INDEX

|  |  |
| --- | --- |
| **Prac.**  **No.** | **Practical** |
| **1** | **Install, configure and run Hadoop and HDFS** |
| **2** | **Implement Decision tree classification techniques** |
| **3** | **Classification using SVM** |
| **4** | **Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python** |
| **5** | **Write Program Naive baye's theorem’s** |
| **6** | **Write a Program showing implementation of Regression**  **model.** |
| **7** | **Write a Program showing clustering.** |

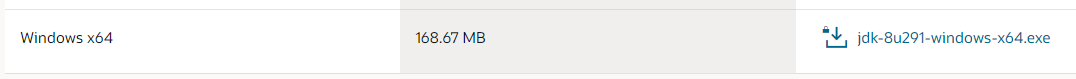
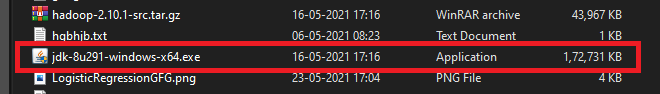
# PRACTICAL NO : 1

**Aim: Install, configure and run Hadoop and HDFS**

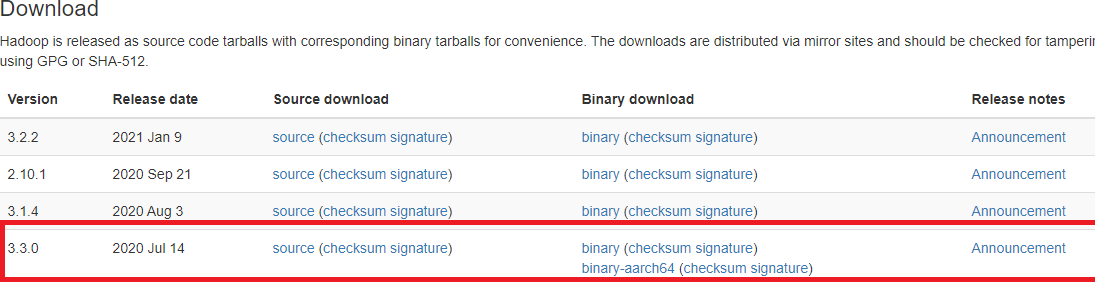
# Description:

Hadoop Installation.

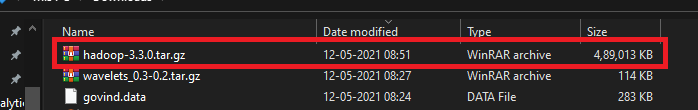
Step 1: downlaod java jdk first .the package size 168.67MB



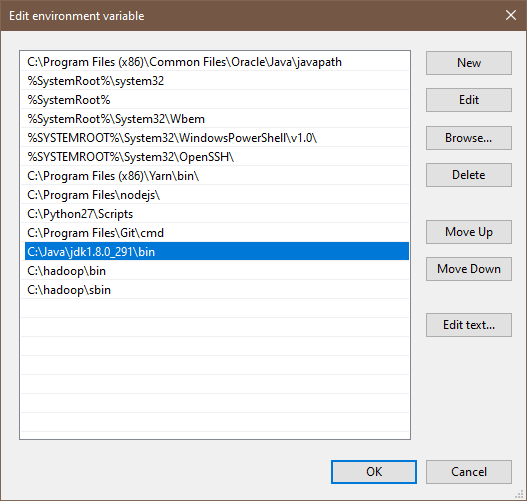
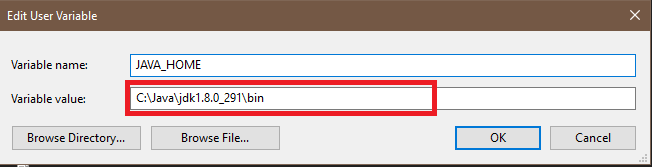
Step 2: download Hadoop binaries from the official website. The binary package size is about 342 MB.



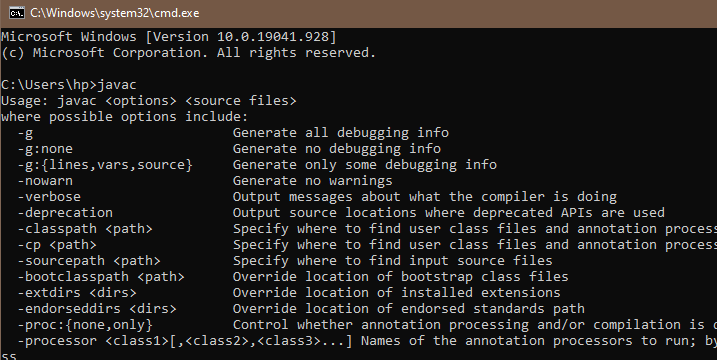
Step 3: After finishing the file download, we should unpack the package using 7zip int two steps. First, we should extract the hadoop-3.2.1.tar.gz library, and then, we should unpack the extracted tar file:

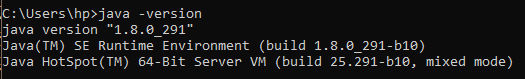


Step 4: When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.

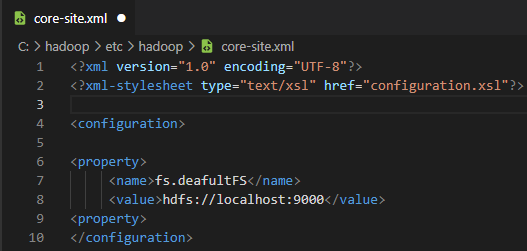


Step 5: Check the version of java



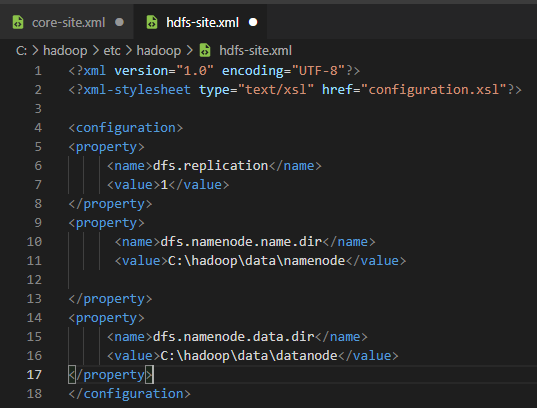


Step 6: Configuration core-site.xml

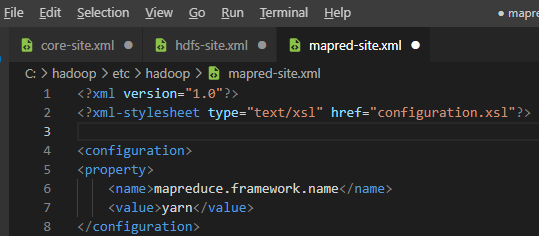


Step 7: Configuration core-site.xml



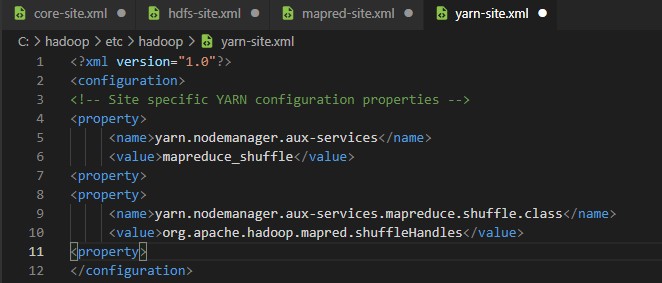


Step 8: Configuration core-site.xml

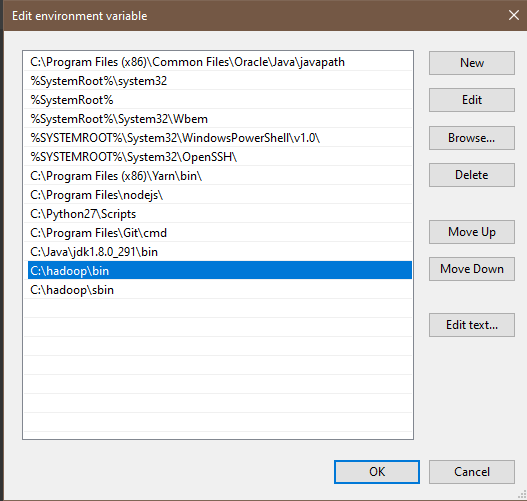


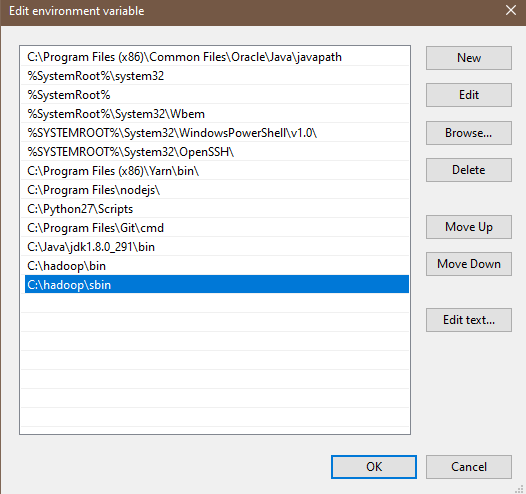
Step 9: Configuration core-site.xml



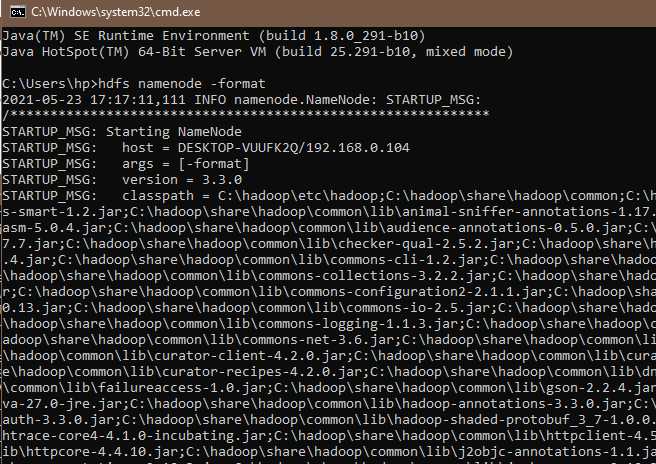


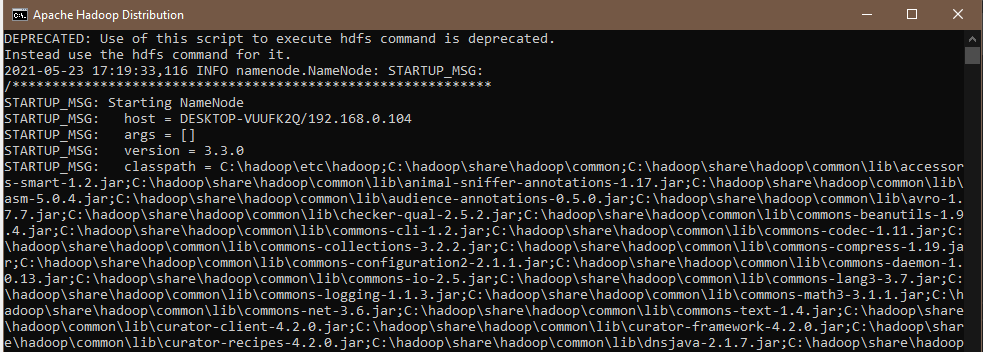
Step 10: When the “Advanced system settings” dialog appears, go to the “Advanced” tab and click on the “Environment variables” button located on the bottom of the dialog.

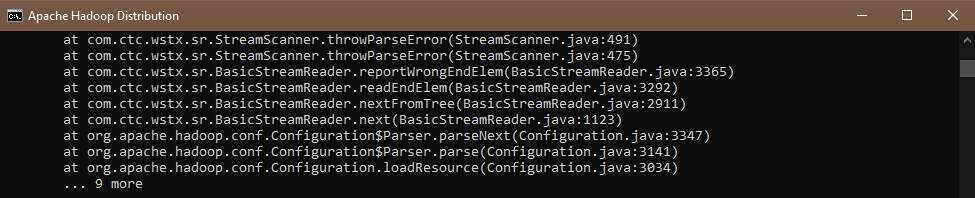




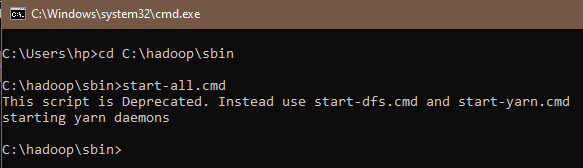
Step 11: let’s check Hadoop install Successfully







Step 12: Let check bin



# PRACTICAL NO : 2

**Aim: Implement Decision tree classification techniques**

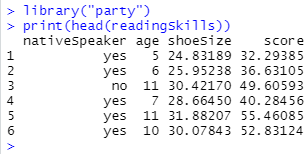
# Description:

Decision tree builds classification or regression 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**

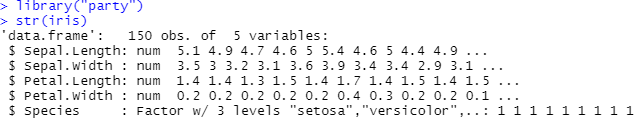
**Step 1: The package "party" has the function ctree() which is used to create and analyze decison tree.**

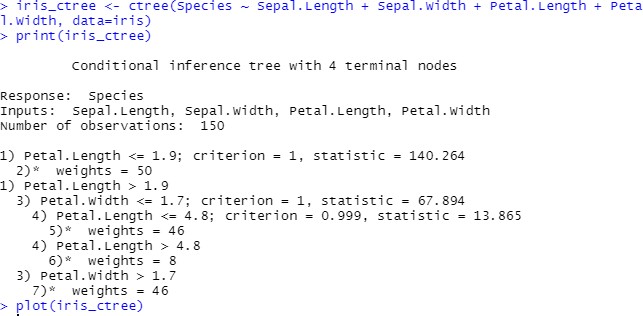
****

**Step 2: Load the party package. It will automatically load other# dependent packages Print some records from data set readingSkills.**

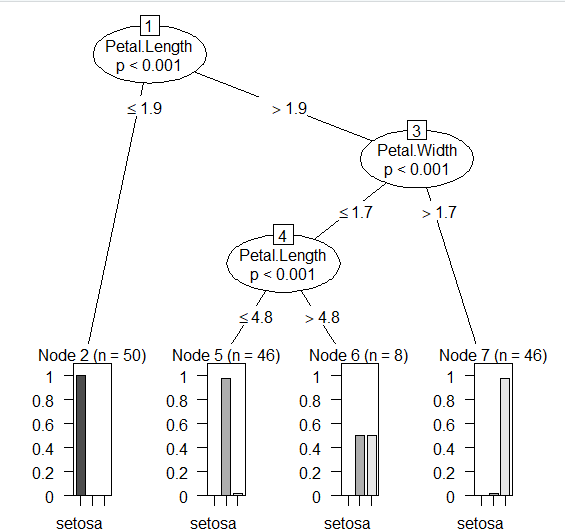
****

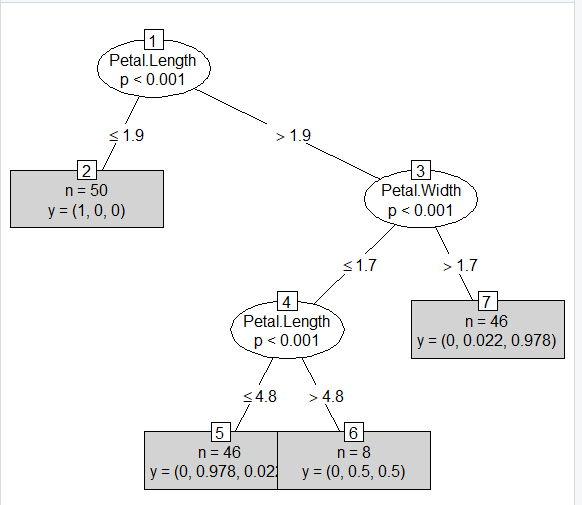
**Step 3 : Call function ctree to build a decision tree. The first parameter is a formula, which defines a target variable and a list of independent variables.**

****



# Output :

****



**PRACTICAL NO : 3**

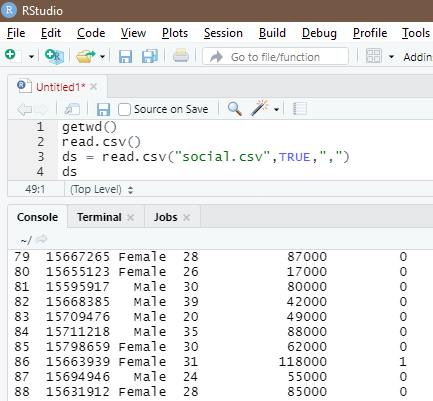
**Aim: Classification using SVM**

# Description:

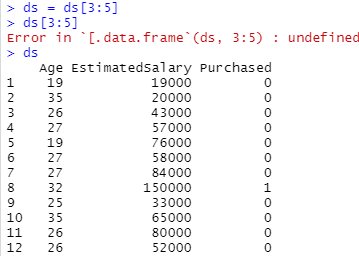
A support vector machine (SVM) is a supervised [machine learning](https://monkeylearn.com/machine-learning/) model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they’re able to categorize new text

The implementation is explained in the following steps:

**Step 1: Importing the dataset**

****

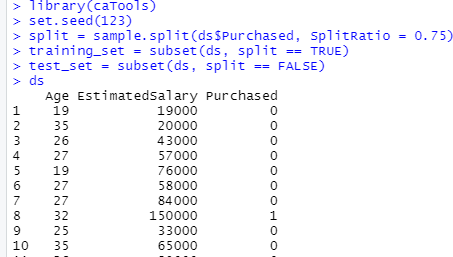
**Step 2: Selecting columns 3-5**



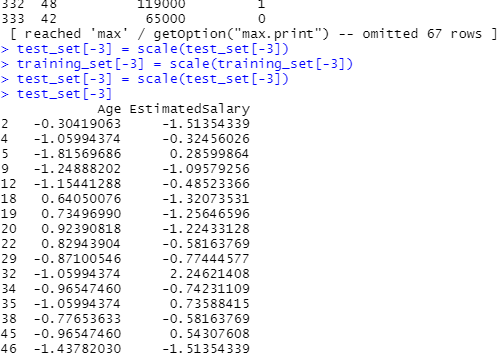
**Step 3: install package**

****

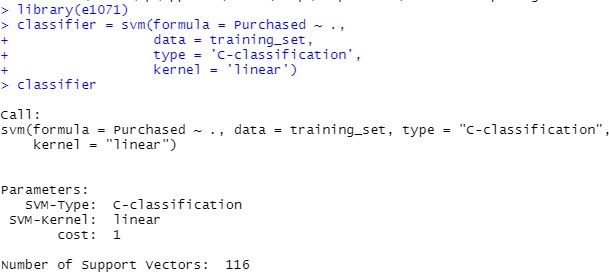
**Step 4: Splitting the dataset**

****

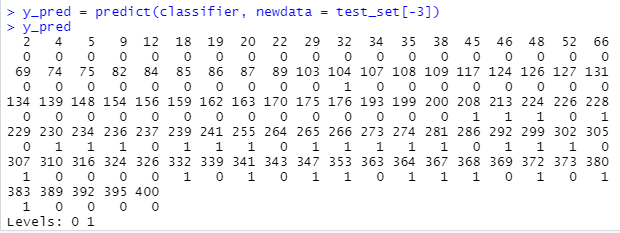
**Step 5: Feature Scaling**

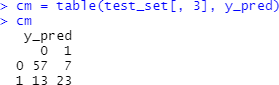


**Step 6: Fitting SVM to the training set**

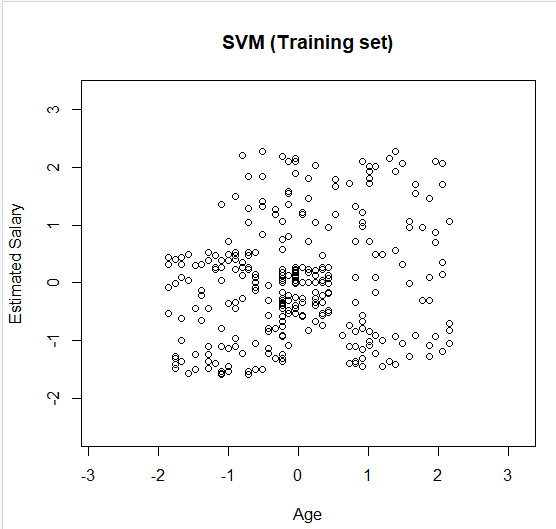
****

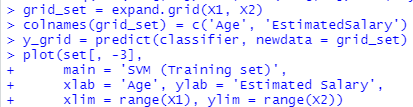
**Step 7: Predicting the test set result**

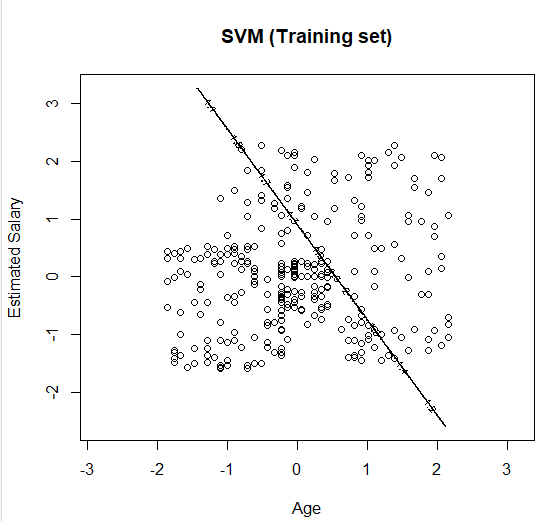


****

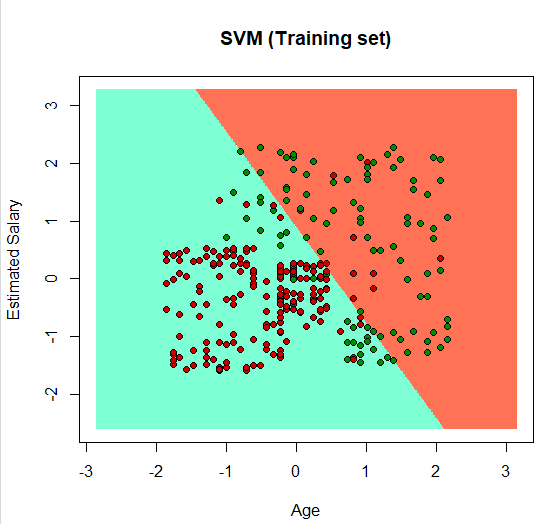
**Step 8: Visualizing the Training set results**

****



****

**Output:**

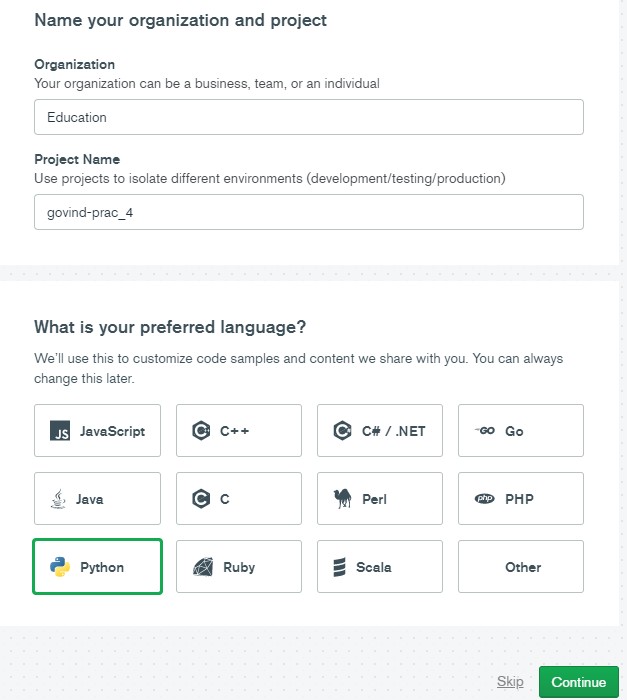


# PRACTICAL NO : 4

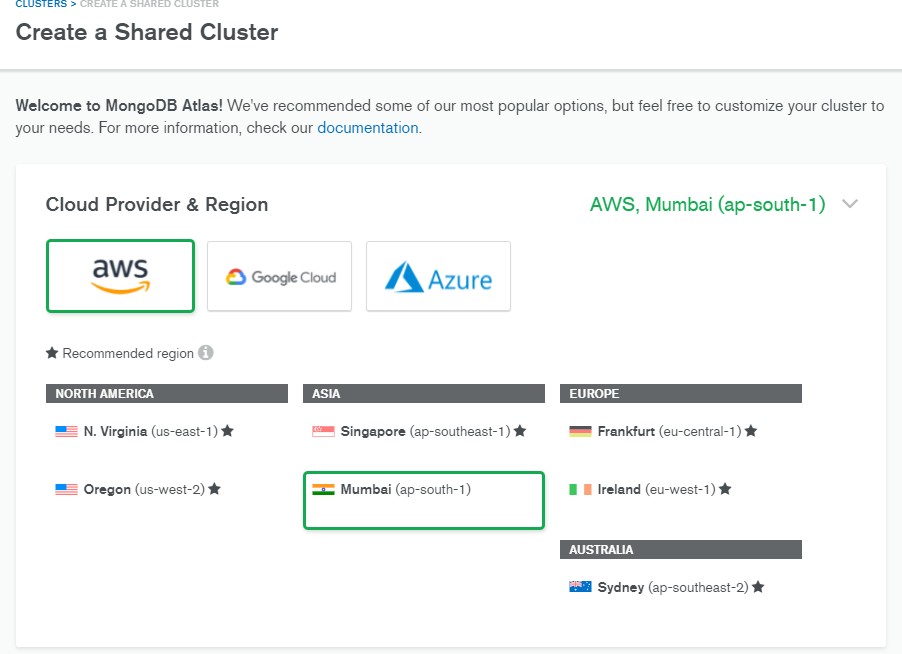
**Aim: Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python**

# Description:

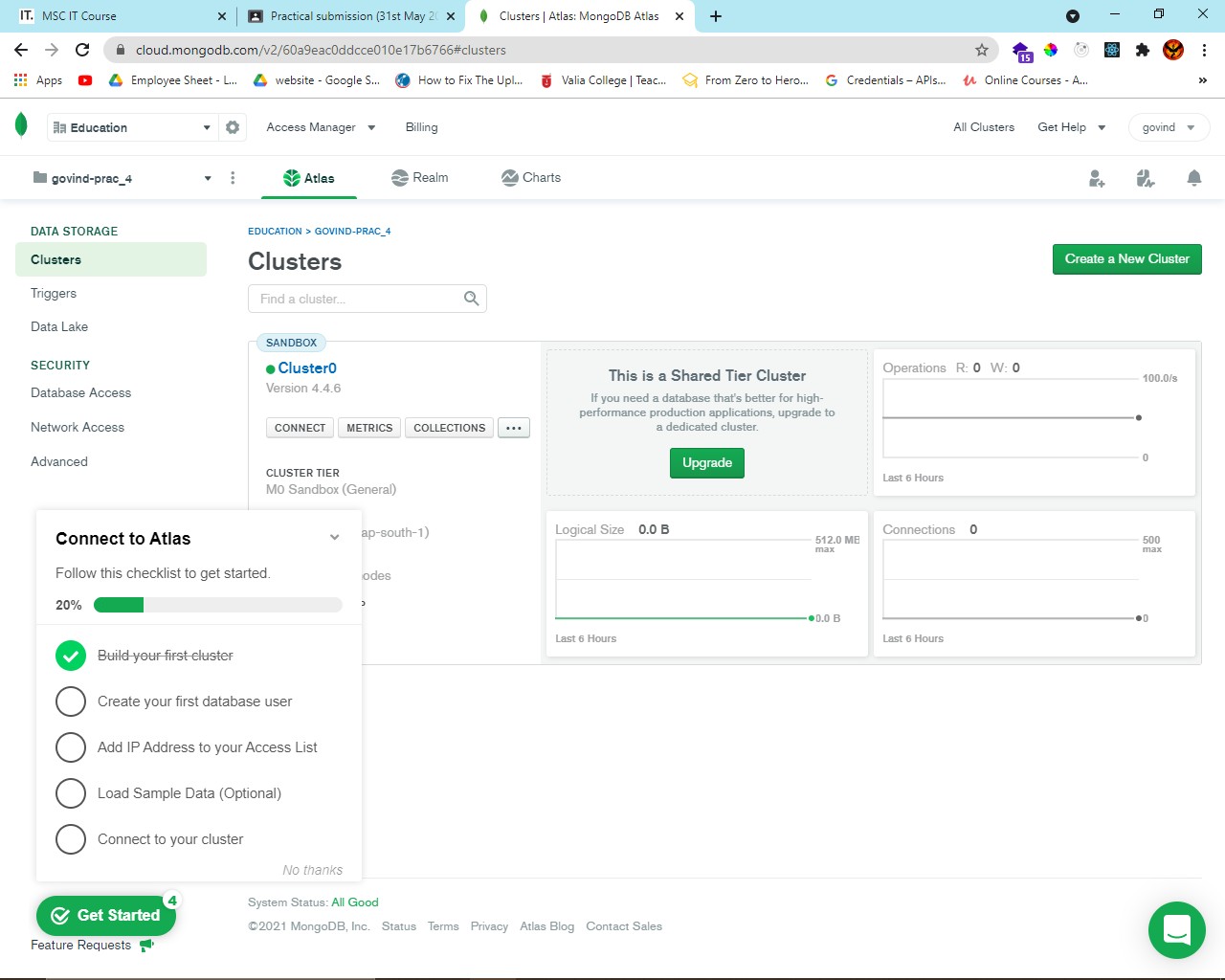
MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License



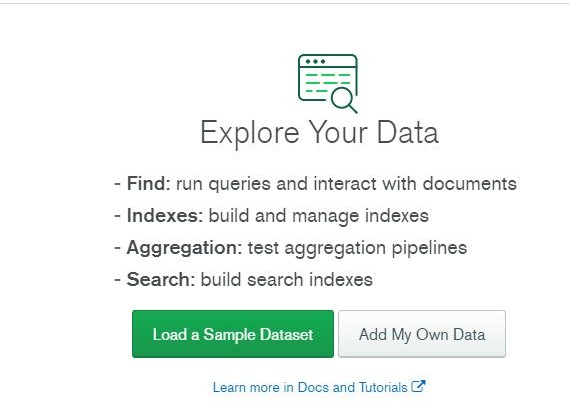
**Step 1 : Sign up and create a cluster.**



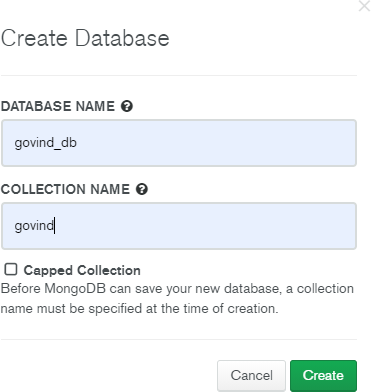
This is the home page of mongoDB Atlas.



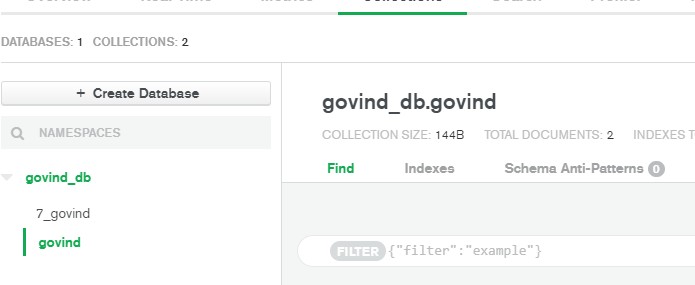
**Step 2 : Click on collections to create and view existing databases.**

****

**Step 3 : Click on ‘Add My Own Data’ to create a database.**

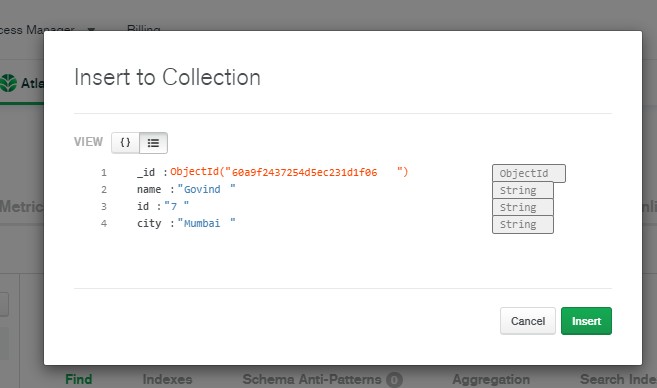
****

**Step 4 : Click on insert document to add records.**

****

Since MongoDB is a No-SQL database, so you can add ‘n’ number of columns for any

row/record.



Perform updating data



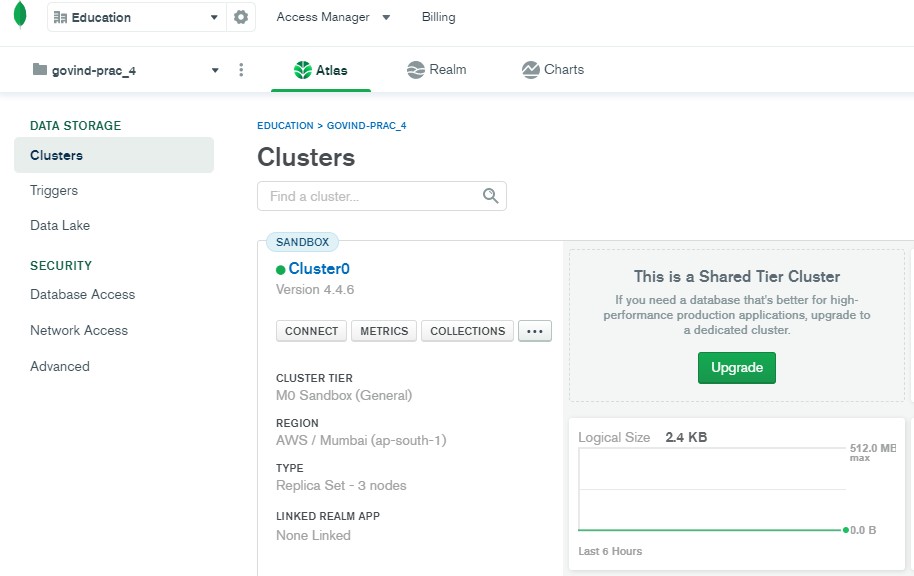
Performing deleting data



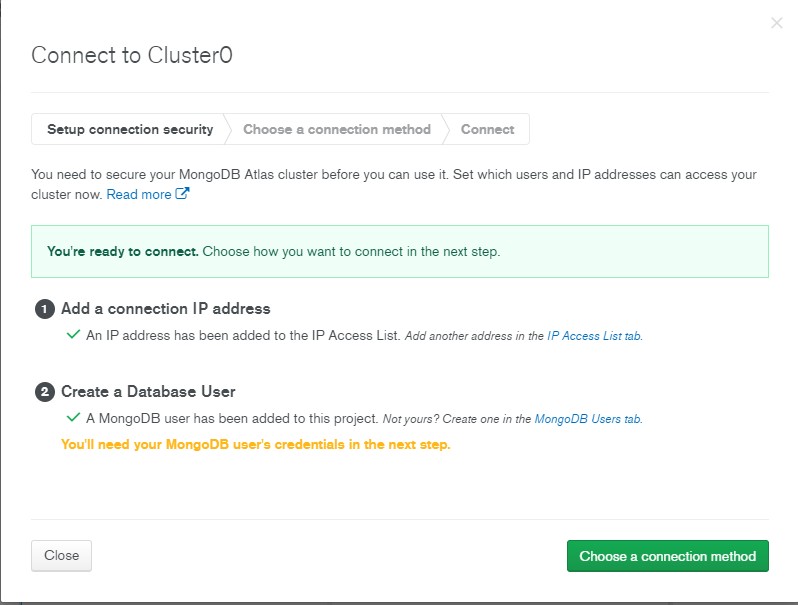
**Performing Insert data**



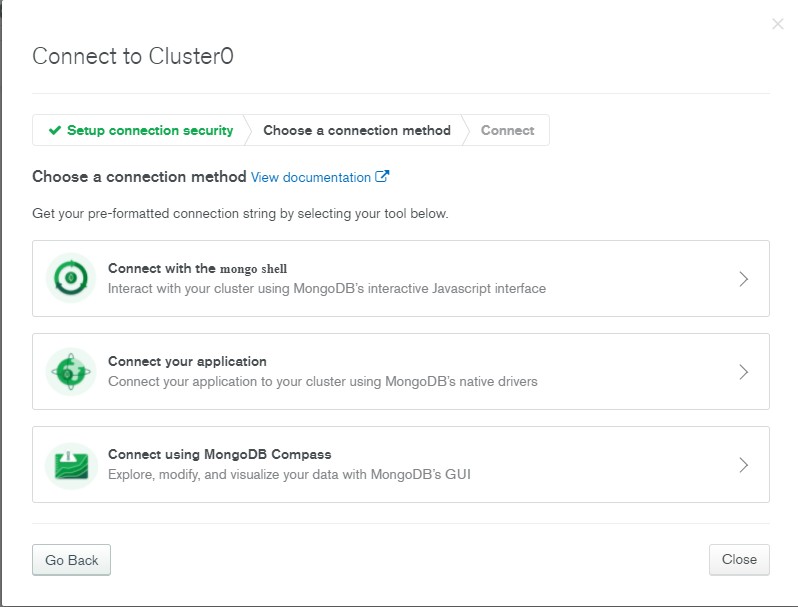
**Step 5 : To start with the connection click on Overview, and then click on Connect.**

****

**Step 6 : Select on add your current IP and create a MongoDB user.**

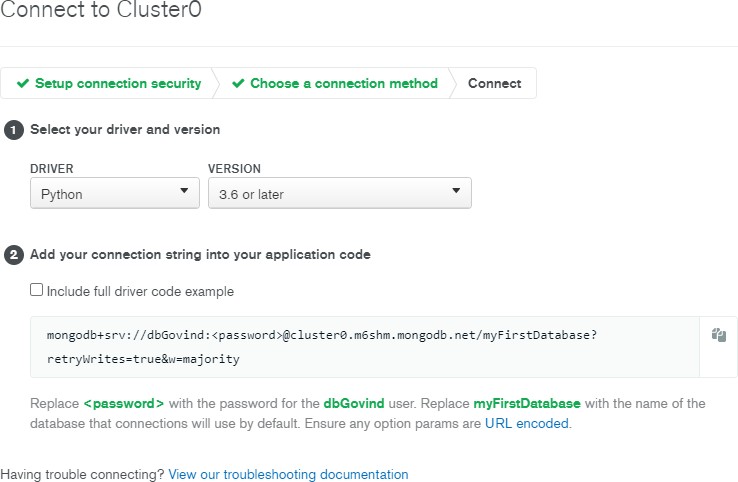


**Step 7 : Click on ‘Connect your application’.**

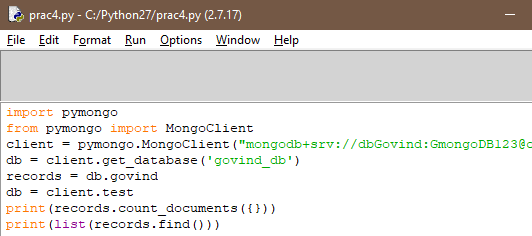
****

**Step 8 : Select the driver as ‘Python’ and version as ‘3.6 or later’. (Select the version as**

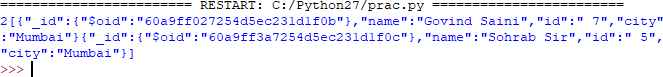
**3.6 or later only if your Python’s version is 3.6 or later.)**



**Step 9 : Write the code given below in a Python file.**

****

# Output :

****

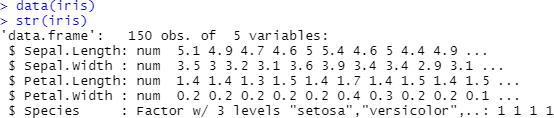
**PRACTICAL NO : 5**

**Aim: write program in R of Naive baye's theorem**

# Description:

Naive Bayes is a Supervised Non-linear classification algorithm in [R Programming](https://www.geeksforgeeks.org/introduction-to-r-programming-language/). Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Baye’s theorem with strong(Naive) independence assumptions between the features or variables

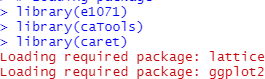
**# Loading data**

****

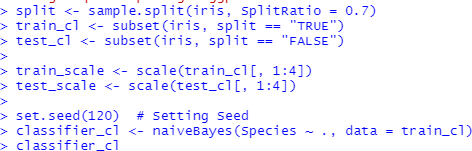
**# Installing Packages**

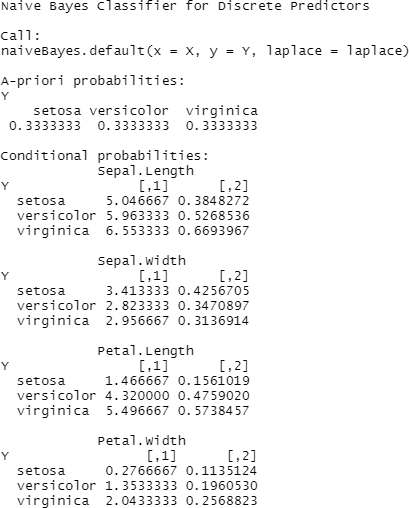
****

**# Loading package**

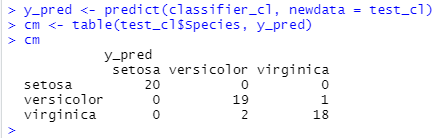
****

**# Splitting data into train and test data**

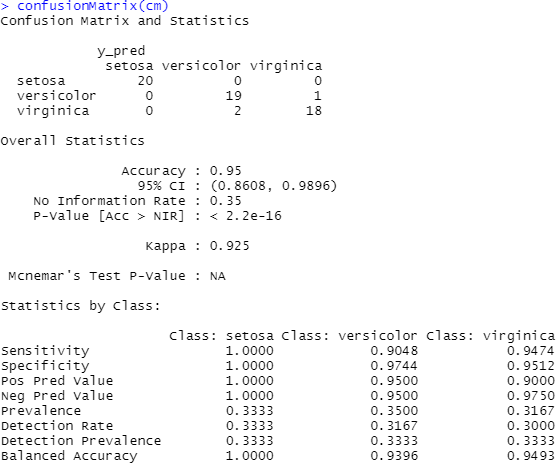


****

**# Predicting on test data'**

****

**# Model Evauation**

****

# PRACTICAL NO : 6

**Aim: Write a Program showing implementation of Regression model.**

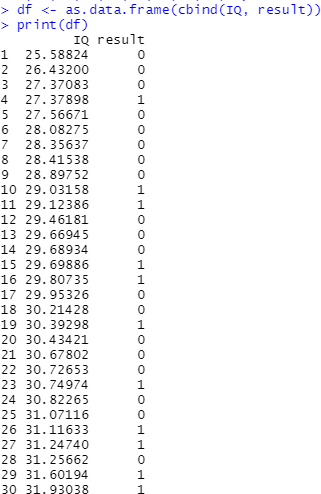
# Description:

Regression is a method to mathematically formulate relationship between variables that in due course can be used to estimate, interpolate and extrapolate. Suppose we want to estimate the weight of individuals, which is influenced by height, diet, workout, etc.

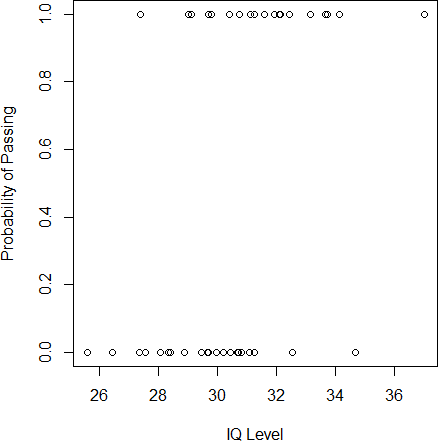
Here, *Weight* is the **predicted** variable

Lets implementation of Regression Model some Example:

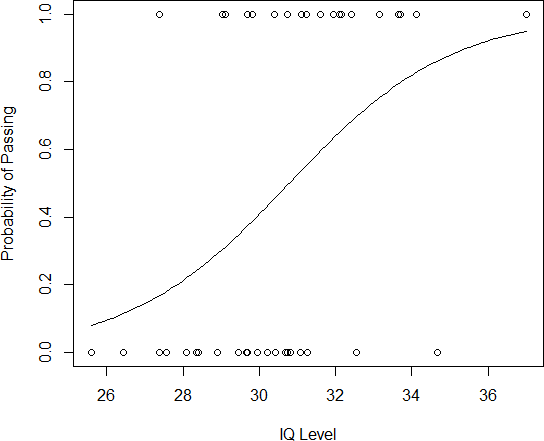


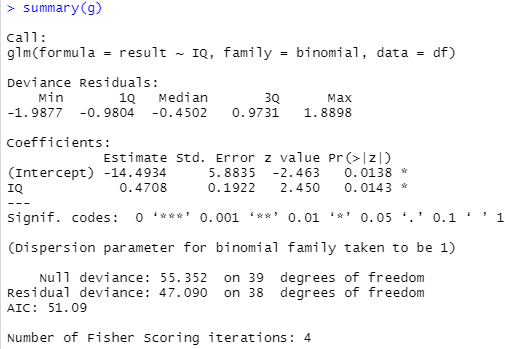












# PRACTICAL NO : 7

**Aim: Write a Program showing clustering.**

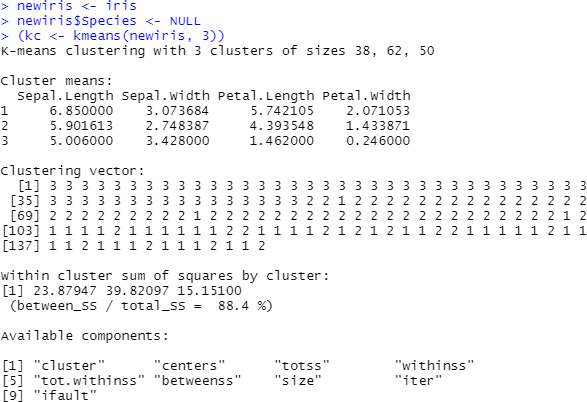
# Description:

**# In this Program we understand about K-Mean Clustering #** **What Does K-Means Clustering Mean?**

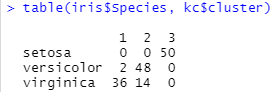
* K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems.
* It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k," which is fixed beforehand.
* The clusters are then positioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted and then the process starts over using the new adjustments until a desired result is reached.

**We Understand in different Steps :**

**Step 1: Apply kmeans to *newiris*, and store the clustering result in *kc*. The cluster number is set to 3.**

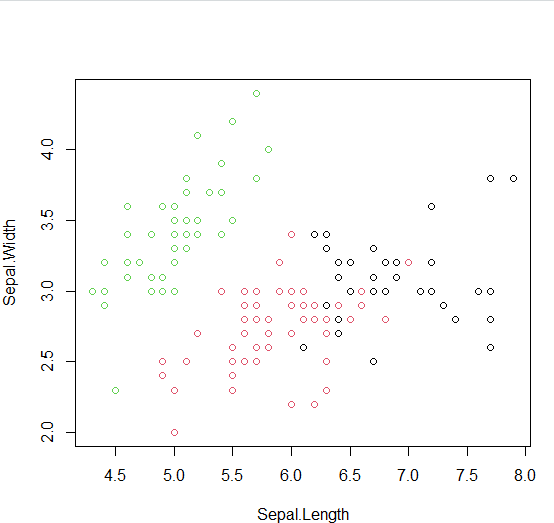


**Step 2: Compare the Species label with the clustering result**

****

**Step 3 : Plot the clusters and their centres. Note that there are four dimensions in the data and that only the first two dimensions are used to draw the plot below.**

****



**Step 4: Some black points close to the green centre (asterisk) are actually closer to the black centre in the four dimensional space.**

****

