A Project

Report On

Prediction Using Supervised and Unsupervised Learning

By

Umama Valli Shaikh
B.E (Computer Engineering)
Batch: - 2021 – 5129
Center:- Thane

Under the Guidance of,

Mr Mathivanan Balakrishnan

Technical Trainer

EduBridge

(School of coding)

Software Requirements:

- **Software : Spyder 4.1.5, RStudio Version 1.4.1106**
- **Back End : MongoDB 4.4.4**
- **Operating System: Window 10,64-bit Operating System**

Linear Regression

Linear regression is used for finding linear relationship between target and one or more predictors. There are two types of linear regression- Simple and Multiple. linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables. For example, relationship between height and weight.

- **Example of Linear regression using python**

Student Grades Prediction

We have a dataset which contains information about relationship between 'number of hours studied' and 'marks obtained'. Many students have been observed and their hours of study and grade are recorded. This will be our training data. Goal is to design a model that can predict marks if given the number of hours studied. Using the training data, a regression line is obtained which will give minimum error. This linear equation is then used for any new data. That is, if we give number of hours studied by a student as an input, our model should predict their mark with minimum error.

$$Y(pred) = b0 + b1*x$$

The values b0 and b1 must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

$$\text{Error} = \sum_{i=1}^{n} (actual\_output - predicted\_output) ** 2$$

If we don't square the error, then positive and negative point will cancel out each other.

For model with one predictor,

$$b_0 = \bar{y} - b_1\bar{x}$$

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Co-efficient Formula

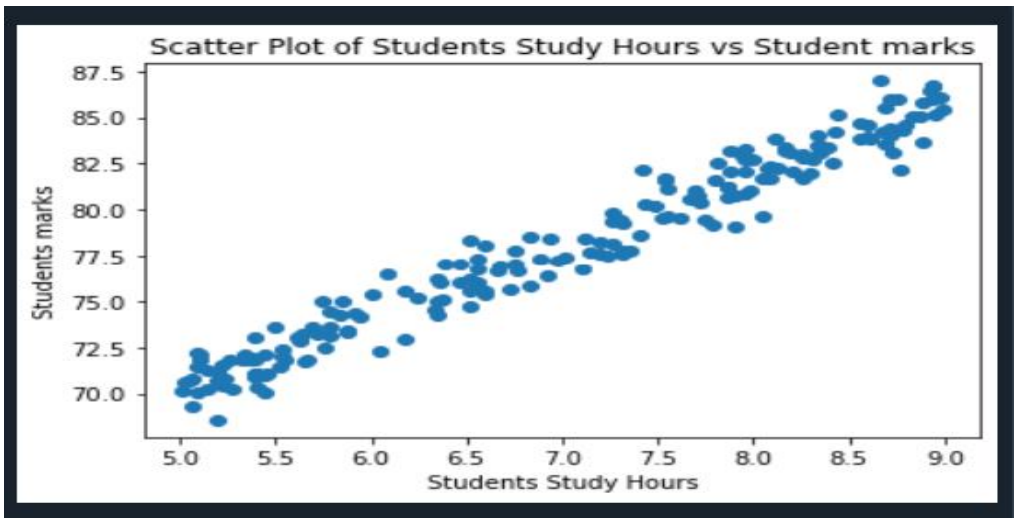
Exploring 'b1'

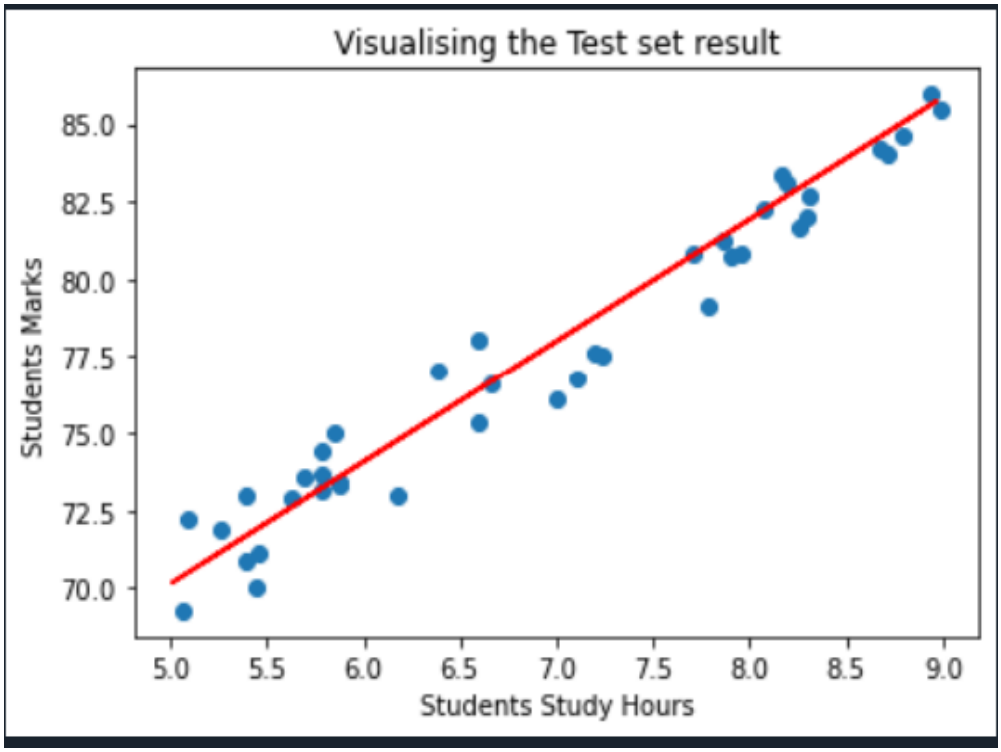
- If b1 > 0, then x(predictor) and y(target) have a positive relationship. That is increase in x will increase y.
- If b1 < 0, then x(predictor) and y(target) have a negative relationship. That is increase in x will decrease y.

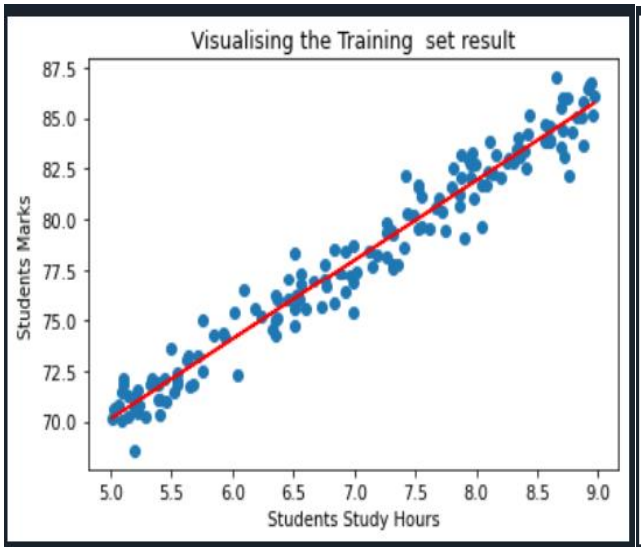
Exploring 'b0'

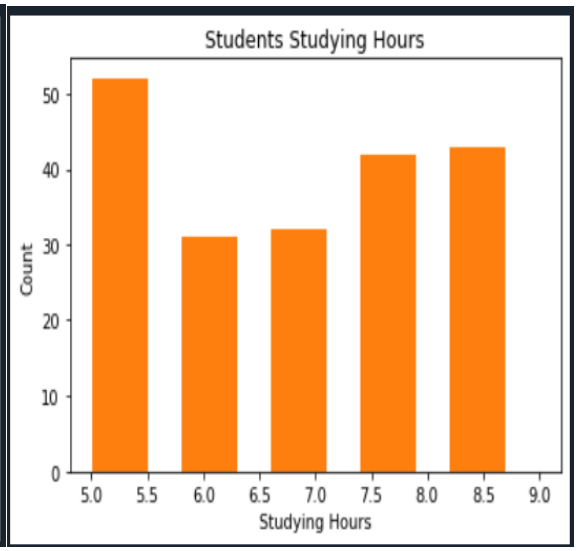
- If the model does not include x=0, then the prediction will become meaningless with only b0. For example, we have a dataset that relates height(x) and weight(y). Taking x=0(that is height as 0), will make equation have only b0 value which is completely meaningless as in real-time height and weight can never be zero. This resulted due to considering the model values beyond its scope.
- If the model includes value 0, then 'b0' will be the average of all predicted values when x=0. But, setting zero for all the predictor variables is often impossible.

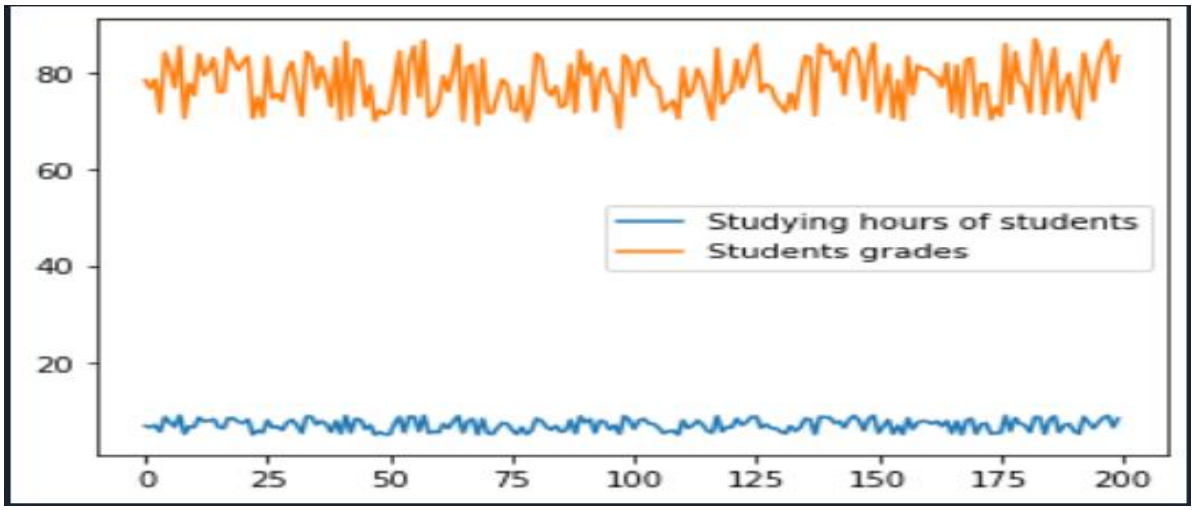
- The value of bo guarantee that residual have mean zero. If there is no ‘bo’ term, then regression will be forced to pass over the origin. Both the regression co-efficient and prediction will be biased.
- **<u>Experimental Results</u>**

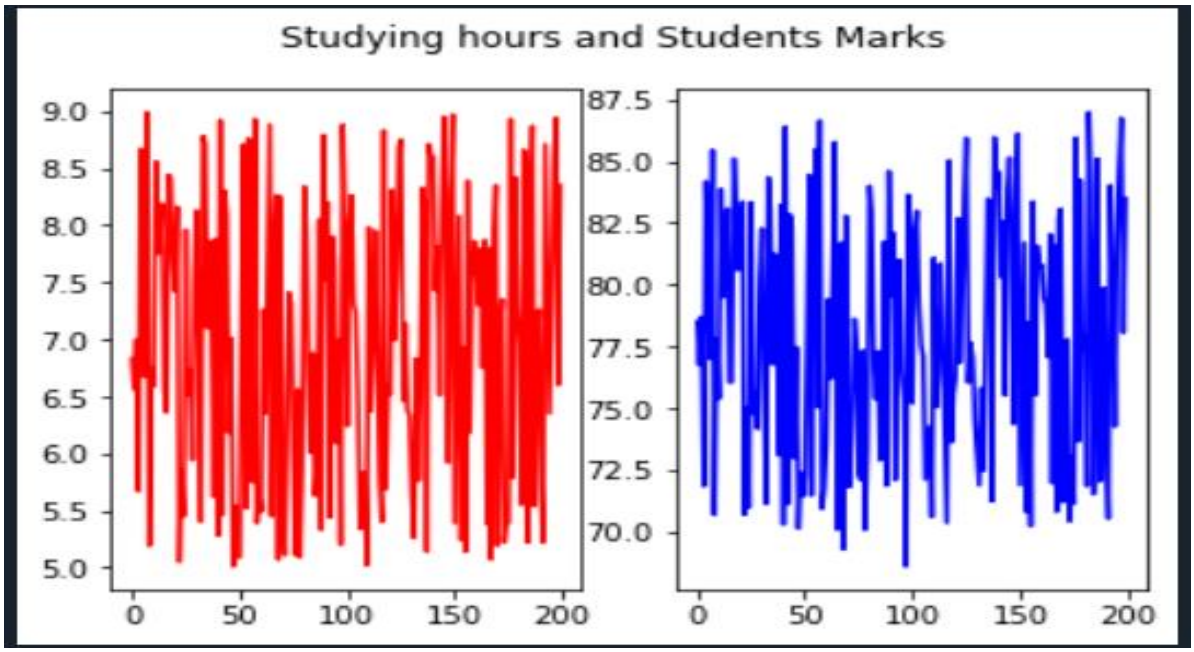





Visualising the Training set result
Students Marks
Students Study Hours
87.5
85.0
82.5
80.0
77.5
75.0
72.5
70.0
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


Students Studying Hours
Count
Studying Hours
50
40
30
20
10
0
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


80
60
40
20
Studying hours of students
Students grades
0
25
50
75
100
125
150
175
200


Studying hours and Students Marks
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
87.5
85.0
82.5
80.0
77.5
75.0
72.5
70.0
0
50
100
150
200

- **Example of Linear regression using R programming**

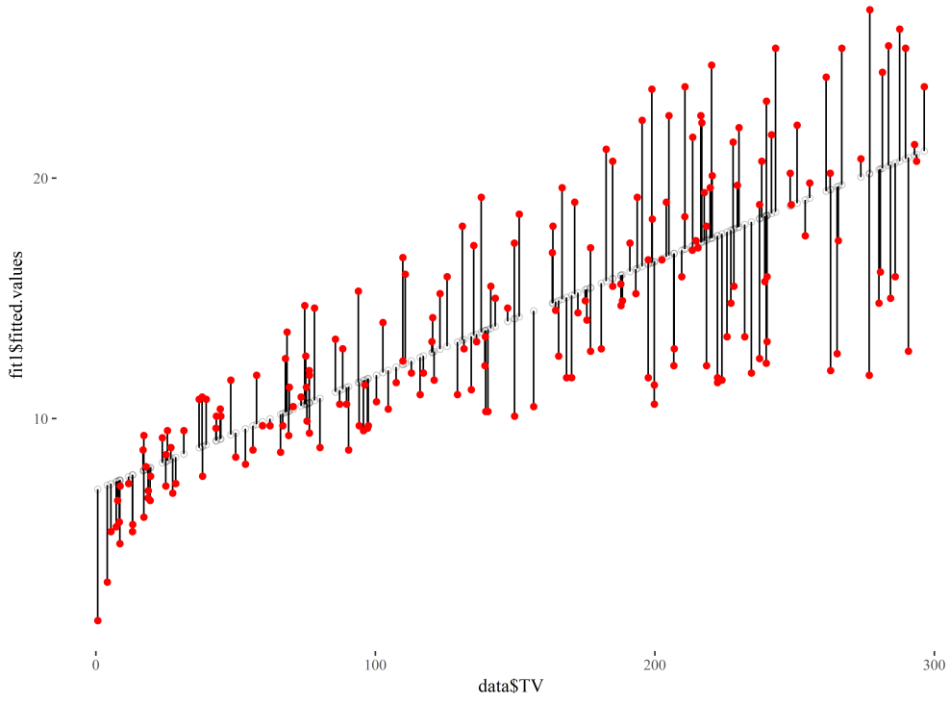
Adverting Data Prediction

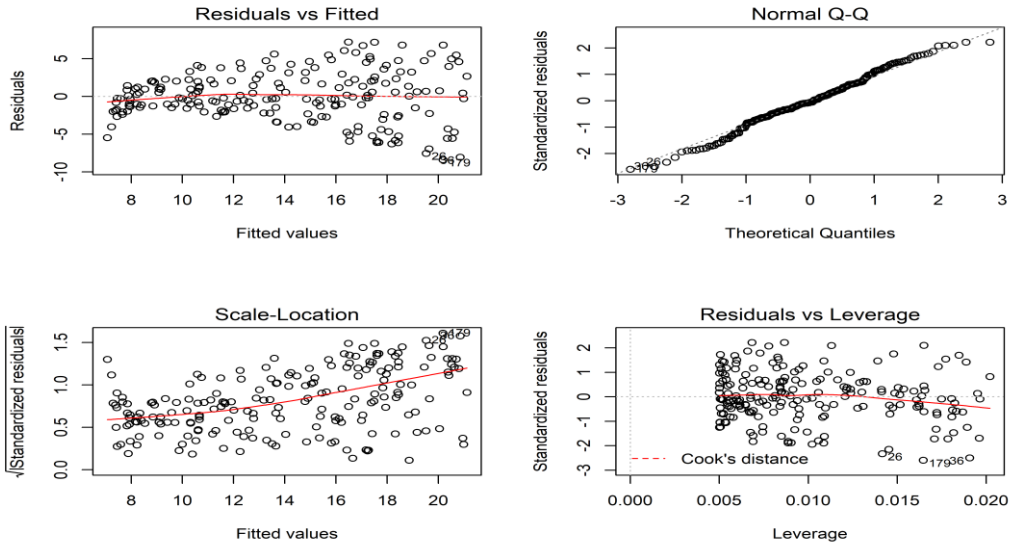
The dataset contains statistics about the sales of a product in 200 different markets, together with advertising budgets in each of these markets for different media channels: TV, radio and newspaper. The sales are in thousands of units and the budget is in thousands of dollars. We'll have here a deeper look at the data and what it means to apply a regression model to it.

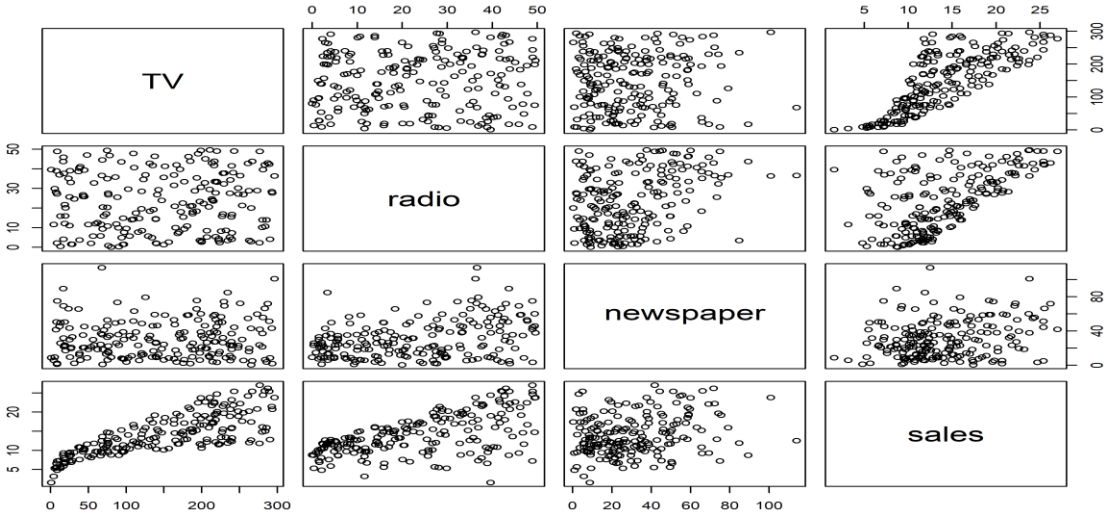
Look at: TV and Sales

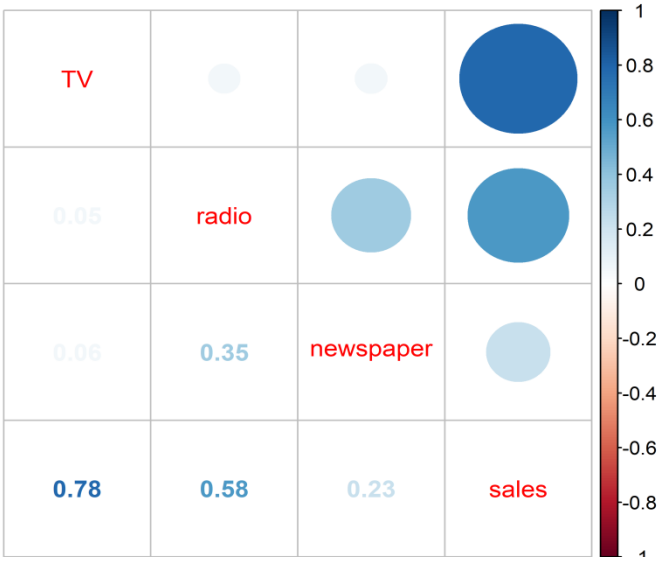
This is not yet the big picture, but let's first see how the TV spendings and the Sales numbers look like. In the graph, you'll see a straight-forward fit of a linear regression to it - very basic.

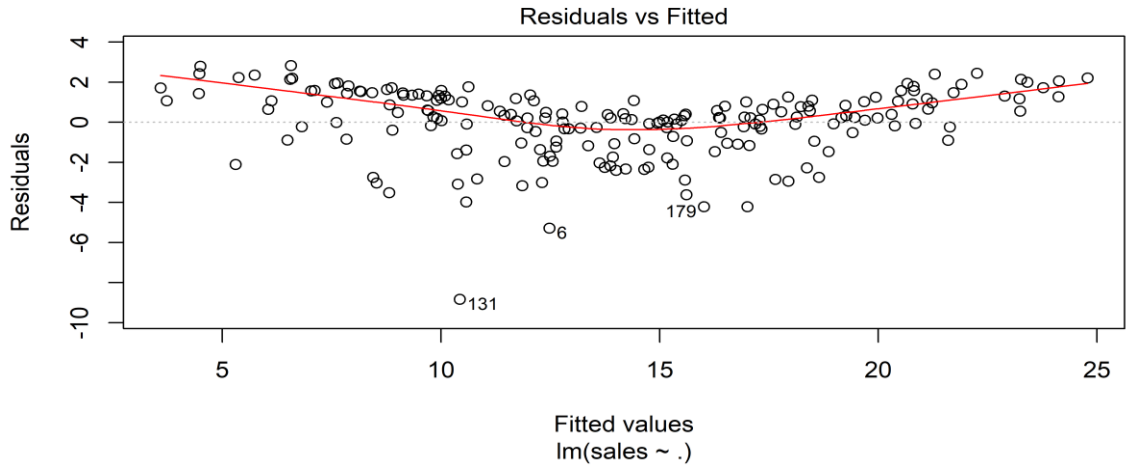
- **Experimental Results**

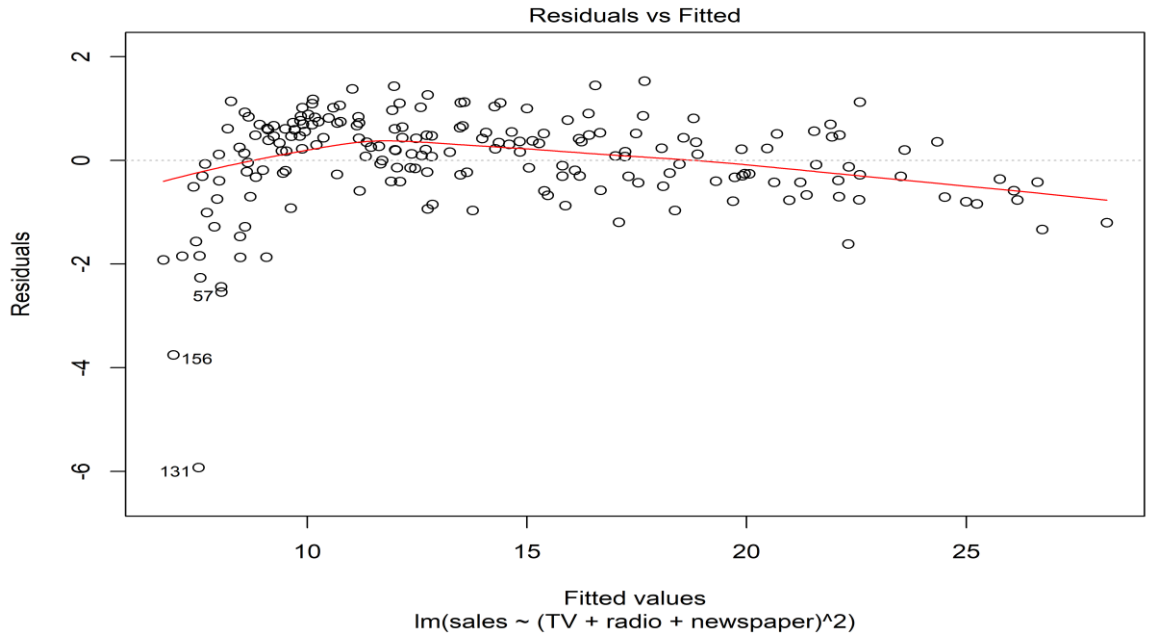



Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage


TV
radio
newspaper
sales

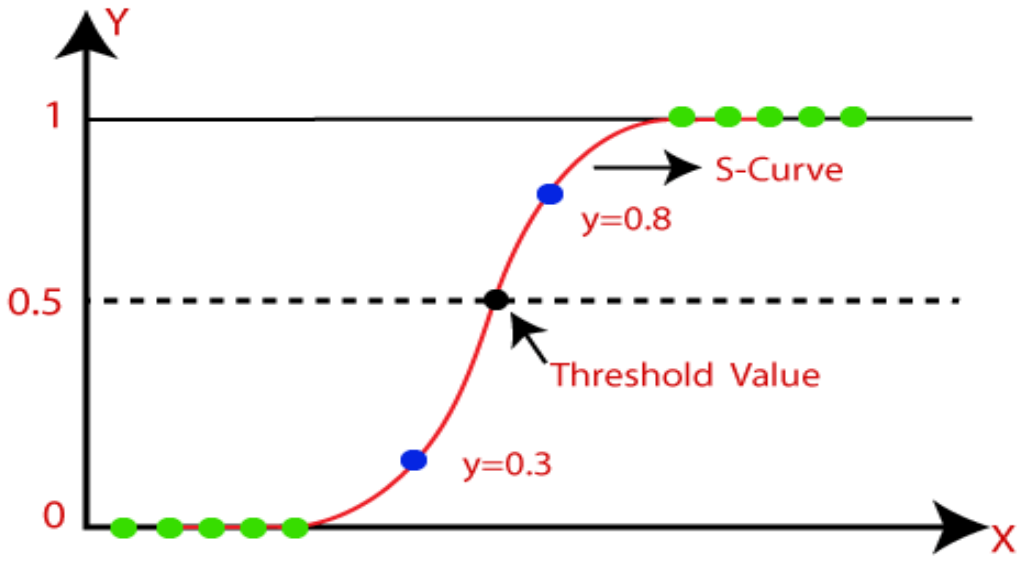

TV
radio
newspaper
sales
0.05
0.06
0.35
0.78
0.58
0.23
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8


Residuals vs Fitted
Residuals
Fitted values
lm(sales ~ .)
179
6
131


Residuals vs Fitted
Residuals
Fitted values
lm(sales ~ (TV + radio + newspaper)^2)
57
156
131

Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1**, it gives the probabilistic values which lie between 0 and 1**.Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.



The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}$$; 0 for y= 0, and infinity for y=1

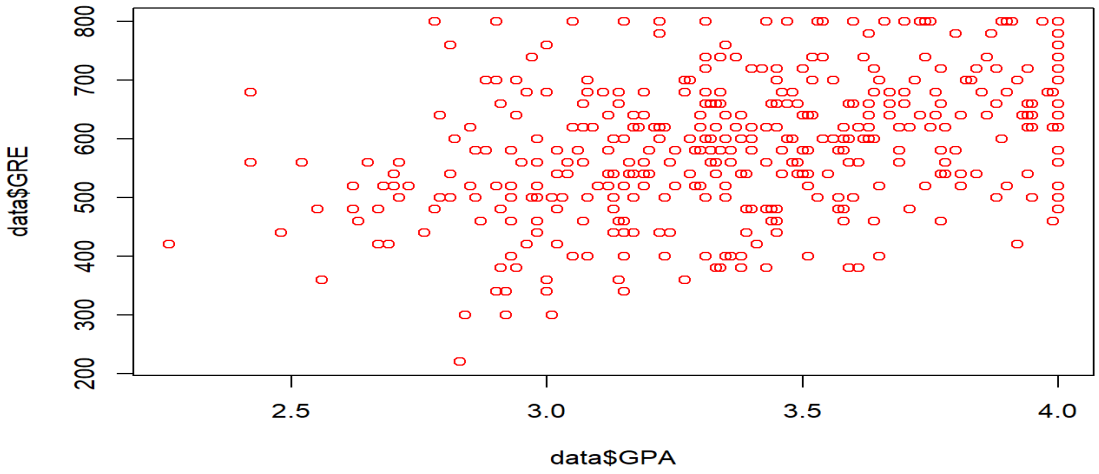
- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

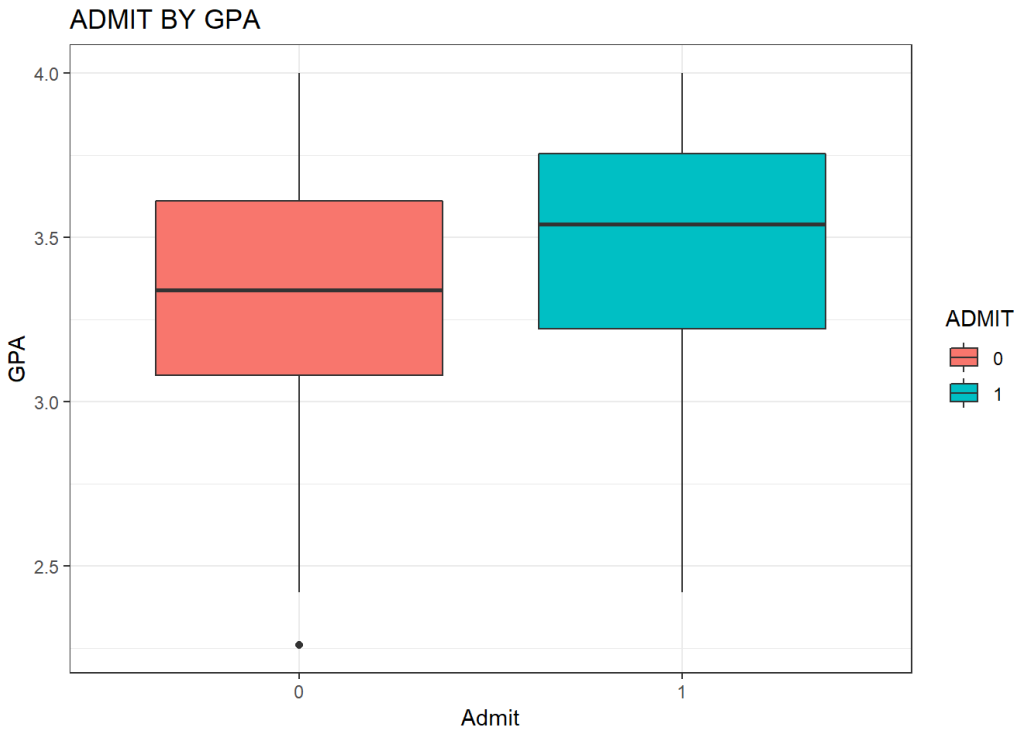
$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

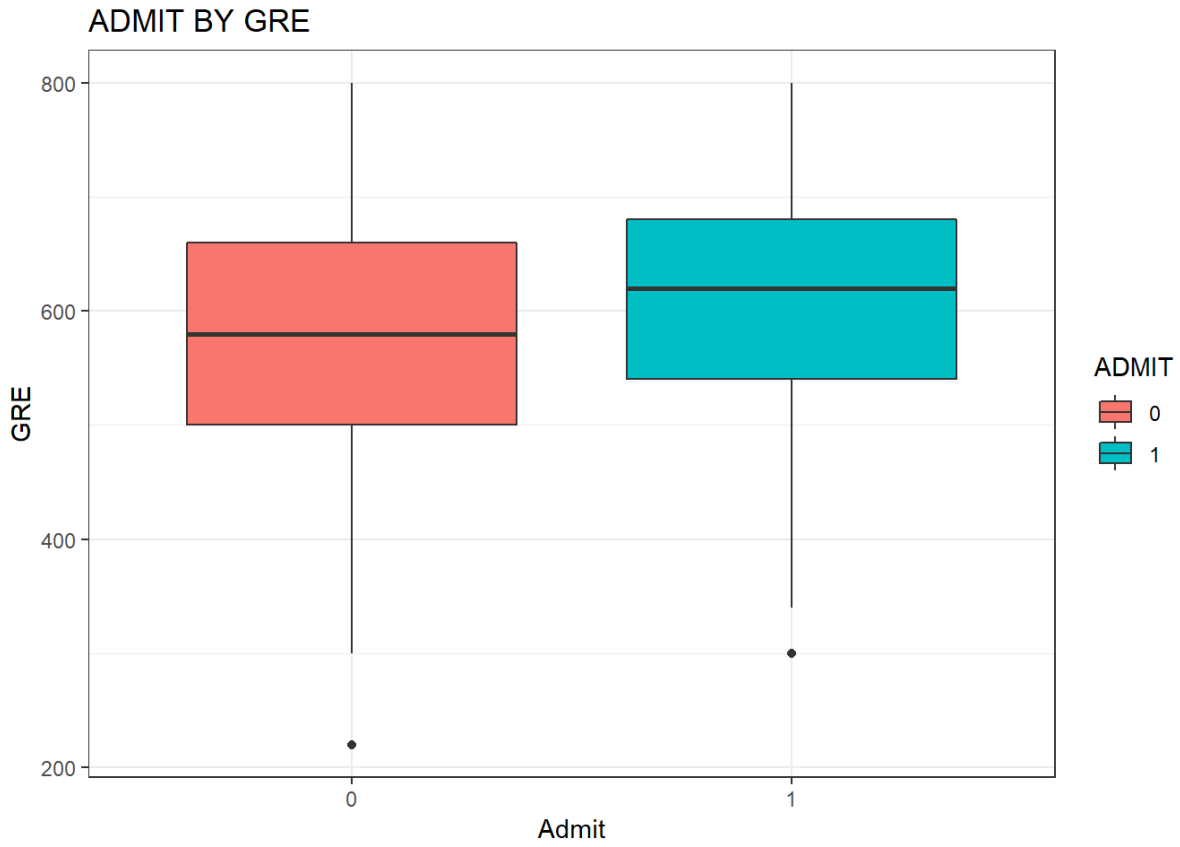
- **Example of Logistic regression using R programming**

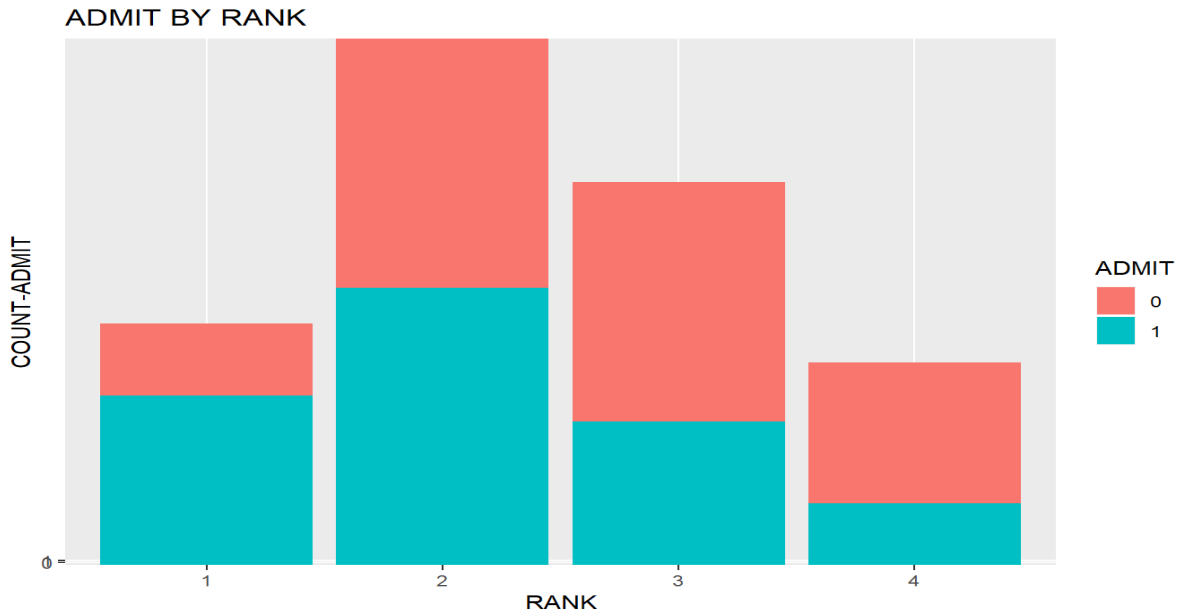
This data set has a binary response (outcome, dependent) variable called admit, which is equal to 1 if the individual was admitted to graduate school, and 0 otherwise. There are three predictor variables: gre, gpa, and rank. We will treat the variables gre and gpa as continuous. The variable rank takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.

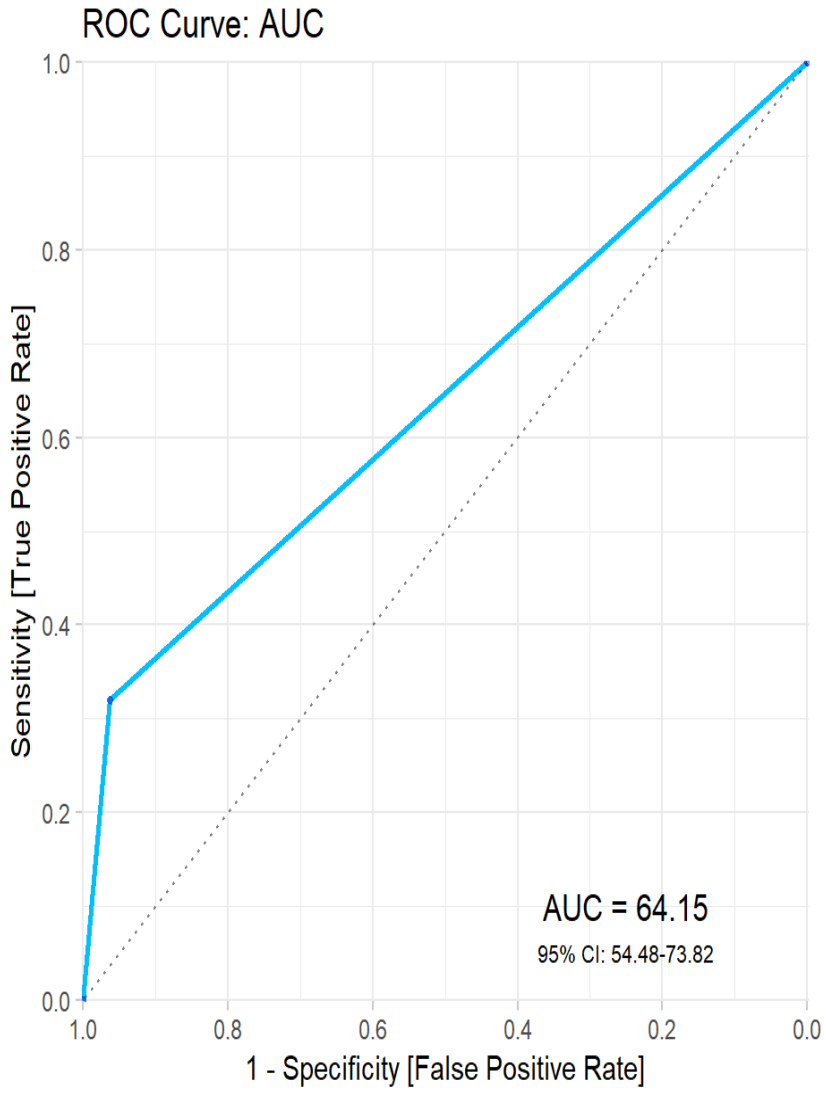
- **Experimental Results**




ADMIT BY GPA
4.0
3.5
3.0
2.5
GPA
0
1
Admit
ADMIT
0
1


ADMIT BY GRE
800
600
400
200
GRE
0
1
Admit
ADMIT
0
1


ADMIT BY RANK
COUNT-ADMIT
0
1
2
3
4
RANK
ADMIT
0
1


ROC Curve: AUC
Sensitivity [True Positive Rate]
1.0
0.8
0.6
0.4
0.2
0.0
AUC = 64.15
95% CI: 54.48-73.82
1.0
0.8
0.6
0.4
0.2
0.0
1 - Specificity [False Positive Rate]

K-Means Clustering

K-Means Clustering is an Unsupervised Learning algorithm which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabelled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K centre points or centroids by an iterative process.
- Assigns each data point to its closest k-centre. Those data points which are near to the particular k-centre, create a cluster.

Hence each cluster has data points with some commonalities, and it is away from other clusters.

- **Example of K-Mean Clustering using python**

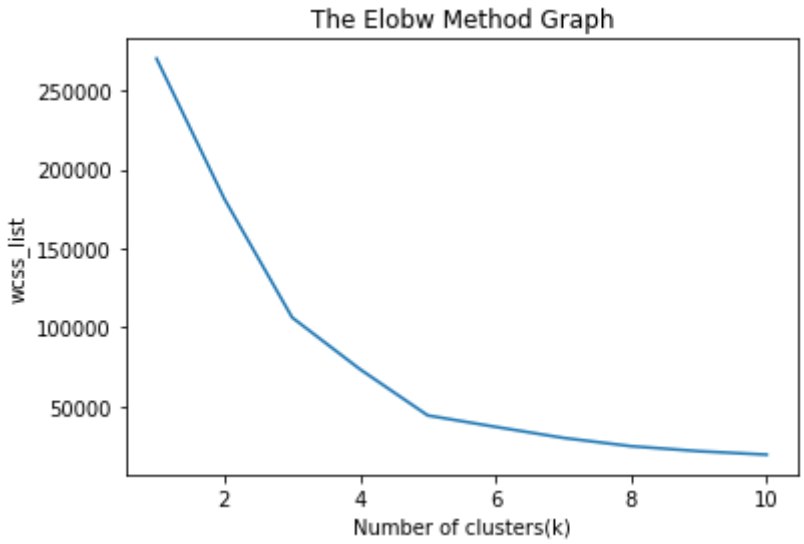
Mall Customer Prediction

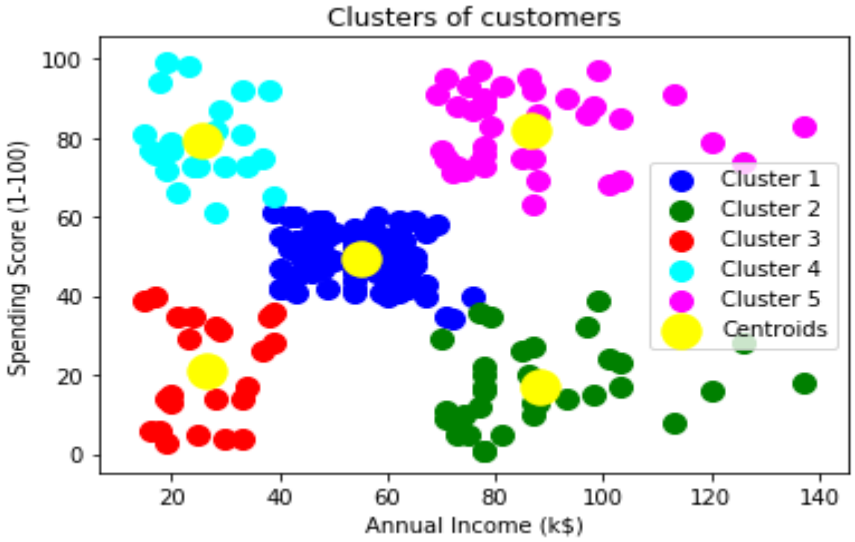
We have a dataset of **Mall_Customers**, which is the data of customers who visit the mall and spend there.

In the given dataset, we have **Customer_Id, Gender, Age, Annual Income ($), and Spending Score** (which is the calculated value of how much a customer has spent in the mall, the more the value, the more he has spent). From this dataset, we need to calculate some patterns, as it is an unsupervised method, so we don't know what to calculate exactly.

The steps to be followed for the implementation are given below:

- **Data Pre-processing**
- **Finding the optimal number of clusters using the elbow method**
- **Training the K-means algorithm on the training dataset**
- **Visualizing the clusters**.
- **Experimental Results**



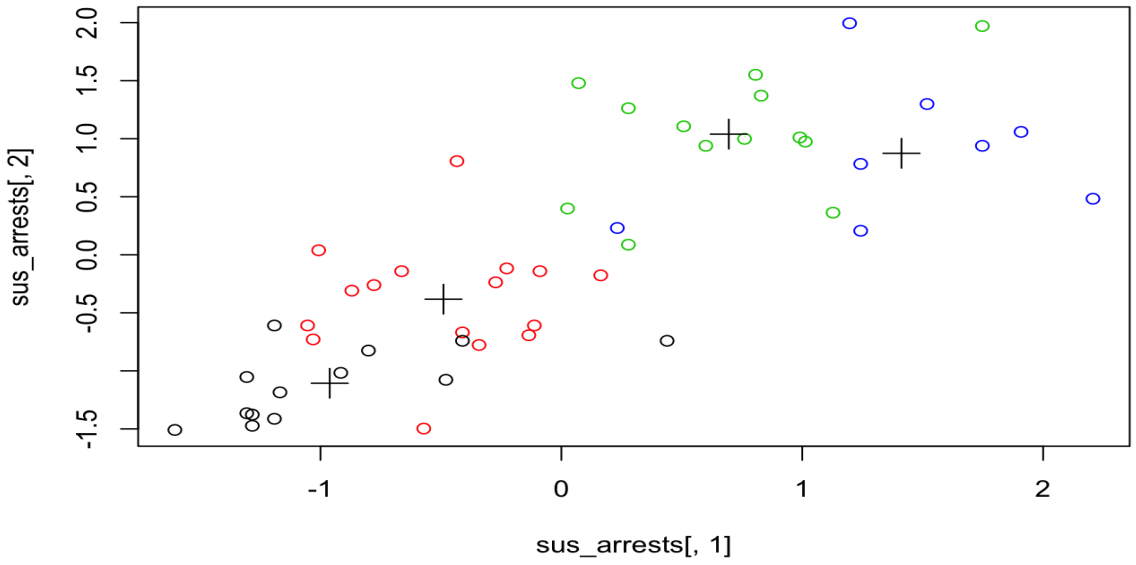


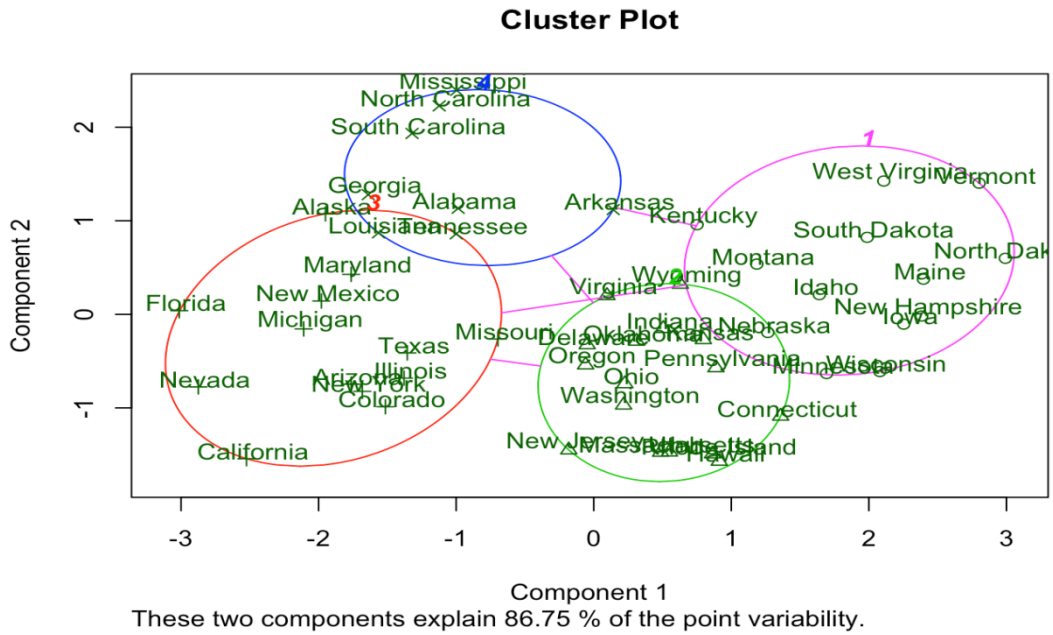
- **Example of K-Mean Clustering using R programming**

US Arrests Prediction

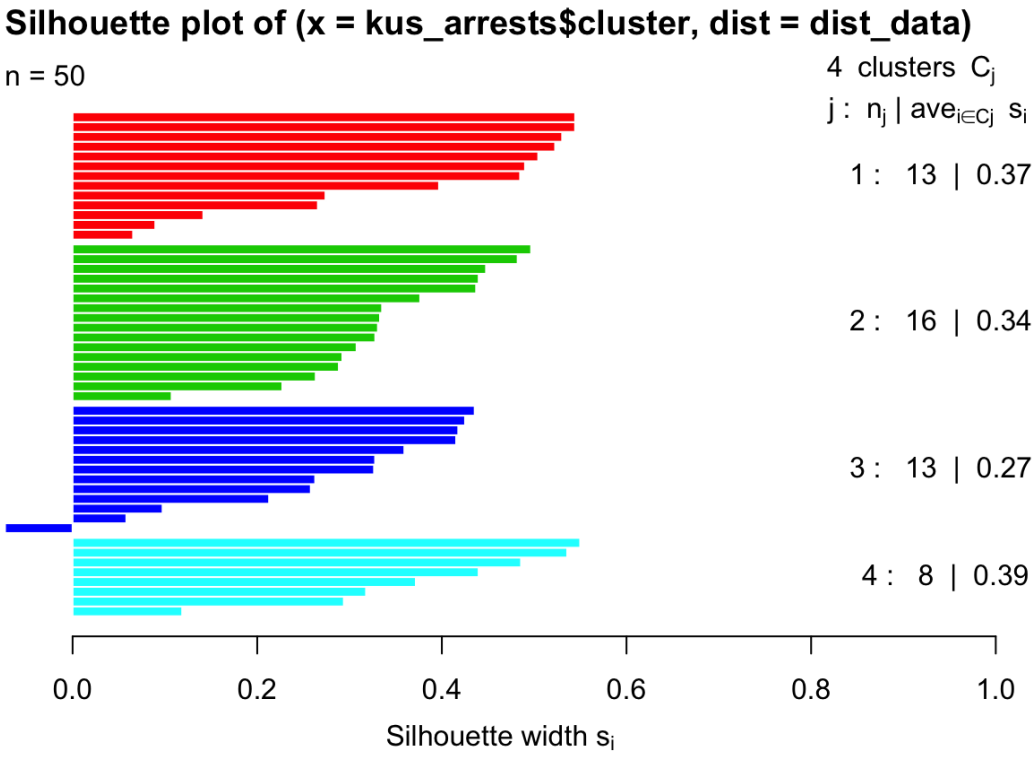
Here, we'll use the built-in R data set US Arrests, which contains statistics in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. It includes also the present of the population living in urban areas

- **Experimental Result**

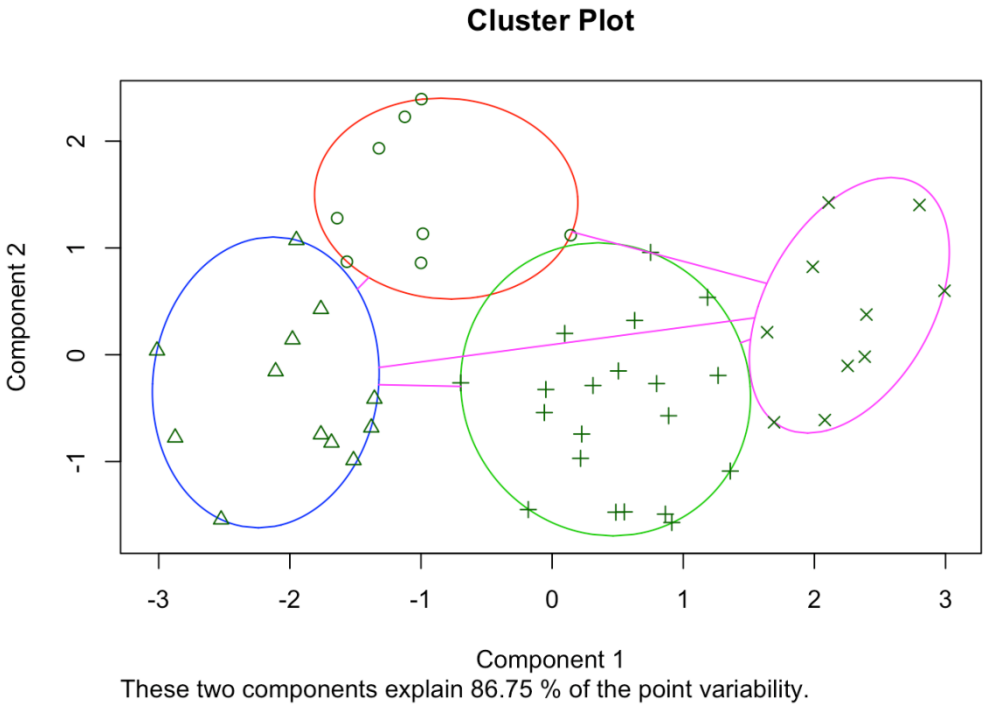




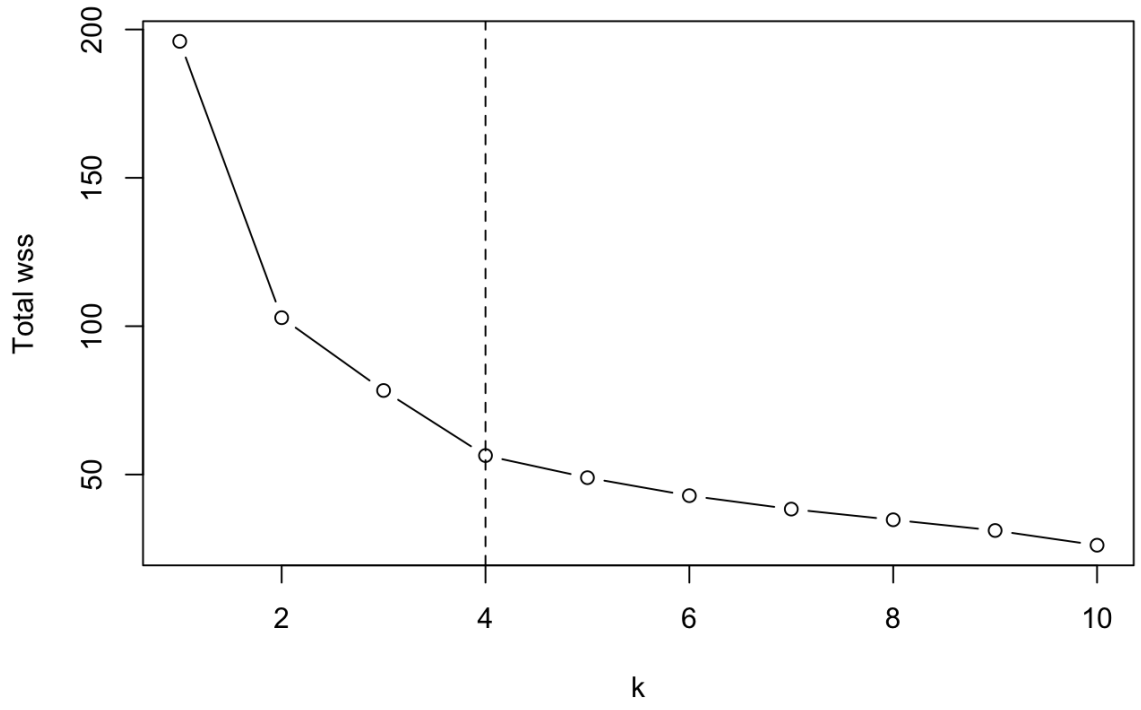
These two components explain 86.75 % of the point variability.

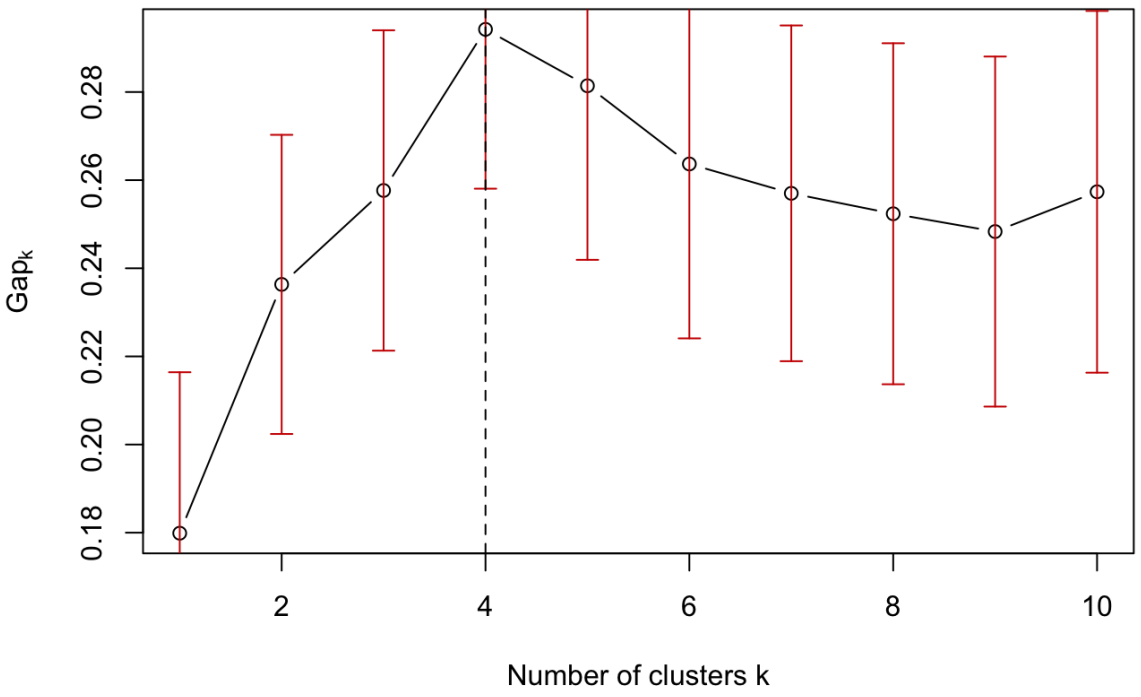


Average silhouette width : 0.34



These two components explain 86.75 % of the point variability.


Total wss
200
150
100
50
2
4
6
8
10
k


Gap$_k$
0.18
0.20
0.22
0.24
0.26
0.28
2
4
6
8
10
Number of clusters k