A PROJECT REPORT
ON

STATISTICAL MODELING FOR WATER REGRESION

Submitted to

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

MASTER OF SCIENCE
IN
STATISTICS

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

BY

T.VINAY	Roll No:100717507023
K.NARENDAR	Roll No:100717507024
S.PREMALATHA	Roll No:100717507027
J.RAJASHEKAR	Roll No:100717507028
G.RAMA DEVI	Roll No:100717507029
M.SUJATHA	Roll No:100717507030
H.PRIYANKA	Roll No:100717507032

Under the Supervision of
Submitted to
Dr.M. VENUGOPALA RAO
2018

A PROJECT REPORT
ON

Statistical modelling for water reggression

Osmania University
in partial fulfilment of the
requirements for the award of
Master of Science in Statistics

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

BY

T.VINAY	Roll No:100717507023
K.NARENDAR	Roll No:100717507024
S.PREMALATHA	Roll No:100717 507027
J.RAJASHEKAR	Roll No:100717507028
G.RAMA	Roll No:100717507029
M.SUJATHA	Roll No:100717507030
H.PRIYANKA	Roll No:100717507032

Under the Supervision of

Dr. M. VENUGOPALA RAO
2018

CERTIFICATE

This is to certify that

T.VINAY	Roll No:100717507023
K.NARENDAR	Roll No:100717507024
S.PREMALATHA	Roll No:100717 507027
J.RAJASHEKAR	Roll No:100717507028
G.RAMADEVI	Roll No:100717507029
M.SUJATHA	Roll No:100717507030
H.PRIYANKA	Roll No:100717507032

havee submitted the project titled **"Statistical modelling water regression"** in partial fulfilment for the degree of Master of Science in Statistics.

Head

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

T.VINAY
K.NARENDAR
S.PREMALATHA
J.RAJASHEKAR
G.RAMADEVI
M.SUJATHA
H.PRIYANKA

Department of Statistics

Osmania University

Hyderabad – 500 007, T.S.

INDIA

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, 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.JAYALAXMI,** Head of Department, Depatment of statistics, Osmania University for providing the necessary support and facilities in the department for completion of this work successfully.

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

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

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

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

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

CONTENTS

CHAPTER 1

INTRODUCTION

Scope Of The Problem:

Can Southern California's water supply in future years be predicted from past data? One factor affecting water availability is stream runoff. If runoff could be predicted, engineers, planners and policy makers could do their jobs more efficiently. Multiple linear regression models have been used in this regard. This dataset contains 43 years worth of precipitation measurements taken at six sites in the Owens Valley.

1.1 Variable Description:

This data frame contains the following columns:

Year

collection year

APMAM

Snowfall in inches measurement site

APSAB

Snowfall in inches measurement site

APSLAKE

Snowfall in inches measurement site

OPBPC

Snowfall in inches measurement site

OPRC

Snowfall in inches measurement site

OPSLAKE

Snowfall in inches measurement site

BSAAM

Stream runoff near Bishop, CA, in acre-feet

- **Source**

Source: http://www.stat.ucla.edu.

1.2 REVIEW OF THE CHAPTER

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

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

Section 4 describes the summary and conclusions followed by Bibliography

Chapter 2

Review of Machine Learning Process

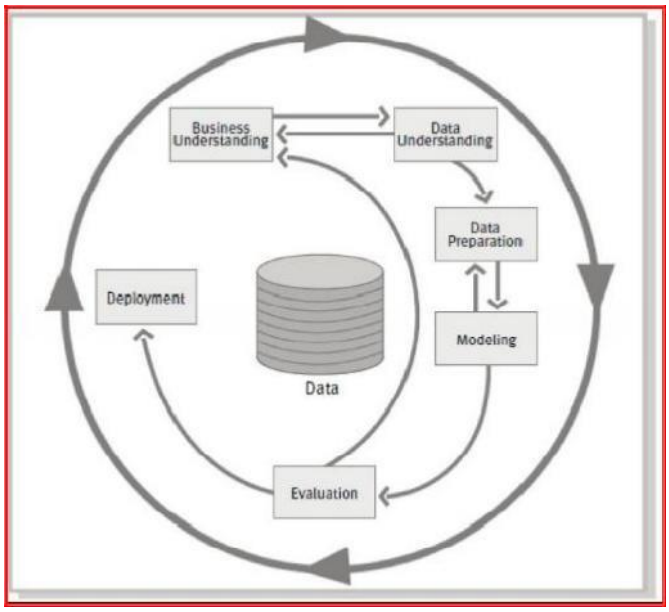
2.0 Need of Machine Learning

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 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

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 steps:

Select a modelling technique

1) Generate a test design
2) Build a model
3) Assess a Model

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

2.1.5)Evaluation

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

1) Evaluate the results
2) Review the process
3) Determine the next steps

2.1.6)Deployment

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

1) Deploying the plan
2) Monitoring and maintenance of the plan
3) Producing the final report

2.2 Types of Machine Learning

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

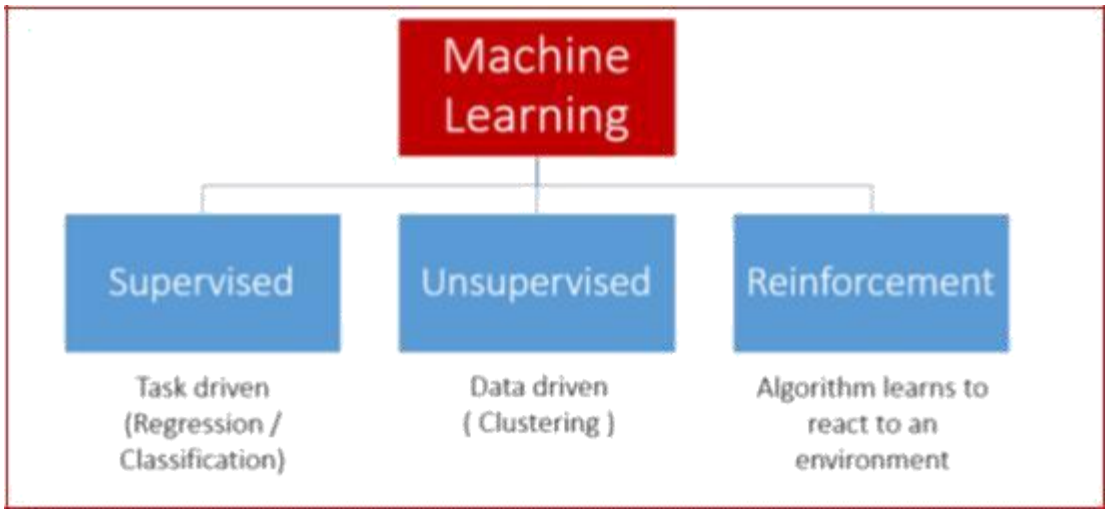


Fig 2.2 Types of Machine Learning

2.2.1) Supervised Learning

This algorithm consists of a target / outcome / dependent variable which is to be predicted from a given set of predictors / independent variables. Using these set of variables,

we generate a function that maps inputs to desired output. The training process continues until the model achieves a desired level of accuracy on the training data.

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

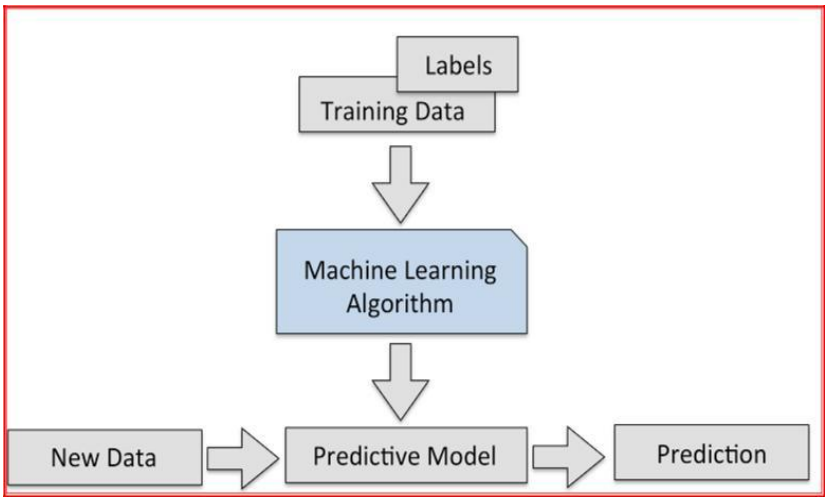


Fig 2.2.1 Supervised Learning

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

Classification **Regression**

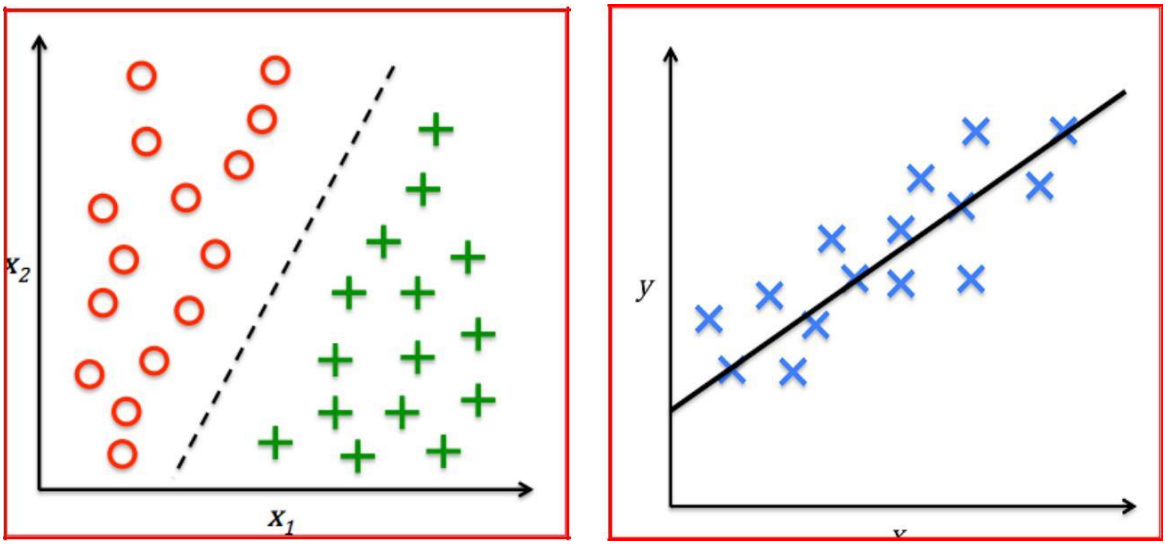


Fig 2.2.1 classification and regresion

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)

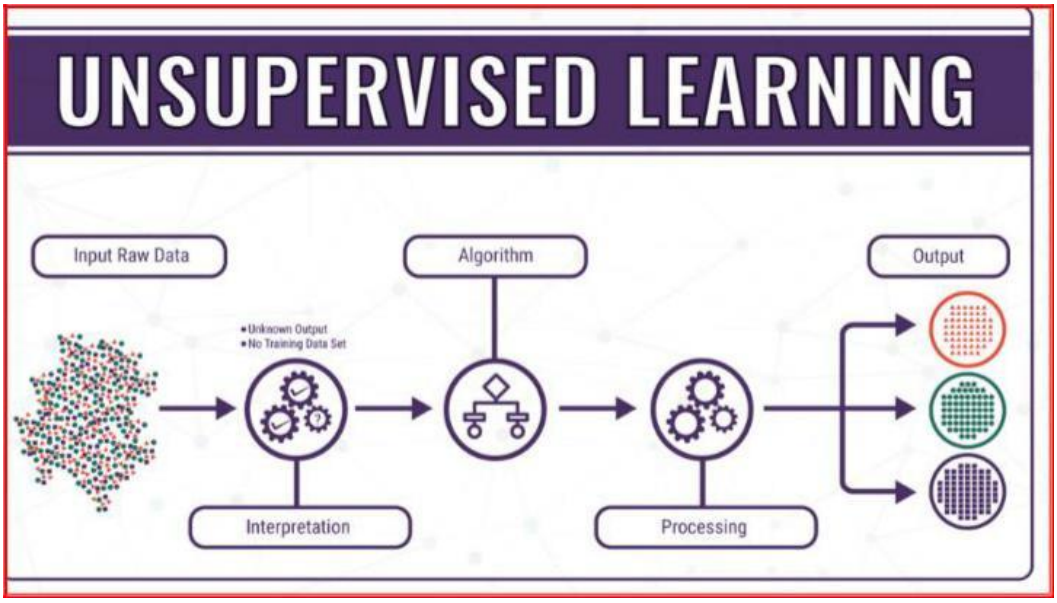


Fig 2.2.2 Unsupervised Learning

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

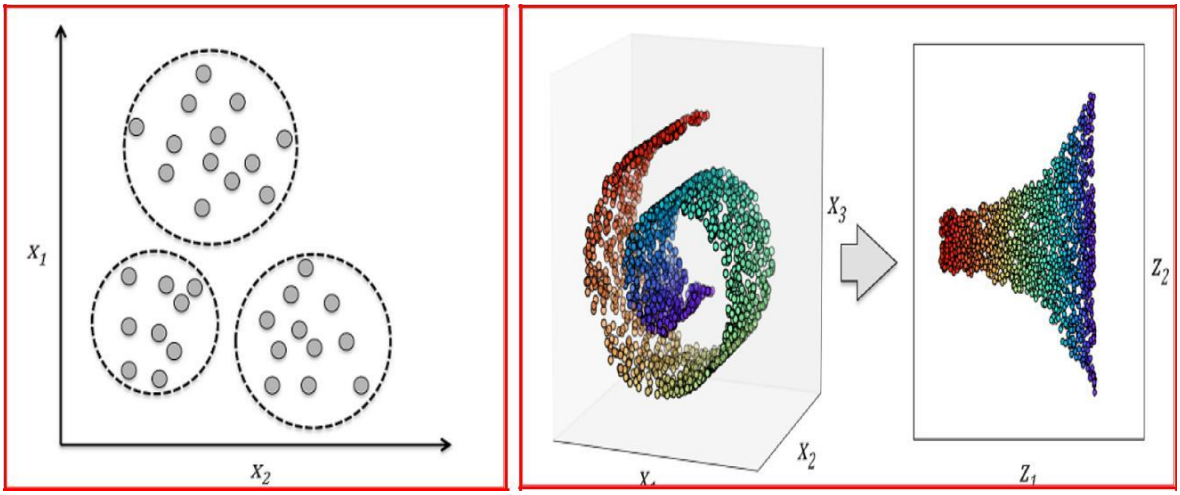


Fig 2.2.2 Cluster Analysis & Data Reduction Techniques

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:

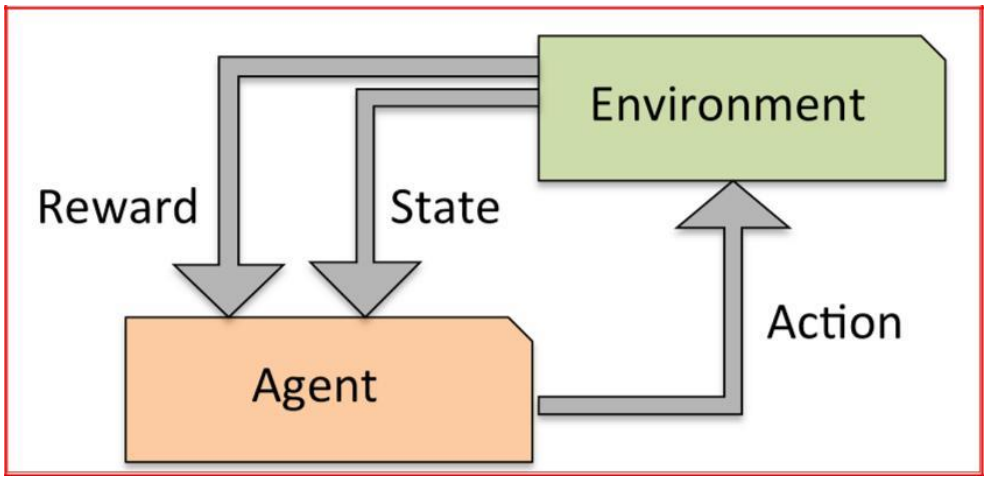


Fig 2.2.3 Reinforcement Learning

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:

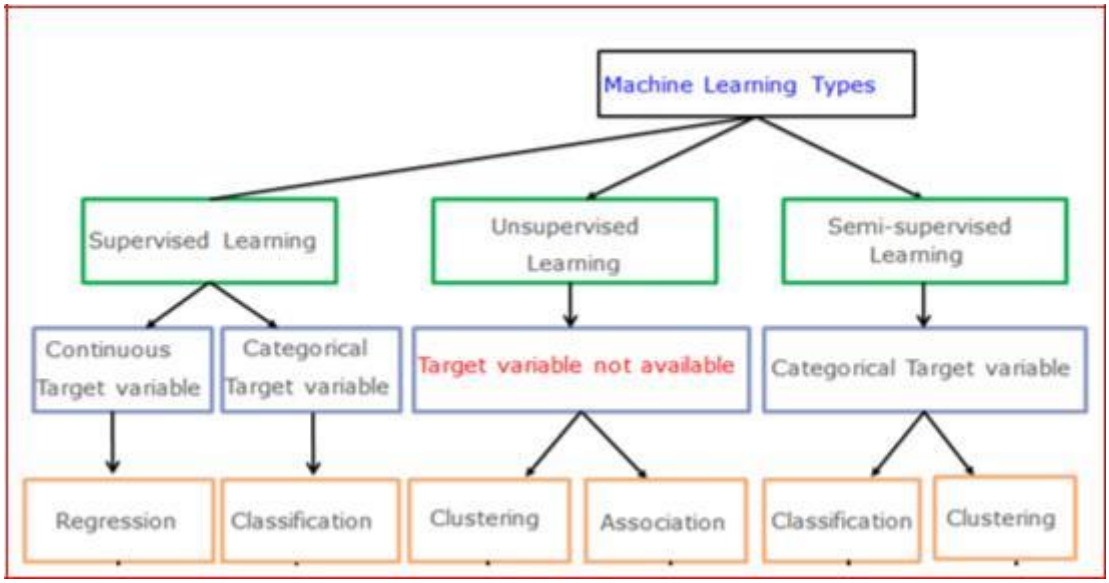


Fig 2.3: choosing the right algorithm

2.3.1) Types of Regression Algorithms

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

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

b) **Multiple Linear Regression:-** Multiple linear regression is one of the algorithms of regression technique, and is the most common form of linear regression analysis. As a predictive analysis, the multiple linear 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.

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

d) **Support Vector Machines:-** Support Vector Machines can be applied to regression problems as well as Classification. It contains all the features that characterises maximum margin algorithm. Linear learning machine maps a non-linear function into high dimensional 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 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 hyperplane. The algorithm outputs an optimal hyperplane which categorises new examples. In two dimensional space, this hyperplane 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 algorithmthat 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.

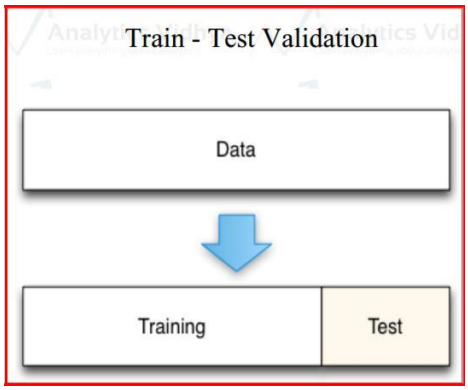


Fig: 2.4.1 model validation

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 are : 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.

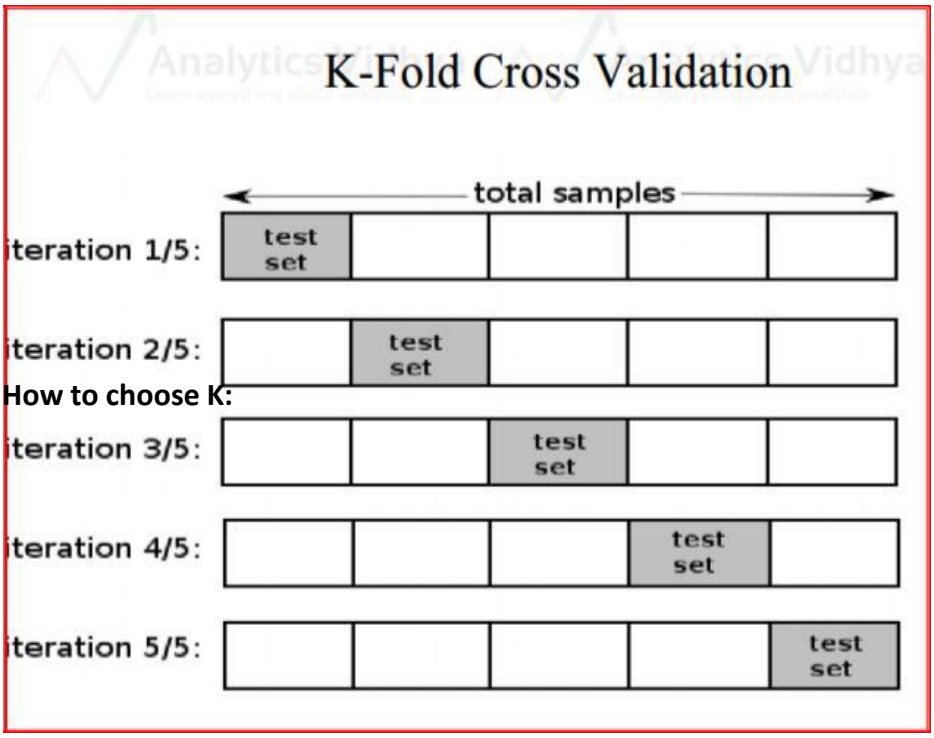


Fig: 2.4.1 a k-fold validation

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

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

1) 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 Underfitting and over fitting like mentioned in the below picture.

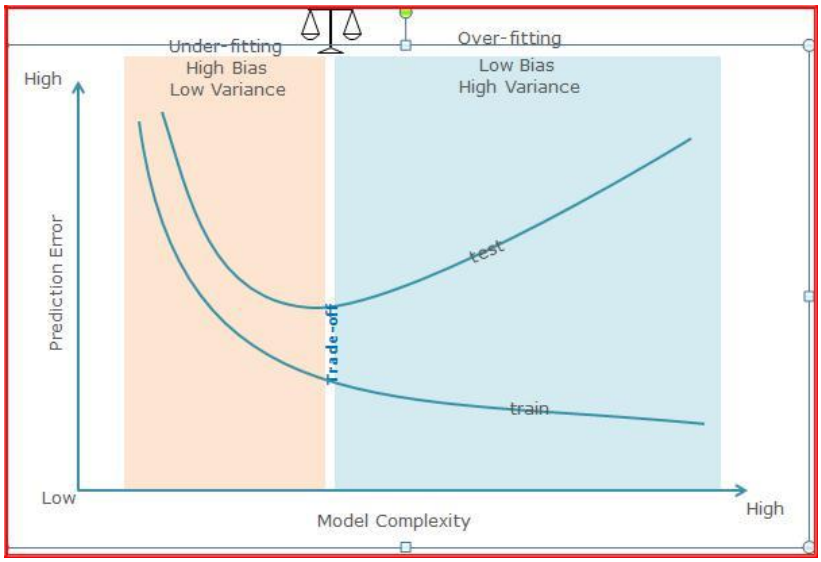


Fig : 2.5.1 over-fitting issue

And, the Hyperparameters 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

Marix

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 : confussion 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 : confussion matrix

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

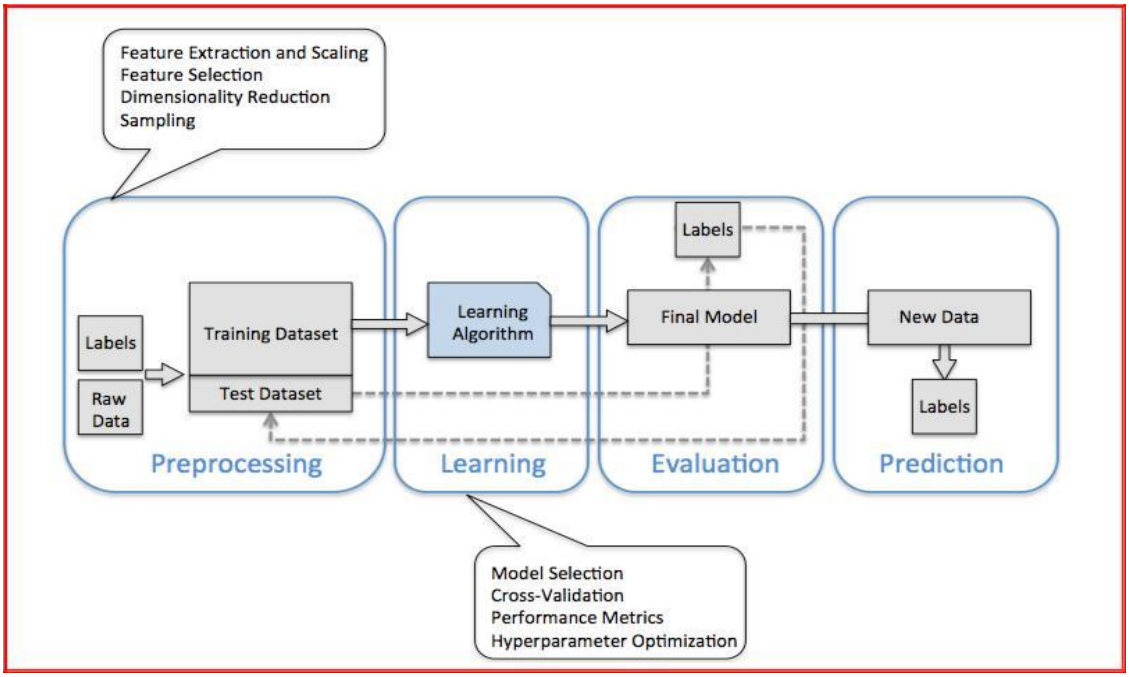


Fig 2.6 road map for building ML Systems

CHAPTER 3

MACHINE LEARNING-ATWORK

3.1 An approach to the problem:

In order to carry out the analysis, we have extracted WATER records from the SOUTHERN CALIFORNIA 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.

Data understanding and preparation:

To begin,we will load the dataset named water and define the structure of the str() function as follows:

```
'data.frame':   43 obs. of  8 variables:
$ Year   : int  1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 ...
$ APMAM  : num  9.13 5.28 4.2 4.6 7.15 9.7 5.02 6.7 10.5 9.1 ...
$ APSAB  : num  3.58 4.82 3.77 4.46 4.99 5.65 1.45 7.44 5.85 6.13 ...
$ APSLAKE: num  3.91 5.2 3.67 3.93 4.88 4.91 1.77 6.51 3.38 4.08 ...
$ OPBPC  : num  4.1 7.55 9.52 11.14 16.34 ...
$ OPRC   : num  7.43 11.11 12.2 15.15 20.05 ...
$ OPSLAKE: num  6.47 10.26 11.35 11.13 22.81 ...
$ BSAAM  : int  54235 67567 66161 68094 107080 67594 65356 67909 92715 70024 ...
```

Out put 3.1 : decrisption of data

Here we have eightn features and one response variable,BSAAM. The observations start in 1943 and run for 43 consecutiveyears. Since we are not concerned with what year the observations occurred, it makes sense to create a new dataframe,excluding the year vector .This is quite easy to do. With one line of code,we can creat thedata frame ,and then that verify that it worked with the head()function as follows:

```
The socal.water=water[,-1] #new dataframe  With
deletion  of  column  1
```

Understanding data using Descriptive Statistics:

We will look at the head of the data.

```
> head(socal.water)
  APMAM APSAB APSLAKE OPBPC  OPRC OPSLAKE  BSAAM
1  9.13  3.58    3.91  4.10  7.43    6.47  54235
2  5.28  4.82    5.20  7.55 11.11   10.26  67567
3  4.20  3.77    3.67  9.52 12.20   11.35  66161
4  4.60  4.46    3.93 11.14 15.15   11.13  68094
5  7.15  4.99    4.88 16.34 20.05   22.81 107080
6  9.70  5.65    4.91  8.88  8.15    7.41  67594
```

Out put :3.1.1: head of the data

We will look at the names of the data.

```
> names(socal.water)
[1] "APMAM"   "APSAB"   "APSLAKE" "OPBPC"   "OPRC"    "OPSLAKE" "BSAAM"
```

Output 3.1.2 : names of variables

We look at the dimentional of the data

```
> dim(socal.water)
[1] 43  7
```

Output3.1.3:dimentional of the data

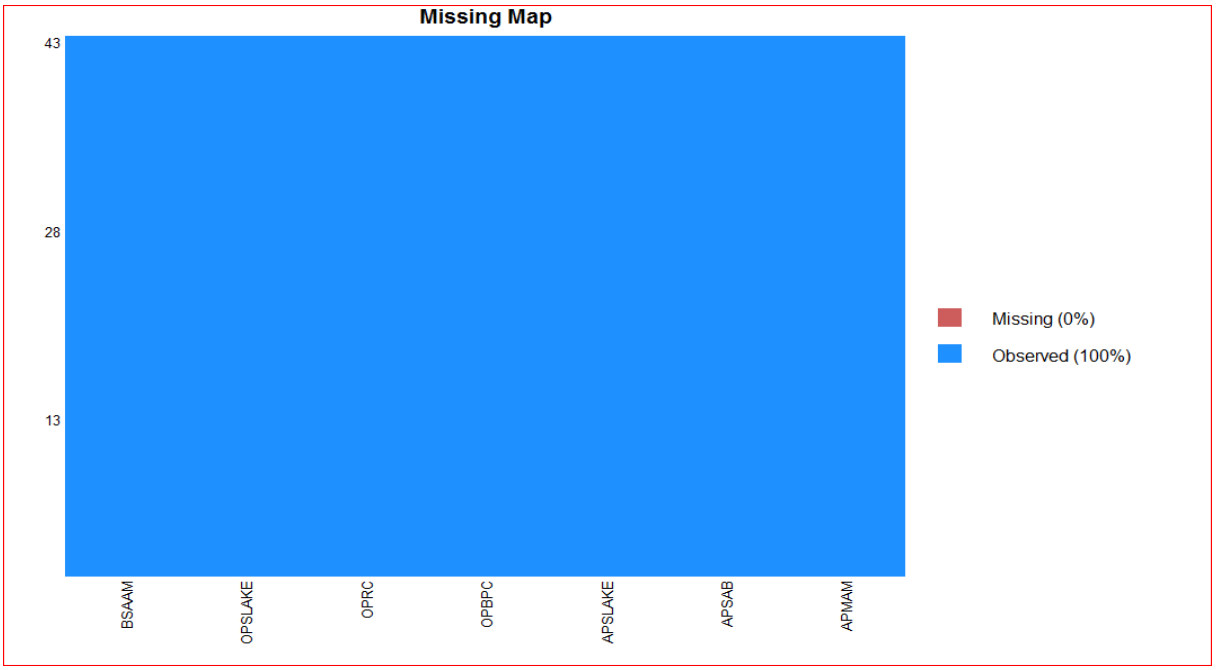
Checking for missing Values

APMAM	APSAB	APSLAKE	OPBPC	OPRC	OPSLAKE	BSAAM
0	0	0	0	0	0	0

Their are no missing values

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

Visually:

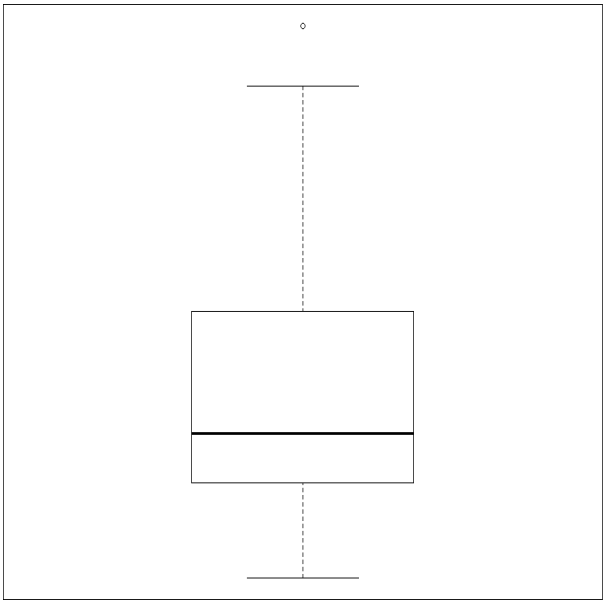


Output 3.1.4: missing values

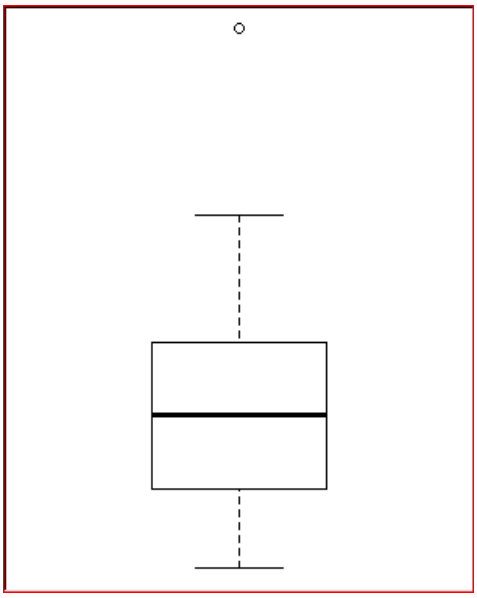
Checking for Outliers:

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

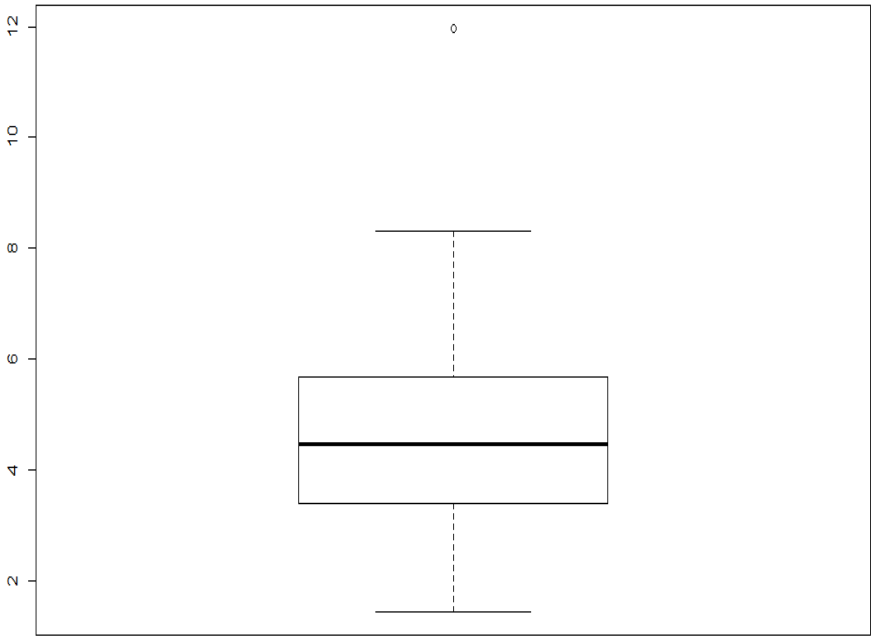
We used Box-plots to check for Outliers for independent variable **BSAAM**.



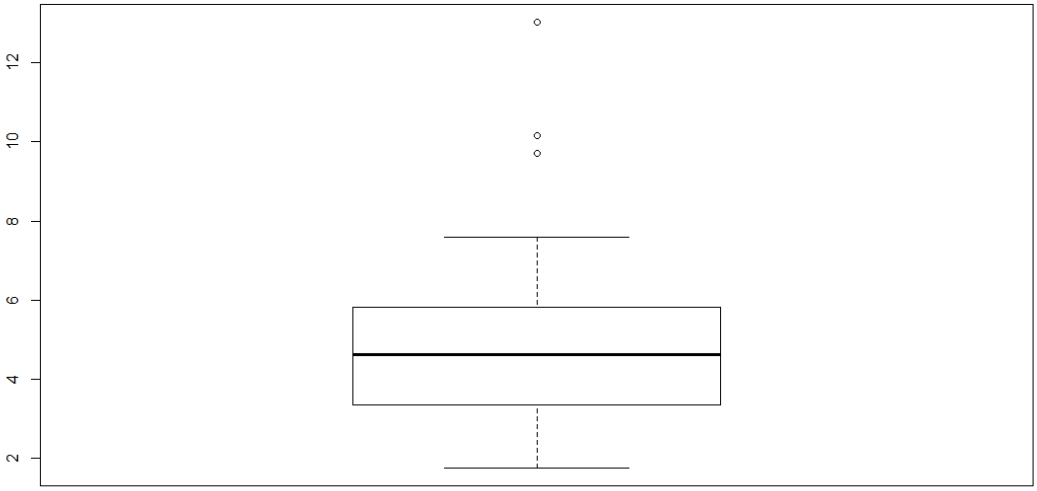
Boxplot for APMAM



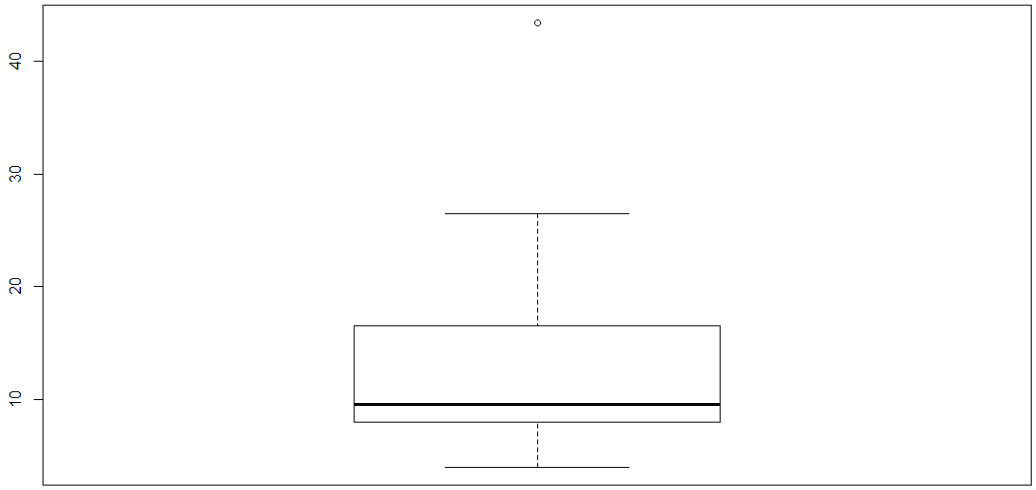
Boxplot for APSAB



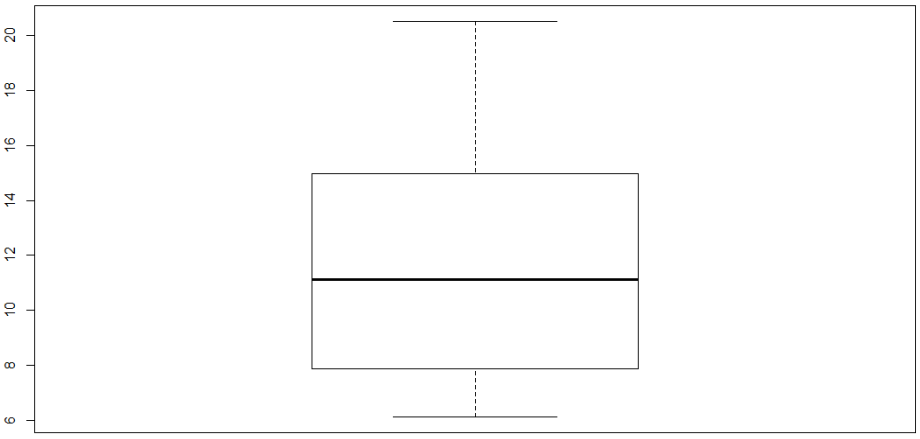
Boxplot for APSLAKE



Boxplot for OPBPC

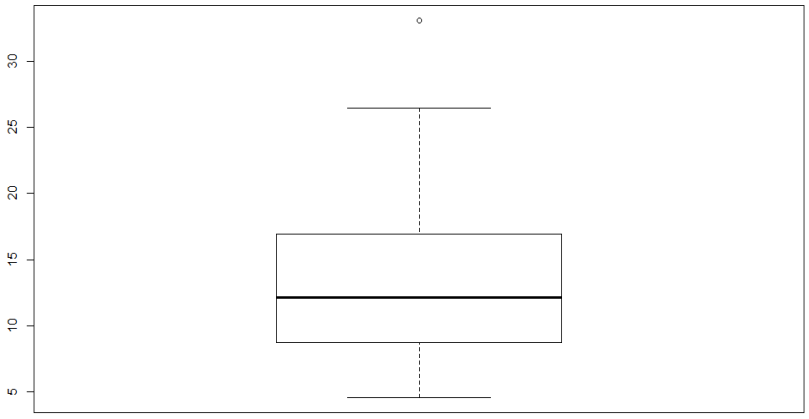


Boxplot for OPRC



There is no outliers

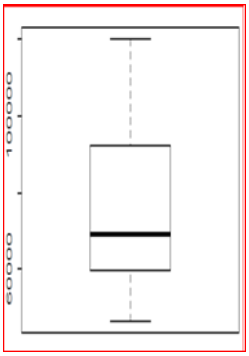
Boxplot for OPSLAKE



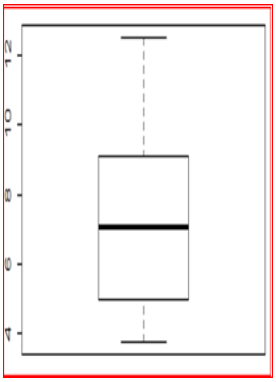
Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile will be imputed using 5th percentile value.

Boxplots after removing Outliers:

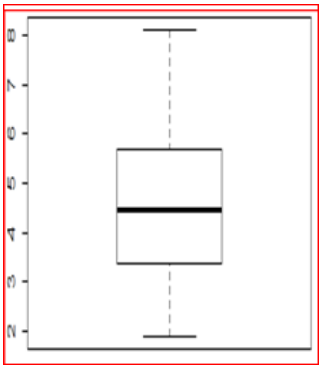
BSAAM



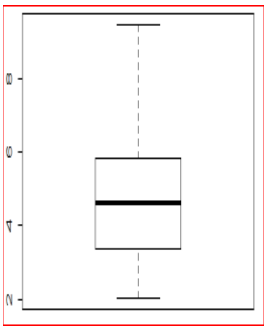
APMAM



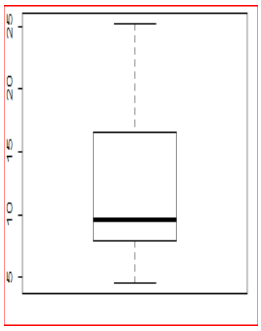
APSAB



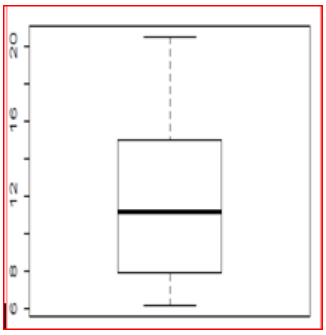
APSLAKE



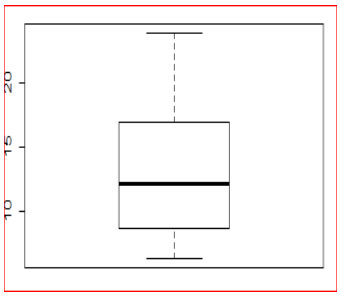
OPBPC



OPRC



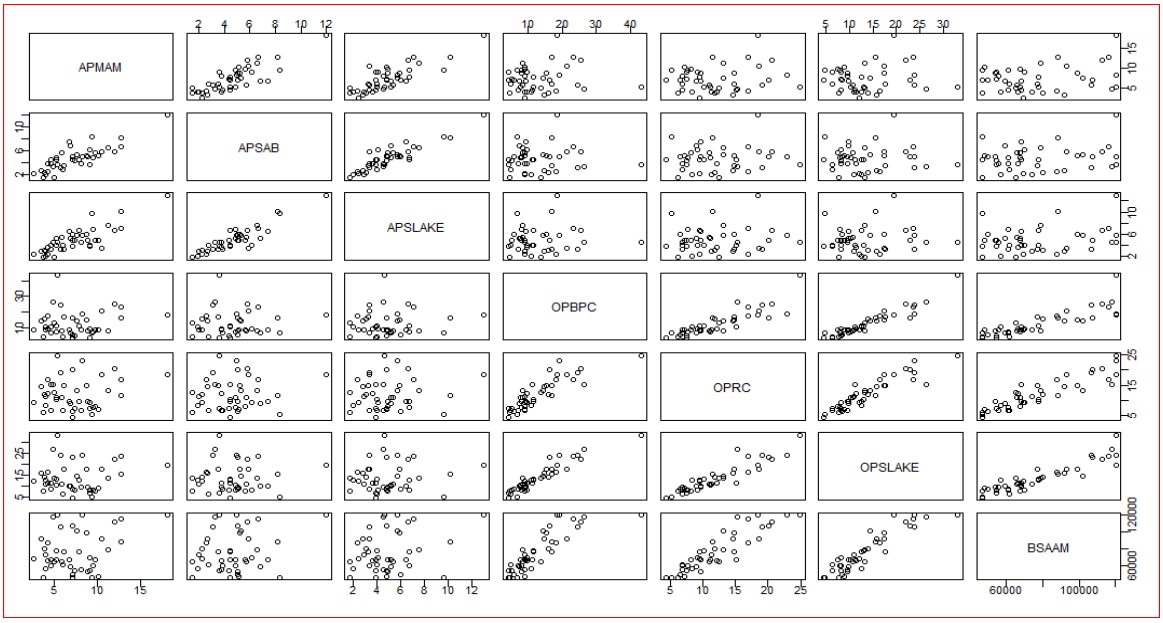
OPSALAKE



Output `3.1.5` : removed outliers

Understanding data visually:

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



Output 3.1.6 :relartionship between variables...

From the above , we can see some relationship between the variables

STANDERDIZING DATA

In oder to carry out the analysis we have split the data into train & test and the below are the train & test records

Split the data into train & test

```
> train_rows
 [1]  9 20  3 10 16 28 35 21  2 27 15 12 40 18 30  4  1 17 25 37  5 41  6 11 42 26 33 1
4 31 19 39
[32]  7 24 23  8 22 36 43
```

```
> dim(training)
[1] 38  7
```

```
> dim(test)
[1] 5 7
```

```
> head(test)
   APMAM APSAB APSLAKE  OPBPC  OPRC OPSLAKE    BSAAM
13 3.763 1.908   2.036  4.593  6.14   7.650 46493.86
29 9.380 8.097   9.490  6.800  6.14   6.371 46493.86
32 3.880 2.260   3.100 15.970 11.83  13.880 79975.00
34 3.763 2.220   2.480  8.990  9.45  12.140 69177.00
38 5.220 4.420   4.040 11.450 10.16  13.060 77790.00
```

```
> summary(socal.water)
     APMAM            APSAB           APSLAKE          OPBPC            OPRC
 Min.   : 3.763   Min.   :1.908   Min.   :2.036   Min.   : 4.593   Min.   : 6.140
 1st Qu.: 4.975   1st Qu.:3.390   1st Qu.:3.360   1st Qu.: 7.975   1st Qu.: 7.875
 Median : 7.080   Median :4.460   Median :4.620   Median : 9.550   Median :11.110
 Mean   : 7.222   Mean   :4.577   Mean   :4.841   Mean   :12.407   Mean   :11.907
 3rd Qu.: 9.115   3rd Qu.:5.685   3rd Qu.:5.830   3rd Qu.:16.545   3rd Qu.:14.975
 Max.   :12.577   Max.   :8.097   Max.   :9.490   Max.   :25.210   Max.   :20.500
    OPSLAKE          BSAAM
 Min.   : 6.371   Min.   : 46494
 1st Qu.: 8.705   1st Qu.: 59857
 Median :12.140   Median : 69177
 Mean   :13.326   Mean   : 76942
 3rd Qu.:16.920   3rd Qu.: 92206
```

Output 3.1.7:summary of data

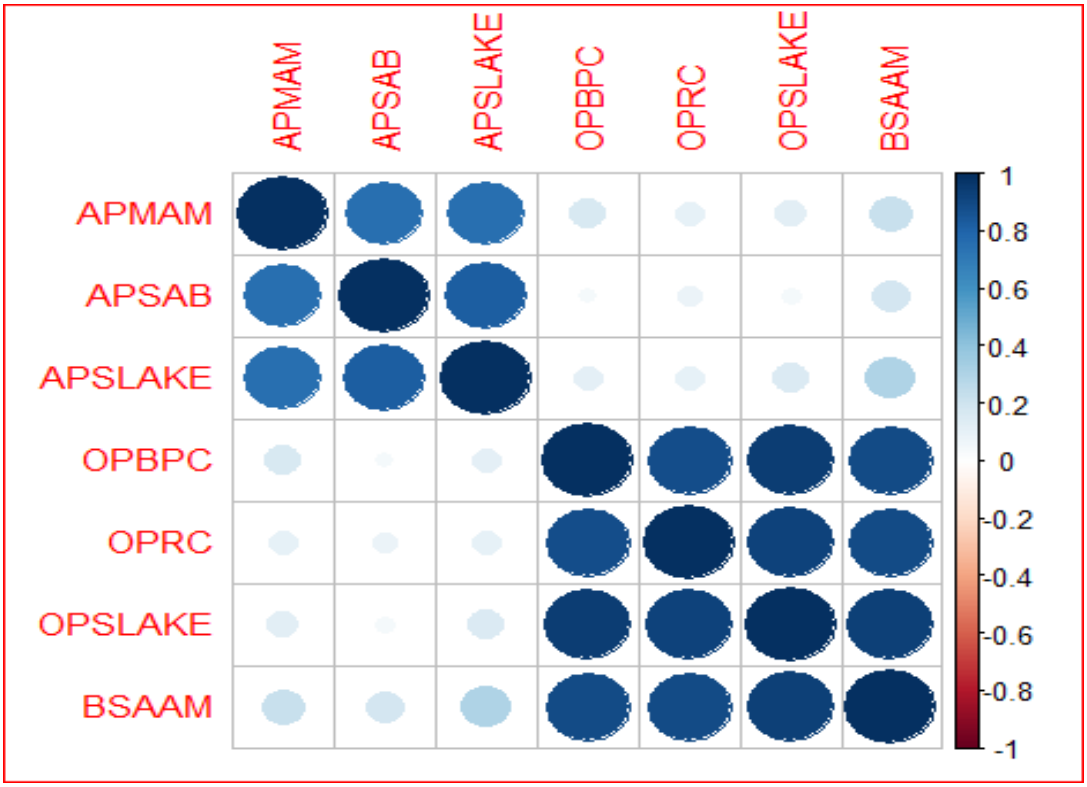
Coreplot

```
> water.cor     =       cor(training)
> water.cor
              APMAM         APSAB       APSLAKE       OPBPC          OPRC       OPSLAKE
APMAM    1.00000000  0.744265736  0.715545926 0.16344658  0.071674432  0.08200793
APSAB    0.74426574  1.000000000  0.840639881 0.01618442  0.004778007 -0.02792937
APSLAKE  0.71554593  0.840639881  1.000000000 0.07122699 -0.008137834  0.05396362
OPBPC    0.16344658  0.016184423  0.071226986 1.00000000  0.895484405  0.94684627
OPRC     0.07167443  0.004778007 -0.008137834 0.89548441  1.000000000  0.92317394
OPSLAKE  0.08200793 -0.027929373  0.053963624 0.94684627  0.923173940  1.00000000
BSAAM    0.13222714  0.038960999  0.117828274 0.92557919  0.914947795  0.95523669
             BSAAM
APMAM    0.1322271
APSAB    0.0389610
APSLAKE  0.1178283
OPBPC    0.9255792
OPRC     0.9149478
OPSLAKE  0.9552367
```

Output 3.1.8 : coreplot

Understanding relationships between variables

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



Output :3.1.9: coreplot

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

So, what does this tell us ? first of all, the response variable. Is highly and positively correlated with the OP features with OPBPC as 0.9255, OPRC as 0.9149,and OPSLAKE as 0.9552.Also note that the AP features are highly correlated with each other and the OP features as well. The implication is that we may run into the issues of multicollenearity. The correlation plot matrix provides a nice visual of the correlations as follows:

```
# Base Tthe Equation
fit=lm(BSAAM~.,      data=training)
names(fit)
[1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values"
[6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
```

```
> summary(fit)

Call:
lm(formula = BSAAM ~ ., data = training)

Residuals:
   Min     1Q Median     3Q    Max
-10993  -4794  -1339   6112  16822

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  16819.8     5116.0   3.288  0.00252 **
APMAM         -587.1      806.9  -0.728  0.47231
APSAB        -1008.6     1588.7  -0.635  0.53018
APSLAKE       3858.5     1477.2   2.612  0.01375 *
OPBPC         1028.5      643.0   1.600  0.11984
OPRC          1749.3      729.7   2.397  0.02273 *
OPSLAKE       1261.1      897.8   1.405  0.17010
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7508 on 31 degrees of freedom
Multiple R-squared:  0.9143,    Adjusted R-squared:  0.8977
F-statistic: 55.11 on 6 and 31 DF,  p-value: 3.537e-15
```

Output 3.1.10 : summary of fit

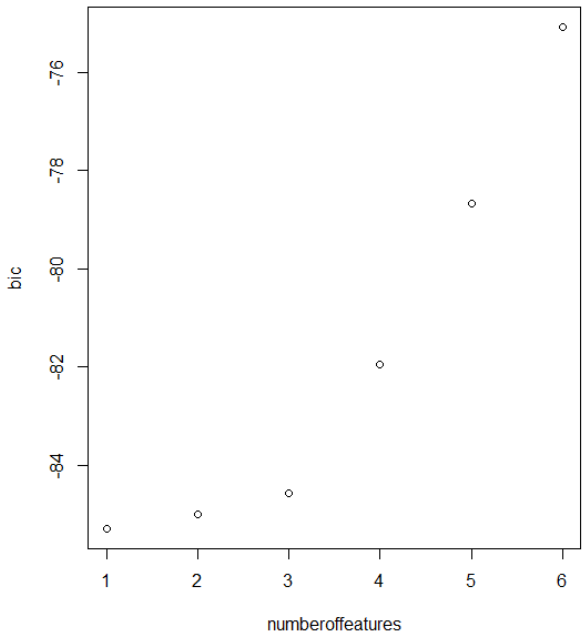
From the above we are getting an accuracy of 91% , when we run regression with all variables

In order to find the best variable we have applied step wise regression and the below are the outputs for the same.

From the above table, as the p value is >0.05, we can conclude that there is no significant relationship between the variables

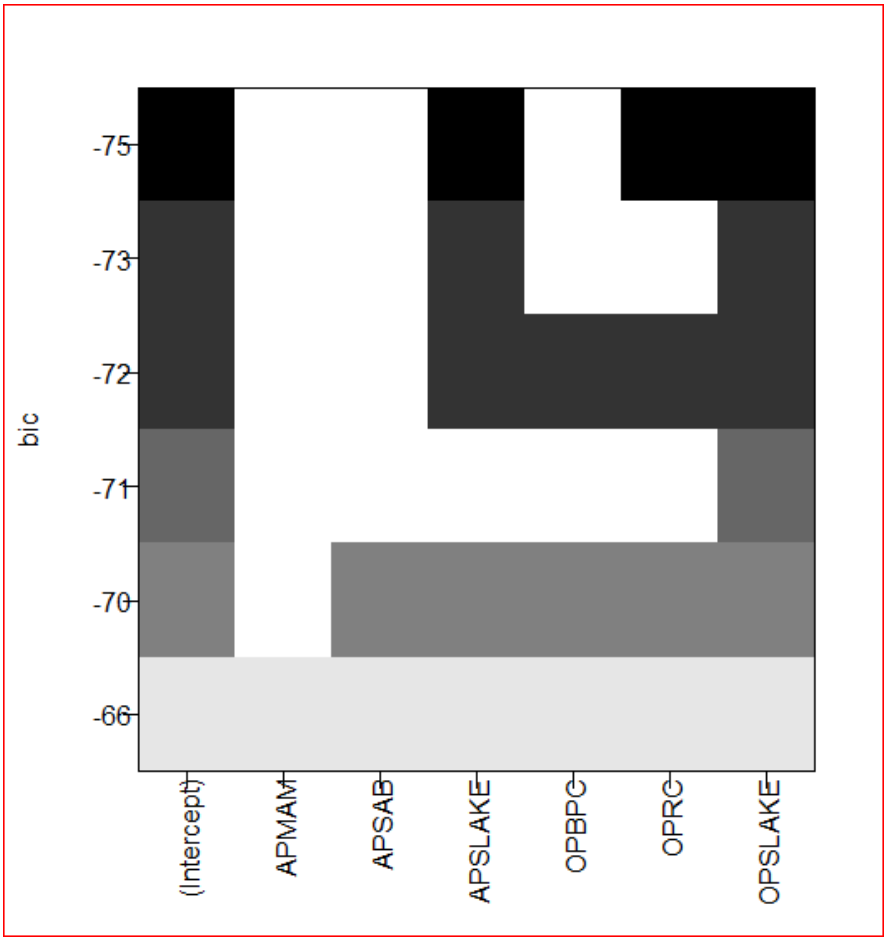
```
library(leaps)
# Best subsets
sub.fit      =      regsubsets(BSAAM~.,     data=training)
best.summary  =     summary(sub.fit)
names(best.summary)
] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
par(mfrow=c(1,2))
plot(best.summary$bic,      xlab="number    of     features",     ylab="bic")
```



Output 3.1.11: number of features

```
plot(sub.fit, scale="bic")
```



Output 3.1.12 : sub fit

```
> which.min(best.summary$bic)
[1] 1
```

```
> which.max(best.summary$adjr2)
[1] 3
```

From the above , we have short listed the three variables base BIC value which are OPSSLAKE, OPRC , APSLAKE

```
> best.fit      =       lm(BSAAM~APSLAKE+OPRC+OPSLAKE,  data=training)
> summary(best.fit)

Call:
lm(formula = BSAAM ~ APSLAKE + OPRC + OPSLAKE, data = training)

Residuals:
     Min       1Q   Median       3Q      Max
-10488.3  -4509.5    714.1   3744.9  12369.6

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  18916.8     4018.2   4.708 4.10e-05 ***
APSLAKE       1033.1      598.9   1.725   0.0936 .
OPRC          1251.3      598.9   2.089   0.0442 *
OPSLAKE       2839.0      487.4   5.825 1.45e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6548 on 34 degrees of freedom
Multiple R-squared:  0.9263,    Adjusted R-squared:  0.9198
F-statistic: 142.5 on 3 and 34 DF,  p-value: < 2.2e-16
```

```
> best.fit1     =       lm(BSAAM~APSLAKE+OPSLAKE,       data=training)
> summary(best.fit1)

Call:
lm(formula = BSAAM ~ APSLAKE + OPSLAKE, data = training)

Residuals:
   Min     1Q Median     3Q    Max
-14651  -5144   1665   3331  13446

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21987.5     3915.4   5.616 2.48e-06 ***
APSLAKE        844.2      619.8   1.362    0.182
OPSLAKE       3780.8      193.9  19.500  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6856 on 35 degrees of freedom
Multiple R-squared:  0.9169,    Adjusted R-squared:  0.9121
```

Output 3.1.13 : summary of best fit

Now with the three variables we are getting R2 arround 92% with three variables

Now we need to check for multicollinearity

```
> library(car)
> vif(best.fit)
 APSLAKE     OPRC  OPSLAKE
1.026320 6.926109 6.945877
>
```

From the above we have some multicollinearity , Hence we can drop one variable and run regression two variables .

From the above table, , we can conclude APSLAKE OPRC OPSLAKE

```
> best.fit1     =       lm(BSAAM~APSLAKE+OPSLAKE,       data=training)
> summary(best.fit1)

Call:
lm(formula = BSAAM ~ APSLAKE + OPSLAKE, data = training)

Residuals:
   Min     1Q Median     3Q    Max
-14651  -5144    1665   3331  13446

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21987.5     3915.4   5.616 2.48e-06 ***
APSLAKE        844.2      619.8   1.362    0.182
OPSLAKE       3780.8      193.9  19.500  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6856 on 35 degrees of freedom
Multiple R-squared:  0.9169,    Adjusted R-squared:  0.9121
F-statistic:   193 on 2 and 35 DF,  p-value: < 2.2e-16
```

Output 3.1.13: summary of best fit

```
> vif(best.fit1)
 APSLAKE  OPSLAKE
1.002921 1.002921
```

From the above table we obtain best fit for APSLAKE OPSLAKE

Therefore , the final model with two variables explaining around 92% variance and there is no multicollinearity and hence this is the best model

```
> # RMSE: Mean Squared Error
> sqrt(mean((test$BSAAM - predicted)^2))
[1] 13607.37
> # MAPE: Mean Absolute Percentage Error
> mean(abs(test$BSAAM - predicted)/test$BSAAM)
[1] 0.1125565
```

Output 3.1.14: output of mean square error and mean absolute percentage error

From the above table we obtain mean square eror & mean absolute percentage error

CHAPTER 4

4.1 SUMMARY:

In order to solve the above problem we have applied STEP WISE REGRESSION To find the best variable

Predicting WATER REGRESSION. hence we have applied STEP WISE REGRESSION

For this key variable and obtained the train and test data and computed the accuracy

R- Square Test = 93.75%

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 of the data

APPENDIX

R –CODE

DATA SET

BIBLIOGRAPHY

R PROGRAM

```
setwd("C:/batch11/Batch11 water")

getwd()

water=read.csv("input.csv")

str(water)

socal.water = water[ ,-1] #new dataframe with the
    deletion of column 1

head(socal.water)

names(socal.water)

dim(socal.water)


# Missing values

sapply(socal.water, function(df) {

  (sum(is.na(df)==TRUE)/ length(df))*100;

})


#install.packages("Amelia")

require(Amelia)

missmap(socal.water, main="Missing Map")
```

```
#AmeliaView()

##############################################3

# Removing outliers

boxplot(socal.water$BSAAM)

pairs(socal.water)

socal.water$BSAAM[socal.water$BSAAM>quantile(socal.water$BSAAM
, 0.95)] <- quantile(socal.water$BSAAM, 0.95)

socal.water$BSAAM[socal.water$BSAAM<quantile(socal.water$BSAAM
, 0.05)] <- quantile(socal.water$BSAAM, 0.05)

boxplot(socal.water$BSAAM)

pairs(socal.water)

boxplot(socal.water$APMAM)

pairs(socal.water)

socal.water$APMAM[socal.water$APMAM>quantile(socal.water$APM
AM, 0.95)] <- quantile(socal.water$APMAM, 0.95)

socal.water$APMAM[socal.water$APMAM<quantile(socal.water$APM
AM, 0.05)] <- quantile(socal.water$APMAM, 0.05)

boxplot(socal.water$APMAM)

pairs(socal.water)

boxplot(socal.water$APSAB)

pairs(socal.water)

socal.water$APSAB[socal.water$APSAB>quantile(socal.water$APSAB,
0.95)] <- quantile(socal.water$APSAB, 0.95)
```

```
socal.water$APSAB[socal.water$APSAB<quantile(socal.water$APSAB,0.05)] <- quantile(socal.water$APSAB, 0.05)

boxplot(socal.water$APSAB)

pairs(socal.water)

boxplot(socal.water$APSLAKE)

socal.water$APSLAKE[socal.water$APSLAKE>quantile(socal.water$APSLAKE, 0.95)] <- quantile(socal.water$APSLAKE, 0.95)

socal.water$APSLAKE[socal.water$APSLAKE<quantile(socal.water$APSLAKE, 0.05)] <- quantile(socal.water$APSLAKE, 0.05)

boxplot(socal.water$APSLAKE)

boxplot(socal.water$OPBPC)

socal.water$OPBPC[socal.water$OPBPC>quantile(socal.water$OPBPC,0.95)] <- quantile(socal.water$OPBPC, 0.95)

socal.water$OPBPC[socal.water$OPBPC<quantile(socal.water$OPBPC, 0.05)] <- quantile(socal.water$OPBPC, 0.05)

boxplot(socal.water$OPBPC)

boxplot(socal.water$OPRC)

socal.water$OPRC[socal.water$OPRC>quantile(socal.water$OPRC, 0.95)] <- quantile(socal.water$OPRC, 0.95)

socal.water$OPRC[socal.water$OPRC<quantile(socal.water$OPRC, 0.05)] <- quantile(socal.water$OPRC, 0.05)

boxplot(socal.water$OPRC)

boxplot(socal.water$OPSLAKE)

socal.water$OPSLAKE[socal.water$OPSLAKE>quantile(socal.water$OPSLAKE, 0.95)] <- quantile(socal.water$OPSLAKE, 0.95)
```

```
socal.water$OPSLAKE[socal.water$OPSLAKE<quantile(socal.water$OPS
LAKE, 0.05)] <- quantile(socal.water$OPSLAKE, 0.05)

boxplot(socal.water$OPSLAKE)

##############################################3

#Split the data in to train andtests

##############################################3

train_rows<- sample(1:nrow(socal.water), size=0.9*nrow(socal.water))

train_rows

training <- socal.water[train_rows, ]

test <- socal.water[-train_rows, ]

dim(training)

dim(test)

head(test)

##############################################3

summary(socal.water)

##############################################3

library(corrplot)

water.cor   =       cor(training)

water.cor

corrplot(water.cor,      method="circle")
```

```
# BAse final
library(leaps)
# Base line Equation
fit=lm(BSAAM~., data=training)
names(fit)
summary(fit)
##################################################3
library(leaps)
# Best subsets
sub.fit = regsubsets(BSAAM~., data=training)
best.summary = summary(sub.fit)
names(best.summary)

par(mfrow=c(1,2))
plot(best.summary$bic, xlab="number of features",
     ylab="bic")
plot(sub.fit, scale="bic")
which.min(best.summary$bic)
which.max(best.summary$adjr2)

##################################################3

base.mod = lm(BSAAM ~ 1, data=training)
```

```
all.mod = lm(BSAAM ~ .,       data=training)


stepMod<- step(base.mod,
 scope = list(lower = base.mod, upper = all.mod),
  direction = "both", trace = 1, steps = 1000)


summary(stepMod)
################################################3
best.fit      =      lm(BSAAM~APSLAKE+OPRC+OPSLAKE,   data=training)
summary(best.fit)


library(car)


vif(best.fit)
################################################3
best.fit1     =      lm(BSAAM~APSLAKE+OPSLAKE,     data=training)
summary(best.fit1)
vif(best.fit1)
################################################3
predicted <- predict(best.fit1, newdata = test)
xx=cbind(test$BSAAM,predicted)
write.csv(xx,"output.csv")
```

```
# RMSE: Mean Squared Error
sqrt(mean((test$BSAAM - predicted)^2))

# MAPE: Mean Absolute Percentage Error
mean(abs(test$BSAAM - predicted)/test$BSAAM)
```

INPUT DATA

Year	APMAM	APSAB	APSLAKE	OPBPC	OPRC	OPSLAKE	BSAAM
1948	9.13	3.58	3.91	4.1	7.43	6.47	54235
1949	5.28	4.82	5.2	7.55	11.11	10.26	67567
1950	4.2	3.77	3.67	9.52	12.2	11.35	66161
1951	4.6	4.46	3.93	11.14	15.15	11.13	68094
1952	7.15	4.99	4.88	16.34	20.05	22.81	107080
1953	9.7	5.65	4.91	8.88	8.15	7.41	67594
1954	5.02	1.45	1.77	13.57	12.45	13.32	65356
1955	6.7	7.44	6.51	9.28	9.65	9.8	67909
1956	10.5	5.85	3.38	21.2	18.55	17.42	92715
1957	9.1	6.13	4.08	9.55	9.2	8.25	70024
1958	8.75	5.23	5.9	15.25	14.8	17.48	99216
1959	8.1	3.77	4.56	9.05	6.85	9.56	55786
1960	3.75	1.47	1.78	4.57	6.1	7.65	46153
1961	10.15	5.09	4.86	8.9	7.15	9	47947
1962	6.15	3.52	3.3	16.9	14.75	17.68	76877
1963	12.75	8.17	10.16	16.75	11.55	15.53	88443
1964	7.35	4.33	4.85	5.25	7.45	8.2	54634
1965	11.25	6.56	7.6	8.4	13.2	13.29	78806
1966	4.05	1.9	2	10.85	8.25	12.56	56542
1967	12.65	6.62	7.14	23.25	17	23.66	116244
1968	4.65	3.84	3.34	7.1	6.8	8.28	60857
1969	5.35	3.62	4.62	43.37	24.85	33.07	146345
1970	4.05	1.98	2.94	8.95	11.25	11	73726
1971	5.9	5.72	5.42	8.45	10.9	10.82	65530
1972	9.45	4.82	6.79	7.9	7.6	8.06	60772
1973	3.45	2.63	2.88	14.8	14.7	15.86	91696
1974	4.25	2.54	2.36	18.05	16.9	16.42	87377
1975	7.9	4.42	6.78	11.5	9.55	12.56	77306
1976	9.38	8.3	9.7	6.8	5.25	4.73	44756
1977	7.08	4.4	3.9	4.05	4.35	4.6	41785
1978	11.92	5.78	6.7	25.3	20.55	21.94	112653
1979	3.88	2.26	3.1	15.97	11.83	13.88	79975
1980	5.8	3.1	3.34	24.4	19.15	23.78	106821
1981	2.7	2.22	2.48	8.99	9.45	12.14	69177
1982	18.08	11.96	13.02	18.55	18.4	19.45	120463
1983	8.2	4.98	5.76	19.25	22.9	23.86	135043
1984	7.65	5.3	5.74	14.45	13.15	14.42	102001
1985	5.22	4.42	4.04	11.45	10.16	13.06	77790
1986	4.93	3.26	4.58	26.47	15.33	26.46	118144

1987	5.99	2.76	3.98	4.8	6.85	6.36	61229
1988	6.83	6.82	5.18	7.2	9.01	9.88	58942
1989	8.8	5.06	4.92	8.05	9.6	9.58	53965
1990	7.1	5.06	6.05	5.8	6.5	8.41	49774

‘

CHAPTER 6

BIBLIOGRAPHY

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

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

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

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

5. machine_learning_mastery_with_r. – Jason Brownlee

6. master_machine_learning_algorithms -- Jason Brownlee

7. statistical_methods_for_machine_learning - Jason Brownlee

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

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

10. Deep learning with Deep learning R by François Chollet