A PROJECT REPORT ON

PREDICTING THE PHYSICAL EXAMINATION STATUS FOR THE TRAINEES OF SUB – INSPECTOR OF POLICE.

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

G.SANDEEP	**Roll No: 1007-17-508-012**
B.VIJAYA SRI	**Roll No: 1007-17-508-013**
G.CHANDANA PRIYA	**Roll No: 1007-17-508-016**
R.C.APOORVA	**Roll No: 1007-17-508-009**
CH.KOMALA PRIYANKA	**Roll No: 1007-17-508-015**
B.SRILATHA	**Roll No: 1007-17-508-027**

Under the Supervision of
Ms. T. SANDHYA
2018

A PROJECT REPORT ON

PREDICTING THE PHYSICAL EXAMINATION STATUS FOR THE TRAINEES OF SUB – INSPECTOR OF POLICE.

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

G.SANDEEP	**Roll No: 1007-17-508-012**
B.VIJAYA SRI	**Roll No: 1007-17-508-013**
G.CHANDANA PRIYA	**Roll No: 1007-17-508-016**
R.C.APOORVA	**Roll No: 1007-17-508-009**
CH.KOMALA PRIYANKA	**Roll No: 1007-17-508-015**
B.SRILATHA	**Roll No: 1007-17-508-027**

Under the Supervision of
T. SANDHYA
2018

CERTIFICATE

This is to certify that

G.SANDEEP	**Roll No: 1007-17-508-012**
B.VIJAYA SRI	**Roll No: 1007-17-508-013**
G.CHANDANA PRIYA	**Roll No: 1007-17-508-016**
R.C.APOORVA	**Roll No: 1007-17-508-009**
CH.KOMALA PRIYANKA	**Roll No: 1007-17-508-015**
B.SRILATHA	**Roll No: 1007-17-508-027**

have submitted the project titled **"PREDICTING THE PHYSICAL EXAMINATION STATUS FOR THE TRAINEES OF SUB – INSPECTOR OF POLICE"** in partial fulfillment for the degree of Master of Science in Statistics.

Head of Department of Statistics

Internal Examiner

External Examiner

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.

G.SANDEEP

B.VIJAYASRI

G.CHANDANA PRIYA

R.C.APOORVA

CH.KOMALA PRIYANKA

B.SRILATHA

Department of Statistics

Osmania University

Hyderabad – 500007, T.S.

INDIA

ACKNOWLEDGEMENTS

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

We also express our 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 we record our 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.

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

We would like to express our deepest gratitude to **Ms.T.SANDHYA** and **Mr.BALA KARTHIK** for their advice, guidance and involvement at various stages of this work, We would also like to thank them for their understanding and constant encouragement throughout this project.

We thank all Non-Teaching members of the Department of Statistics, who helped us during our Thesis work.

We are thankful to the Osmania University for permitting us to carry out this work.

CONTENTS

CHAPTER-1
INTRODUCTION AND SCOPE OF THE PROBLEM

INTRODUCTION

1.1 Scope of the problem

This problem refers to predicting the physical examination status of the trainees of Sub Inspector of Police, from the data collected at Osmania University premises.

1.2 Data Description:

To achieve this, we have prepared a questionnaire which consists of following questions:

1. Name: _____
2. Age: _____
3. Height (in cms):_____
4. Weight (in kgs):_____
5. Running (in meters):_____
6. Long Jump (in meters):_____
7. Shot Put (in meters):_____
8. Rice :

- ❑ Once
- ❑ Twice
- ❑ Thrice

9. Chapatti (in grams):_____
10. Ragi Malt (no. of glasses):_____
11. Eggs (In number):_____
12. Non Veg., (no. of times):_____
13. Milk (no. of liters):_____
14. Fruits (banana):_____
15. Practicing from (how many days):_____
16. Daily workout timings (average):_____

We calculated below variables from the data collected and considered these as Input for the further analysis

- ✓ BMI calculated from Height and Weight of the Trainees
- ✓ Calories calculated from the daily food intake
- ✓ Protein calculated from the daily food intake
- ✓ Practicing since how many days
- ✓ Workout Number of hours per day
- ✓ Qualify_PhysicalTest Extracted from tests(running ,long jump and shot put) conducted to the trainees

1.3 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.

Appendix: Describes the Data used for the analysis as well as the Code in R.

Source: Data is Collected from S.I. trainees in the C-ground of Osmania University.

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

- Business Understanding
- Data Understanding
- Data preparation
- Modelling
- Evaluation
- Deployment

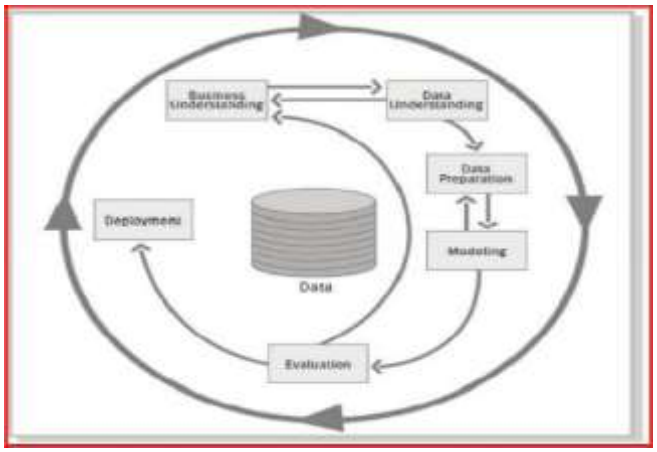


Fig 2.1: Machine learning process

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 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 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

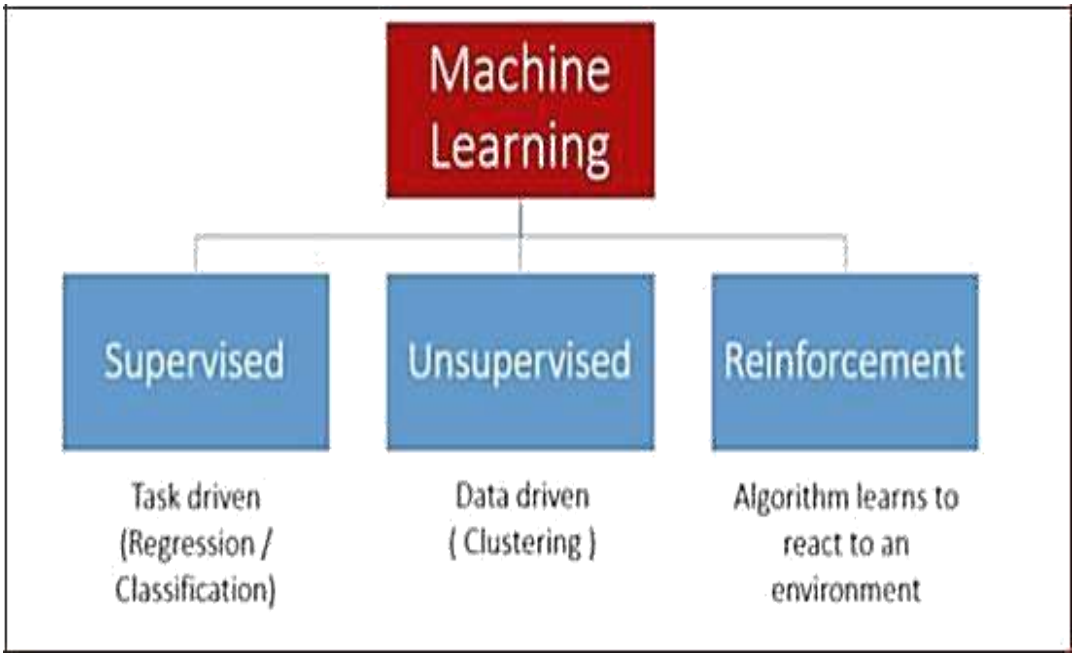


Fig 2.2 Types of Machine Learning

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:

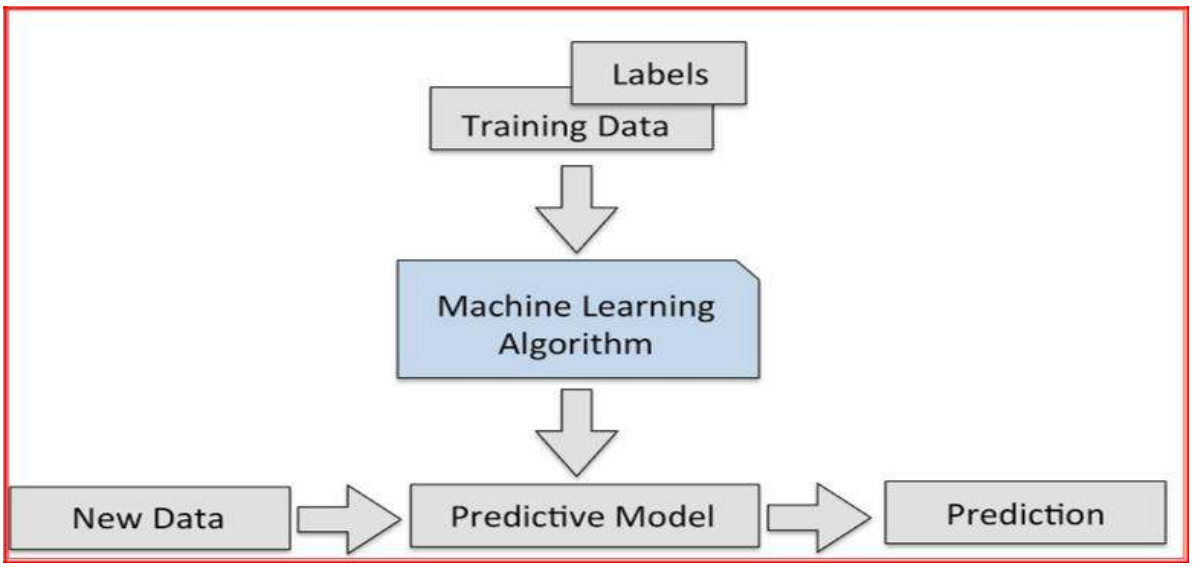


Fig 2.2.1 Supervised Learning

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

Classification and Regression

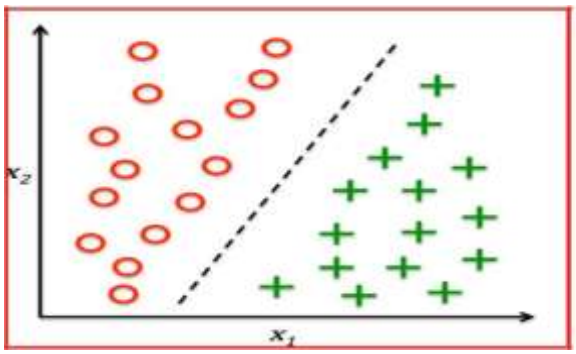


Fig 2.2.1 Classification

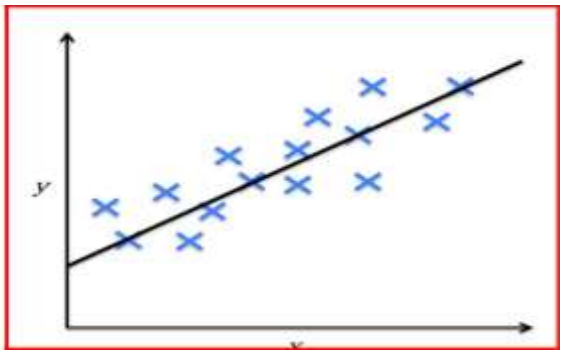


Fig 2.2.1 Regression

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)

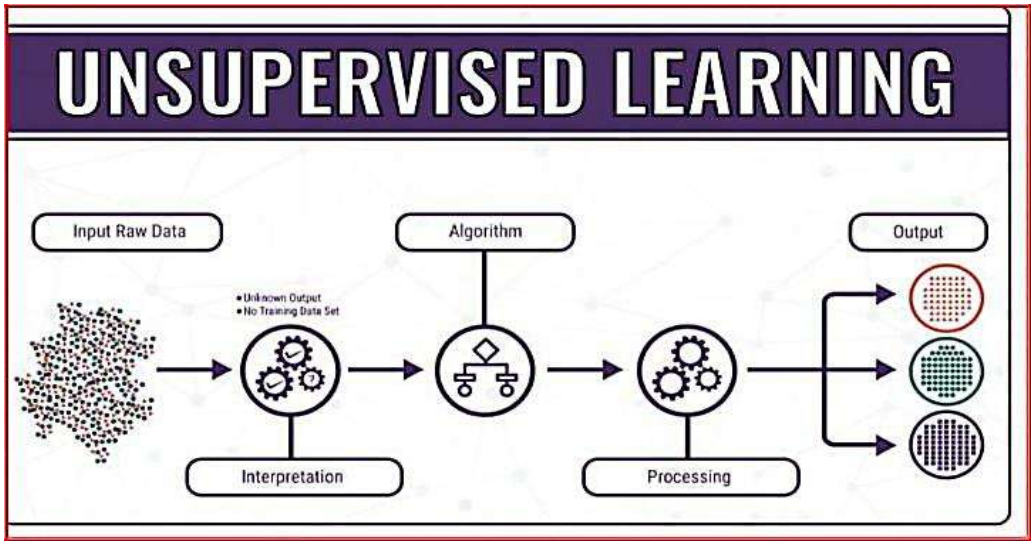


Fig 2.2.2 Unsupervised Learning

Examples of unsupervised learning: data reduction techniques, CA etc.

Cluster Analysis and Data reduction techniques:

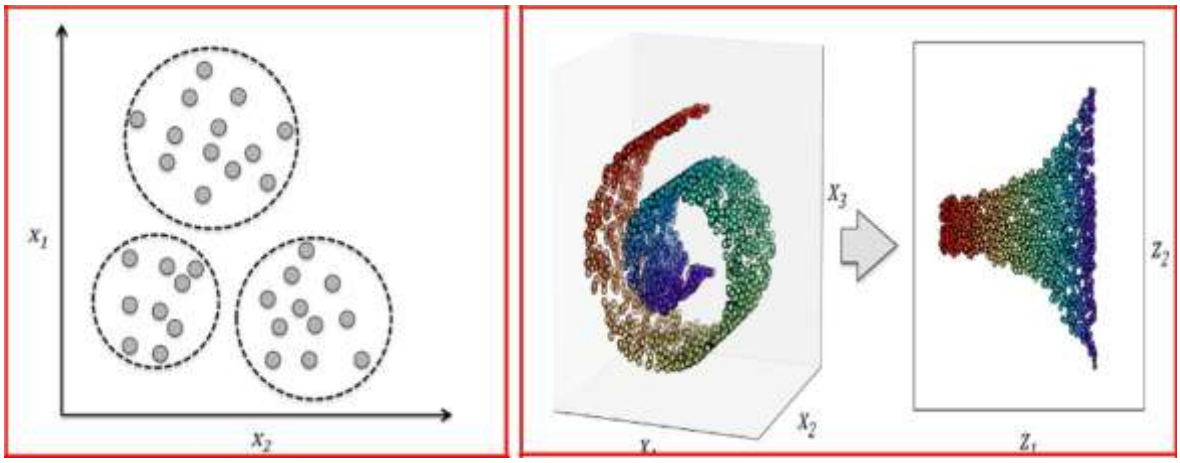


Fig 2.2.2 Cluster analysis and data reduction techniques

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:

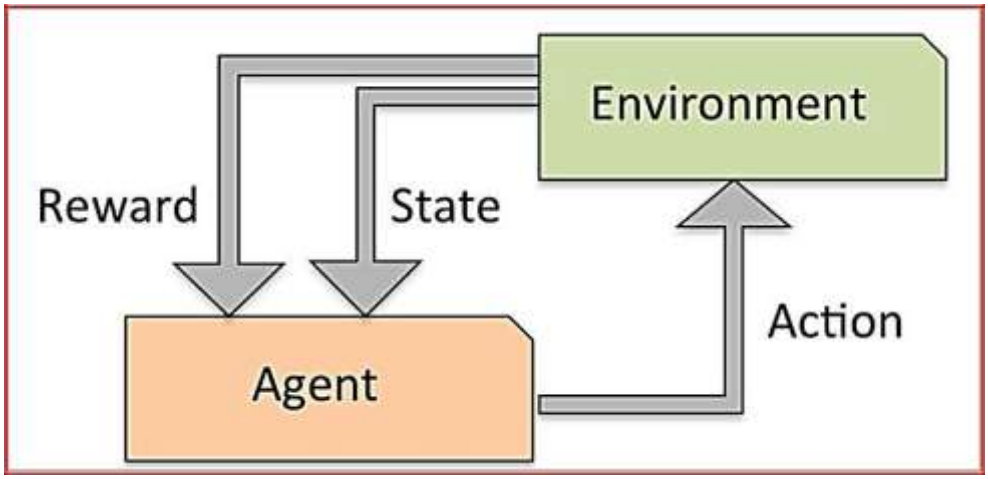


Fig 2.2.3 Reinforcement Learning

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 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:

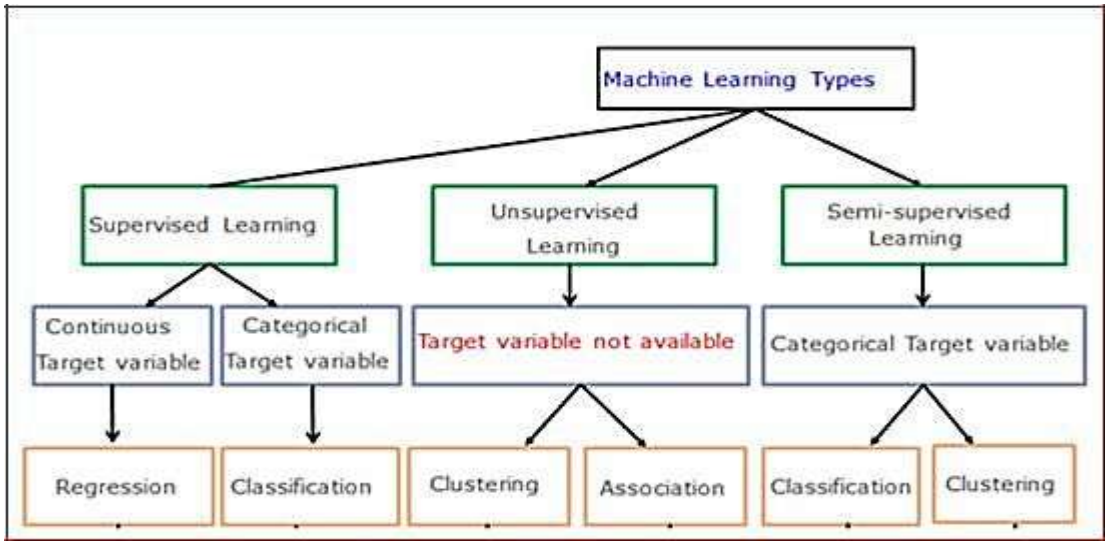


Fig 2.3 Choosing the algorithm

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:

- **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.
- **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 regression is used to explain the relationship between one dependent variable with two or more independent variables. The independent variables can be either continuous or categorical.

- **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 inthe form of a curve.
- **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 kernel-induced feature space. The system capacity will be controlled by parameters that do not depend on the dimensionality of feature space.
- **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.
- **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 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 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.

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 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 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 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.

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.

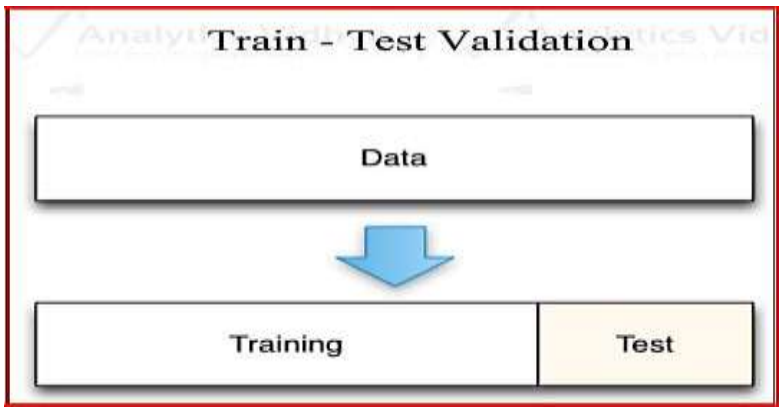


Fig 2.4.1 Model Validation

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 kth 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.

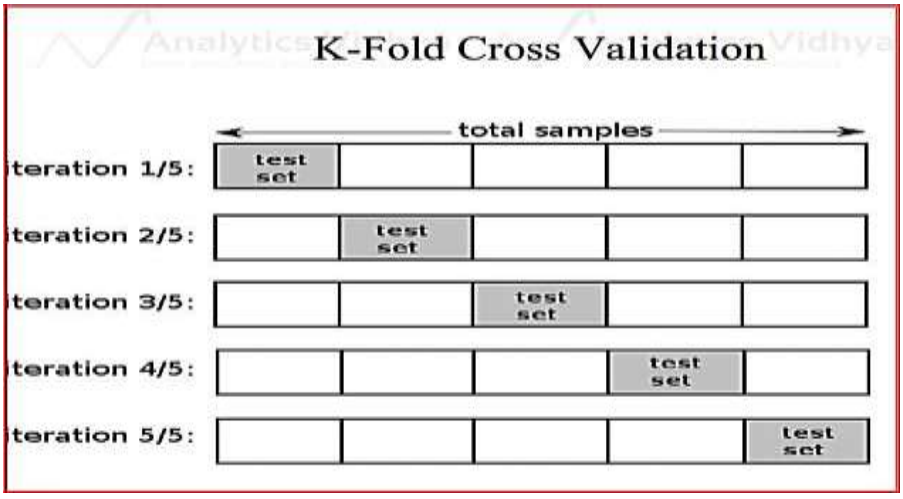


Fig 2.4.1 K-fold cross validation

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

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.

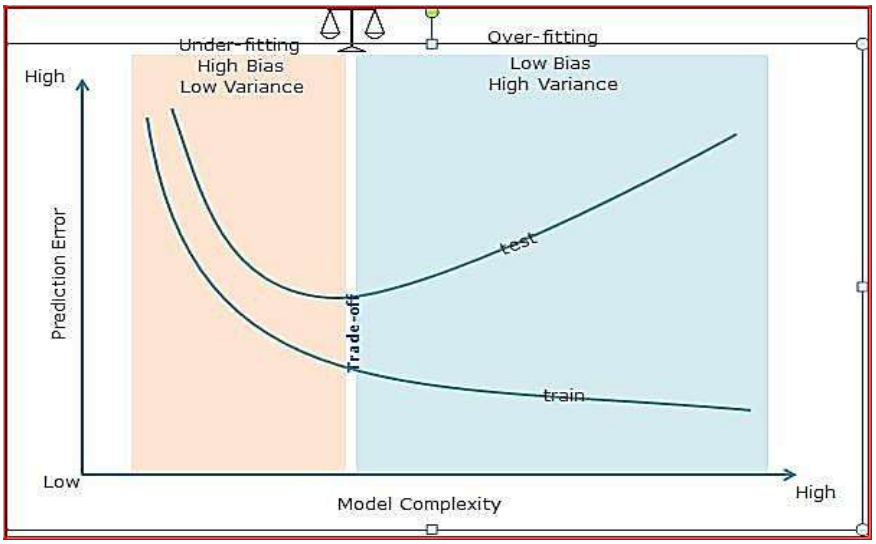


Fig 2.5.1 Bias and Variance

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 1. 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.

shows an example confusion matrix.

. 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.

Table 2. Measures to assess the performance of the model

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:

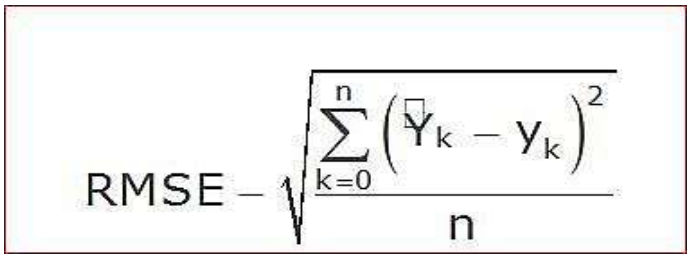


Fig 2.5.2 Root Mean square Error

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

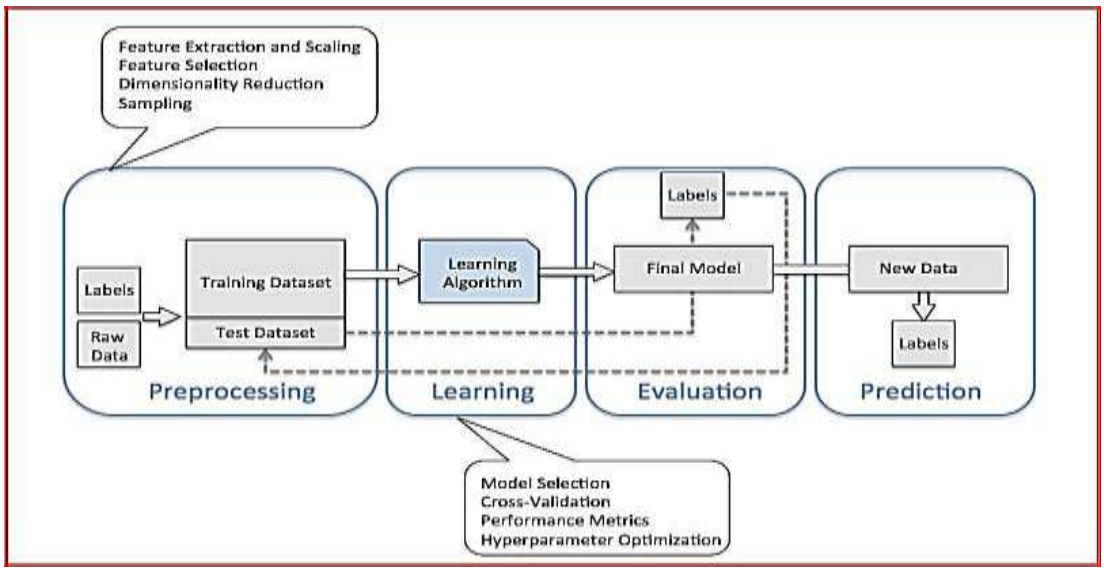


Fig 2.6 Overall Process of Machine Learning

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 **the records of performance status of S.I. trainees** from **C-ground of Osmania University** 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.

1.Structure of the data:

We will first look at the structure of the data

```
'data.frame':       140 obs. of  6 variables:
$ BMI               : num  0.68 1.774 0.712 -0.701 -0.607 ...
$ calories          : num  -1.42 -1.42 -1.36 -1.31 -1.24 ...
$ Protein           : num  -1.134 -0.959 -1.03 -0.926 -0.886 ...
$ practice          : num  0.0506 0.1849 2.1984 -0.9427 -0.3521 ...
$ workout            : num  -1.339 0.915 0.164 -0.588 -0.588 ...
$ Qualify_PhysicalTest: int  1 1 1 1 0 1 0 0 1 1 ...
```

Output 1.1 :Structure of the data before conversion

From the above structure BMI , Calories , Protein , Practice , Workout are in numeric data type , because those are continuous variables but, Qualify_PhysicalTest is in integer data type. According to our data, it must be a categorical variable. So, converting it into a categorical variable.

```
'data.frame':       140 obs. of  6 variables:
$ BMI                       : num  23.9 28.5 2.4 19.2 19.5 …....
$ calories                  : num  390 392 427 464 505 …...
$ Protein                   : num  8.1 15.4 12.5 16.8 18.5 …....
$ practice                  : int    40 45 120 3 25 60 20 30 14 7 ……...
$ workout                   : num  2 5 4 3 3 6 4 3 2 2  ……..…...
$ Qualify_PhysicalTest : Factor w/2 levels “0”,”1”:1 1 2 2 1 2 1 1 2 2 ...
```

Output 1.2 : Structure of the data after conversion

2.Understanding data using Descriptive Statistics:

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

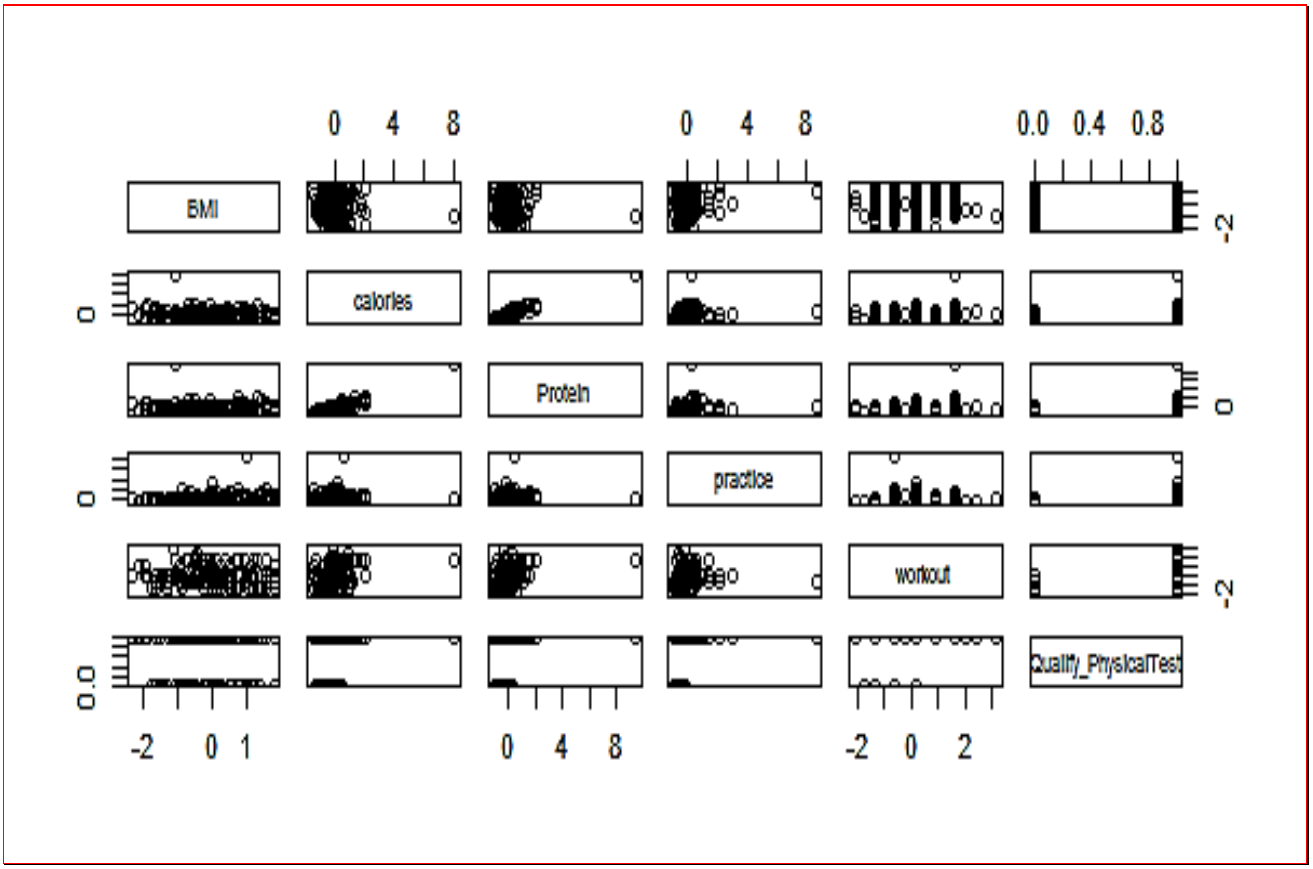
```
      BMI          calories       Protine          practice        workout
 Min.   :13.67   Min.   : 390   Min.   :  8.10   Min.   :  3.00   Min.   :1.000
 1st Qu.:19.19   1st Qu.:1002   1st Qu.: 37.76   1st Qu.: 20.00   1st Qu.:3.000
 Median :21.53   Median :1228   Median : 49.91   Median : 30.00   Median :4.000
 Mean   :21.80   Mean   :1315   Mean   : 55.50   Mean   : 38.11   Mean   :3.782
 3rd Qu.:24.00   3rd Qu.:1521   3rd Qu.: 65.11   3rd Qu.: 45.00   3rd Qu.:4.000
 Max.   :36.28   Max.   :6558   Max.   :451.22   Max.   :360.00   Max.   :8.000
 Qualify_PhysicalTest
 0:60
 1:80
```

Output 2.1 :Summary of the data

From the above table, we can see the counts 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



Output 2.2: Understanding data through Visualization

3.Checking for missing Values:

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

```
    BMI            calories             Protine            practice
      0                   0                   0                   0
workout Qualify_PhysicalTest
      0                   0
```

Output 3.1: Missing values in the data

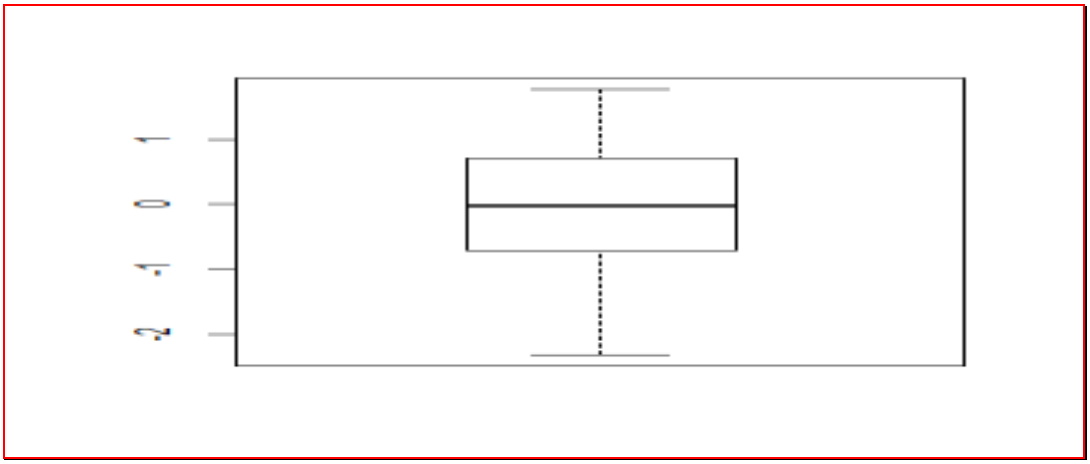
The missing values for the continuous variables will be imputed using Mean / Median value of the valid records and the categorical variables will be imputed using Mode value.

As we do not have any missing values in our data, Imputation is not carried out.

3.2 Checking for Outliers:

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

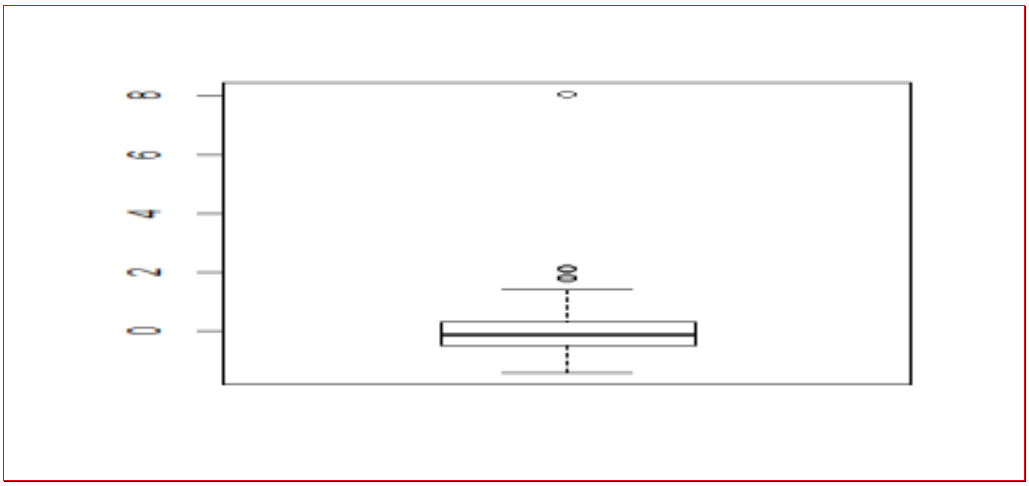
Box plot for BMI:



Output 3.2.1 :With outliers

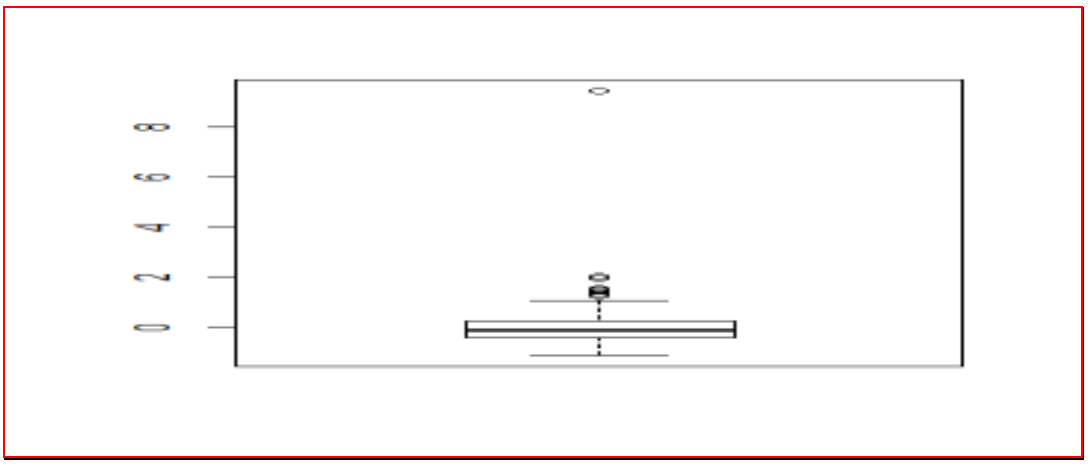
As there are no outliers in the column of BMI, we see no outliers present in the boxplot of BMI.

Calories:



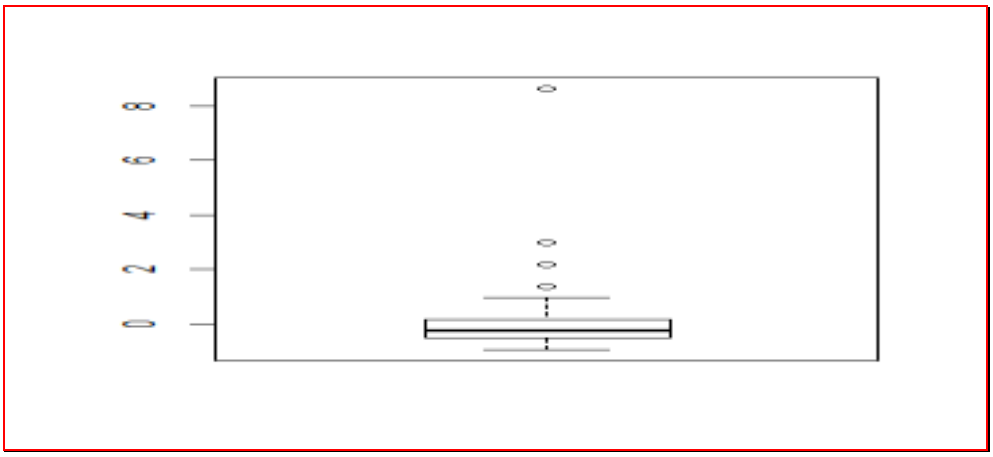
Output 3.2.2: With outliers

Protein:



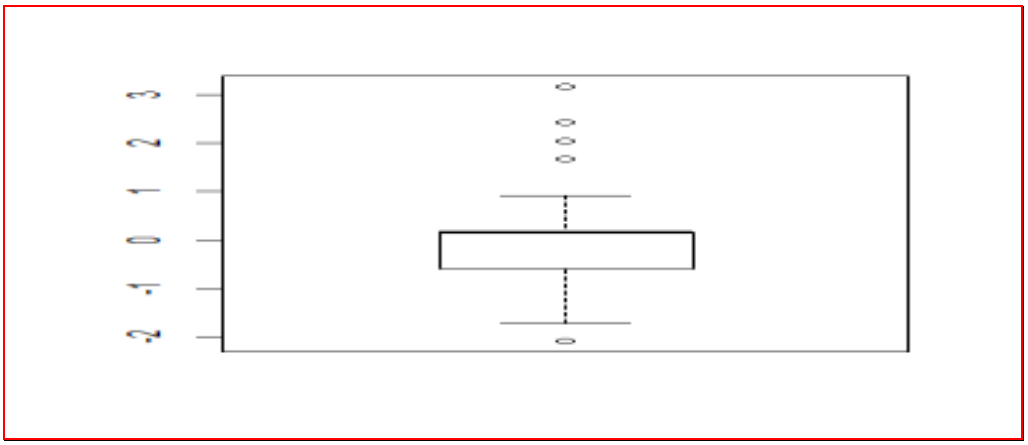
Output 3.2.3: With outliers

Practice:



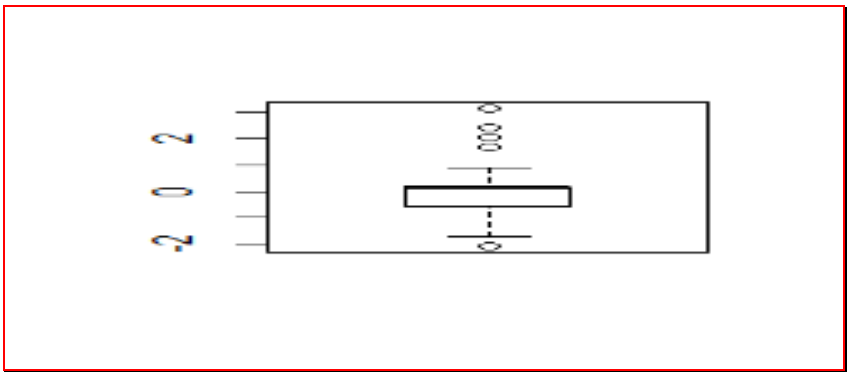
Output 3.2.4: With outliers

Workout:



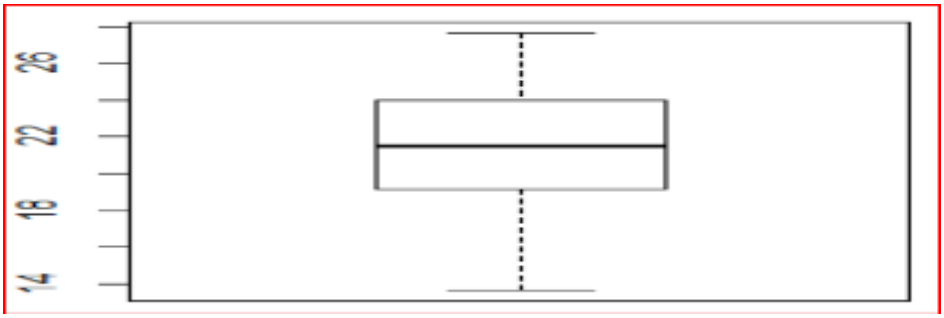
Output 3.2.5: With outliers

3.3 Treating outliers:



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

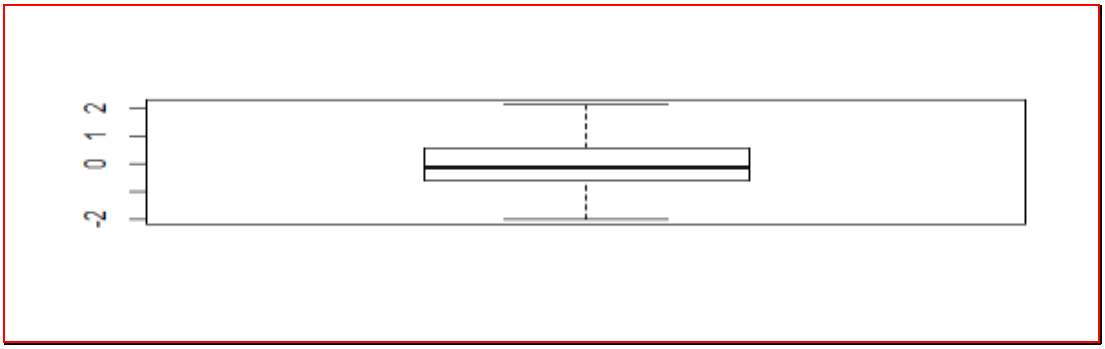
Box plots after treating Outliers:



For BMI:

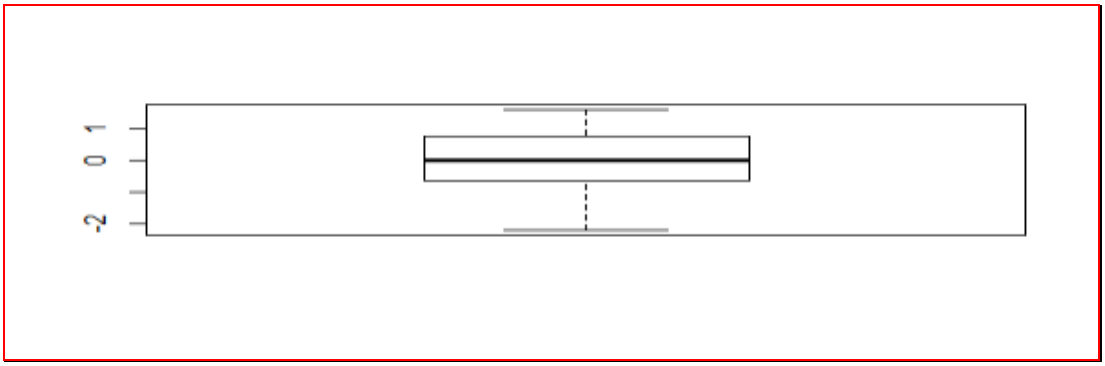
Here we can see, there are no outliers in BMI.

For Calories:



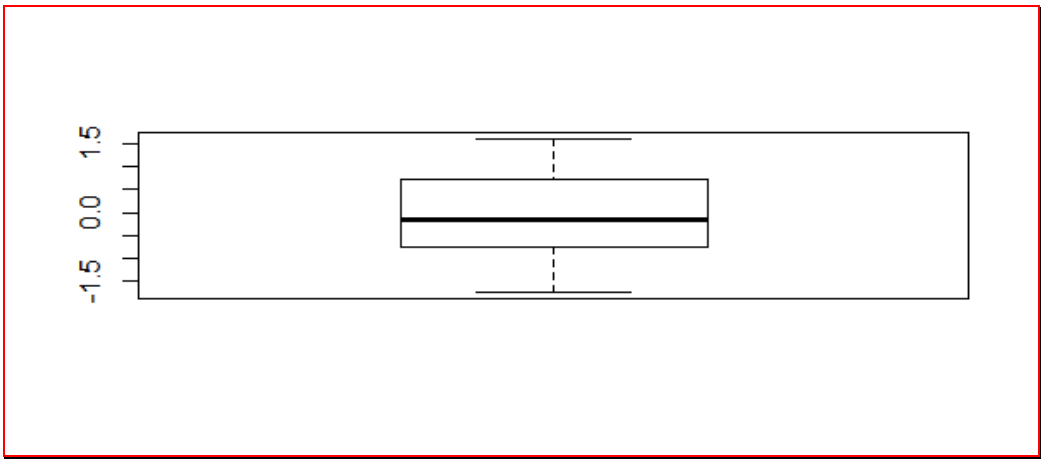
Output 3.3.1: After replacing outliers

Protein:



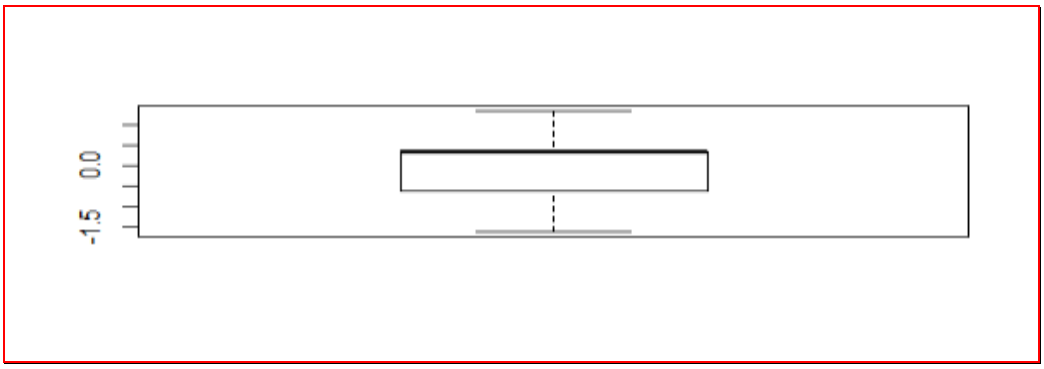
Output 3.3.2: After replacing outliers

Practice:



Output 3.3.3 : After replacing outliers

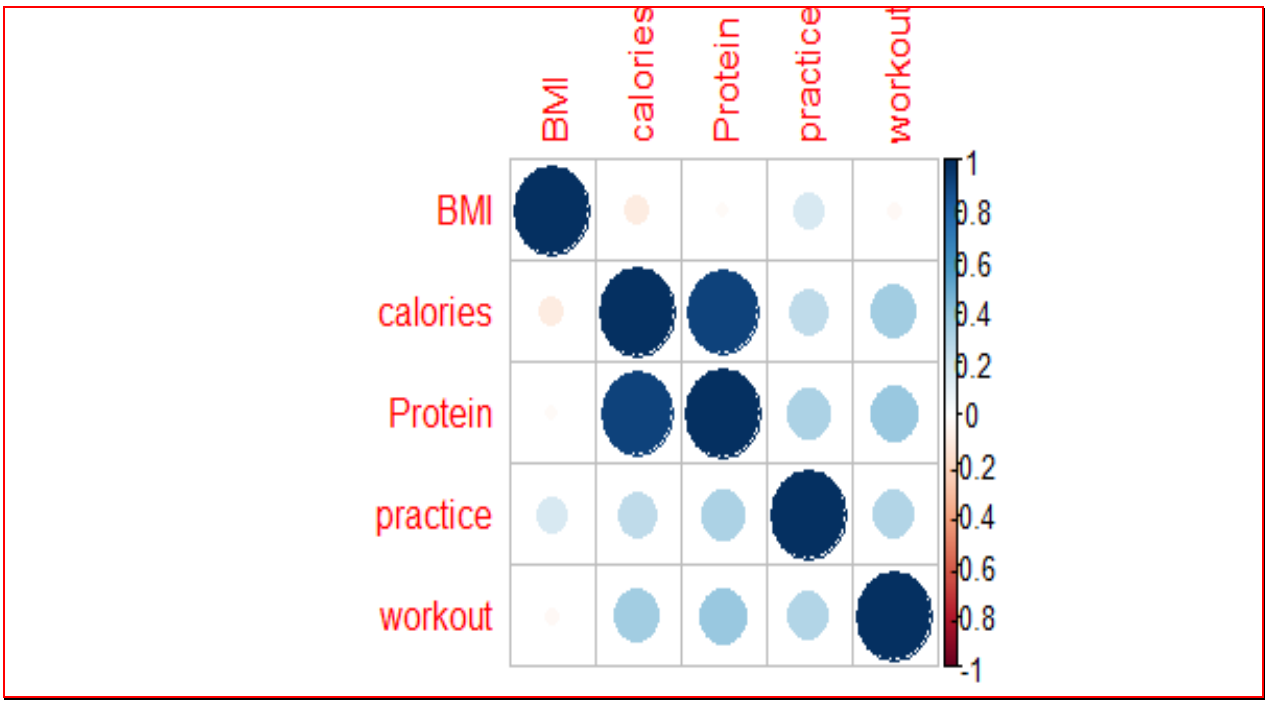
Workout:



Output 3.3.4: After replacing outliers

4.Understanding relationships between variables:

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



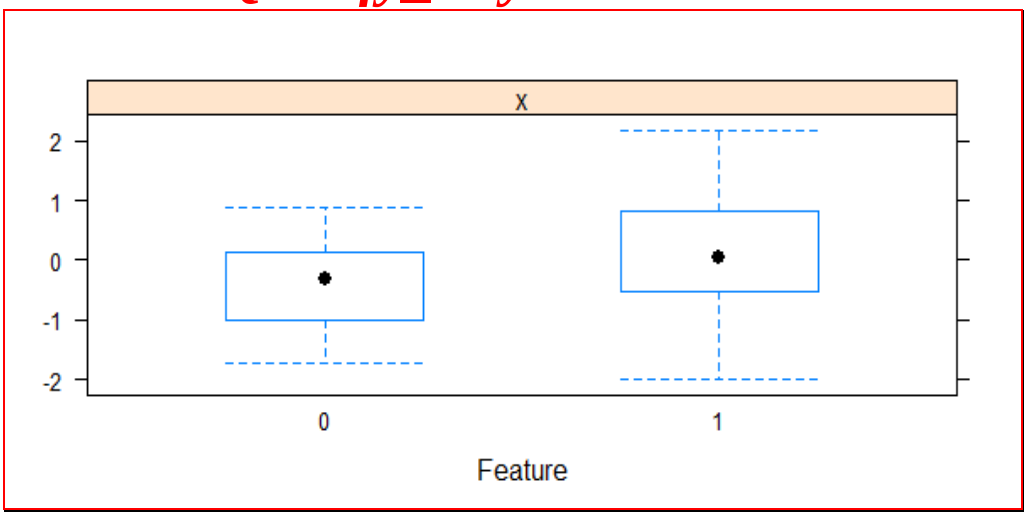
Output 4.1 : Relationship between all the variables

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

From the plot, we can see that Protein intake and Calories are highly positively correlated.

For the Continuous VS Categorical variable, we will look at **Feature plots** to understand the relationships

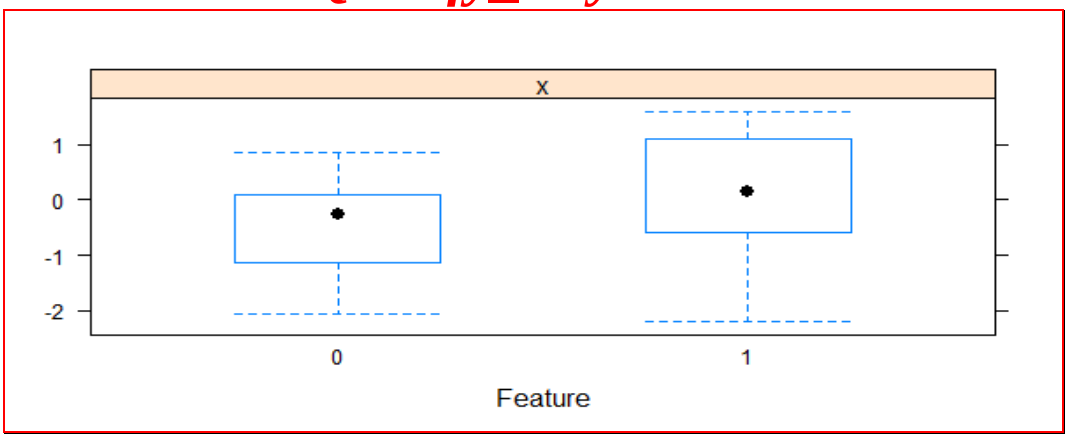
BMI Vs. Qualify_PhysicalTest:



Output 4.2 : Relationship between the variables using feature plot

From the plot, we can understand that there is not much difference between BMI and Qualifying for the physical test.

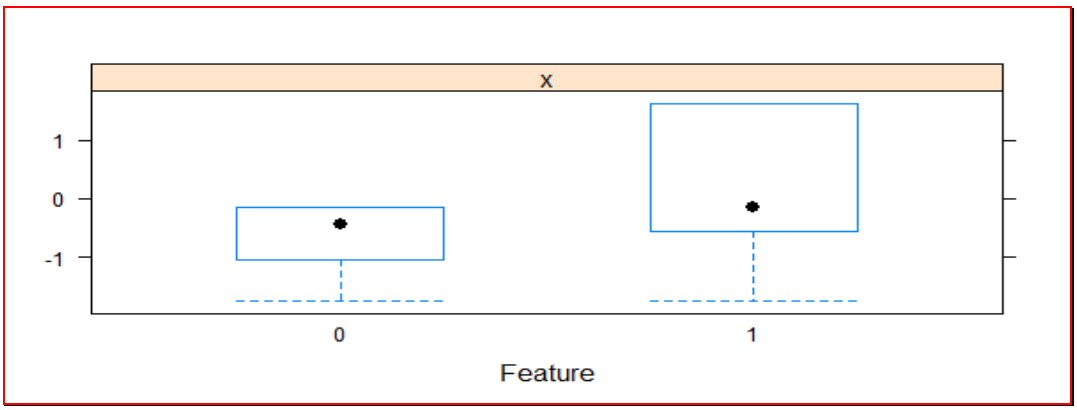
Calories Vs. Qualify_PhysicalTest:



Output 4.3 : Relationship between the variables using feature plot

From the plot, we can understand that there is not much difference between Calories and Qualifying for the physical test.

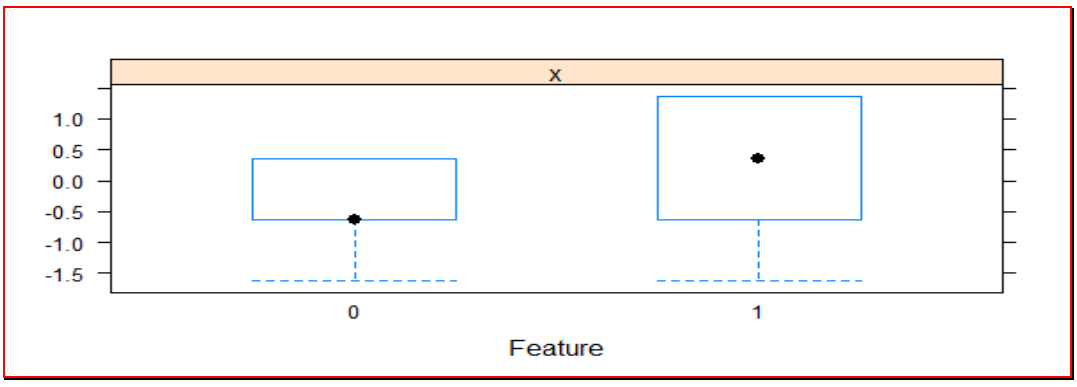
Protein Vs. Qualify_PhysicalTest:



Output 4.4 : Relationship between the variables using feature plot

From the plot, we can understand that there is not much difference between Protein and Qualifying for the physical test.

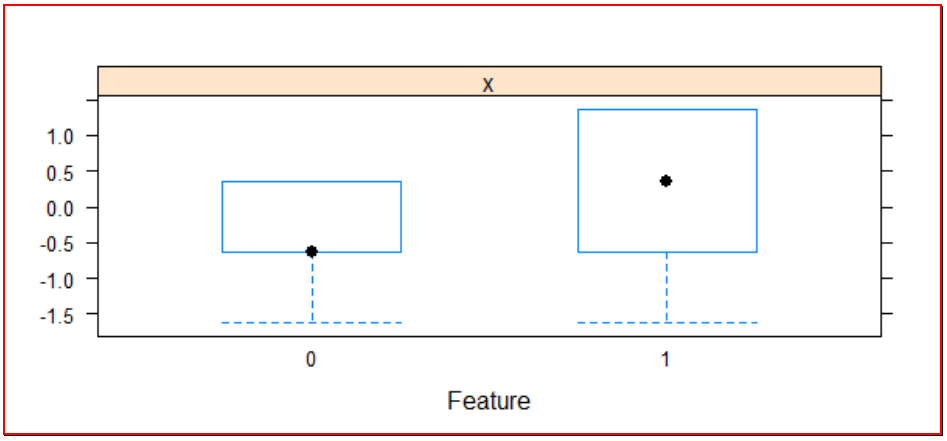
Practice Vs. Qualify_PhysicalTest:



Output 4.5 : Relationship between the variables using feature plot

From the plot, we can understand that there is a difference between Practice and Qualifying for the physical test.

Workout Vs. Qualify_PhysicalTest:



Output 4.6 : Relationship between the variables using feature plot

From the plot, we can understand that there is a difference between workout and Qualifying for the physical test.

5.Checking for the significance difference between variables:

To test the significant difference between Continuous vs categorical variables, we will look at the t-test value.

BMI Vs. Qualify:

```
Df Sum Sq Mean Sq F value Pr(>F)
sports$Qualify_PhysicalTest   1   1.89  1.8878     1.9   0.17
Residuals                   138 137.11  0.9936
```

Output 5.1:Significance of BMI and Qualify variables using t-test

From the above table, as the p value is >0.05, we can conclude that there is no significant relationship between BMI and Qualify_Physical Test.

Calories Vs. Qualify_PhysicalTest:

```
  Df Sum Sq Mean Sq F value  Pr(>F)
sports$Qualify_PhysicalTest   1   3.79   3.790   8.545 0.00405 **
Residuals                   138  61.20   0.443
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 5.2:Significance of Calories and Qualify variables using t-test

From the above table, as the p value is >0.05, we can conclude that there is a significant relationship between Calories and Qualify_Physical Test.

Protein Vs. Qualify_PhysicalTest:

```
 Df Sum Sq Mean Sq F value  Pr(>F)
sports$Qualify_PhysicalTest   1  1.068  1.0680    6.99  0.00914 **
Residuals                   138 21.084 0.1528
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 5.3:Significance of Protein and Qualify variables using t-test

From the above table, as the p value is >0.05, we can conclude that there is a significant relationship between Protein and Qualify_PhysicalTest.

Practice Vs. Qualify_PhysicalTest:

```
 Df Sum Sq Mean Sq F value  Pr(>F)
sports$Qualify_PhysicalTest   1  5.567   5.567   32.66 6.5e-08 ***
Residuals                   138 23.523   0.170
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 5.4:Significance of Practice and Qualify variables using t-test

From the above table, as the p value is >0.05, we can conclude that there is a significant relationship between Practice and Qualify_Physical Test.

Workout Vs. Qualify_PhysicalTest:

```
 Df Sum Sq Mean Sq F value   Pr(>F)
sports$Qualify_PhysicalTest   1   7.73   7.733   14.69 0.000192 ***
Residuals                   138  72.64   0.526
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 5.5:Significance of Workout and Qualify variables using t-test

From the above table, as the p value is >0.05, we can conclude that there is no significant relationship between Workout and Qualify_Physical Test.

6.Standardizing the data:

As the data is measured in different units, we ideally need to do standardize the data before carrying out the analysis. Hence, we standardized the data with mean 0 and Standard deviation 1.

Below is the summary of data after Standardizing:

```
'data.frame':       140 obs. of  5 variables:
 $ BMI     : num  0.68 1.774 0.712 -0.701 -0.607 ...
 $ calories: num  -1.98 -1.97 -1.89 -1.81 -1.72 ...
 $ Protein : num  -2.42 -1.98 -2.16 -1.9 -1.8 ...
 $ practice: num  0.433 0.727 1.607 -1.738 -0.447 ...
 $ workout : num  -1.613 1.352 0.364 -0.625 -0.625 ...
```

Output 6.1:Structure of the data after Standardization

7.Cross validation:

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 and 30 respectively.

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

```
>dim(training)
[1] 98  7
```

Output 7.1: Summary of the train data

Dimensions for Test data is

```
>dim(test)
[1] 42  7
```

Output 7.2: Summary of the test data

Further we use k-fold validation for splitting train data into 10 folds as below

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

8.Running Pipeline using k-fold validation:

Here, we will use a pipeline of algorithms for classification to compare accuracies between 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

98 samples
6 predictor
 2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 88, 88, 89, ...
Resampling results:

  Accuracy   Kappa    0.6705724  0.2308734
```

Output 8.1 : The Logistic Regression Model

When logistic model is fitted for the train data, the accuracy obtained is 67.05%

Decision Tree:

```
CART

98 samples
 6 predictor
 2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 88, 88, 89, ...
Resampling results across tuning parameters:

cp          Accuracy   Kappa
  0.04411765  0.6926936  0.3422691
  0.14705882  0.7217172  0.4482883
  0.17647059  0.6741077  0.2855958

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.1470588.
```

Output 8.2: The Decision Tree Model

By fitting decision tree the accuracy level obtained is 72.17% at the hyper parameter value 0.14705

k-Nearest Neighbors:

```
k-Nearest Neighbors

98 samples
 6 predictor
 2 classes: '0', '1'

Pre-processing: centered (6), scaled (6)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 88, 88, 89, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.7303030  0.4246498
  7  0.7116498  0.3879442
  9  0.6729966  0.2971143

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

Output 8.3: The kNN Model

Best accuracy i.e., 73.03% is obtained when k=5 nearest neighbors is chosen from a given list of neighbors by K-Nearest neighborhood method.

Support Vector Machines:

```
Support Vector Machines with Radial Basis Function Kernel

98 samples
 6 predictor
 2 classes: '0', '1'

Pre-processing: centered (6), scaled (6)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 88, 88, 89, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.6537374  0.0000000
  0.50  0.6310101  0.1170985
  1.00  0.7033670  0.3556740

Tuning parameter 'sigma' was held constant at a value of 0.1571569
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.1571569 and C = 1.
```

Output 8.4: The SVM Model

70.33% accuracy level is obtained when the sigma value selected is 0.1571and c=1 from a list of parameters using support vector machines

Random Forest:

Random Forest

98 samples
 6 predictor
 2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 88, 88, 89, ...
Resampling results across tuning parameters:

mtry	Accuracy	Kappa
2	0.7646128	0.4826565
4	0.7665993	0.4907091
6	0.7628956	0.4872268

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 4.

Output 8.5 :The Random Forest Model

If the number of variables selected for split is 4 then we get the best decision tree from large number of decision trees which are generally used in random forest with an accuracy of 76.28%.

Comparing algorithms:

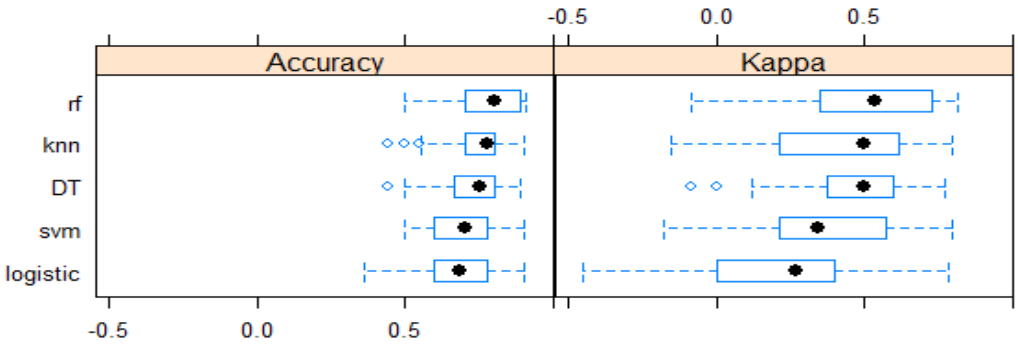
```
Call:
summary.resamples(object = results)

Models: logistic, svm, knn, DT, rf
Number of resamples: 30

Accuracy
Min.1st Qu.Median      Mean   3rd Qu.      Max. NA's
logistic   0.3636364   0.6000000   0.6833333   0.6705724   0.7777778
0.9000000   0
svm             0.5000000   0.6090909   0.7000000   0.7033670   0.7777778
0.9000000   0
knn             0.4444444   0.7000000   0.7777778   0.7303030   0.8000000
0.9000000   0
DT              0.4444444   0.6666667   0.7525253   0.7217172   0.8000000
0.8888889   0
rf       0.5000000 0.7000000 0.8000000 0.7665993 0.8712121 0.9090909
0

Kappa
Min.1st Qu.Median      Mean   3rd Qu.      Max. NA's
logistic  -0.45283019  0.01190476  0.2665222  0.2308734  0.4000000
0.7826087   0
svm            -0.17647059  0.21052632  0.3425254  0.3556740  0.5595238
0.8000000   0
knn            -0.15384615  0.23481781  0.5000000  0.4246498  0.6133242
0.8000000   0
DT             -0.08695652  0.38301887  0.4961538  0.4482883  0.6000000
0.7692308   0
rf             -0.08695652  0.34782609  0.5346320  0.4907091  0.6993007
0.8135593   0
```

Output 8.6 : The comparison of all the Models

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

9.1 Boxplot Comparison:

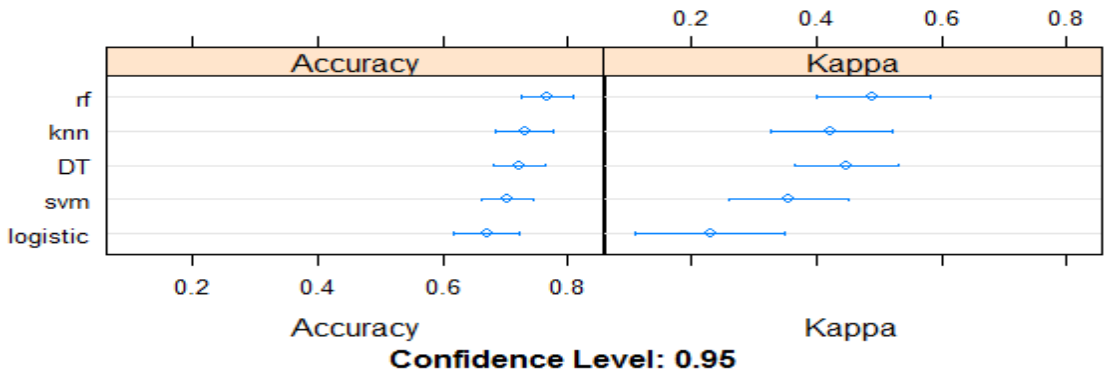
The below plot visualizes the accuracies of above 5 models using Boxplot Comparison.



Output 9.1: Box plot comparison

9.2 Dot-Plot Comparison:

The below plot visualizes the accuracies of above 5 models using Dot-Plot Comparison.



Output 9.2: Dot-Plot comparison

10.1 Finding the key variables using Random Forest

In order to find the key variables , we run the Random Forest using grid search and obtain key hyperparameters.Using these key parameters,we have run Random Forest to get the variable importance.

10.2 Tuning the parameters in random forest

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

- The number of trees as 100 or 200 or300
- The number of variables in each tree would range from 1 to 5

98 samples
6 predictor
2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (20 fold, repeated 3 times)
Summary of sample sizes: 93, 93, 93, 94, 94, 93, ...
Resampling results across tuning parameters:

mtry	ntree	Accuracy	Kappa
1	100	0.7619444	0.4593348
1	200	0.7808333	0.4962279
1	500	0.7786111	0.4942799
2	100	0.7786111	0.5260348
2	200	0.7655556	0.4802520
2	500	0.7658333	0.4851382
3	100	0.7816667	0.5225314
3	200	0.7863889	0.5349051
3	500	0.7763889	0.5066903
4	100	0.7694444	0.4953086
4	200	0.7666667	0.4886816
4	500	0.7844444	0.5396487
5	100	0.7730556	0.5159796
5	200	0.7680556	0.4970693
5	500	0.7741667	0.5140895

Output 10.2: Obtained result after tuning the parameters

Accuracy was used to select the optimal model with the largest value. The final values used for the model were mtry = 3 and ntree = 200.

From the above , we got the best accuracy when the no. of trees are 200 with each tree containing three variables.

10.3 Finding important variables using Random Forest:

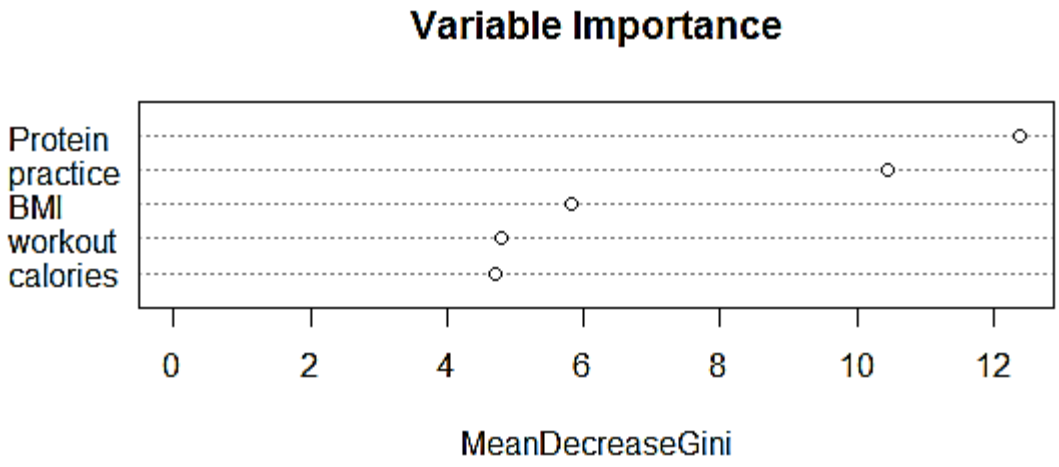
	MeanDecreaseGini
S.NO	3.539408
BMI	5.807997
calories	4.697394
Protein	12.392884
practice	10.452002
workout	4.797499

Output 10.3: Variable Importance using Random Forest

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 protein> practice> BMI >workout>Calories.

11. Visualization of key variables :

A plot visualizing the top 10 important variables among the above variables is as follows:



Output 11.1 : Variable Importance through Visualization

Above plot describes the importance of variables actually affecting the Physical Test .

Further we used 3 variables i.e., Protein , Practice ,BMI in the model.

12. Final model:

We considered the top 3 variables 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
-1.6673  -1.0210   0.3658   0.8812   1.7098

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0707     0.2819   3.798 0.000146 ***
Protein       0.1663     0.2689   0.619 0.536157
practice      1.0973     0.3357   3.268 0.001082 **
BMI           0.1043     0.2337   0.446 0.655539
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 122.32  on 97  degrees of freedom
Residual deviance: 102.29  on 94  degrees of freedom
AIC: 110.29


Number of Fisher Scoring iterations: 5
```

```
Generalized Linear Model

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

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 88, 88, 88, 89, 88, 89, ...
Resampling results:

  Accuracy  Kappa
  0.680101  0.2041492
```

Output 12.1: The Final 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
Actual
Predicted  0  1
        0 13  7
        1 18 60

               Accuracy : 0.7449
                 95% CI : (0.6469, 0.8276)
    No Information Rate : 0.6837
    P-Value [Acc > NIR] : 0.1148

                  Kappa : 0.3481
 Mcnemar's Test P-Value : 0.0455

            Sensitivity : 0.4194
            Specificity : 0.8955
         Pos Pred Value : 0.6500
         Neg Pred Value : 0.7692
```

```
          Prevalence : 0.3163
      Detection Rate : 0.1327
Detection Prevalence : 0.2041
   Balanced Accuracy : 0.6574

    'Positive' Class : 0
```

Output 12.2 :The Confision Matrix of the train data

From the above confusion matrix, we obtained the accuracy for the train data as 74.49%. We conclude that 73 observations are correctly classified out of 98 observations when the logistic model is applied for the train data.

Validating the model using test data:

We fitted Logistic regression on the Test data which is of 42 observations to validate the final model.

The confusion matrix of Test data is:

```
Confusion Matrix and Statistics

          Actual
Predicted  0  1
        0  4  2
        1 10 26

               Accuracy : 0.7143
                 95% CI : (0.5542, 0.8428)
    No Information Rate : 0.6667
    P-Value [Acc > NIR] : 0.31700

                  Kappa : 0.25
 Mcnemar's Test P-Value : 0.04331

            Sensitivity : 0.28571
            Specificity : 0.92857
         Pos Pred Value : 0.66667
         Neg Pred Value : 0.72222
             Prevalence : 0.33333
         Detection Rate : 0.09524
   Detection Prevalence : 0.14286
      Balanced Accuracy : 0.60714

       'Positive' Class : 0
```

Output 12.3 : The Confusion Matrix of the test data

Out of 42 observations, 30 are correctly classified for Test data by using the same Logistic regression model i.e. we attained the accuracy of 71.43% 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 3 important variables Practice ,Protein, BMI.

CHAPTER 4

SUMMARY

SUMMARY:

For the given data, we have applied the pipeline of technique and finalized the best model . We also identified the key variables using Random Forest. The identified key variables are protein, practice and BMI which we use to predict the Qualify_PhysicalTest .

We have then applied the Logistic Regression model for these key variables using train data set and validated the model using the test data set and the accuracies for the train and test data are as follow:

ACCURACY FOR TRAIN DATA IS:- 74.49% ~75%

ACCURACY FOR TEST DATA IS:-71.43% ~71%

Since, the accuracy of train and test data are more or less similar. Hence, the model is a Generalised model, so we can use this model to predict the future data.

CHAPTER-5

APPENDIX

R code

Data

R code

```
getwd()
#setting working directory
setwd("D:/r project on si events/project")
getwd()
#reading the data
sports<-read.csv('primary data.csv',stringsAsFactors = T,
na.strings = c("","  ","NA","?",NA))
#Viewing the data
View(sports)
#viewing the head, tail of the data
head(sports,10)
tail(sports,10)
#checking the dimension of data
dim(sports)
#checking for the variables names in the data
names(sports)
#To know the structure of the data
str(sports)
```

#changing the data types into factors

sports$Qualify_PhysicalTest<-

as.factor(sports$Qualify_PhysicalTest)

str(sports)

#To get descriptive statistics

summary(sports)

#Checking for missing values

sum(is.na(sports))

colSums(is.na(data))

#Checking the percentage of missing values in each column (variable)

sapply(sports, function(df) {(sum(is.na(df)==TRUE)/ length(df))*100; })

#Understanding data using Descrptive

pairs(sports)

#Checking for outliers

boxplot(sports$BMI)

boxplot(sports$calories)

boxplot(sports$Protein)

boxplot(sports$practice)

boxplot(sports$workout)

Removing outliers

```
sports$calories[sports$calories>quantile(sports$calories, 0.95)] <- quantile(sports$calories, 0.95)
sports$Protein[sports$Protein>quantile(sports$Protein, 0.80)] <- quantile(sports$Protein, 0.80)
sports$practice[sports$practice>quantile(sports$practice, 0.90)] <- quantile(sports$practice, 0.90)
sports$workout[sports$workout>quantile(sports$workout, 0.80)] <- quantile(sports$workout, 0.80)
sports$workout[sports$workout<quantile(sports$workout, 0.05)] <- quantile(sports$workout, 0.05)
```

```
boxplot(sports$calories)
boxplot(sports$Protein)
boxplot(sports$practice)
boxplot(sports$workout)
```

Continuous vs Continuous

#Correlation

```
pre1=sports[c(2:6)]
#installing packages
install.packages("corrplot")
```

```
library(corrplot)
pre1.cor   =      cor(pre1)
corrplot(pre1.cor,    method="circle")
```

 # Continuous vs categories

#Feature plot

```
library(caret)
x <- pre1[,2]
str(sports)
y <- sports[,7]
featurePlot(x=x, y=y, plot="box")
library(caret)
x <- pre1[,3]
str(sports)
y <- sports[,7]
featurePlot(x=x, y=y, plot="box")
library(caret)
x <- pre1[,4]
str(sports)
y <- sports[,7]
featurePlot(x=x, y=y, plot="box")
library(caret)
x <- pre1[,5]
```

```
str(sports)
y <- sports[,7]
featurePlot(x=x, y=y, plot="box")
library(caret)
x <- pre1[,6]
str(sports)
y <- sports[,7]
featurePlot(x=x, y=y, plot="box")
```

#Statistical tests/ Filter Method

#ANOVA

```
x1=aov(sports$BMI ~ sports$Qualify_PhysicalTest)
summary(x1)
x2=aov(sports$calories ~ sports$Qualify_PhysicalTest)
summary(x2)
x3=aov(sports$Protine ~ sports$Qualify_PhysicalTest)
summary(x3)
x4=aov(sports$practice~ sports$Qualify_PhysicalTest)
summary(x4)
x5=aov(sports$workout~ sports$Qualify_PhysicalTest)
summary(x5)
```

##**choosing continuous variables**

```
str(sports)
cont<-subset(sports,select = -c(Qualify_PhysicalTest))
str(cont)
#standardizing continuous variables
scale_training <- as.data.frame(scale(cont[,],
                                      center = TRUE, scale = TRUE))
str(scale_training)
scale_training<-cbind(scale_training,
sports$Qualify_PhysicalTest)
str(scale_training)
names(scale_training)
write.csv(scale_training,'primary data.csv')
scaledata=read.csv('primary data.csv')
str(scaledata)
scaledata$Qualify_PhysicalTest=as.factor(scaledata$Qua
lify_PhysicalTest)
str(scaledata)
```

Preparing data for Training and testing

```
train_rows<-sample(1:nrow(scaledata),
size=0.7*nrow(scaledata))
train_rows
```

```
training <- scaledata[train_rows, ]
test <- scaledata[-train_rows, ]
dim(scaledata)
dim(training)
dim(test)
names(training)
names(test)
```

Evaluate Algorithms: Baseline

Run algorithms using 10-fold cross validation

```
control<-trainControl(method="repeatedcv",
number=10,repeats = 3)
seed <- 7
metric <- "Accuracy"
print(trainControl)
```

General linear model

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

```
# CART
set.seed(seed)
fit.cart <- train(Qualify_PhysicalTest~.,data=training, method="rpart", metric=metric, trControl=control)
print(fit.cart)
# kNN
set.seed(seed)
fit.knn <- train(Qualify_PhysicalTest~.,data=training, method="knn", metric=metric, preProc=c("center", "scale"), trControl=control)
print(fit.knn)
# SVM
set.seed(seed)
fit.svm <- train(Qualify_PhysicalTest~.,data=training, metric=metric, preProc=c("center", "scale"), trControl=control, fit=FALSE)
print(fit.svm)

# Random Forest
set.seed(seed)
```

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

print(fit.rf)

Compare algorithms

results <- resamples(list(logistic=fit.glm,svm=fit.svm, knn=fit.knn, DT=fit.cart,rf=fit.rf))

Table comparison

summary(results)

boxplot comparison

bwplot(results)

Dot-plot comparison

dotplot(results)

Checking 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
```

#Random Forest for checking variable importance

```
library(caret)
library(randomForest)
control <- trainControl(method="repeatedcv", number = 10 , repeats=3)
 tunegrid <- expand.grid(.mtry=c(1:5),ntree=c(100,200,500))
set.seed(7)
```

```
custom <- train(Qualify_PhysicalTest~.,data = training ,
method=customRF, tuneGrid=tunegrid,
trControl=control)
print(custom)
```

#variable importance

```
library('randomForest')
control <- trainControl(method="repeatedcv",
number=5)
rf_model <- randomForest(Qualify_PhysicalTest~.,data
= training,ntree=500,mtry=4,trControl=control)
varImpPlot(rf_model,sort = T,  n.var = 5,main =
"Variable Importance")
importance(rf_model)
```

Fitting logistic regression with selected variables

```
set.seed(100)
trainControl <- trainControl(method="repeatedcv",
number=10, repeats=3)
final.glm <-train (Qualify_PhysicalTest ~  Protein
+practice+BMI , data=training, method="glm",
trControl=trainControl)
summary(final.glm)
```

```
print (final.glm)
#confusion matrix for train data
predslog <- predict(final.glm, data=training, type =
"raw")
tabtrain <- table(Predicted = predslog, Actual =
training$Qualify_PhysicalTest)
confusionMatrix(tabtrain)
#confusion matrix for test data
predstest <- predict(final.glm, newdata=test, type =
"raw")
tabtest <- table(Predicted = predstest, Actual =
test$Qualify_PhysicalTest)
confusionMatrix(tabtest)
```

PRIMARY DATA

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
1	M. Ramesh	23.92	13	3.8	6.52	2.00	0	0	0	0	0	0	390.00	8.10	40.00	2.00
2	Thanneeru chandra shekar	28.47	15	3.6	7	1.00	2	0	1	1	0	0	392.14	15.41	45.00	5.00
3	G. ramesh	24.03	14	3.8	7	2.00	0	0	0	1	0	0	427.14	12.46	120.00	4.00
4	P. Narsimlu	19.20	13	5	7	2.00	0	0	0	2	0	0	464.29	16.81	3.00	3.00
5	Mallesh	19.52	15	5	5.8	2.00	0	0	1	1	0	0	505.14	18.46	25.00	3.00
6	K manoj	22.71	14	4.2	6.5	2.00	0	0	1	1	0	0	505.14	18.46	60.00	6.00
7	B. Srikanth	18.24	14	3.8	6.5	2.00	0	0	0	0	0	2	600.00	10.70	20.00	4.00
8	Madhu kar	21.20	14	3.8	7.8	2.00	0	0	0	1	2.8	0	603.14	24.46	30.00	3.00
9	Arun kumar	21.50	14	5	7	3.00	0	0	0	1	0	0	622.14	16.51	14.00	2.00
10	Jagadish	20.59	13	5	7.2	3.00	0	0	0	1	0	0	622.14	16.51	7.00	2.00
11	B. Sreenu	17.89	12.5	4	6	2.00	0	0	0	1	2	1	657.86	22.33	15.00	1.50
12	D. Ramesh	21.87	14	4	6.8	2.00	0	0	0	2	0	2	674.29	19.41	20.00	2.00
13	Balaraju	16.96	13	4.5	6	2.00	0	0	1	1	0	2	715.14	21.06	14.00	2.00
14	J. Omkar	18.46	13	4.3	5.8	3.00	0	0	1	2	0	0	737.29	26.86	20.00	2.00
15	Anil	22.40	14	4.5	4.7	2.00	0	0	2	0	0	2	756.00	22.70	10.00	4.00
16	Sampath	31.45	13	4.7	9	3.00	0	0	1	3	0	0	774.43	31.22	5.00	2.00
17	M. Ashok	21.53	13	4	7.21	3.00	0	0	2	1	0	0	778.14	28.51	90.00	3.00
18	C. Sai	20.80	18	3.8	5.6	2.00	0	0	2	1	0	2	793.14	27.06	15.00	3.00
19	Martha laxman	24.28	14	3.8	6.5	2.00	0	0	2	1	0	2	793.14	27.06	20.00	4.00
20	S. Gopi	18.22	14	5	7	2.00	0	0	2	1	0	2	793.14	27.06	25.00	4.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
21	Ramesh	21.44	12.5	5	6.5	1.00	2	0	2	2	0	2	795.29	34.36	25.00	3.00
22	P. Ramesh	23.34	16	3	5	3.00	0.00	0	1	1	0	1	805.14	23.81	20.00	2.00
23	Asif	26.38	12.5	4.8	7.5	2.00	0.00	0	0	0	0	4	810.00	13.30	10.00	2.00
24	Macha sai kumar	19.20	13	4.6	7.7	3.00	0.00	0	2	2	0	0	815.29	32.86	30.00	3.00
25	B. Nagesh	22.40		4	5.6	2.00	0.00	1	2	1	0	0	820.14	36.46	3.00	2.00
26	G. Goutham raj	25.41	15	4	6.5	2.00	2	0	0	0	2.8	1	831.00	28.40	25.00	4.00
27	Bhaskar	19.48	18	5	5.6	2.00	2	0	1	0	0	2	838.00	23.70	20.00	3.00
28	S. Ajay	27.62	15	4.4	7	2.00	2	0	1	1	2.8	0	841.14	37.46	15.00	3.00
29	Kishore	24.12	15	4	8	1.00	2	1	0	2	2.8	0	842.29	43.76	60.00	4.00
30	Pagolu vidyasagar	23.36	14	5	8	2.00	0	0	2	3	0	2	867.43	35.77	15.00	1.00
31	Kamalakar	22.65	13	4.5	7.5	1.00	3	0	2	2	0	2	875.29	37.86	30.00	4.00
32	E maheshwar	14.83	13	4.5	6	2.00	2	0	2	0	0	2	916.00	29.70	50.00	5.00
33	Sheelam bhoomeshwer	23.00	13	4.3	6.5	1.00	2	1	0	3	0.15	2	922.54	39.34	25.00	3.00
34	D. Geevan	22.32	14.7	3.8	4	3.00	0	0	2	2	2.5	0	972.43	43.58	7.00	2.00
35	G srinivas	20.99	15	4	7.5	2.00	2	1	1	0.5	0	1	988.57	36.58	60.00	3.00
36	Srinath	20.80	14	160	7	2.00	2	1	1	1	0	1	1007.14	38.76	20.00	4.00
37	L. Raju	26.17	15	3.8	5.6	3.00	0	0	2	1	2	1	1008.86	38.38	7.00	2.00
38	Maesh	23.19	13	4.8	8.5	1.00	6	0	1	1	3.5	0	1010.14	50.41	90.00	4.00
39	Dhakshakumar	19.48	12.28	5.56	6.82	2.00	0	2	0	4	0	0	1012.57	49.53	90.00	3.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
40	P. Ramesh	36.28	14	5	8.1	1.00	5	1	2	1	0	0	1025.14	49.91	40.00	3.00
41	C. Mahesh	22.40	13	4	8	3.00	2	0	1	1	2.8	0	1036.14	41.51	30.00	5.00
42	Ravi	18.55	13	4.5	7	3.00	0	0	1	2	0	3	1052.29	30.76	30.00	3.00
43	Srikanth	20.80	13.5	4	6	3.00	0	1	2	2	0	0	1052.29	44.86	45.00	6.00
44	D. Rakesh	29.80	13	5	7.4	2.00	1	1	2	1	1	1	1068.00	45.54	40.00	4.00
45	Arjun	17.08	17			2.00	3	0	2	2	0	2	1070.29	41.91	14.00	4.00
46	Sai reja	21.44	15.4	3.6	5.8	2.00	4	1	0	1	1.5	0	1078.43	44.89	21.00	5.00
47	Wajeed	25.41	14	3.8	6	3.00	0	1	2	3	0	0	1089.43	49.22	20.00	5.00
48	J. Karunakar	24.85	17	3	4	2.00	0	2	1	4	0	0	1090.57	55.53	60.00	4.00
49	Hari	20.16	13	4.5	7	3.00	0	0	2	1	0	3	1093.14	32.41	30.00	4.00
50	Prudvi	17.85	17	3.8	5.6	2.00	0	0	1	1	2.8	4	1101.14	35.66	14.00	4.00
51	Chatan	16.08	14	1.3	7	2.00	0	0	4	2	3.5	1	1101.29	57.11	10.00	4.00
52	C. Anjanailu	15.46	14	4	6	3.00	0	0	2	1	3.5	1	1103.14	44.81	30.00	2.00
53	Sandeep	19.88	15	3.8	5.8	2.00	1	1	2	1	0	2	1110.14	42.56	25.00	3.00
54	P. Vykuntam	23.85	16	3.6	7.2	2.00	4	0	2	1	0	2	1113.14	41.06	45.00	2.00
55	Balu	21.87	13	4.5	7	3.00	0	1	2	1	0	1	1120.14	41.81	90.00	4.00
56	Masum	24.91	15	4	6	2.00	1	0	2	2	3.5	2	1130.29	49.91	30.00	4.00
57	Masum	24.91	15	4	6	2.00	1	0	2	2	3.5	2	1130.29	49.91	30.00	4.00
58	D. Satish	16.00	14	5	6	3.00	0	0	1	7	0	2	1133.00	51.25	10.00	3.00
59	K. Anjanailu	23.25	16	3	4	2.00	0	1	2	4	0	2	1141.57	52.13	30.00	3.00
60	K. Shankar	17.32	12.9	5	6.5	2.00	0	1	3	2	0	2	1145.29	49.41	45.00	4.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
61	Dhakay kumar	19.88	12.5	4	7	3.00	0	1	1	1	0	2	1147.14	37.11	30.00	4.00
62	Veresh	26.90	13	4.6	8	3.00	2	0	2	1	0	2	1148.14	38.11	20.00	2.00
63	Kalyan	21.28	12	4.2	7.5	2.00	4	0	2	2	0	2	1150.29	45.41	20.00	4.00
64	Chawan gopal	26.79	13	5.3	7.8	2.00	0	2	2	1	0	1	1162.14	49.76	120.00	4.00
65	Devendar	25.06	13	4.5	8	3.00	0	1	3	4	0	0	1204.57	59.58	15.00	2.00
66	Sagar	23.34	14	4.28	7.2	2.00	1	1	2	2	2.8	1	1218.29	57.61	90.00	4.00
67	Mg. Dattu yadav	16.56	14	3.8	5.6	2.00	0	1	4	2	0	2	1223.29	55.41	12.00	4.00
68	V. Ashok	26.77	13	4.8	8.4	2.00	2	1	1	4	0	2	1223.57	53.13	60.00	4.00
69	Vinod	19.17	14.5	6	6.5	2.00	2	1	2	2	0	2	1227.29	50.41	30.00	3.00
70	A. Vinay	20.10	14	4	7	3.00	0	1	2	2	2.8	0	1228.29	56.86	30.00	3.00
71	Anil	19.84	16	5	7.2	2.00	3	0	2	1	1.5	3	1232.43	45.29	20.00	6.50
72	K. Gopal	18.55	13	4.4	6	3.00	0	0.5	1	4	0	3	1245.07	45.48	15.00	4.00
73	T. Srinivas	25.66	14.5	5	7	2.00	3	0	2	1	3.5	2	1253.14	52.56	30.00	2.00
74	K. Shiva kumar	17.60	14.5	4	7	1.00	3	0	2	1	0	6	1258.14	38.71	30.00	8.00
75	B venkanna	18.22	13	5	7	3.00	0	1	2	2	0	2	1262.29	47.46	30.00	3.00
76	Shiva	21.53	14	5	5.6	3.00	0	1	2	3	2.8	0	1265.43	61.22	30.00	4.00
77	Allepu vinod	20.59	18	3.6	5.8	3.00	0	2	1	1	0	1	1279.14	47.81	30.00	3.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
78	Sanjeev	22.42	15	3.5	6	2.00	0	1	2	2	3.5	2	1287.29	58.41	30.00	5.00
79	Kathoji umesh	20.94	13.5	4.1	7.2	2.00	2	1	4	1	2.8	0	1312.14	67.46	20.00	5.00
80	Lashmman	14.34	12.03	4.5	7	2.00	0	1	2	3	0	4	1314.43	50.37	20.00	5.00
81	Ranjit	21.95	15	4.2	7	1.00	2	2	0	2	0	4	1323.29	48.96	20.00	4.00
82	Kiran kumar	32.96	15	4	8	0.00	9	0	4	2	3.5	0	1326.29	79.21	60.00	5.00
83	Vinod	26.90	13	4.2	7.5	3.00	0	1	2	1	0	3	1330.14	44.41	30.00	3.00
84	Bharath	21.87	13	3.8	7.5	3.00	0	1	2	1	0	3	1330.14	44.41	30.00	4.00
85	Ram	18.88	14	4.3	6	2.00	2	1	2	2	0	3	1332.29	51.71	30.00	3.00
86	Madhu	21.76	15	3.9	5.9	3.00	0	1	2	2	2.8	1	1333.29	58.16	30.00	4.00
87	Karthik	21.87	15	4	6	1.00	3	1	3	2	2.8	2	1366.29	67.86	90.00	6.00
88	B. Venkat	22.02	16	3.7	6	3.00	0	1	1	1	3.5	2	1367.14	52.11	30.00	4.00
89	Nerella pavan	31.04	16	4	6.5	2.00	0	2	2	1	0	3	1372.14	52.36	30.00	3.00
90	Chandra shekar	21.53	14	3.8	6.2	2.00	0	1	2	3	2.8	3	1385.43	61.07	40.00	3.00
91	J Mahesh	18.22	14	3.8	6	2.00	2	1	2	1	3.5	2	1410.14	61.06	30.00	5.00
92	Karthik	18.55	12	5	7	2.00	0	1	6	3	0	2	1416.43	71.77	120.00	3.00
93	B. Chanti	21.53	14	4.5	7	3.00	0	1	2	1	0	4	1435.14	45.71	150.00	4.00
94	Sainath	18.86	14	4	5.7	2.00	2	1	2	2	0	4	1437.29	53.01	60.00	6.00
95	Ashok	24.00	14	1.2	7.5	3.00	4	0	1	1	0	4	1440.14	41.71	20.00	2.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
96	M. Shiva kumar	24.74	20	6.5	6	2.00	5	2	1	3	0	0	1453.43	68.67	35.00	6.00
97	M. Pulya	16.44	12	5	8	2.00	2	1	2	3	0	4	1474.43	57.37	30.00	3.00
98	Shiva	21.53	14	6	5.6	3.00	0	1	2	3	2.8	2	1475.43	63.82	30.00	4.00
99	G. Chakrapani	21.53	14	3.8	6.8	3.00	0	1	1	1	2	4	1482.86	48.28	60.00	3.50
100	Rama murthi	26.79	14	4.5	7	3.00	0	2	2	2	0	2	1499.29	59.46	40.00	4.00
101	Karnakar	24.00	13	5.18	7	2.00	0	2	5	4	0	1	1507.57	80.83	60.00	4.00
102	V srikanth	19.88	13.5	4.2	6.4	2.00	2	1	3	3	2.8	2	1518.43	72.77	30.00	4.00
103	Srikanth	22.08	15	3.9	6.3	3.00	1	1	1	2	3.5	3	1589.29	61.26	30.00	3.00
104	Praveen kumar	23.19	13.84	3.8	6	2.00	0	2	4	0	0	4	1596.00	61.30	20.00	5.00
105	Rama krishna	16.39	15	4	5.6	3.00	0	0	2	2	2.5	6	1602.43	51.38	30.00	3.00
106	C. Shiva	22.53	14	4.2	5.9	2.00	2	2	2	2	2.2	2	1602.57	71.84	30.00	3.00
107	K. Anjani ksrishna	20.10	13	4.2	8	2.00	2	0	4	3	0	6	1603.43	59.97	14.00	4.00
108	R. Ravi Nayak	20.10	13	4.6	7.3	3.00	2	1	2	1	3.5	2	1605.14	65.11	65.00	3.00
109	Abhilash	20.54	20	5	7.2	2.00	4	0	3	4	3.5	3	1627.57	76.43	20.00	6.00
110	Jitendra	26.79	12	5	7	2.00	0	2	4	1	0	4	1633.14	65.66	30.00	4.00
111	Sandeep	16.96	15.3	4.8	5	3.00	0	2	2	3	0	3	1641.43	65.12	25.00	3.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
112	K. Shiva kumar	26.49	12	4.5	8	1.00	0	0	4	7	14	0	1647.00	118.55	75.00	4.00
113	C. Kurumayya	16.00	13	4.8	6	3.00	0	1	2	1	3.5	4	1655.14	60.71	30.00	2.00
114	V. Uday kiran	24.95	13	3.7	7	2.00	2	1	4	0	3.5	4	1739.00	71.30	360.00	3.00
115	Linga	17.92	12	5.2	7	2.00	5	1	4	7	0	2	1809.00	94.70	30.00	4.00
116	Raju	25.66	14	3.5	7	3.00	0	0	5	1	2.8	6	1818.14	66.31	10.00	6.00
117	B Nagaraju	21.53	13	4.6	7.8	2.00	3	2	3	3	0	4	1869.43	78.87	15.00	1.00
118	M. Rajeev	26.90	17	7	6	2.00	0	2	4	4	2.8	4	1920.57	90.73	10.00	6.00
119	Sai kiran	20.10	13	5	7.2	2.00	2	2	3	1	0	6	1925.14	69.26	15.00	7.00
120	Ksrishna	23.99	17	3.7	6	3.00	1	1	2	2	2.8	6	1938.29	68.16	50.00	2.00
121	Ramdas	13.67	16.5	3.8	7	3.00	0	3	2	3	2.8	2	1949.43	87.82	30.00	4.00
122	Sharat	24.00	13	4.2	8	2.00	6	2	2	1	3.5	2	1967.14	87.06	60.00	4.00
123	M. Satish	16.08	13	4.8	6.5	3.00	0	1.5	2	2	2.8	6	1976.79	70.66	15.00	3.00
124	Sk. Saidhul	23.04	14	4.3	7.2	3.00	2	2	2	1	3.5	4	2052.14	79.71	60.00	3.00
125	P. Mahender	14.91	13	5	6.3	2.00	4	1	6	0	3.5	4	2055.00	90.30	30.00	4.00
126	A shekar	24.71	13	3.8	6.5	2.00	4	2	2	2	7	2	2064.29	99.41	60.00	3.00
127	P. Ramesh	20.54	13	5	8	2.00	0	1	4	3	2	10	2226.14	78.74	40.00	6.00
128	Vinod	24.03	14.3	4	8	1.00	2	4	3	7	7	0	2237.00	137.55	45.00	6.00
129	Suresh	21.34	13	5.2	9	2.00	4	4	4	2	3.5	2	2474.29	120.41	60.00	6.00
130	Satish kumar	19.88	14.5	3.8	6	1.00	4	4	4	3	0	6	2516.43	110.92	45.00	6.00

S NO	Name	BMI	Running	Long Jump	Shot Put	RICE	Chapatti	Ragi	Egg	Non Veg	Milk	Fruits	calories	Protein	practice	workout
131	Naveen	15.12	12	3	8	3.00	6	1	6	3	1	7	2679.29	107.61	30.00	4.00
132	Sai nath	25.97	13	4	6	3.00	4	2	10	3	3.5	2	2700.43	140.82	60.00	6.00
133	Prashanth	19.17	13	4	7	2.00	4	4	4	3	0	6	2711.43	114.97	40.00	6.00
134	Naveen kumar	17.92	12.5	5	7	3.00	0	4	63	3	0	0	6558.43	451.22	45.00	6.00
135	Lakshman	23.13	15	3.8	6	2.00	2	2	2	2	1	2	1527.14	66.70	30.00	4.00
136	Akash	18.86	11.9	4.54	7.2	2.00	4	2	3	2	1	2	1765.14	79.70	60.00	6.00
137	Harish	23.85	12.7	4.2	6.5	2.00	4	0	1	1	1	3	1203.00	40.64	60.00	6.00
138	Nagesb	32.15	15.5	4	7	2.00	2	1	2	1	1	3	1358.00	51.64	11.00	4.00
139	Naresh	19.55	14	4	6	2.00	2	2	1	2	1	3	1554.14	62.00	40.00	5.00
140	Srikanth	21.87	14	4	7	1.00	4	1	2	1	1	4	1428.00	55.89	30.00	3.00
141	Prasad	22.26	14	5	8	2.00	2	1	4	2	1	0	1236.14	64.10	45.00	5.00

BIBLIOGRAPHY

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

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

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

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

- Machine_learning_mastery_with_r. – Jason Brownlee

- Master_machine_learning_algorithms -- Jason Brownlee

- Statistical_methods_for_machine_learning - Jason Brownlee Machine Learning Using Karthik Ramasubramanian ,Abhishek Singh

- Data Science for Business - Forster Provost & Tom Fawcett

- Deep learning with Deep learning R by François Chollet