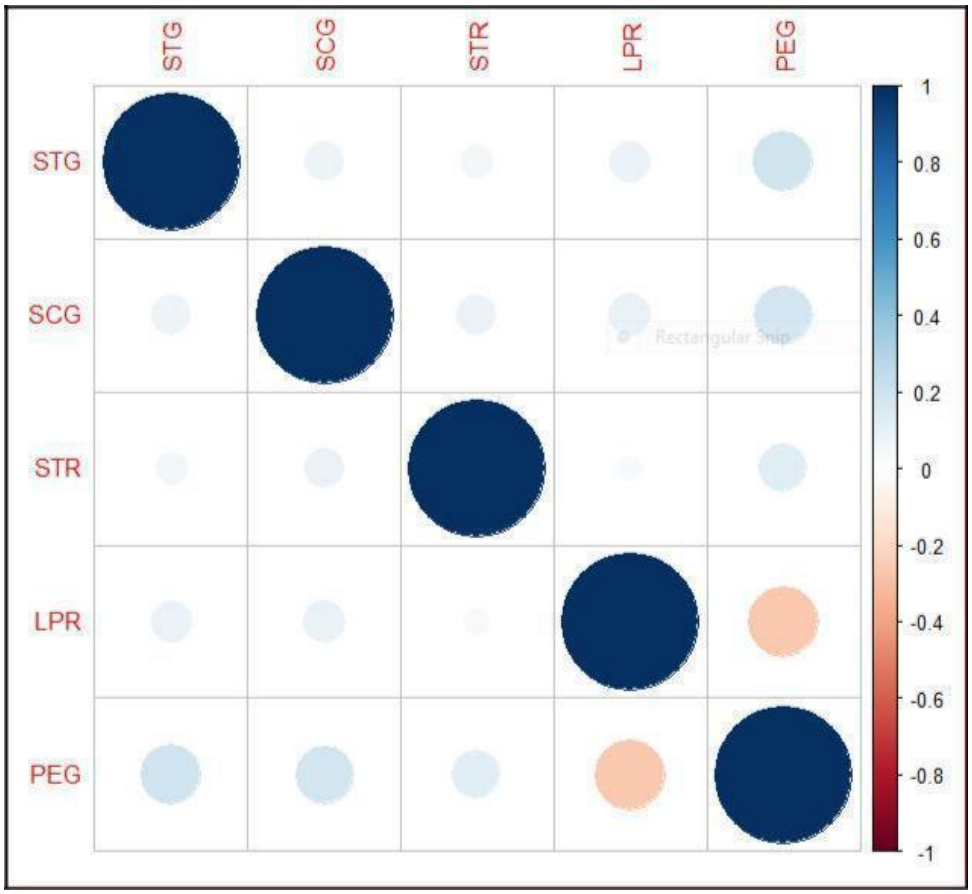


Here there are no Outliers

Output 3.1(11) Boxplot

Understanding relationships between variables:

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



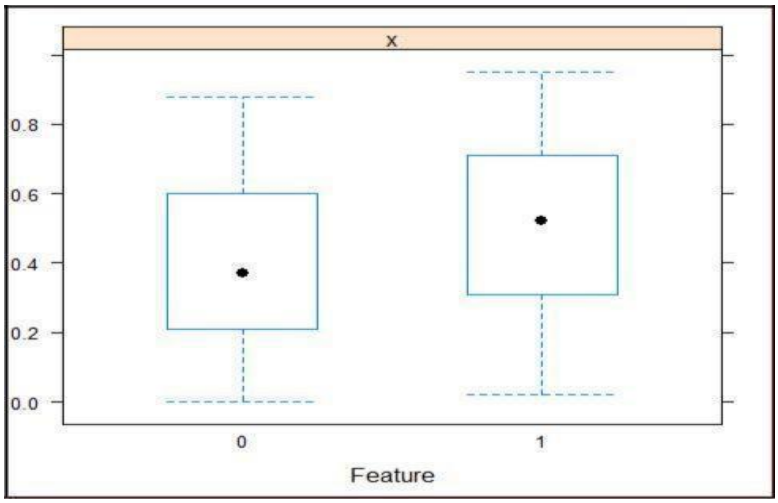
Output 3.1(12) Correlation Plot.

Here, the circle size refers to the strength of the relation between continuous variables and color refers to the direction of the relationship. Blue circles represent a positive correlation; Red circles represent a negative correlation
The small circles indicate that there is very less / no correlation between the variables.

From the plot, we can see that STG and PEG are positively correlated; LPR and PEG are negatively correlated.

For the **continuous vs. categorical variable**, we will look at Feature plots to understand the relationship

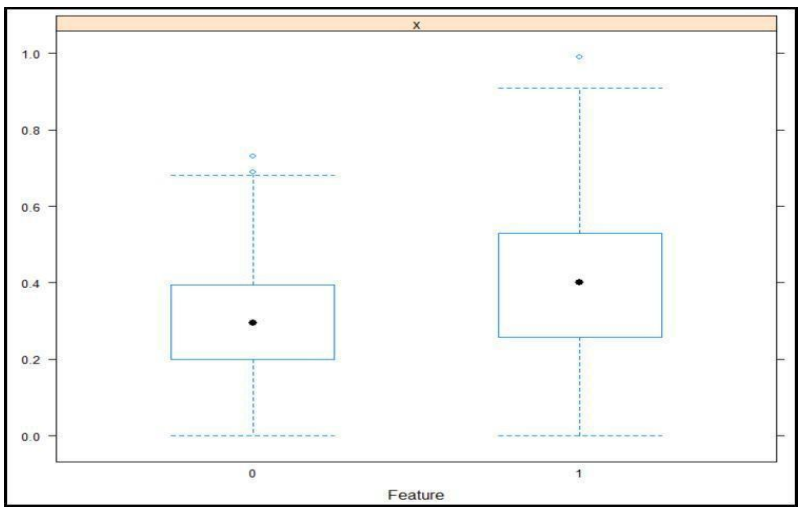
STG VS UNS:



Output 3.1(13) Feature plot for STG vs UNS.

From the plot above, we can conclude that there is not much difference in STG between the User knowledge levels

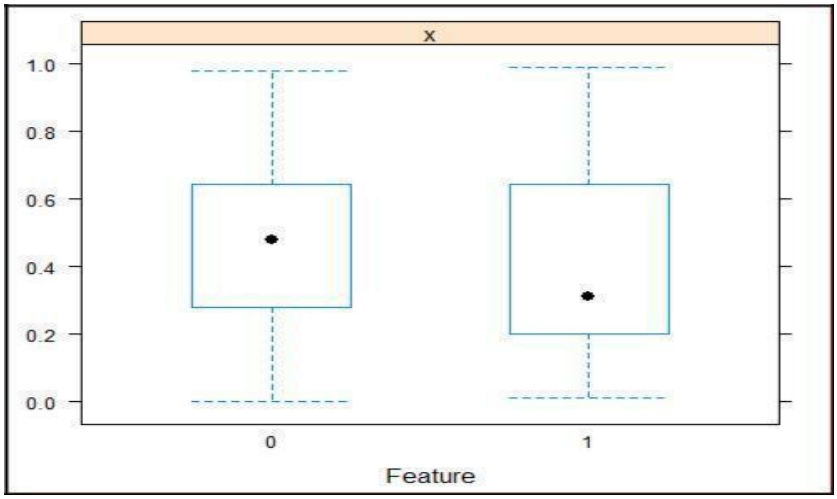
SCG VS UNS



Output 3.1(14) Feature plot for SCG vs UNS

From the plot above, we can conclude that there is some difference in SCG between the User knowledge levels.

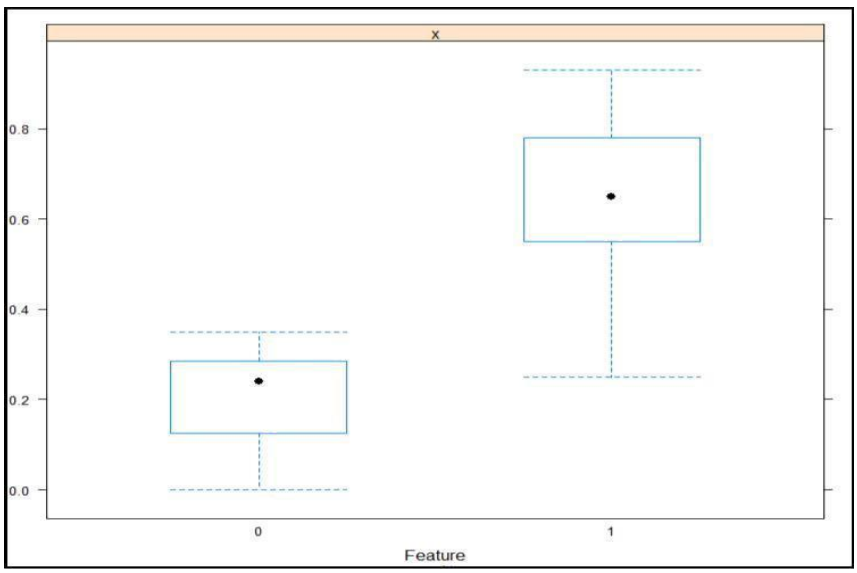
STR VS UNS:



Output 3.1(15) Feature Plot for STR vs UNS

From the plot above, we can conclude that there is some difference in STR between the User knowledge levels

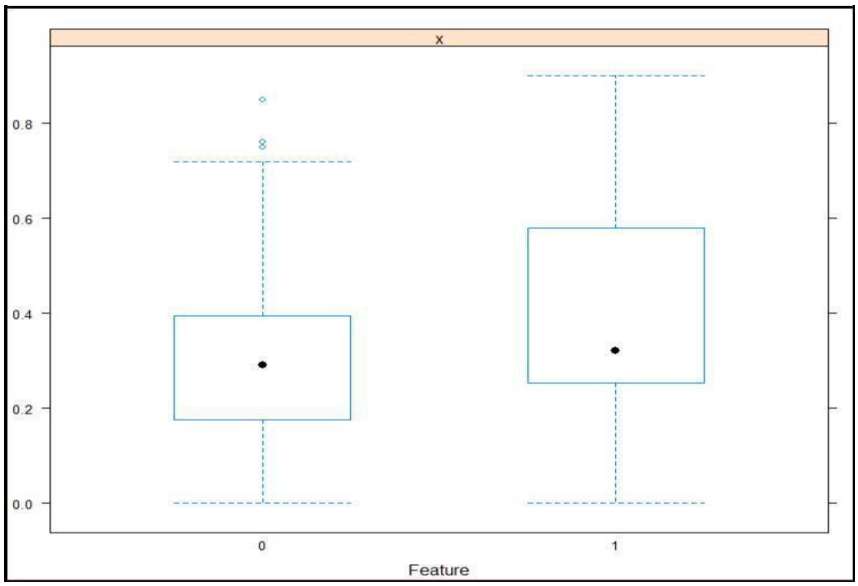
LPR VS UNS



Output 3.1(16) Feature Plot for LPR vs UNS

From the plot above, we can conclude that there is much difference in LPR between the User knowledge levels.

PEG VS UNS



Output 3.1(17) Feature Plot for PEG vs UNS

From the plot above, we can conclude that there is no difference in PEG between the User knowledge levels.

Checking for the significance difference between variables:
To test the significance difference between Continuous vs.categorical variables, we will use ANOVA.

ANOVA between STG / X1 (independent) and UNS (dependent)

```
             Df Sum Sq Mean Sq F value   Pr(>F)
dataset$UNS   1  0.522  0.5222   12.33 0.000526 ***
Residuals   256 10.841  0.0423
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 3.1(18) Significant difference b/w STG and UNS.

From the above result, as $p<0.05$, we may conclude that there is a significant difference between the STG and UNS

ANOVA between SCG / X2 (independent) and UNS (dependent)

```
             Df Sum Sq Mean Sq F value  Pr(>F)
dataset$UNS   1  0.454  0.4538   10.47 0.00137 **
Residuals   256 11.093  0.0433
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Output 3.1(19) Significant difference b/w SCG and UNS

From the above result, as p<0.05 we may conclude that there is significance difference between the variables SCG and UNS.

ANOVA between STR / X3 (independent) and UNS (dependent)

```
             Df Sum Sq Mean Sq F value  Pr(>F)
dataset$UNS   1  0.489  0.4895   8.323 0.00425 **
Residuals   256 15.056  0.0588
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Output 3.1(20) Significant difference b/w STR and UNS

From the above result, as p<0.05 we may conclude that there is significance difference between the variables STR and UNS.

ANOVA between LPR /X4 (independent) and UNS (dependent)

```
             Df Sum Sq Mean Sq F value Pr(>F)
dataset$UNS   1  0.206 0.20577   3.374 0.0674 .
Residuals   256 15.615 0.06099
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Output 3.1(21) Significant difference b/w LPR and UNS

From the above result, as p>0.05, we may conclude that there is no significant difference between LPR and UNS

ANOVA between PEG/X5 (independent) and UNS (dependent):

```
             Df Sum Sq Mean Sq F value Pr(>F)
dataset$UNS   1 11.780  11.780   608.1 <2e-16 ***
Residuals   256  4.959   0.019
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 3.1(22) Significant difference b/w PEG and UNS

From the above result, as $p<0.05$ we may conclude that there is significance difference between the variables PEG and UNS.

Splitting data into Train and Test:

Here we’ll perform the train & test split cross validation technique. So, as part of it, we need to split the original data into train and test considering 70, 30 proportions respectively.

Here, we are taking 70% of the original data randomly as Train data. And, the dimensions of the Train data are:

```
[1] 180   6
```

Output 3.1(23) Train Data

Remaining 30% of the data is considered for Testing. And, the dimensions for the test data are:

```
[1] 78  6
```

Output 3.1(24) Test Data

Further we use k-fold validation for splitting Train data into 5 folds as below

Control <- trainControl (method="repeatedcv", number=5, repeats=3)

Running pipeline using k-fold validation:

Here, we will use a pipeline of algorithms for classification to compare accuracies across different methods. As this is a classification problem, we will use Logistic regression, Decision Tree, SVM, k-NN and Random Forest techniques as part of the pipeline.

Logistic Regression:

```
Generalized Linear Model

180 samples
  5 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results:

  Accuracy   Kappa
  0.9703704  0.93904
```

Output 3.1(25) Logistic Regression

When logistic model is fitted for the train data, the accuracy obtained is 97%

Decision Tree

```
CART

180 samples
  5 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results across tuning parameters:

  cp    Accuracy   Kappa
  0.01  0.9481481  0.8942997
  0.05  0.9407407  0.8804168
  0.10  0.9425926  0.8842630

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01.
```

Output 3.1(26) Decission Tree

By fitting decision tree the accuracy level obtained is 95% at the hyper parameter value of 0.01

Support vector Machines:

```
Support Vector Machines with Radial Basis Function Kernel

180 samples
  5 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results across tuning parameters:

  sigma  Accuracy   Kappa
  0.01   0.9462963  0.8915378
  0.05   0.9592593  0.9180275
  0.10   0.9629630  0.9255041

Tuning parameter 'C' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.1 and C = 1.
```

Output 3.1(27) Support Vector Machine

96% accuracy level is obtained when the sigma value selected is 0.1and c=1 from a list of parameters using support vector machines

K-Nearest Neighborhood

```
k-Nearest Neighbors

180 samples
  5 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  1  0.9407407  0.8787970
  3  0.9388889  0.8760951
  5  0.9462963  0.8909892
  7  0.9259259  0.8502296

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Output 3.1(28) K-Nearest Neighborhood

Best accuracy i.e., 95% is obtained when k=5 nearest neighbors is chosen from a given list of neighbors by K-Nearest neighborhood method.

Random Forest

```
Random Forest

180 samples
  5 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9592593  0.9173285
  3     0.9555556  0.9096382
  5     0.9481481  0.8941865

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Output 3.1(29) Random Forest

If the number of variables selected for split is 2 then we get the best decision tree from large number of decision trees which are generally used in random forest with an accuracy of 95 %.

Comparing algorithms:

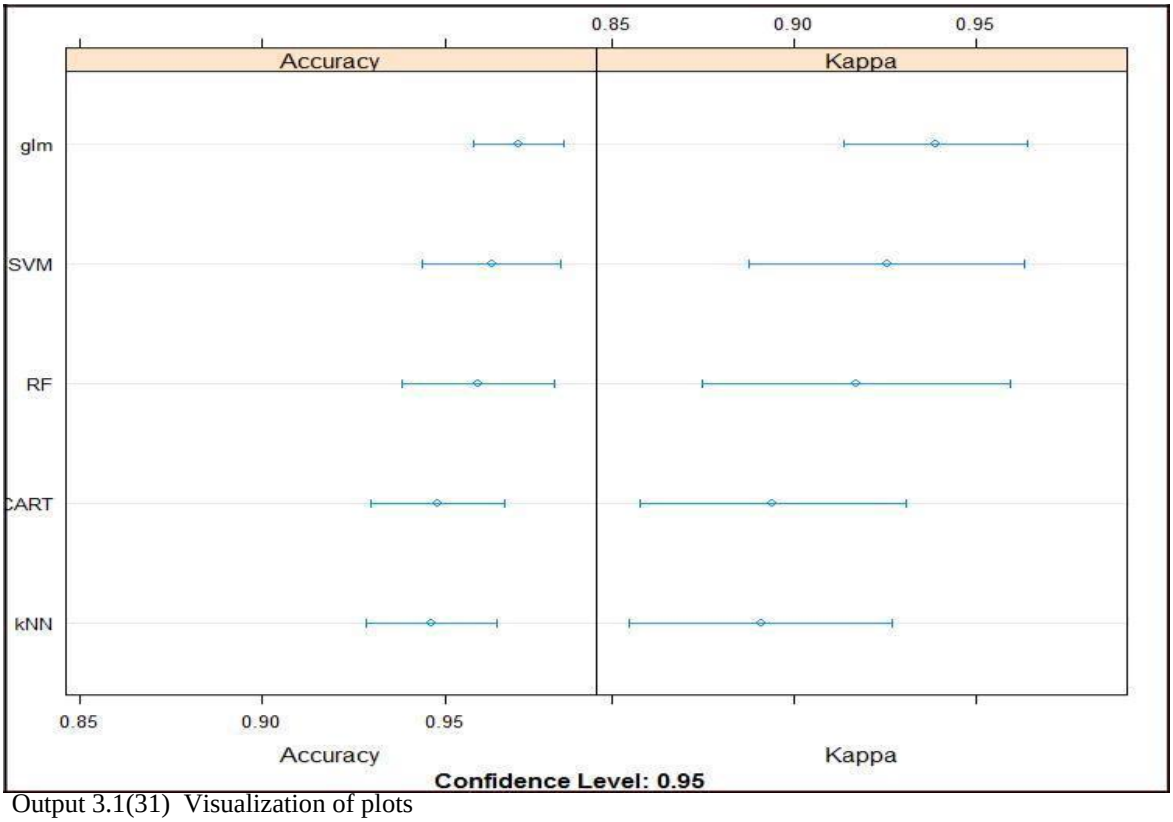
```
Models: SVM, CART, kNN, glm, RF
Number of resamples: 15

Accuracy
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM  0.8888889 0.9444444 0.9722222 0.9629630 0.9861111 1.0000000    0
CART 0.8888889 0.9305556 0.9444444 0.9481481 0.9722222 1.0000000    0
kNN  0.8888889 0.9166667 0.9722222 0.9462963 0.9722222 0.9722222    0
glm  0.9444444 0.9444444 0.9722222 0.9703704 0.9861111 1.0000000    0
RF   0.8611111 0.9444444 0.9722222 0.9592593 0.9861111 1.0000000    0

Kappa
          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
SVM  0.7798165 0.8878505 0.9433962 0.9255041 0.9716981 1.0000000    0
CART 0.7798165 0.8563187 0.8878505 0.8942997 0.9423077 1.0000000    0
kNN  0.7757009 0.8301887 0.9433962 0.8909892 0.9433962 0.9433962    0
glm  0.8834951 0.8867824 0.9423077 0.9390400 0.9716981 1.0000000    0
RF   0.7169811 0.8878505 0.9423077 0.9173285 0.9716981 1.0000000    0
```

Output 3.1(30) Comparing Algorithms

The below plot visualizes the accuracies of above 5 models:



Output 3.1(31) Visualization of plots

When we compare all the models used above, we got best accuracy for Random Forest. So, we used random forest model further to get the variable importance.

Tuning the parameters in random forest

Hyper parameters in Random Forest are tuned to extract the best parameters for final model

We considered below are Inputs to find out the best parameters:

The number of trees as 100 or 200 or300
The number of variables in each tree would range from 1 to 6.

mtry	ntree	Accuracy	Kappa
1	100	0.9572526	0.9140730
1	200	0.9588981	0.9180871
1	300	0.9588981	0.9177712
2	100	0.9608588	0.9214793
2	200	0.9607614	0.9214041
2	300	0.9626132	0.9250175
3	100	0.9607614	0.9214041
3	200	0.9589095	0.9174041
3	300	0.9607614	0.9208401
4	100	0.9589095	0.9170985
4	200	0.9554008	0.9097241
4	300	0.9590070	0.9169963
5	100	0.9552918	0.9092012
5	200	0.9552918	0.9092012
5	300	0.9572526	0.9131640
6	100	0.9552918	0.9092012
6	200	0.9552918	0.9092012
6	300	0.9552918	0.9092012

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were mtry = 2 and ntree = 300.

Output 3.1(32) Accuracy

From the above, we found the best accuracy when the number of trees is 300 with each tree containing two variables.

Finding important variable using Random Forest:

	MeanDecreaseGini
STG	6.245664
SCG	3.588184
STR	4.050709
LPR	10.133376
PEG	62.802067

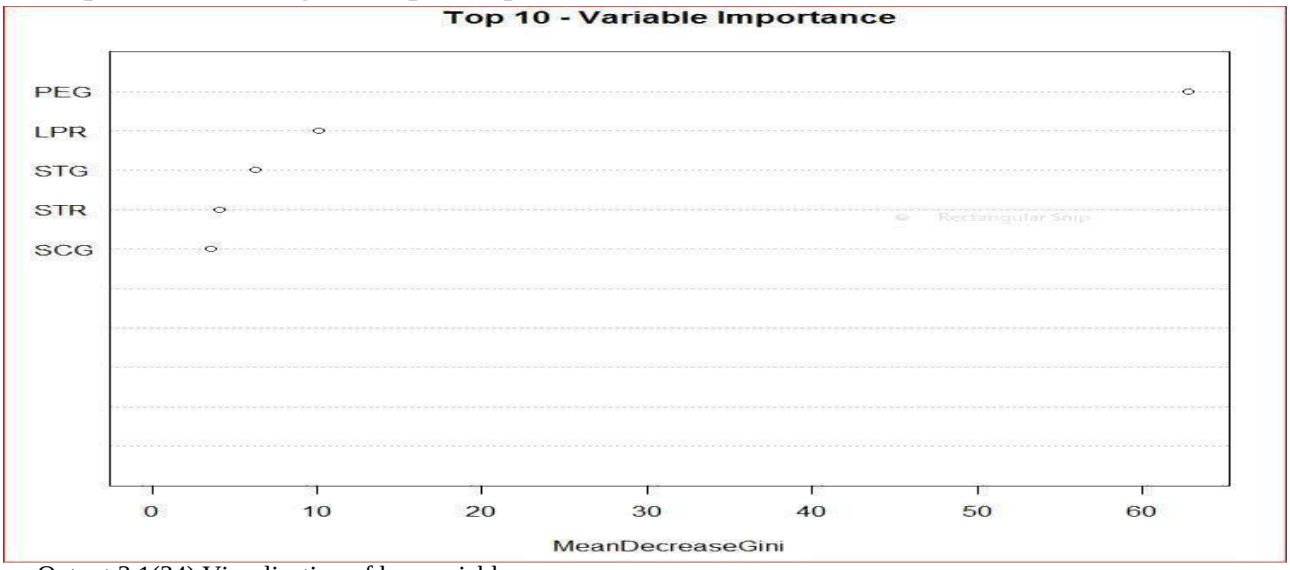
Output 3.1(33) Importance of the variable.

Above table shows the mean-decrease-gini index of each variable, based on which we are choosing the key variables. The variable which is having more gini index is more effective.

The increasing order of effective variables is **PEG> LPR> STG>STR>SCG.**

Visualization of key variables:

A plot visualizing the top 5 important variables is as follows:



Output 3.1(34) Visualization of key variables.

Above plot describes the importance of Top 5 variables actually affecting the User knowledge.

Final Model:

We considered the top 3 variables (PEG,LPR,STG) from Random Forest and fitted Logistic Regression.

Summary of the final fitted logistic model using key variables is shown below:

```
Call:
NULL

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.62352  -0.03176   0.00000   0.00015   2.56263

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -33.389     14.139  -2.361  0.01821 *
PEG           73.114     35.513   2.059  0.03951 *
LPR           16.640      6.409   2.596  0.00942 **
STG            6.749      5.069   1.331  0.18305
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 244.510  on 179  degrees of freedom
Residual deviance:  18.376  on 176  degrees of freedom
AIC: 26.376

Number of Fisher Scoring iterations: 11
```

Output 3.1(35) Summary.

```
Generalized Linear Model

180 samples
  3 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 3 times)
Summary of sample sizes: 144, 144, 144, 144, 144, 144, ...
Resampling results:

  Accuracy   Kappa
  0.9740741  0.9468734
```

Output 3.1(36) Generalized Linear Model.

As this problem is type of classification, we check its accuracy using the confusion matrix.

The confusion matrix for the **Train** data is as follows:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0  74   2
         1   1 103

               Accuracy : 0.9833
                 95% CI : (0.9521, 0.9965)
    No Information Rate : 0.5833
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9658
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9867
            Specificity : 0.9810
         Pos Pred Value : 0.9737
         Neg Pred Value : 0.9904
             Prevalence : 0.4167
         Detection Rate : 0.4111
   Detection Prevalence : 0.4222
      Balanced Accuracy : 0.9838

       'Positive' Class : 0
```

Output 3.1(37) Confusion Matrix.

From the above confusion matrix, we obtained the accuracy for the train data as 98%. We conclude that 176 observations are correctly classified out of 180 observations when the logistic model is applied for the Train data.

Validating the model using test data:

We applied the fitted Logistic regression on the Test data which is of 78 observations to validate the final model.

The confusion matrix of **Test** data is:

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 30  1
         1  2 45

               Accuracy : 0.9615
                 95% CI : (0.8917, 0.992)
    No Information Rate : 0.5897
    P-Value [Acc > NIR] : 3.508e-14

                  Kappa : 0.9201
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9375
            Specificity : 0.9783
         Pos Pred Value : 0.9677
         Neg Pred Value : 0.9574
             Prevalence : 0.4103
         Detection Rate : 0.3846
   Detection Prevalence : 0.3974
      Balanced Accuracy : 0.9579

       'Positive' Class : 0
```

Output 3.1(38) Confusion Matrix and Statistics.

Out of all Observations, 75 are correctly classified for Test data by using the same Logistic regression model i.e. we attained the accuracy of 96% for the test data.

As the accuracy obtained from train and test data do not differ significantly, the obtained logistic model is considered as a generalized model with Three important variables PEG, LPR and STG.

CHAPTER 4

SUMMARY

Summary

In order to predict the knowledge level of a student, we have applied pipeline of techniques and short listed the Logistic Regression based on accuracy value. We also applied Random forest technique to find the key variables for predicting the "Users knowledge level "and the key variables are PEG (The exam performance of user for goal object), LPR (The exam performance of user for related object), STG (The degree of study time for the goal object material).

Hence, we applied the Logistic Regression model using the key variables on the train data and validated the model using the test data. The accuracies of Train and Test data are as below:

Train Data Accuracy: 98% accuracy

Test Data Accuracy:96% accuracy

Since the accuracy of TRAIN and TEST data are more or less similar, the model is generalised. So, we can use this model to predict user knowledge level for any future data.

Appendix

R Code and Data

R CODE

 # Setting the Working directory

getwd()

setwd("D:/batch7 user knowledge")

getwd()

 # Loading required libraries to perform modeling

install. packages("mice")

library (mice)

install. packages("randomForest")

library(randomForest)

install. packages("ggplot2")

library(ggplot2)

install. packages("glmnet")

library(glmnet)

 # reading data from Folder and checking for the data types

dataset<- read.csv(file.choose(), header = T)

str(dataset)

dataset$UNS=as.factor(dataset$UNS)

str(dataset)

dim(dataset)

tail(dataset)

head(dataset)

summary(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;})
boxplot(dataset$STG)
boxplot(dataset$LPR)
boxplot(dataset$PEG)
dataset$STG[dataset$STG>quantile(dataset$STG, 0.95)] <- quantile(dataset$STG, 0.95)
boxplot(dataset$STG)
pairs(dataset)
dim(dataset)
pre1=dataset[c(1:5)]
install. packages("corrplot")
library(corrplot)
pre1.cor=cor(pre1)
pre1.cor
corrplot(pre1.cor, method="circle")
# Continuous vs. categories
install. packages("caret")
library(caret)
install. packages("ggplot2")
library(ggplot2)
x <- dataset[,1:5] y <- dataset[,6]
featurePlot(x=x, y=y,
plot="box") str(dataset)
```

#ANOVA(HYPOTHESIS TESTING)

```
x1=aov(dataset$STG ~ dataset$UNS)
summary(x1)
x2=aov(dataset$SCG ~ dataset$UNS)
summary(x2)
x3=aov(dataset$STR~ dataset$UNS)
summary(x3)
x4=aov(dataset$LPR~ dataset$UNS)
summary(x4)
x5=aov(dataset$PEG~ dataset$UNS)
summary(x5)
```

 # Splting data

```
train_rows<- sample(1:nrow(dataset), size=0.7*nrow(dataset))
train_rows
training<- dataset[train_rows, ]
test<- dataset[-train_rows, ]
dim(dataset)
dim(training)
dim(test)
head(test)
```

Model Pipleline

Run algorithms using 5-fold cross validation

```
control<- trainControl(method="repeatedcv", number=5, repeats=3)
```

 # GLM

```
set.seed(7)
fit.glm<- train(UNS~., data=training, method="glm", metric="Accuracy", trControl=control)
```

```
print(fit.glm)

# CART

set.seed(7)

grid<- expand.grid(.cp=c(0.01,0.05,0.1))

fit.cart<- train(UNS~., data=training, method="rpart",
metric="Accuracy", tuneGrid=grid, trControl=control)

print(fit.cart)

# SVM

set.seed(7)

grid<- expand.grid(.sigma=c(0.01,0.05,0.1), .C=c(1))

fit.svm<- train(UNS~., data=training, method="svmRadial",
metric="Accuracy", tuneGrid=grid, trControl=control)

print(fit.svm)

# KNN

set.seed(7)

grid<- expand.grid(.k=c(1,3,5,7))

fit.knn<- train(UNS~., data=training, method="knn", metric="Accuracy",
tuneGrid=grid, trControl=control)

print(fit.knn)

#RF

fit.rf<- train(UNS~., data=training, method="rf", metric="Accuracy", trControl=control)

print(fit.rf)

# Compare algorithms

results<- resamples(list(SVM=fit.svm, CART=fit.cart, kNN=fit.knn, glm=fit.glm, RF=fit.rf))

summary(results)

dotplot(results)

#Tunning Random Forest
```

```
customRF<- list(type = "Classification", library = "randomForest", loop = NULL)
customRF$parameters<- data.frame(parameter = c("mtry", "ntree"), class
= rep("numeric", 2), label = c("mtry", "ntree"))
customRF$grid<- function(x, y, len = NULL, search = "grid") {}
customRF$fit<- function(x, y, wts, param, lev, last, weights, classProbs, ...) {
randomForest(x, y, mtry = param$mtry, ntree=param$ntree, ...)}
customRF$predict<- function(modelFit, newdata, preProc = NULL, submodels = NULL)
predict(modelFit, newdata)
customRF$prob<- function(modelFit, newdata, preProc = NULL, submodels = NULL)
predict(modelFit, newdata, type = "prob")
customRF$sort<- function(x) x[order(x[,1]),]
customRF$levels<- function(x) x$classes
install.packages("caret")
library(caret)
library(randomForest)
control<- trainControl(method="repeatedcv", number=10, repeats=3)
tunegrid<- expand.grid(.mtry=c(1:6), .ntree=c(100, 200, 300))
set.seed(100)
custom<- train(UNS~., data=training,method=customRF,
tuneGrid=tunegrid, trControl=control)
print(custom)
rf_model3 <- randomForest(UNS ~ ., data =training, ntree=200, 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=5, repeats=3)

final.glm<-train (UNS ~ PEG+LPR+STG, 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$UNS )

caret::confusionMatrix(predslog,training$UNS)
```

#On testing

```
dim(test)

names(test)

p2 <- predict(final.glm,newdata=test,type="raw")

tabtest<- table(Predicted = p2, Actual = test$UNS)

caret::confusionMatrix(p2,test$UNS)
```

Data set

S.No.	STG	SCG	STR	LPR	PEG	UNS
1	0	0	0	0	0	0
2	0.08	0.08	0.1	0.24	0.9	1
3	0.06	0.06	0.05	0.25	0.33	0
4	0.1	0.1	0.15	0.65	0.3	1
5	0.08	0.08	0.08	0.98	0.24	0
6	0.09	0.15	0.4	0.1	0.66	1
7	0.1	0.1	0.43	0.29	0.56	1
8	0.15	0.02	0.34	0.4	0.01	0
9	0.2	0.14	0.35	0.72	0.25	0
10	0	0	0.5	0.2	0.85	1
11	0.18	0.18	0.55	0.3	0.81	1
12	0.06	0.06	0.51	0.41	0.3	0
13	0.1	0.1	0.52	0.78	0.34	1
14	0.1	0.1	0.7	0.15	0.9	1
15	0.2	0.2	0.7	0.3	0.6	1
16	0.12	0.12	0.75	0.35	0.8	1
17	0.05	0.07	0.7	0.01	0.05	0
18	0.1	0.25	0.1	0.08	0.33	0
19	0.15	0.32	0.05	0.27	0.29	0
20	0.2	0.29	0.25	0.49	0.56	1
21	0.12	0.28	0.2	0.78	0.2	0
22	0.18	0.3	0.37	0.12	0.66	1
23	0.1	0.27	0.31	0.29	0.65	1
24	0.18	0.31	0.32	0.42	0.28	0
25	0.06	0.29	0.35	0.76	0.25	0
26	0.09	0.3	0.68	0.18	0.85	1
27	0.04	0.28	0.55	0.25	0.1	0
28	0.09	0.255	0.6	0.45	0.25	0
29	0.08	0.325	0.62	0.94	0.56	1
30	0.15	0.275	0.8	0.21	0.81	1

31	0.12	0.245	0.75	0.31	0.59	1
32	0.15	0.295	0.75	0.65	0.24	0
33	0.1	0.256	0.7	0.76	0.16	0
34	0.18	0.32	0.04	0.19	0.82	1
35	0.2	0.45	0.28	0.31	0.78	1
36	0.06	0.35	0.12	0.43	0.29	0
37	0.1	0.42	0.22	0.72	0.26	0
38	0.18	0.4	0.32	0.08	0.33	0
39	0.09	0.33	0.31	0.26	0	0
40	0.19	0.38	0.38	0.49	0.45	1
41	0.02	0.33	0.36	0.76	0.1	0
42	0.2	0.49	0.6	0.2	0.78	1
43	0.14	0.49	0.55	0.29	0.6	1
44	0.18	0.33	0.61	0.64	0.25	1
45	0.115	0.35	0.65	0.27	0.04	0
46	0.17	0.36	0.8	0.14	0.66	1
47	0.1	0.39	0.75	0.31	0.62	1
48	0.13	0.39	0.85	0.38	0.77	1
49	0.18	0.34	0.71	0.71	0.9	1
50	0.09	0.51	0.02	0.18	0.67	1
51	0.06	0.5	0.09	0.28	0.25	0
52	0.23	0.7	0.19	0.51	0.45	1
53	0.09	0.55	0.12	0.78	0.05	0
54	0.24	0.75	0.32	0.18	0.86	1
55	0.18	0.72	0.37	0.29	0.55	1
56	0.1	0.6	0.33	0.42	0.26	0
57	0.2	0.52	0.36	0.84	0.25	1
58	0.09	0.6	0.66	0.19	0.59	1
59	0.18	0.51	0.58	0.33	0.82	1
60	0.08	0.58	0.6	0.64	0.1	0
61	0.09	0.61	0.53	0.75	0.01	0
62	0.06	0.77	0.72	0.19	0.56	1
63	0.15	0.79	0.78	0.3	0.51	1
64	0.2	0.68	0.73	0.48	0.28	0
65	0.24	0.58	0.76	0.8	0.28	1
66	0.25	0.1	0.03	0.09	0.15	0
67	0.32	0.2	0.06	0.26	0.24	0
68	0.29	0.06	0.19	0.55	0.51	1
69	0.28	0.1	0.12	0.28	0.32	0
70	0.3	0.08	0.4	0.02	0.67	1
71	0.27	0.12	0.37	0.29	0.58	1
72	0.31	0.1	0.41	0.42	0.75	1
73	0.29	0.15	0.33	0.66	0.08	0
74	0.3	0.2	0.52	0.3	0.53	1
75	0.28	0.16	0.69	0.33	0.78	1

76	0.255	0.18	0.5	0.4	0.1	0
77	0.265	0.06	0.57	0.75	0.1	0
78	0.275	0.1	0.72	0.1	0.3	0
79	0.245	0.1	0.71	0.26	0.2	0
80	0.295	0.2	0.86	0.44	0.28	0
81	0.32	0.12	0.79	0.76	0.24	0
82	0.295	0.25	0.26	0.12	0.67	1
83	0.315	0.32	0.29	0.29	0.62	1
84	0.25	0.29	0.15	0.48	0.26	0
85	0.27	0.1	0.1	0.7	0.25	0
86	0.248	0.3	0.31	0.2	0.03	0
87	0.325	0.25	0.38	0.31	0.79	1
88	0.27	0.31	0.32	0.41	0.28	0
89	0.29	0.29	0.4	0.78	0.18	0
90	0.29	0.3	0.52	0.09	0.67	1
91	0.258	0.28	0.64	0.29	0.56	1
92	0.32	0.255	0.55	0.78	0.34	1
93	0.251	0.265	0.57	0.6	0.09	0
94	0.288	0.31	0.79	0.23	0.24	0
95	0.323	0.32	0.89	0.32	0.8	1
96	0.255	0.305	0.86	0.62	0.15	0
97	0.295	0.25	0.73	0.77	0.19	0
98	0.258	0.25	0.295	0.33	0.77	1
99	0.29	0.25	0.29	0.29	0.57	1
100	0.243	0.27	0.08	0.42	0.29	0
101	0.27	0.28	0.18	0.48	0.26	0
102	0.299	0.32	0.31	0.33	0.87	1
103	0.3	0.27	0.31	0.31	0.54	1
104	0.245	0.26	0.38	0.49	0.27	0
105	0.295	0.29	0.31	0.76	0.1	0
106	0.29	0.3	0.56	0.25	0.67	1
107	0.26	0.28	0.6	0.29	0.59	1
108	0.305	0.255	0.63	0.4	0.54	1
109	0.32	0.27	0.52	0.81	0.3	1
110	0.299	0.295	0.8	0.37	0.84	1
111	0.276	0.255	0.81	0.27	0.33	0
112	0.258	0.31	0.88	0.4	0.3	0
113	0.32	0.28	0.72	0.89	0.58	1
114	0.329	0.55	0.02	0.4	0.79	1
115	0.295	0.59	0.29	0.31	0.55	1
116	0.285	0.64	0.18	0.61	0.45	1
117	0.265	0.6	0.28	0.66	0.07	0
118	0.315	0.69	0.28	0.8	0.7	1
119	0.28	0.78	0.44	0.17	0.66	1
120	0.325	0.61	0.46	0.32	0.81	1

166	0.4	0.33	0.12	0.3	0.9	1
167	0.34	0.4	0.38	0.2	0.61	1
168	0.38	0.36	0.46	0.49	0.78	1
169	0.35	0.38	0.32	0.6	0.16	0
170	0.41	0.49	0.34	0.21	0.92	1
171	0.42	0.36	0.63	0.04	0.25	0
172	0.43	0.38	0.62	0.33	0.49	1
173	0.44	0.33	0.59	0.53	0.85	1
174	0.4	0.42	0.58	0.75	0.16	0
175	0.46	0.44	0.89	0.12	0.66	1
176	0.38	0.39	0.79	0.33	0.3	0
177	0.39	0.42	0.83	0.65	0.19	0
178	0.49	0.34	0.88	0.75	0.71	1
179	0.46	0.64	0.22	0.22	0.6	1
180	0.44	0.55	0.11	0.26	0.83	1
181	0.365	0.68	0.1	0.63	0.18	0
182	0.45	0.65	0.19	0.99	0.55	1
183	0.46	0.78	0.38	0.24	0.89	1
184	0.37	0.55	0.41	0.29	0.3	0
185	0.38	0.59	0.31	0.62	0.2	0
186	0.49	0.64	0.34	0.78	0.21	0
187	0.495	0.82	0.67	0.01	0.93	1
188	0.44	0.69	0.61	0.29	0.57	1
189	0.365	0.57	0.59	0.55	0.25	0
190	0.49	0.9	0.52	0.9	0.47	1
191	0.445	0.7	0.82	0.16	0.64	1
192	0.42	0.7	0.72	0.3	0.8	1
193	0.37	0.6	0.77	0.4	0.5	1
194	0.4	0.61	0.71	0.88	0.67	1
195	0.6	0.14	0.22	0.11	0.66	1
196	0.55	0.1	0.27	0.25	0.29	0
197	0.68	0.19	0.19	0.48	0.1	0
198	0.73	0.2	0.07	0.72	0.26	0
199	0.78	0.15	0.38	0.18	0.63	1
200	0.55	0.1	0.34	0.3	0.1	0
201	0.59	0.18	0.31	0.55	0.09	0
202	0.64	0.09	0.33	0.65	0.5	1
203	0.6	0.19	0.55	0.08	0.1	0
204	0.69	0.02	0.62	0.3	0.29	0
205	0.78	0.21	0.68	0.65	0.75	1
206	0.62	0.14	0.52	0.81	0.15	0
207	0.7	0.18	0.88	0.09	0.66	1
208	0.75	0.015	0.78	0.31	0.53	1
209	0.55	0.17	0.71	0.48	0.11	0
210	0.85	0.05	0.91	0.8	0.68	1

211	0.78	0.27	0.13	0.14	0.62	1
212	0.8	0.29	0.06	0.31	0.51	1
213	0.9	0.26	0.19	0.58	0.79	1
214	0.76	0.258	0.07	0.83	0.34	1
215	0.72	0.32	0.48	0.2	0.6	1
216	0.6	0.251	0.39	0.29	0.3	0
217	0.52	0.288	0.32	0.5	0.3	0
218	0.6	0.31	0.31	0.87	0.58	1
219	0.51	0.255	0.55	0.17	0.64	1
220	0.58	0.295	0.62	0.28	0.3	0
221	0.61	0.258	0.56	0.62	0.24	0
222	0.77	0.267	0.59	0.78	0.28	1
223	0.79	0.28	0.88	0.2	0.66	1
224	0.68	0.27	0.78	0.31	0.57	1
225	0.58	0.299	0.73	0.63	0.21	0
226	0.77	0.29	0.74	0.82	0.68	1
227	0.71	0.475	0.13	0.23	0.59	1
228	0.58	0.348	0.06	0.29	0.31	0
229	0.88	0.335	0.19	0.55	0.78	1
230	0.99	0.49	0.07	0.7	0.69	1
231	0.73	0.43	0.32	0.12	0.65	1
232	0.61	0.33	0.36	0.28	0.28	0
233	0.51	0.4	0.4	0.59	0.23	0
234	0.83	0.44	0.49	0.91	0.66	1
235	0.66	0.38	0.55	0.15	0.62	1
236	0.58	0.35	0.51	0.27	0.3	0
237	0.523	0.41	0.55	0.6	0.22	0
238	0.66	0.36	0.56	0.4	0.83	1
239	0.62	0.37	0.81	0.13	0.64	1
240	0.52	0.44	0.82	0.3	0.52	1
241	0.5	0.4	0.73	0.62	0.2	0
242	0.71	0.46	0.95	0.78	0.86	1
243	0.64	0.55	0.15	0.18	0.63	1
244	0.52	0.85	0.06	0.27	0.25	0
245	0.62	0.62	0.24	0.65	0.25	1
246	0.91	0.58	0.26	0.89	0.88	1
247	0.62	0.67	0.39	0.1	0.66	1
248	0.58	0.58	0.31	0.29	0.29	0
249	0.89	0.68	0.49	0.65	0.9	1
250	0.72	0.6	0.45	0.79	0.45	1
251	0.68	0.63	0.65	0.09	0.66	1
252	0.56	0.6	0.6	0.31	0.5	1
253	0.54	0.51	0.55	0.64	0.19	0
254	0.61	0.78	0.69	0.92	0.58	1
255	0.78	0.61	0.71	0.19	0.6	1
256	0.54	0.82	0.71	0.29	0.77	1

257	0.5	0.75	0.81	0.61	0.26	1
258	0.66	0.9	0.76	0.87	0.74	1

6.BIBLIOGRAPHY

1. Multivariate data analysis (Fifth Edition) --- Joseph F.Hair, Rolph E.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 -- Karthik Ramasubramanian, Abhishek Singh

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

10. Deep learning with Deep learning R by François Chollet