MSC CS - I

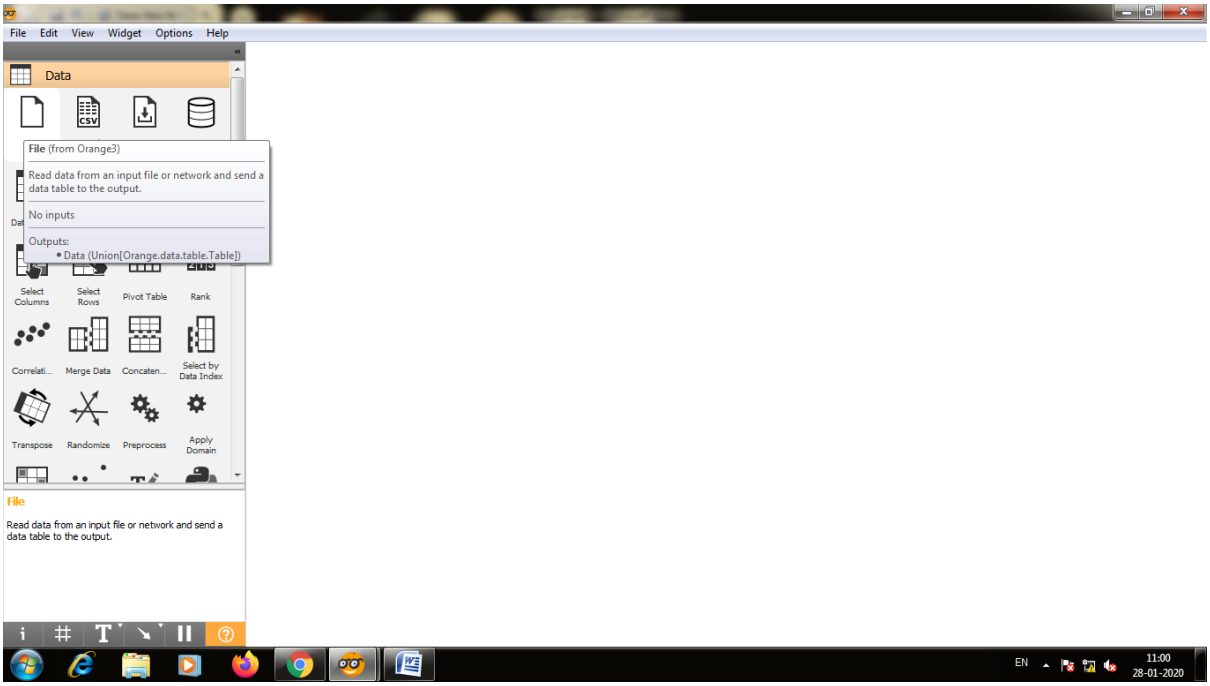
Name: Merin Kurian
Roll No.: 20

BIBD

Practical No.: 1

Aim: Classification using orange tool.

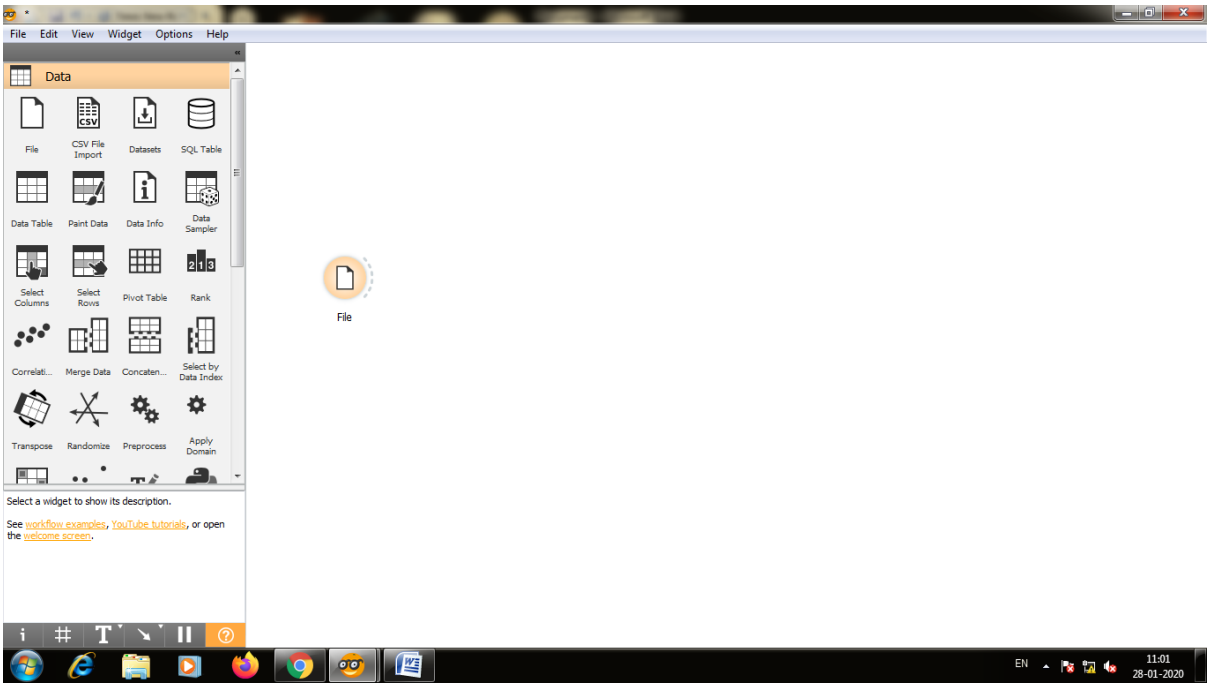
Step1:- We will be taking the available data files in order to predict the future things so open a new data file and drag the file just over it.



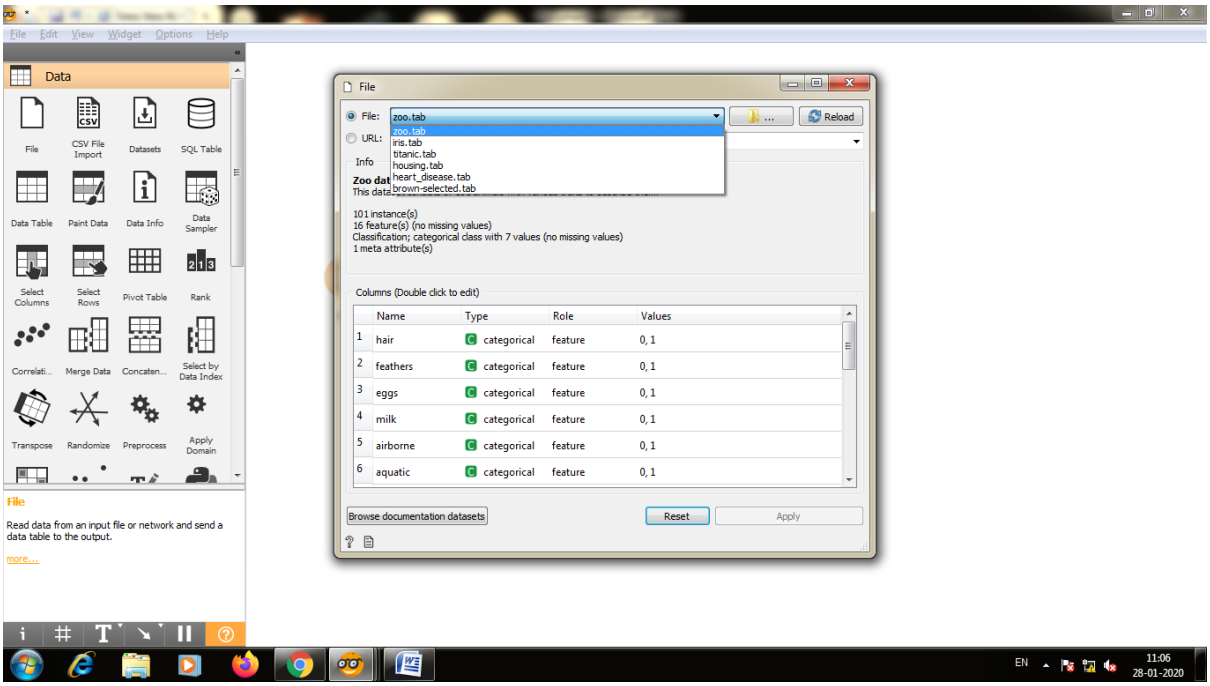
MSC CS - I

Name: Merin Kurian

Roll No.: 20



Step2:-Double click on the data file,from the dropdown list select the zoo.tab data file and close it.

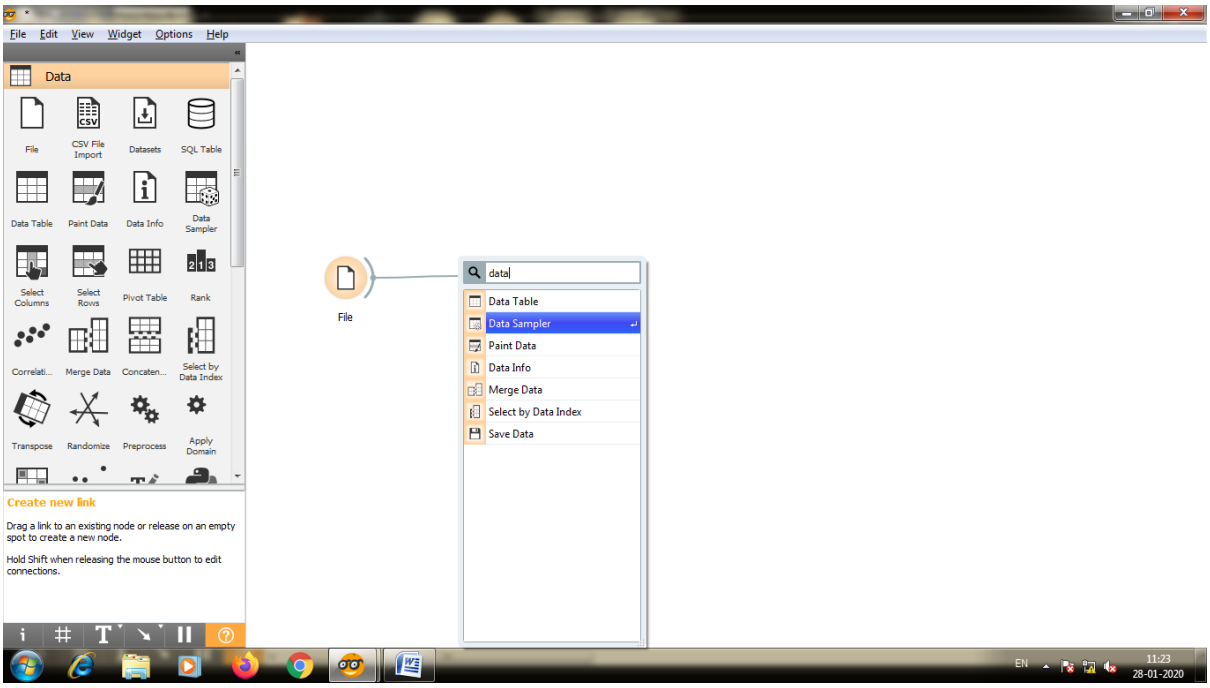


Step3:-Select the file and type data sampler.

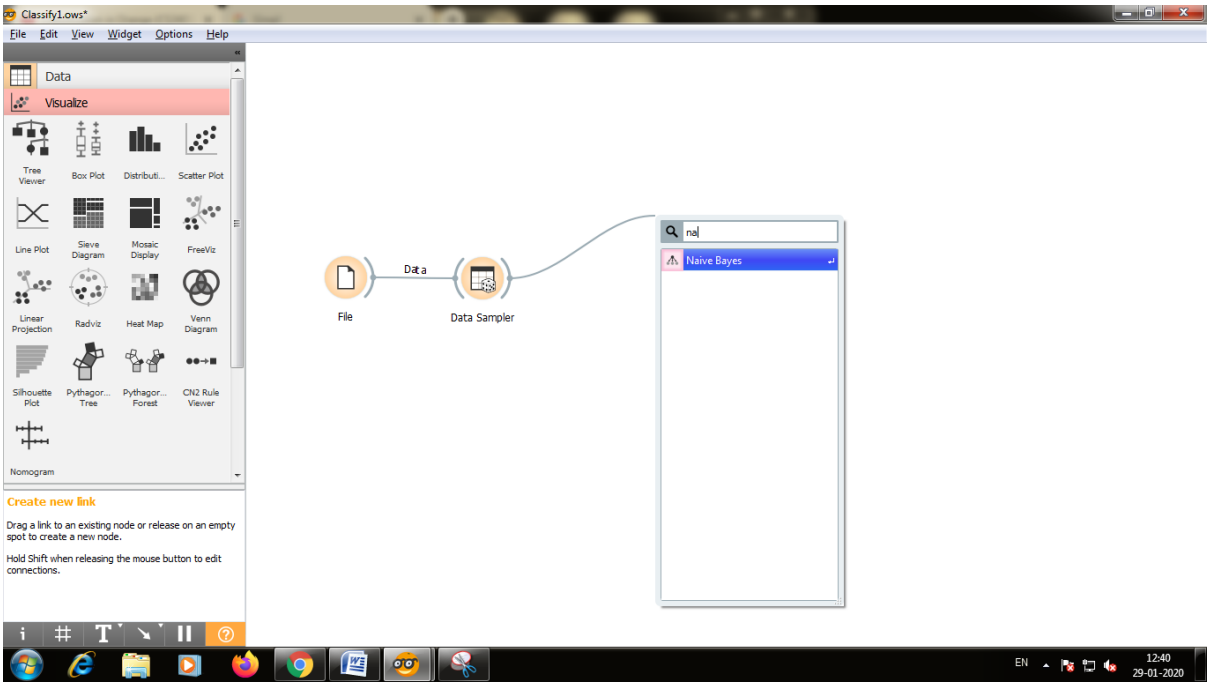
MSC CS - I

Name: Merin Kurian

Roll No.: 20



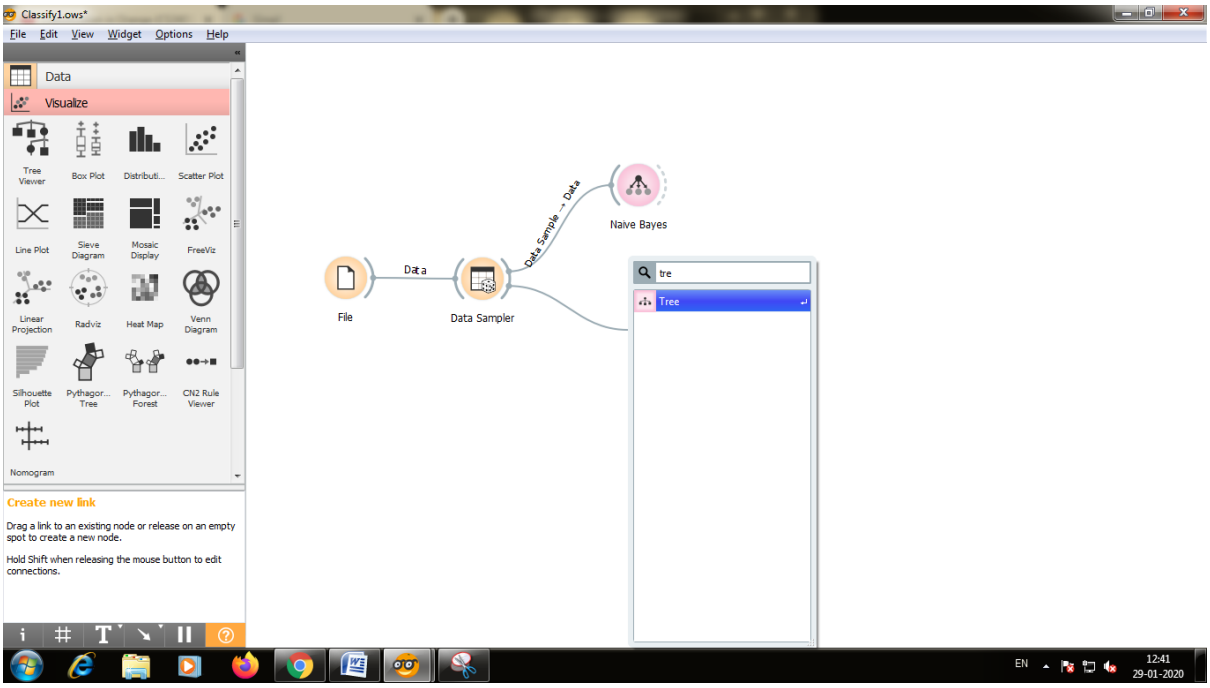
Step4:-Whenever we do prediction we will need to connect them with native bayes and classification tree to the prediction by dragging them across.

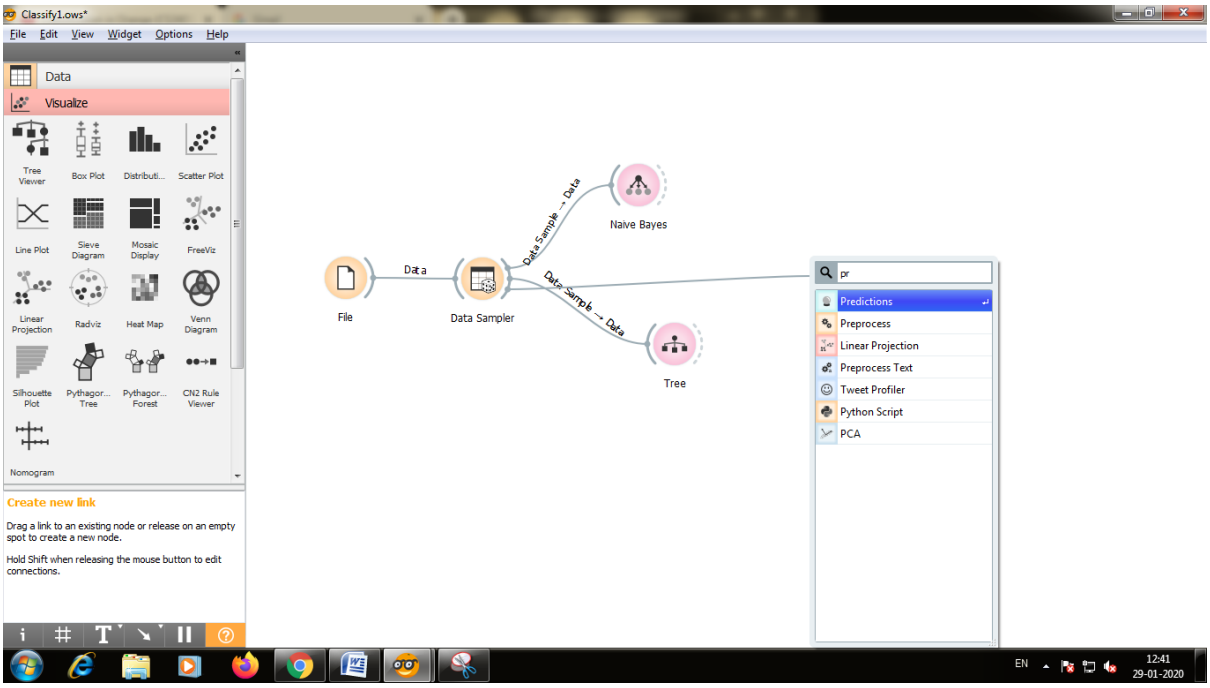


MSC CS - I

Name: Merin Kurian

Roll No.: 20

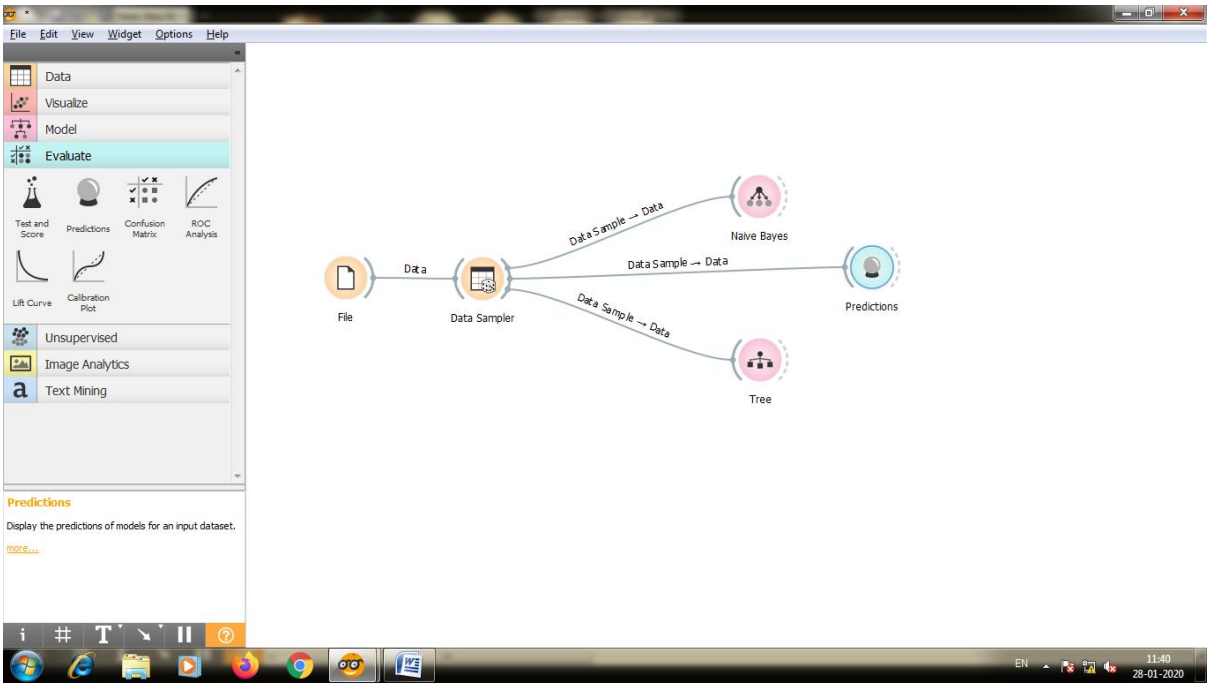




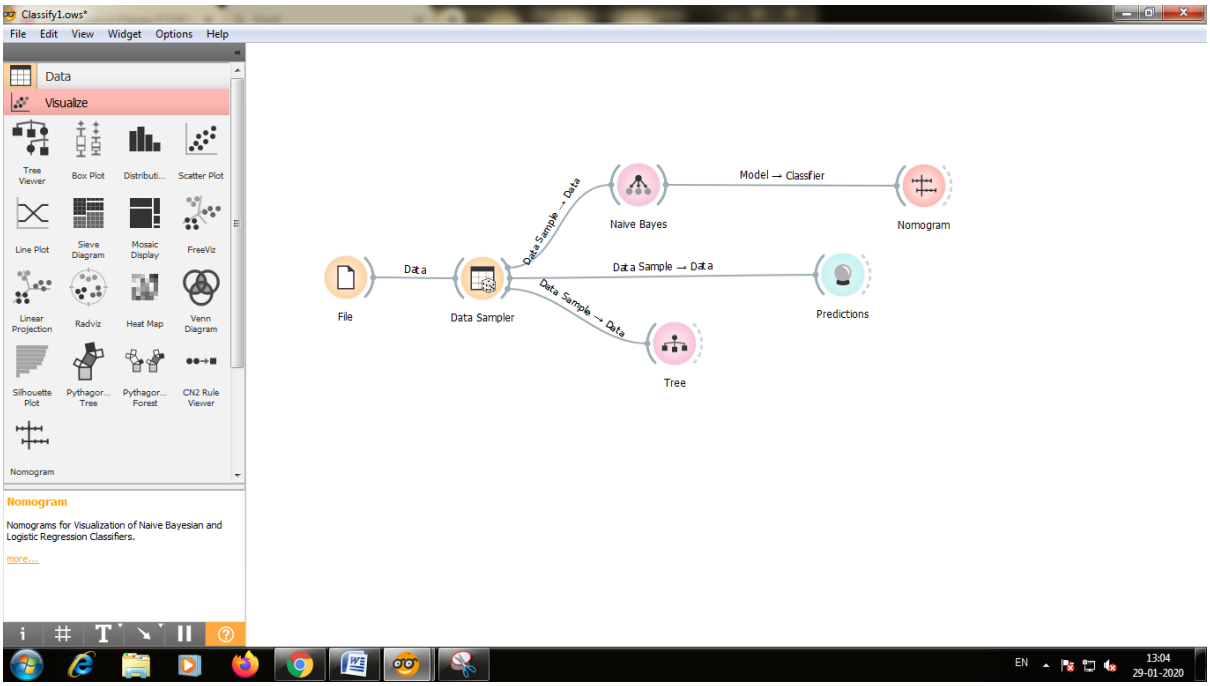
MSC CS - I

Name: Merin Kurian

Roll No.: 20



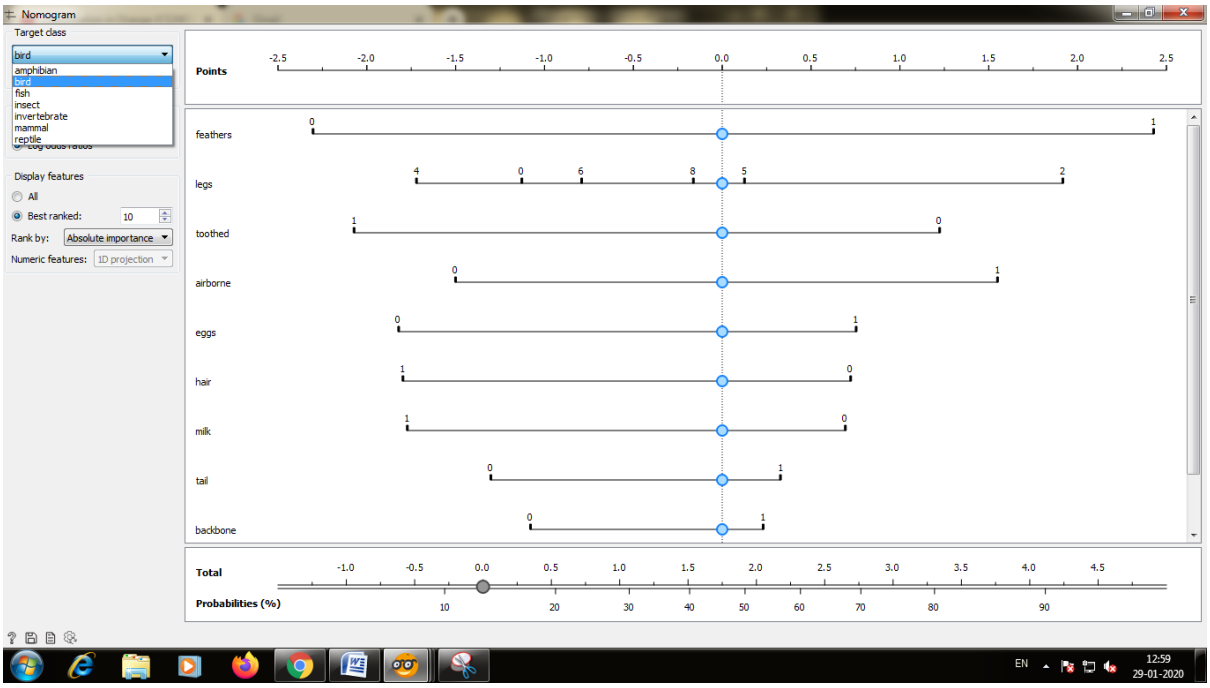
Step5:-The model is also able to show how you can predict the type class of the various attributes so you can use the nomogram for that double click on nomogram and you see the type of target class list so let's see if the you want to see if the data input you can drag these points across by sliding across the data one means yes and zero means no

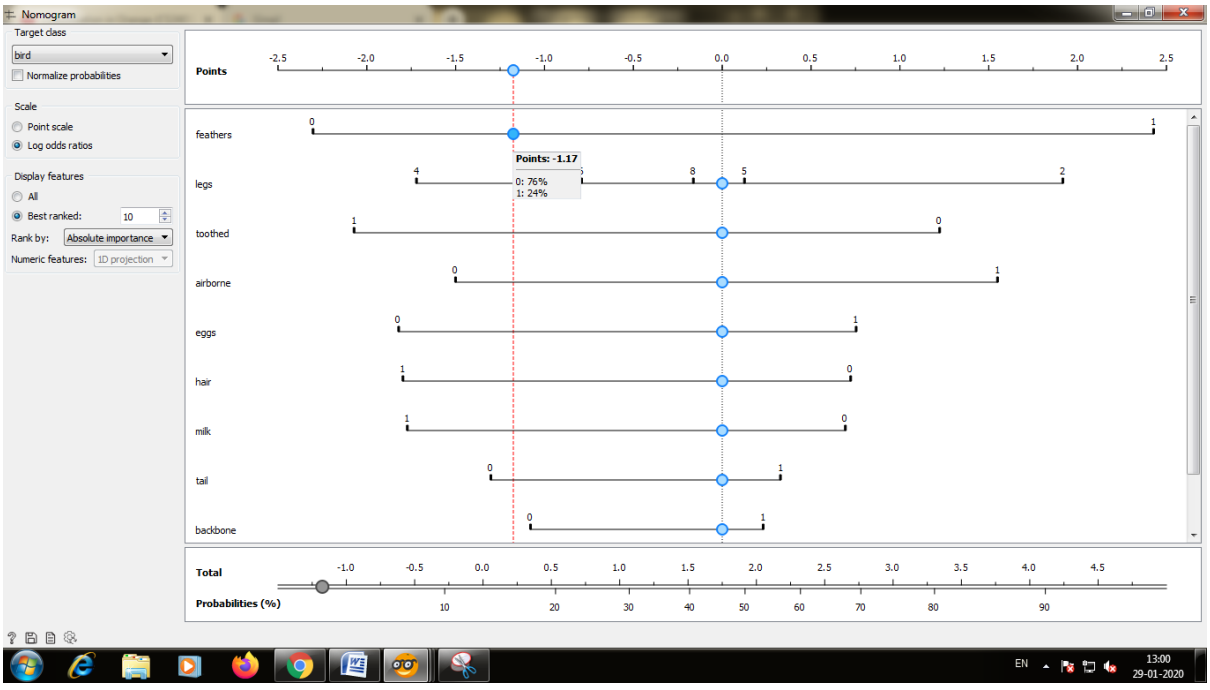


MSC CS - I

Name: Merin Kurian

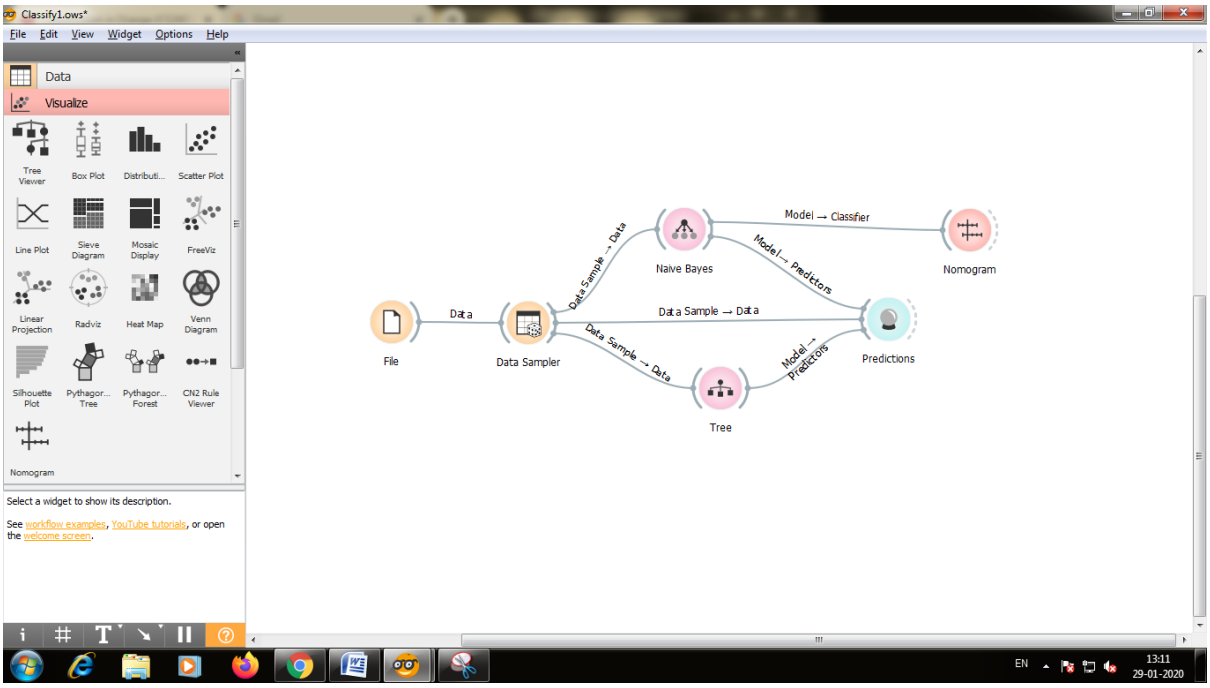
Roll No.: 20



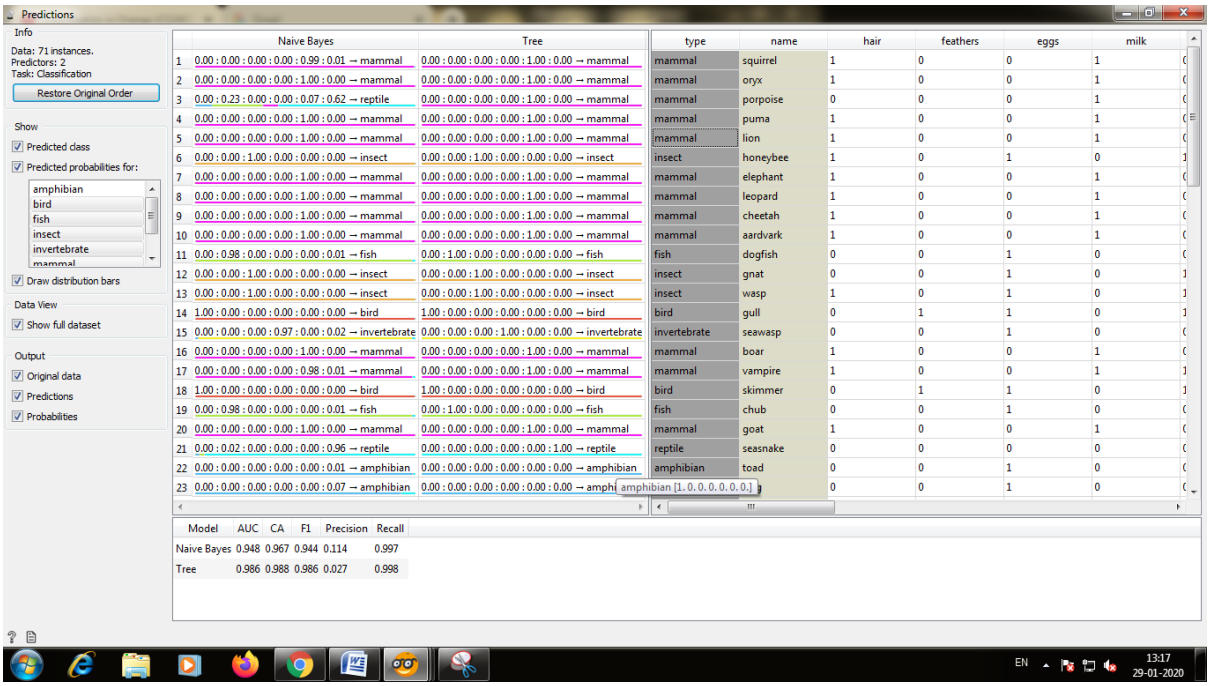


Step6:-Connect the Native Bayes and Classification Tree to Prediction

Name: Merin Kurian
Roll No.: 20



Step7:-Double click on the prediction and you will the data attributes by seeing this input from hair feathers eggs milk and so on.

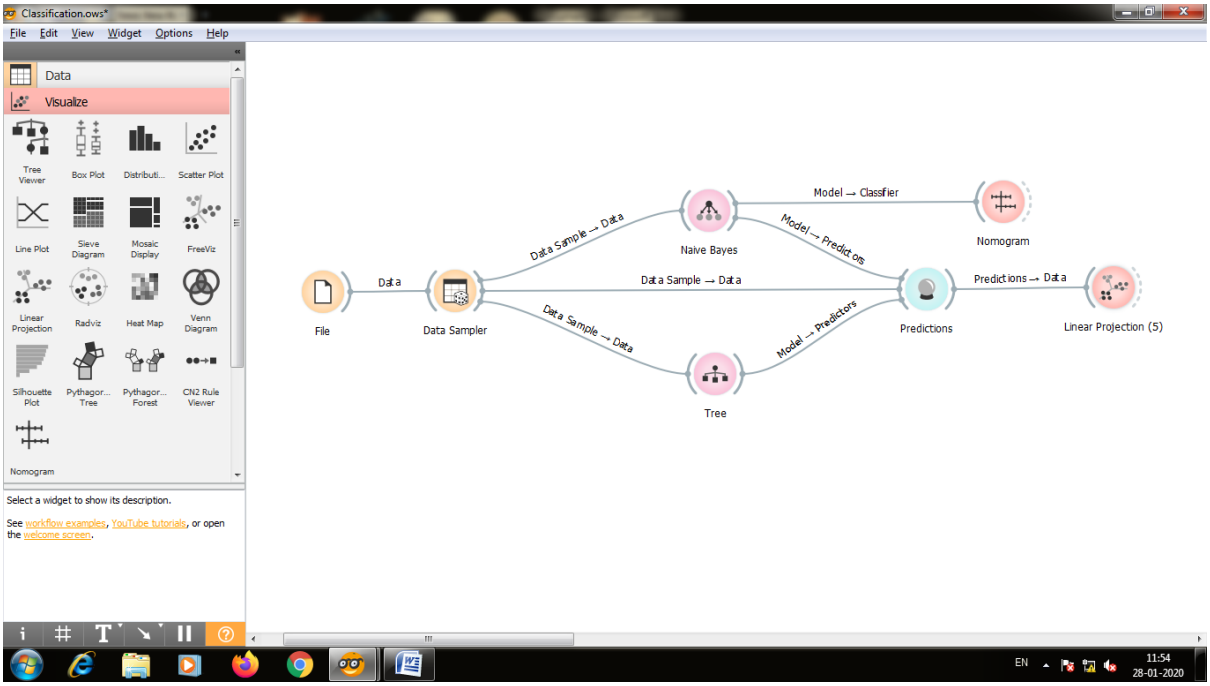


Step8:-There are different ways to visualize the data on the left side of the screen you can scroll down and you will see the visualize step so you click on the tab and you will see several regions that you can use to visualize the data like scatter plot,linear projection etc.

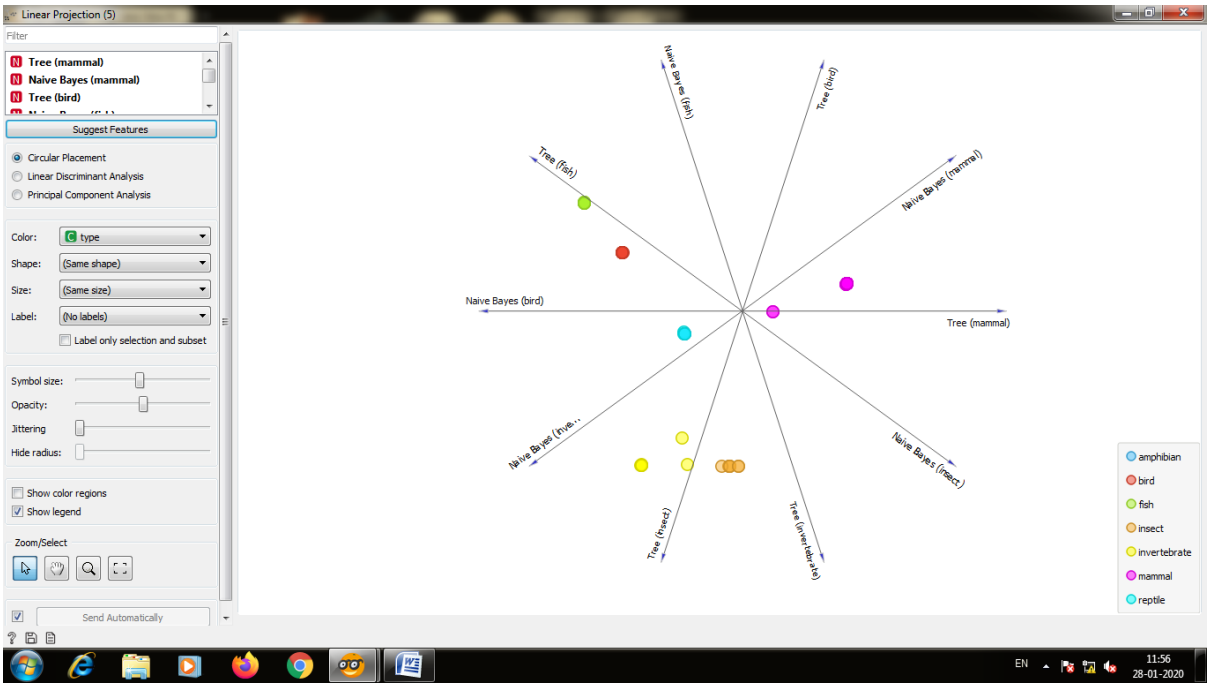
MSC CS - I

Name: Merin Kurian
Roll No.: 20

For this example we are going to use linear projection.



Step9:-Double click on the linear projection and you can the different classes of zoo animal.



MSC CS - I

Name: Merin Kurian
Roll No.: 20

Practical No.: 2

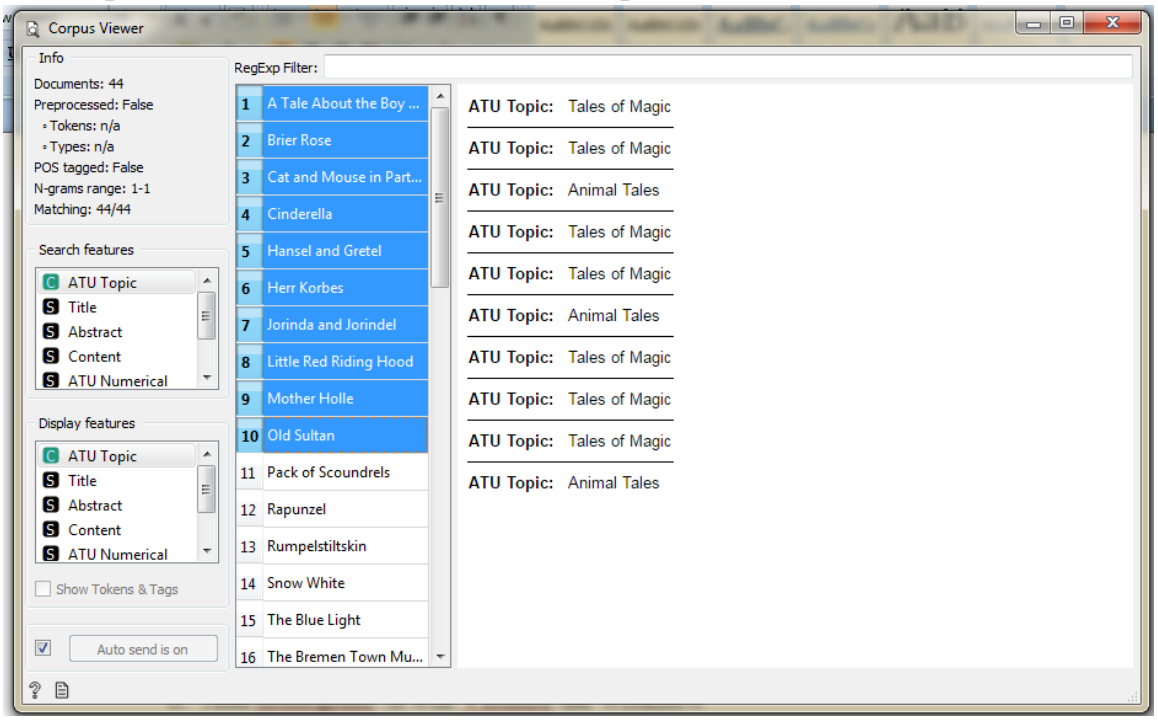
Text Classification

We are going to classify the data based on text . and for that we are going to use unclassify data and check whether the tale is animal tale or tale of magic.

Steps for text classification

1. Take a Corpus and take grimTale.selected and to the corpusViewer.

2. In CorpusViewer take 10 ATU fields to provide data

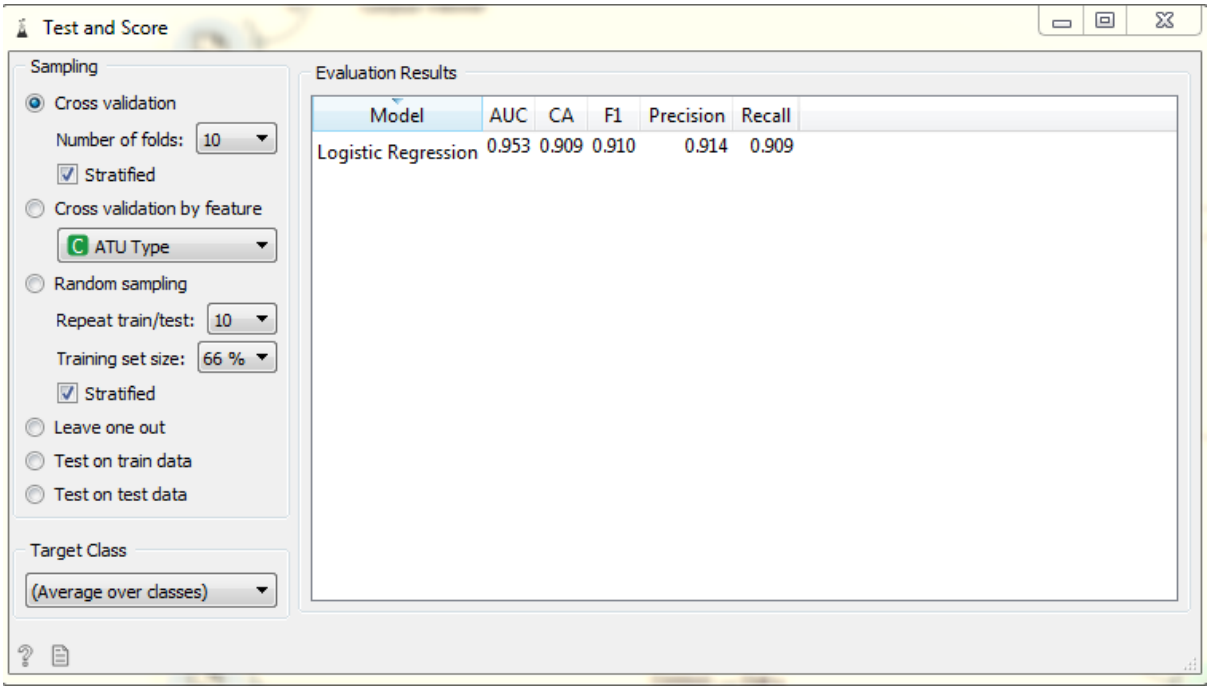


3.
4. And to this add preprocess text for processing the data

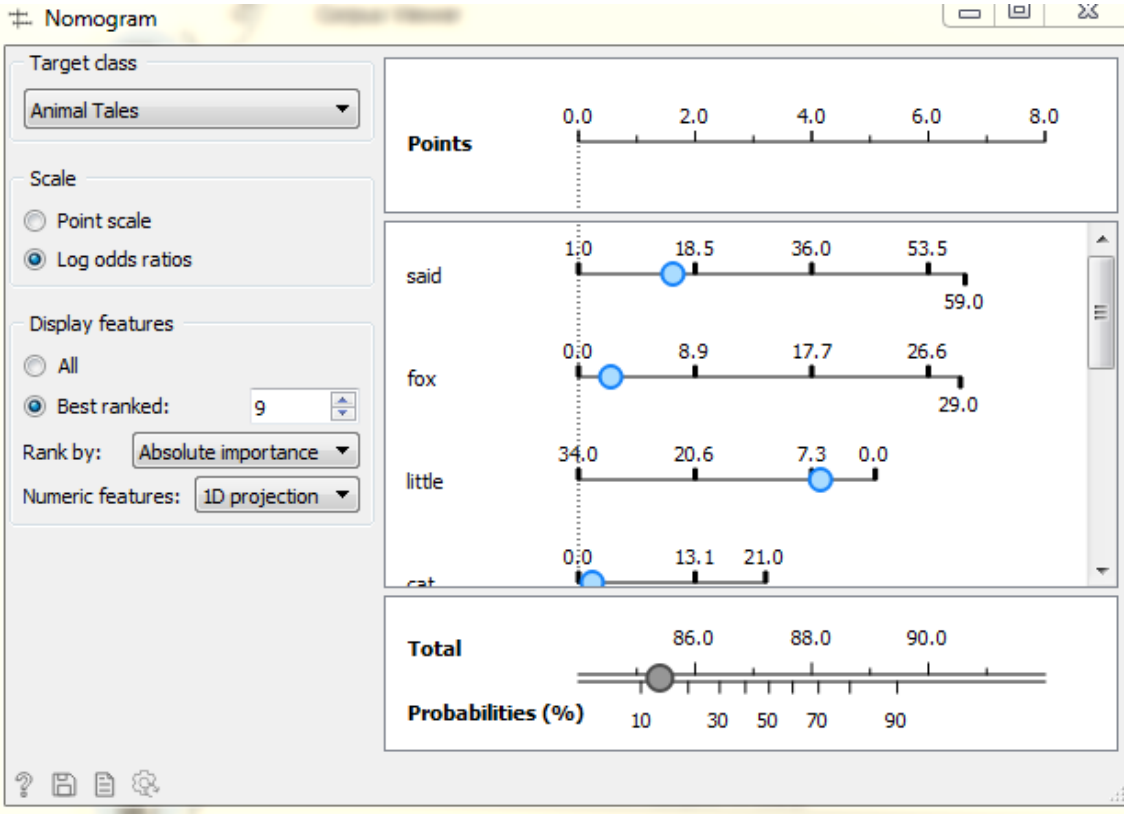
5. And connect this to BagOfWords.

Name: Merin Kurian
Roll No.: 20

6. Take Logistic Regression to bag of words .it construct the model.

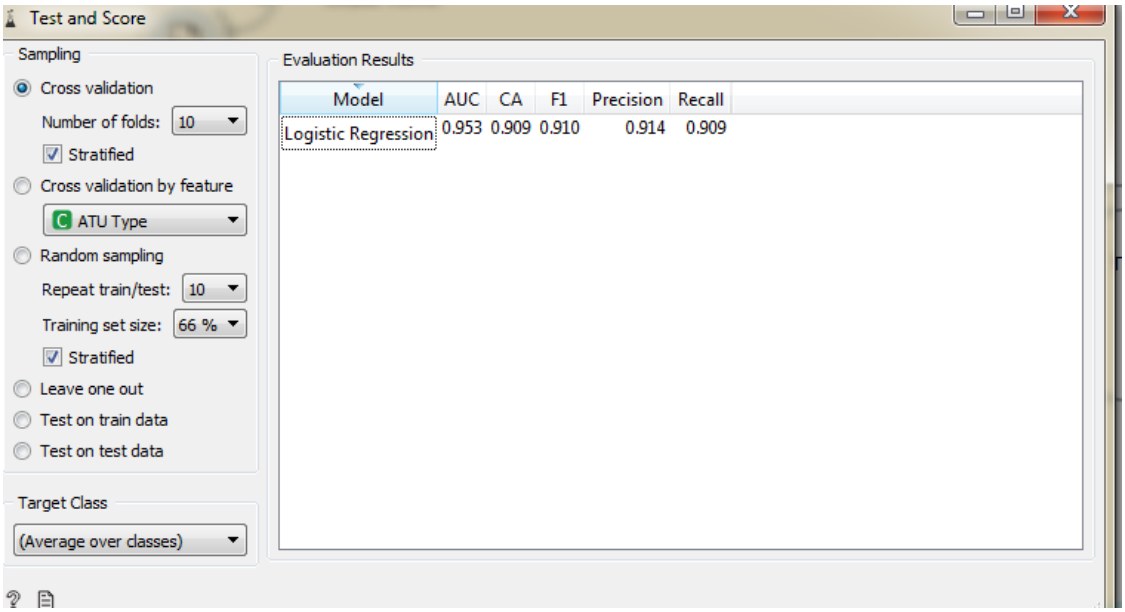


7. Add nomogram .it will Visulize the workflow

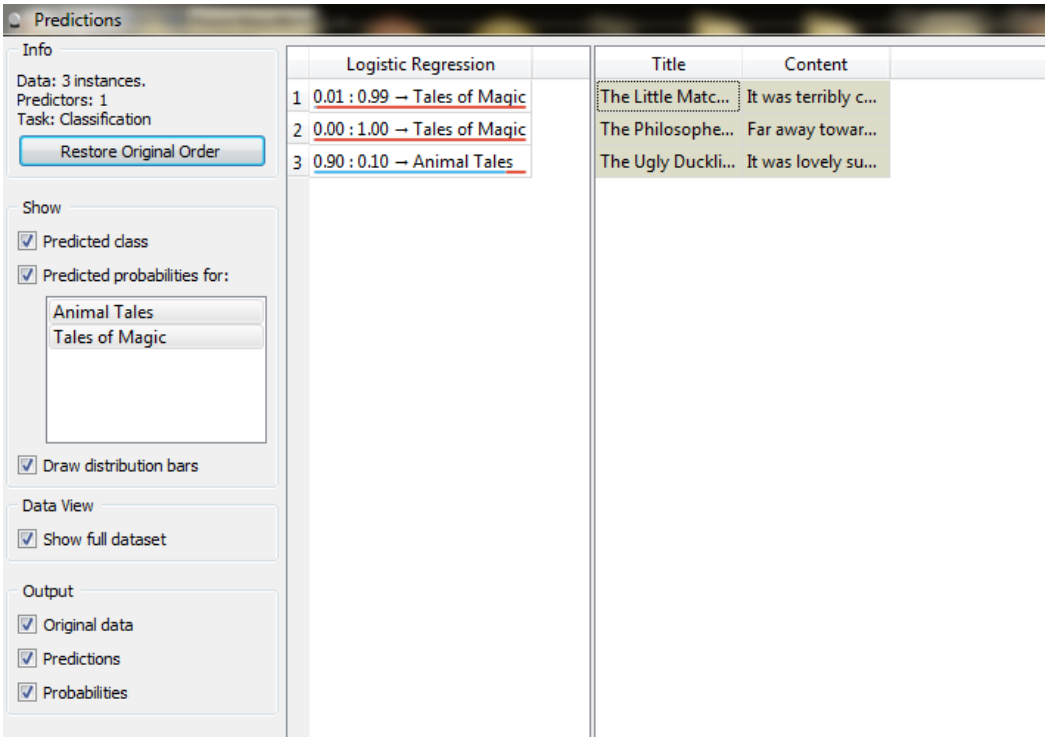


Name: Merin Kurian
Roll No.: 20

8. Add Test and Score it will cross validate the workflow

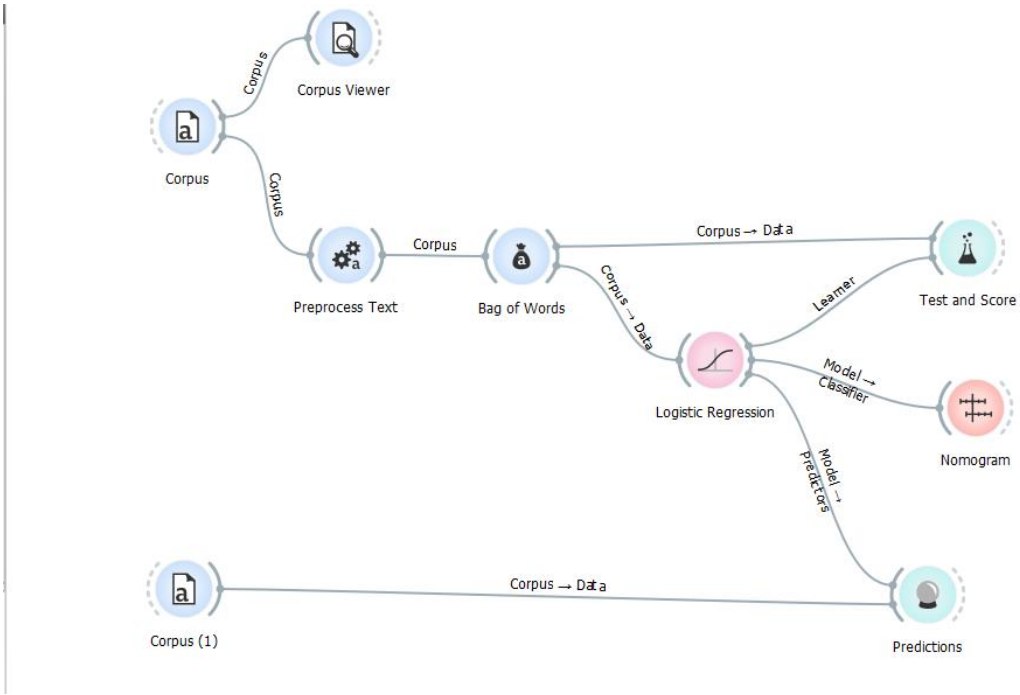


9. and to check our workflow whther it is right or wrong we take another corpus whom we alredy know the classification for that we use Andersen.tab and connect it to prediction



Name: Merin Kurian
Roll No.: 20

10.and finally our workflow looks like this



MSC CS - I

Name: Merin Kurian
Roll No.: 20

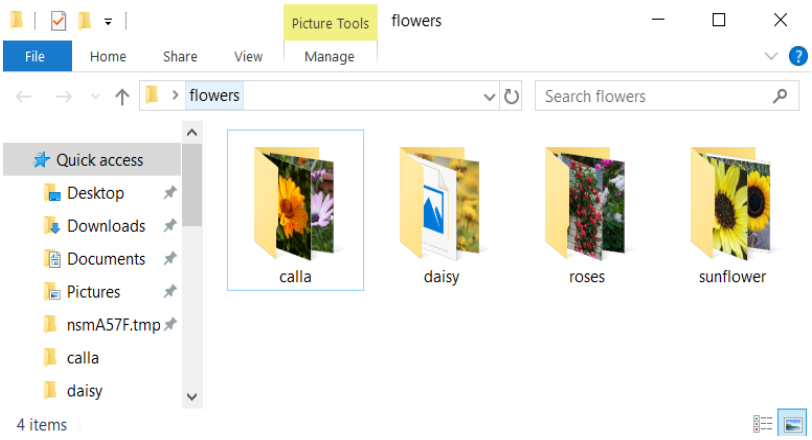
Practical No.: 3

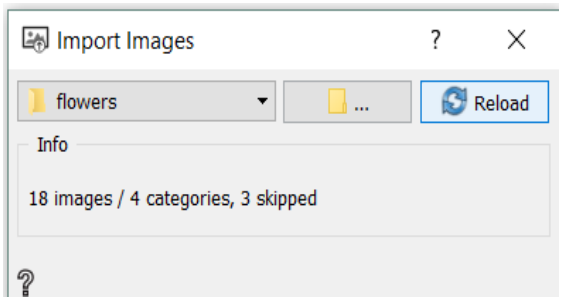
Big Data: Image Analytics-Classification

Image classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: training and testing.

Steps for Image analytics classification:

1. **Import Images:** The first thing to do is to import the image via the **Import Images** widget. You can think this widget as the **File** widget for image. However, **Import Images** widget accepts a directory instead of a file.

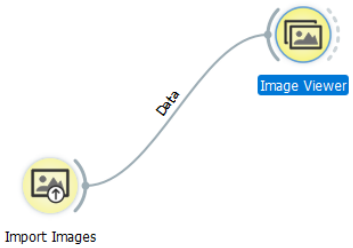


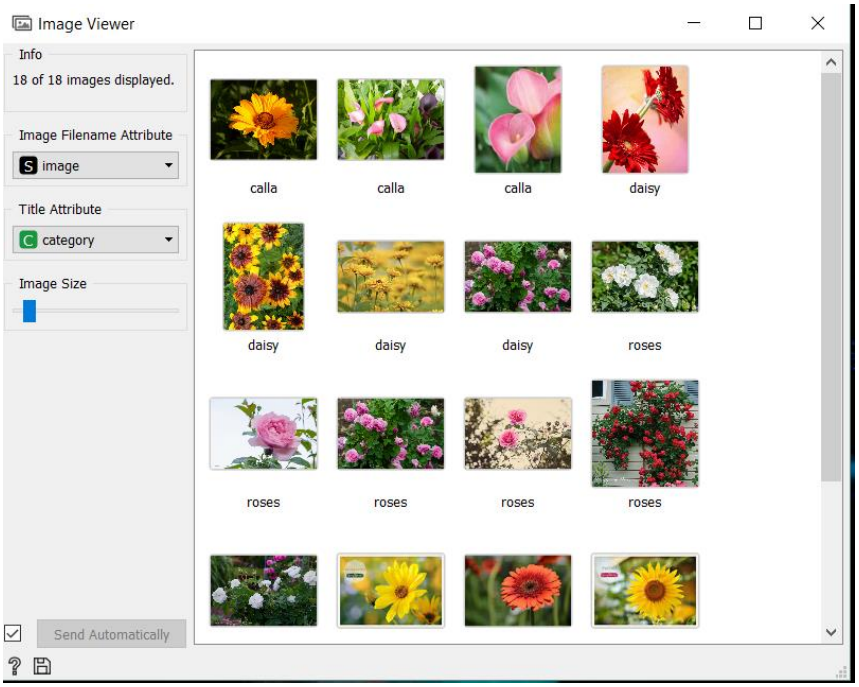


2. **Image Viewer:** Next, we will be relying on the Image Viewer widget to check the content of the directory. This widget will display all of the loaded images.
Connect Image viewer to import images

MSC CS - I

Name: Merin Kurian
Roll No.: 20



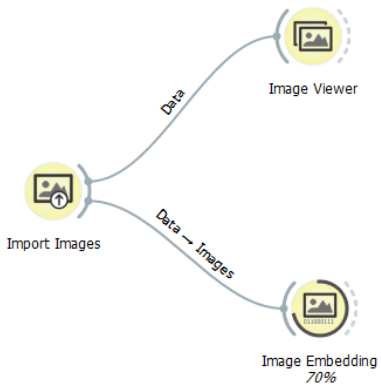


3. **Image Embedding:** Classification and regressions tasks requires data in the form of numbers and there isn't a good way to perform such tasks with images unless we represent it in the form of numbers.

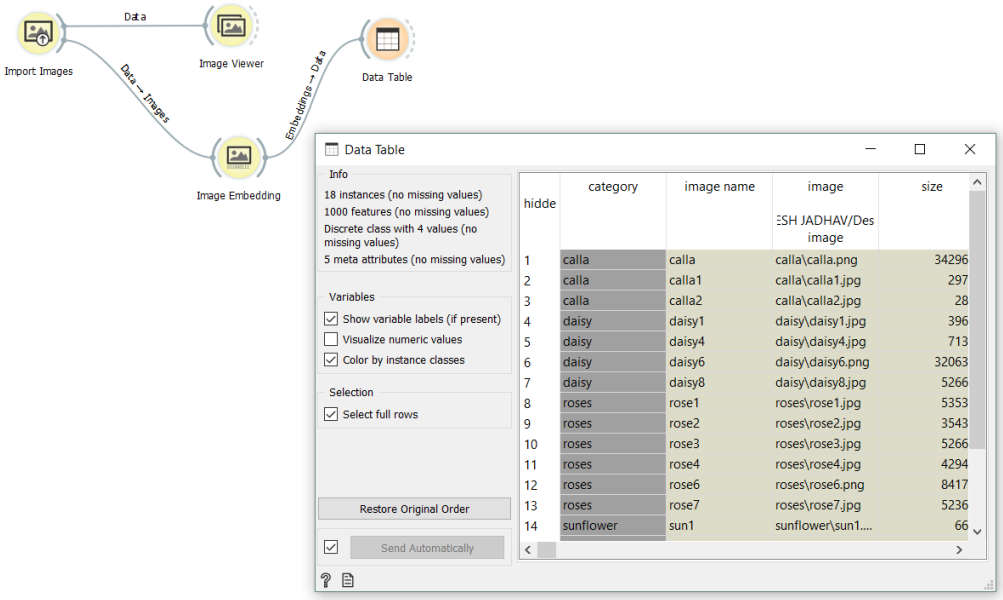
- This is where **Image Embedding** widget works by converting it to a vector of numbers. **Image Embedding** widget reads images and uploads them to a remote server or evaluate them locally.

MSC CS - I

Name: Merin Kurian
Roll No.: 20



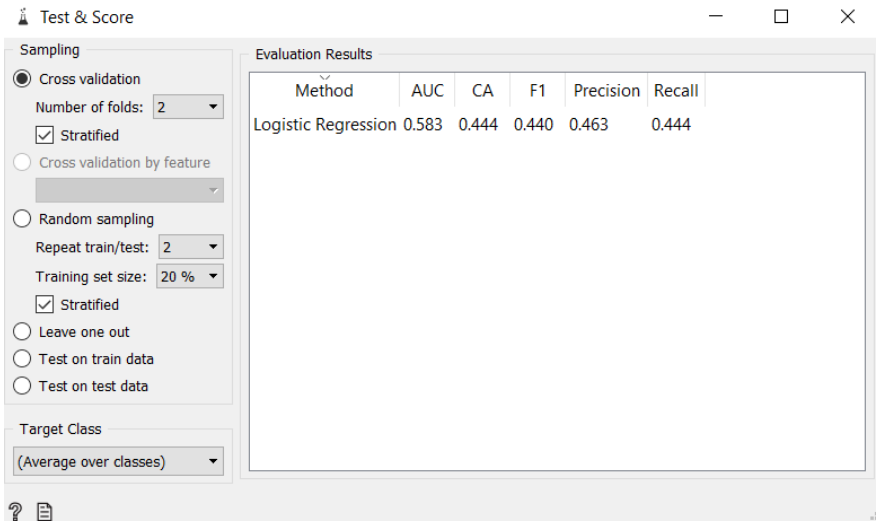
4. **<u>Data Table</u>:** Data Table widget receives one or more data sets for all images with their features and presents them in a spreadsheet format.



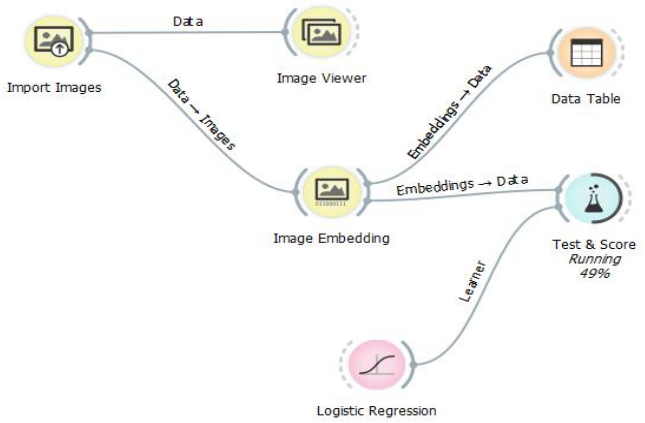
5. **Test and Score:** For cross validation accuracy estimation, we use test and score widgets. But for that, we have to connect test and score to logistic Regression.

MSC CS - I

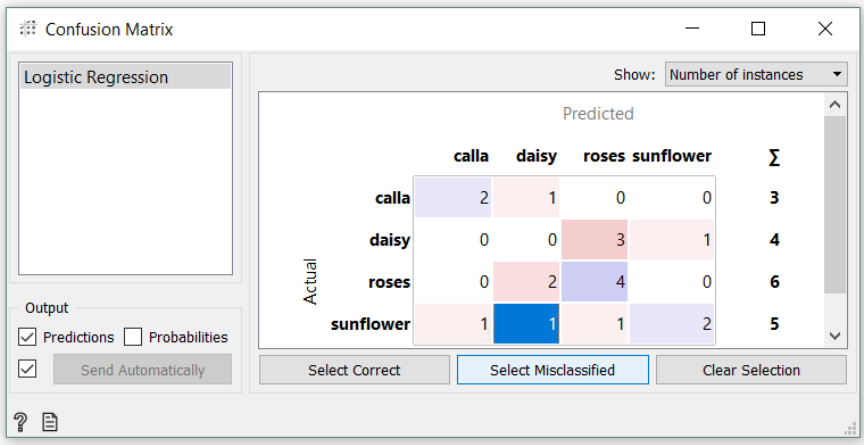
Name: Merin Kurian
Roll No.: 20



6. **Logistic Regression:** The widget is used just as any other widget for inducing a classifier. **Logistic Regression** learns a Logistic Regression model from the data. It only works for classification tasks.



7. **Confusion Matrix:** The Confusion Matrix gives the number/proportion of instances between the predicted and actual class.
From the table, we have to select one cell from misclassified to see the specification.



Name: Merin Kurian
Roll No.: 20

8. **Image Viewer:** After selecting the cell, we can see the specification between actual and predicted classes of images in an image viewer.

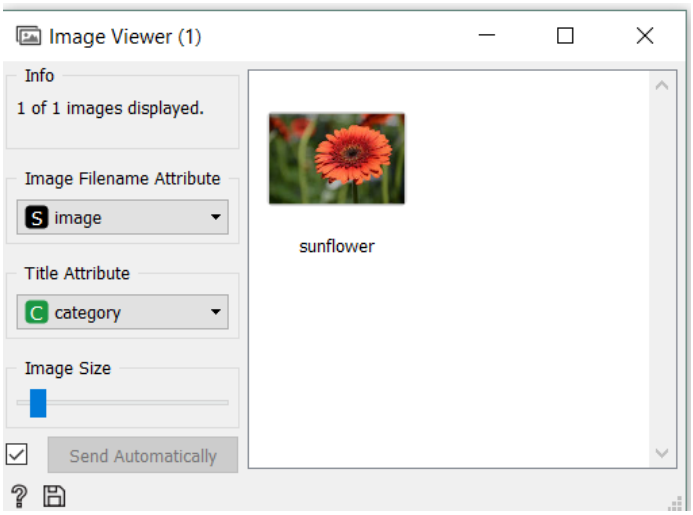
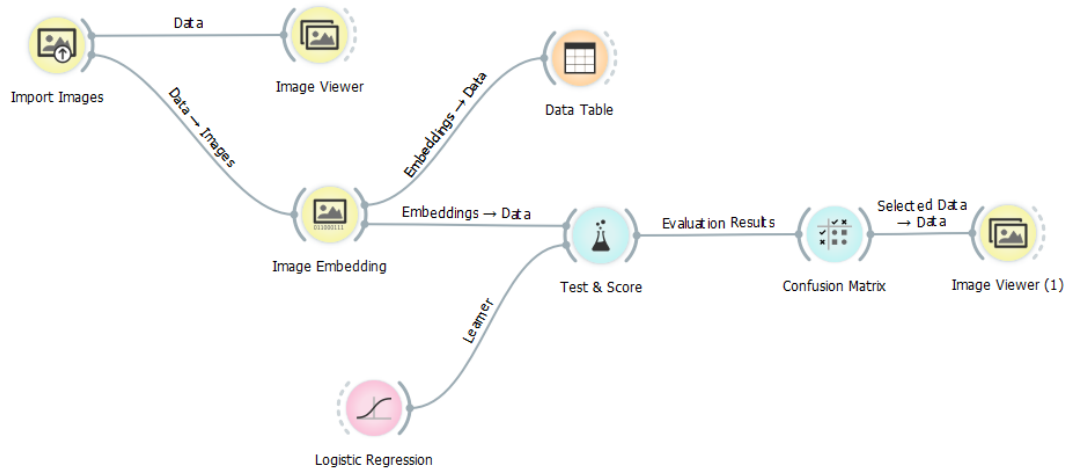


Image Analytics-Classification:



MSC CS - I

Name: Merin Kurian
Roll No.: 20

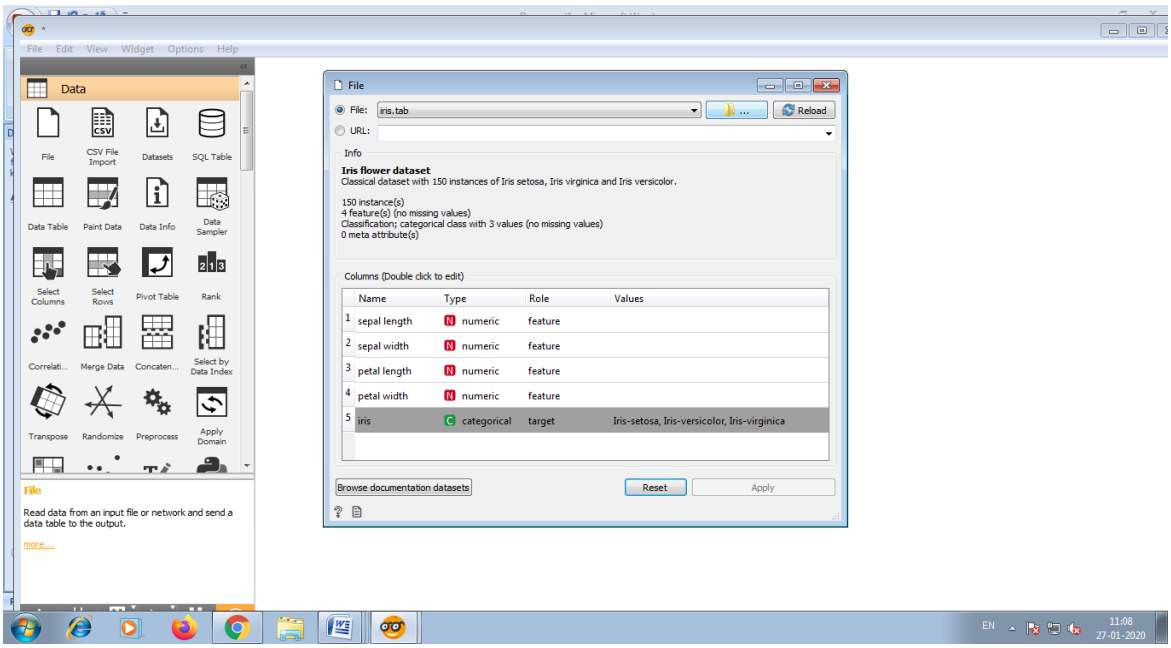
Practical No.: 4

Aim: Hierarchical clustering in orange.

Solution:

Open Application -> Orange -> Select new file

Select File -> double click on it ->Select the table from the file as iris.tab.

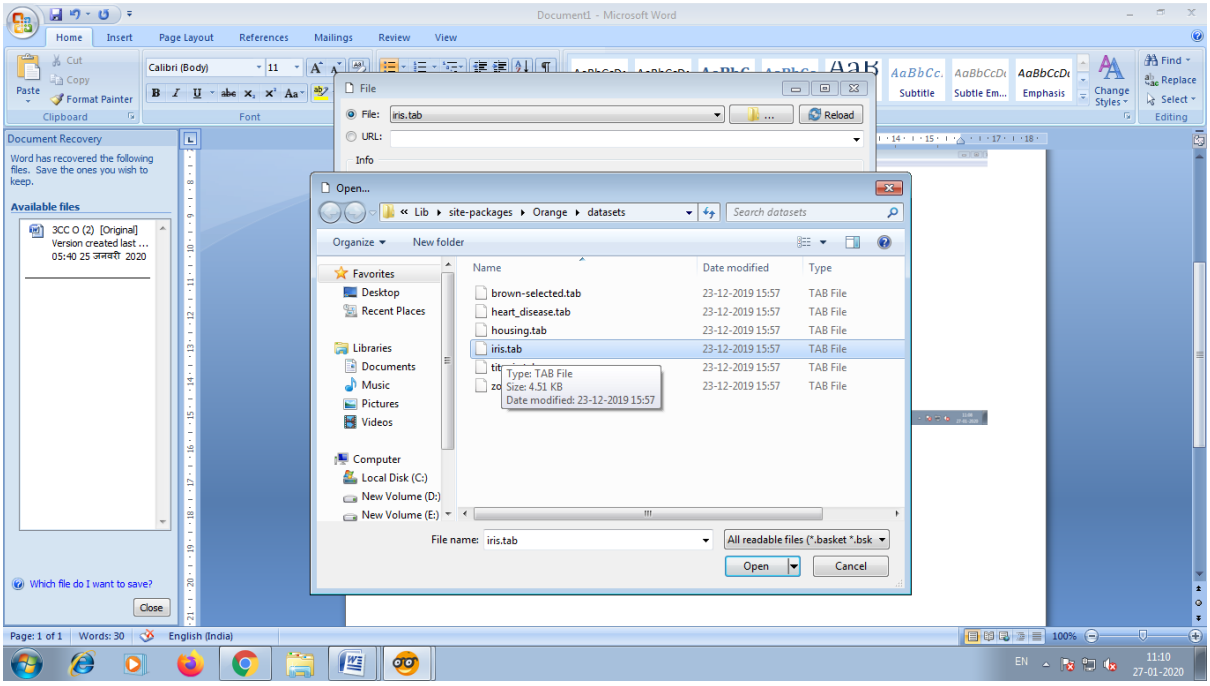


MSC CS - I

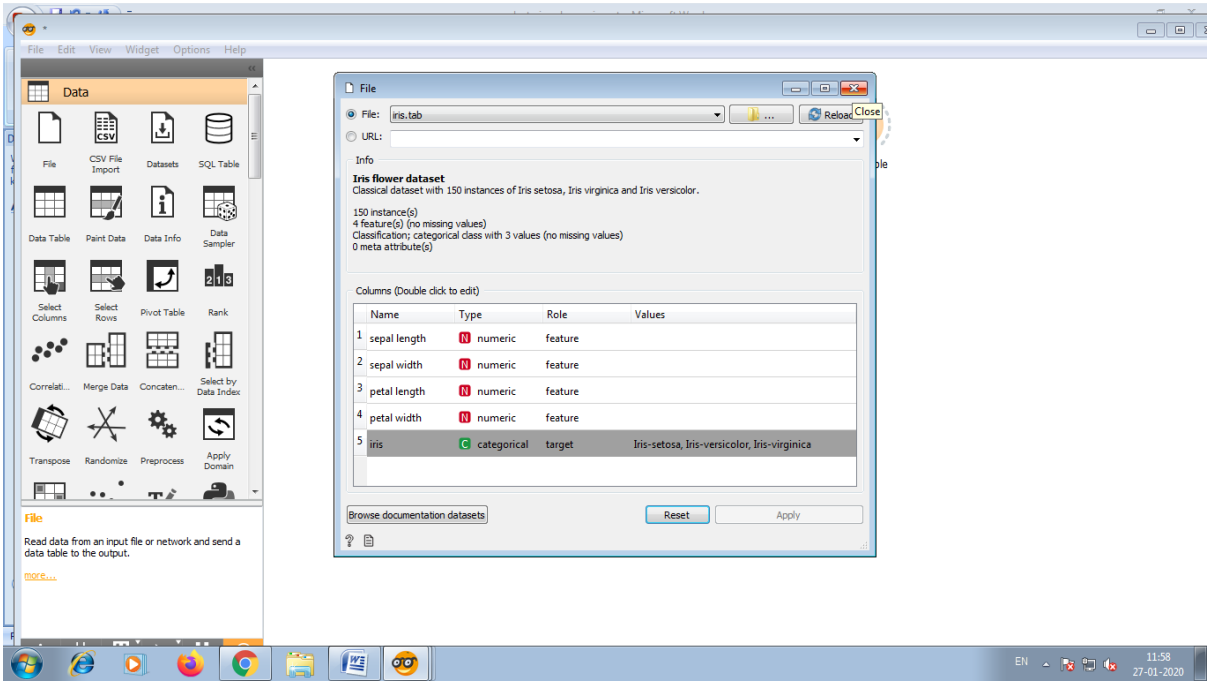
Name: Merin Kurian

Roll No.: 20

After selecting iris.tab, open it .



Directly close the current window as shown in below.

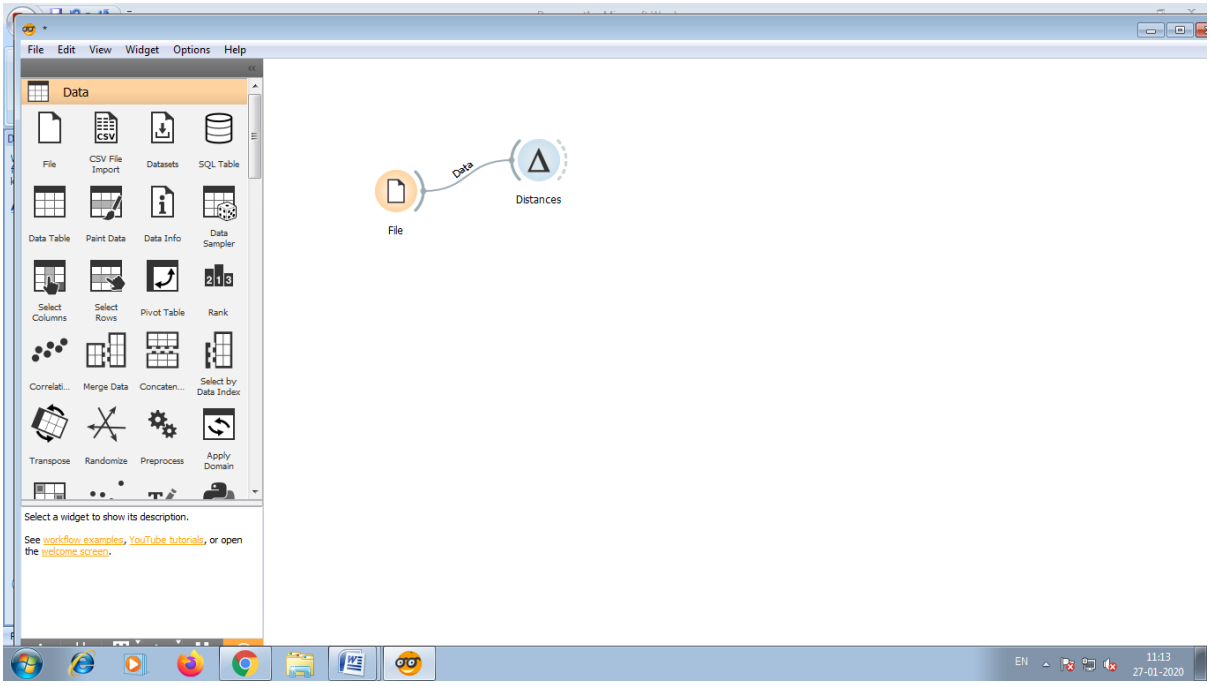


MSC CS - I

Name: Merin Kurian

Roll No.: 20

Click on the file and select Distances from the list.

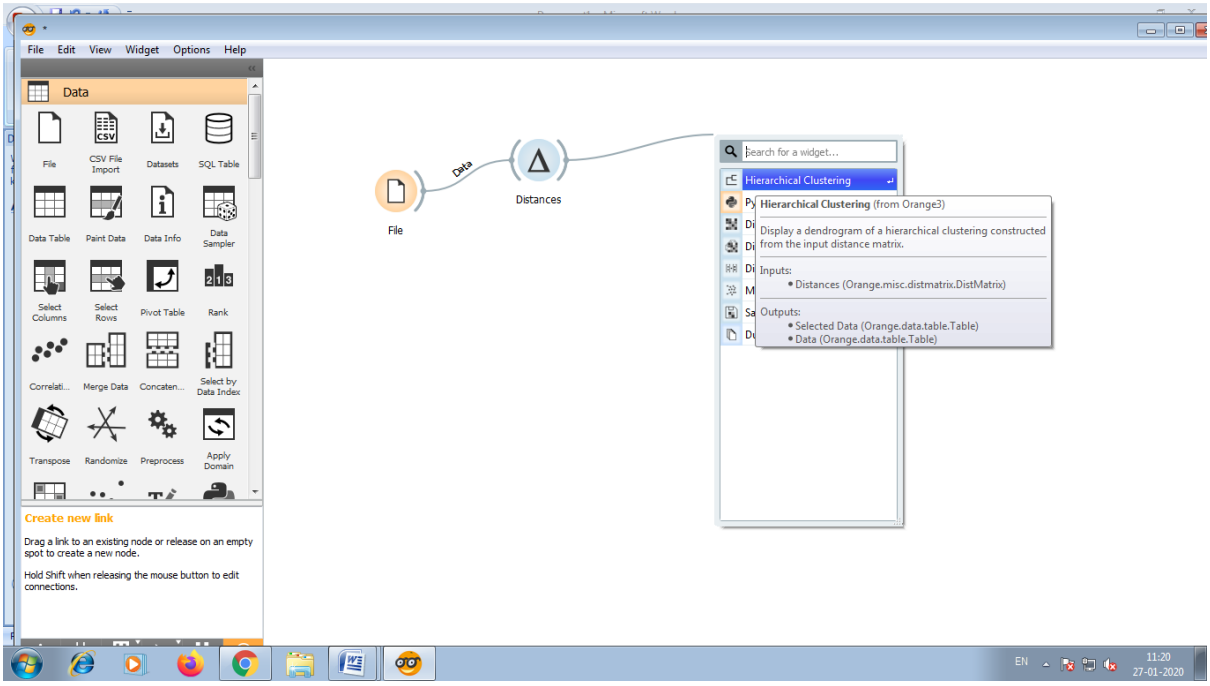


MSC CS - I

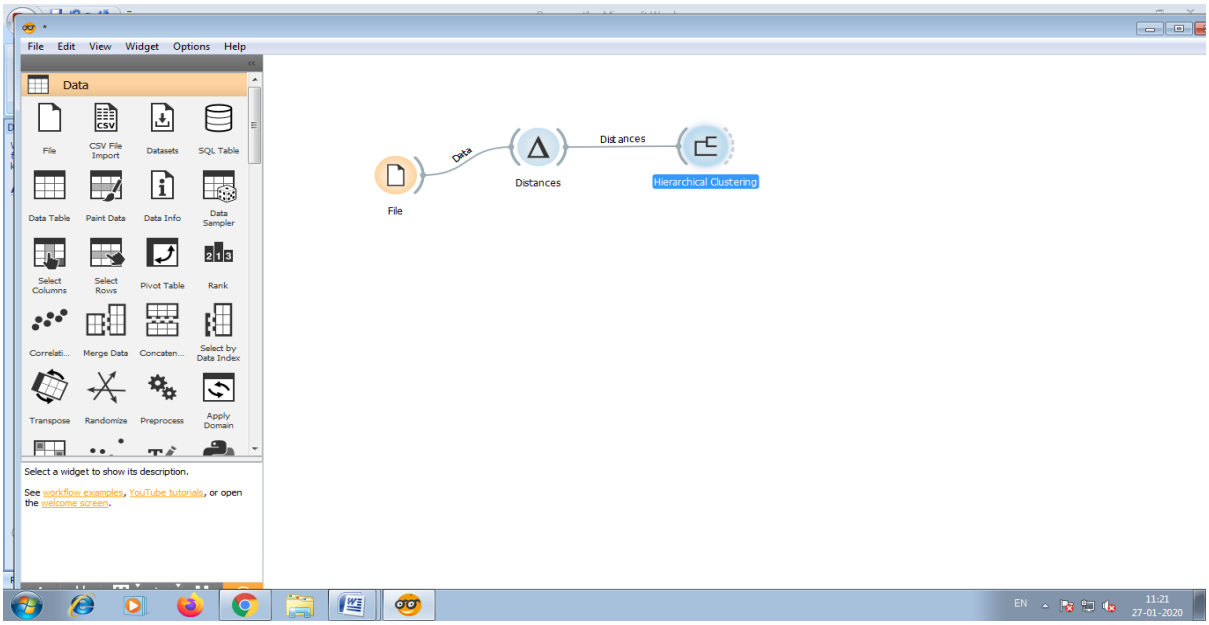
Name: Merin Kurian

Roll No.: 20

Again by clicking on distances select the hierarchical clustering .



After selecting hierarchical clustering it look like this

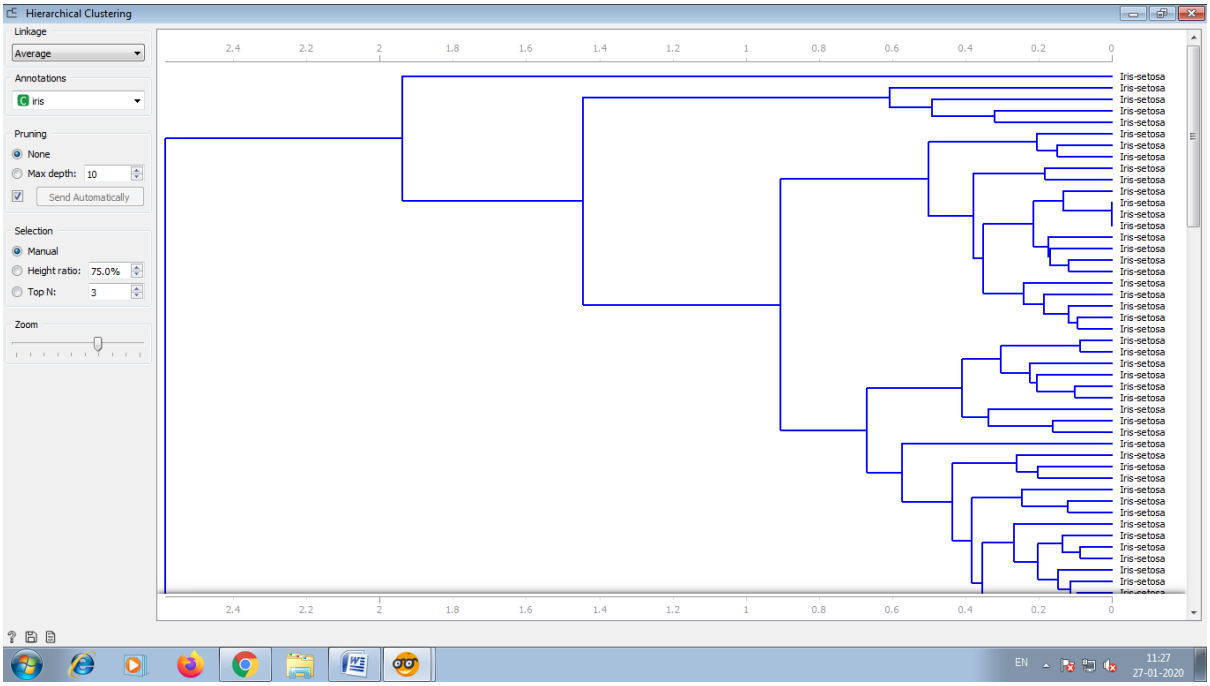


MSC CS - I

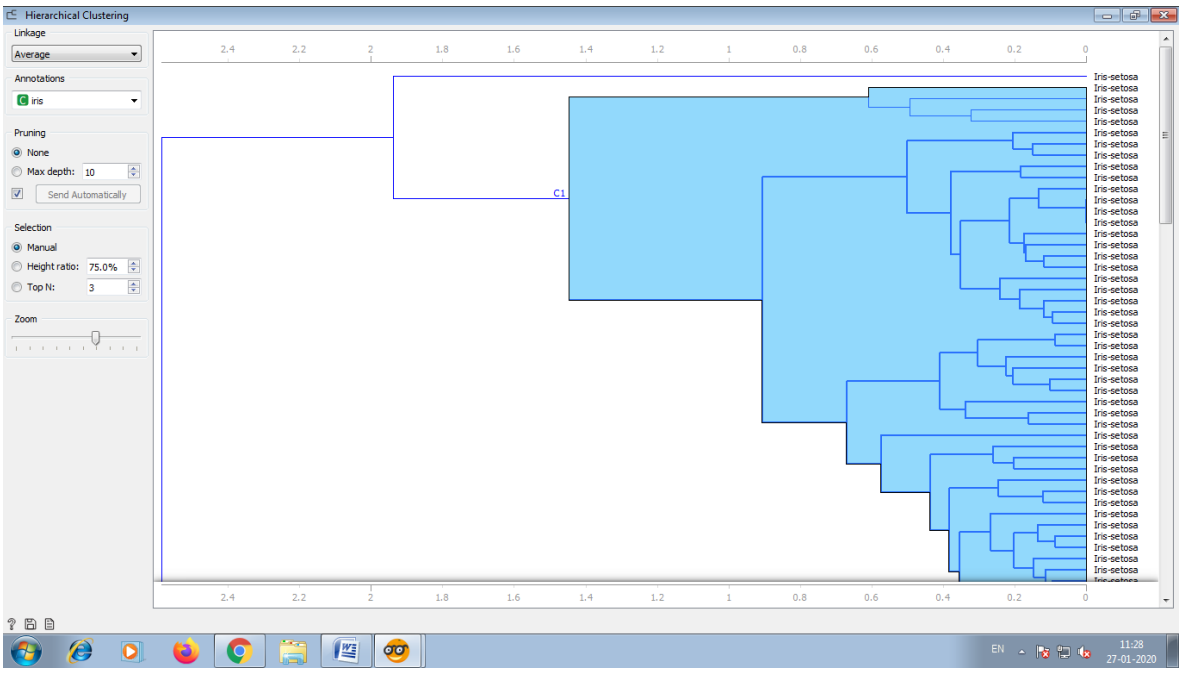
Name: Merin Kurian

Roll No.: 20

Now double click on hierarchical clustering, you will see the picture of hierarchical clustering.



Now select the sub-cluster as shown in below.

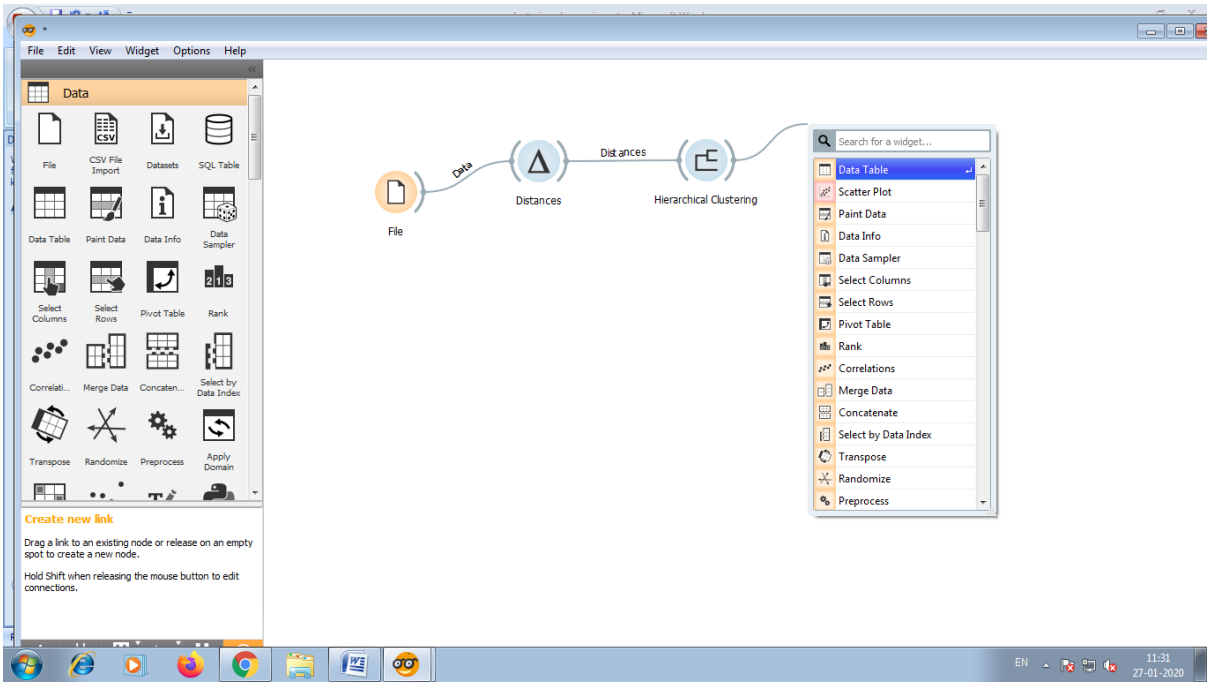


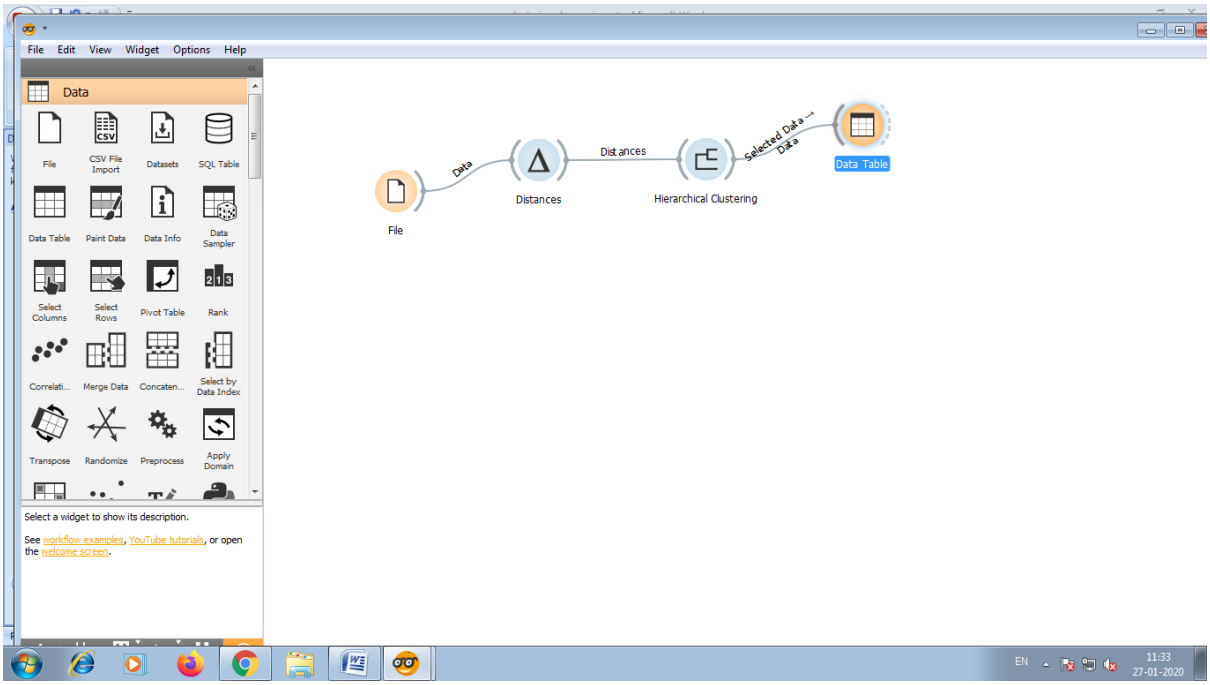
MSC CS - I

Name: Merin Kurian

Roll No.: 20

Now by clicking on hierarchical clustering select the data table.



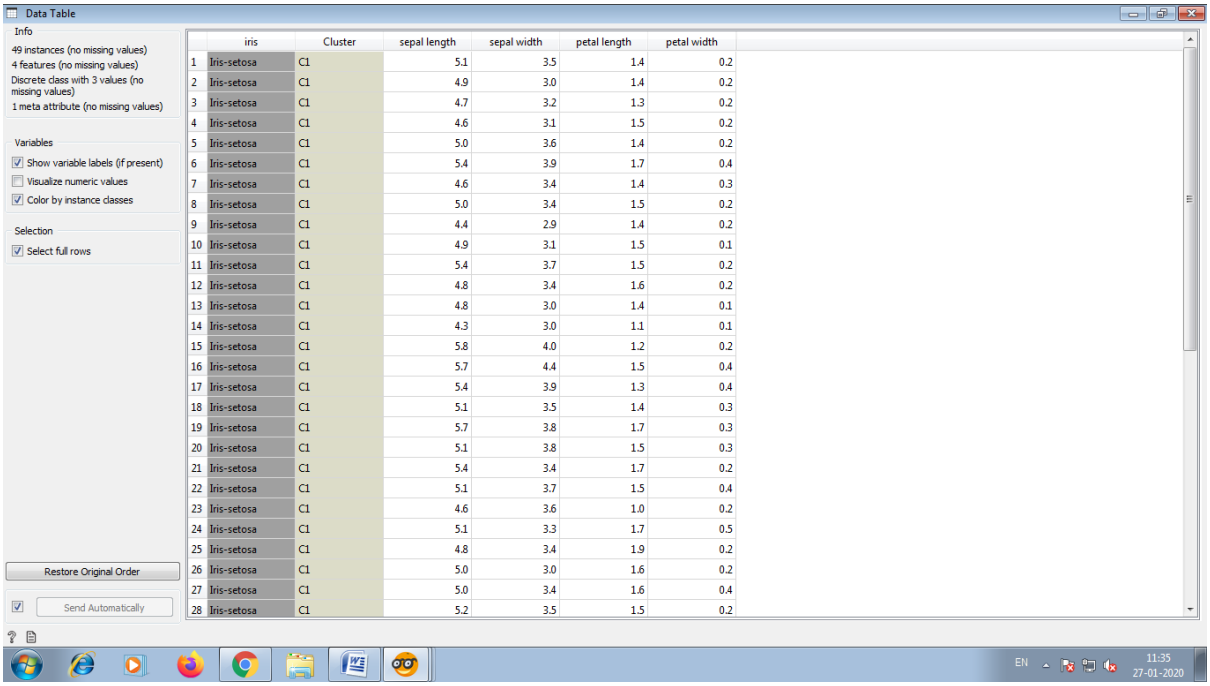


MSC CS - I

Name: Merin Kurian

Roll No.: 20

By double clicking on data table you will see the data which you have selected.

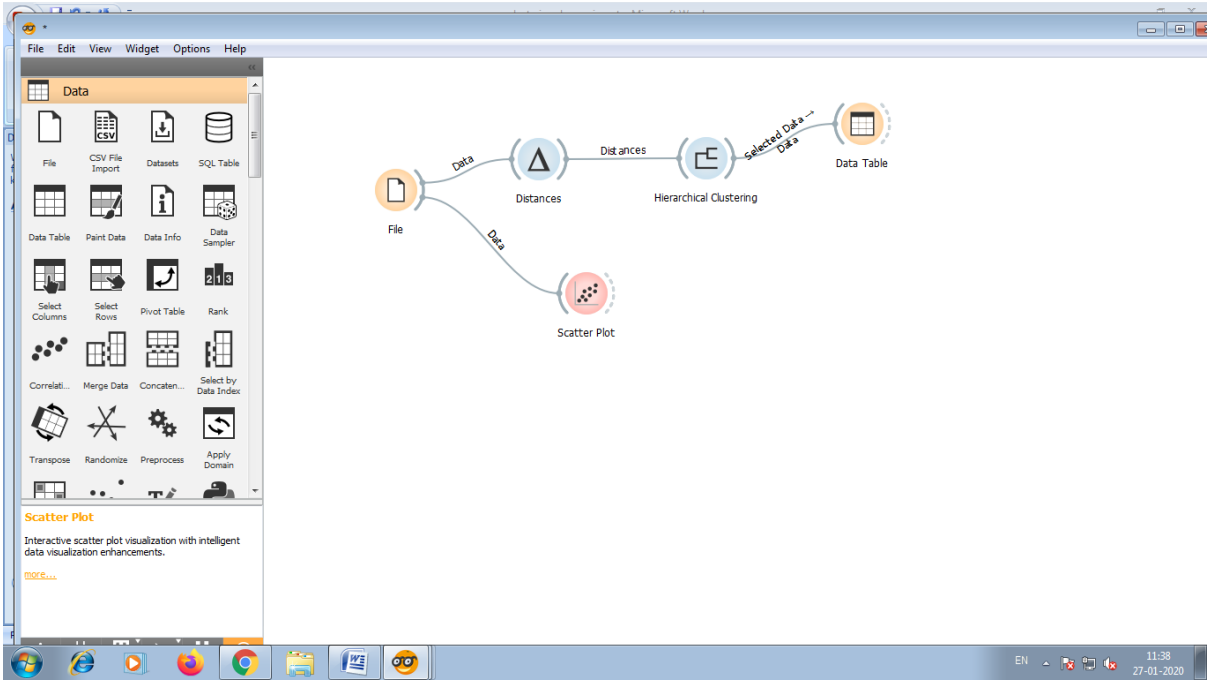


Now by clicking on file select scatter plot.

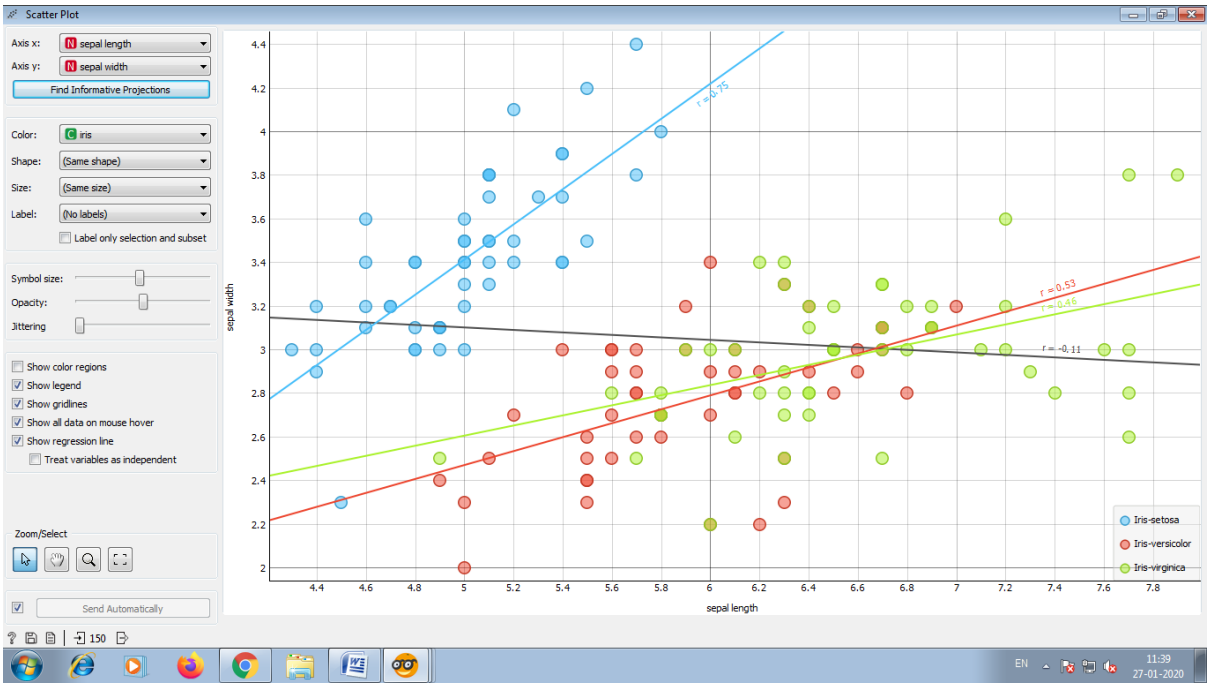
MSC CS - I

Name: Merin Kurian

Roll No.: 20



- By double clicking on scatter plot you will see the different colours , which shows the different type of clusters as shown in below figure.
- Cluster of Iris-setosa is represented by blue colour, Cluster of Iris-versicolor is represented by red colour and Cluster of Iris-virginica is represented by yellow green.



MSC CS - I

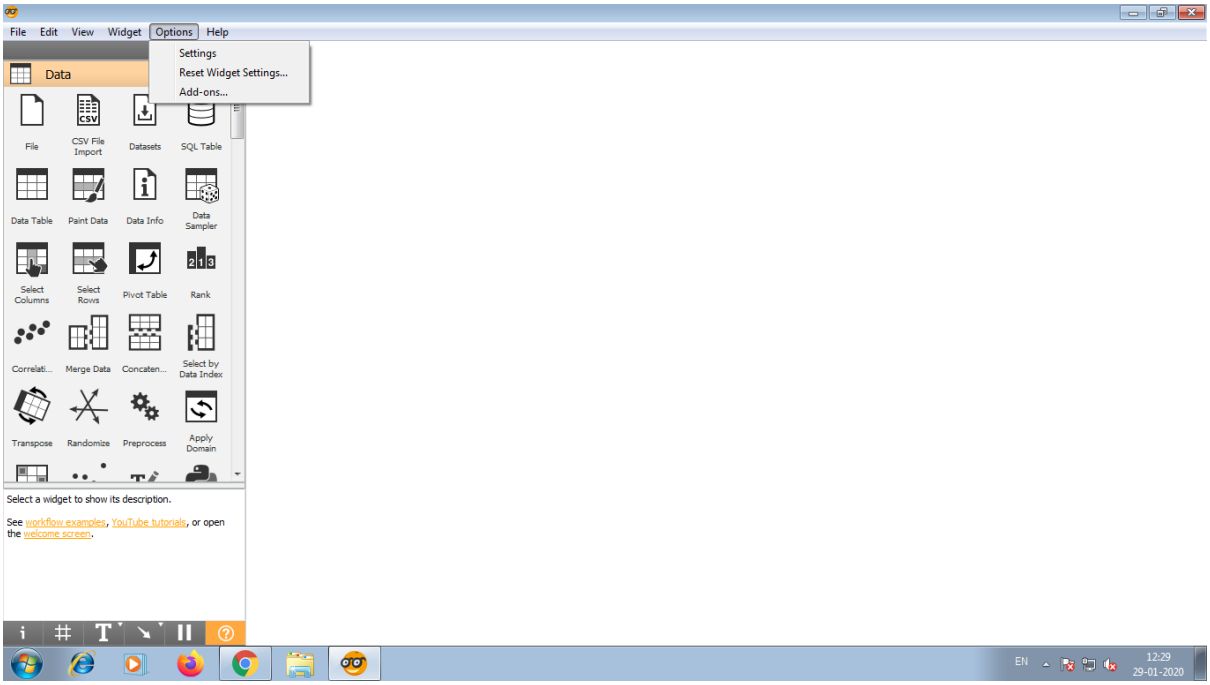
Name: Merin Kurian
Roll No.: 20

Practical No.: 5

Aim : Text preprocessing in Orange

Solution : Open->Orange->File->New

1)First we install the Text-Add-ons , Go to Options->Add-ons.

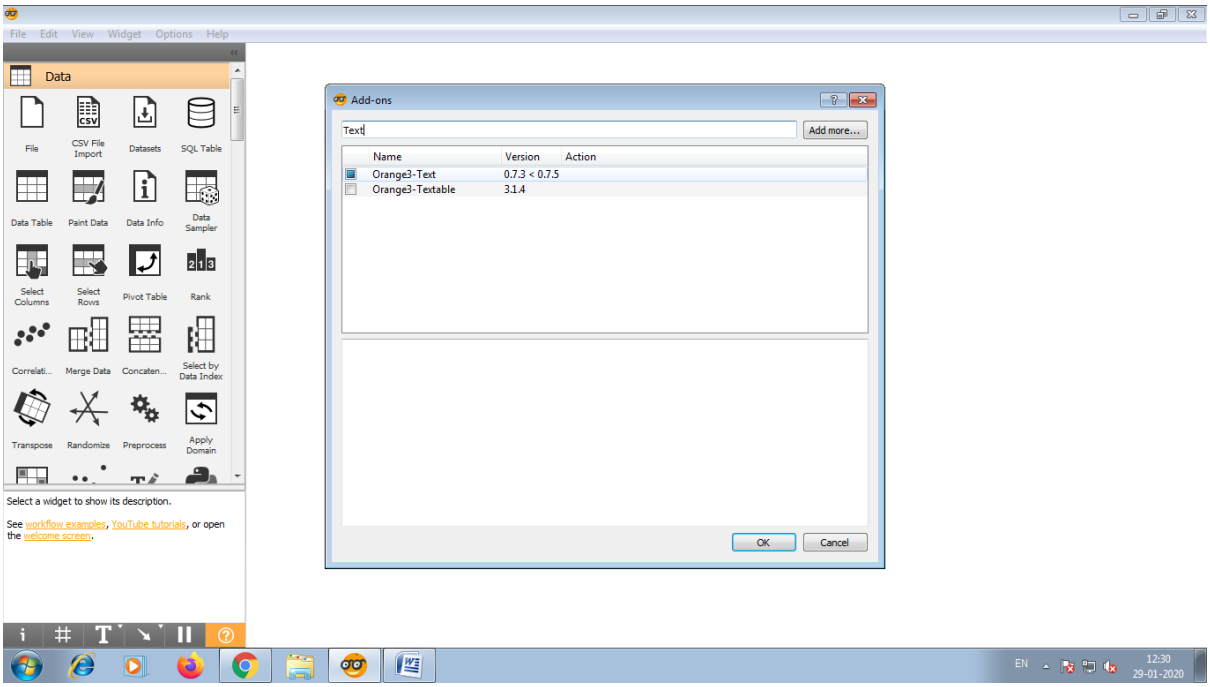


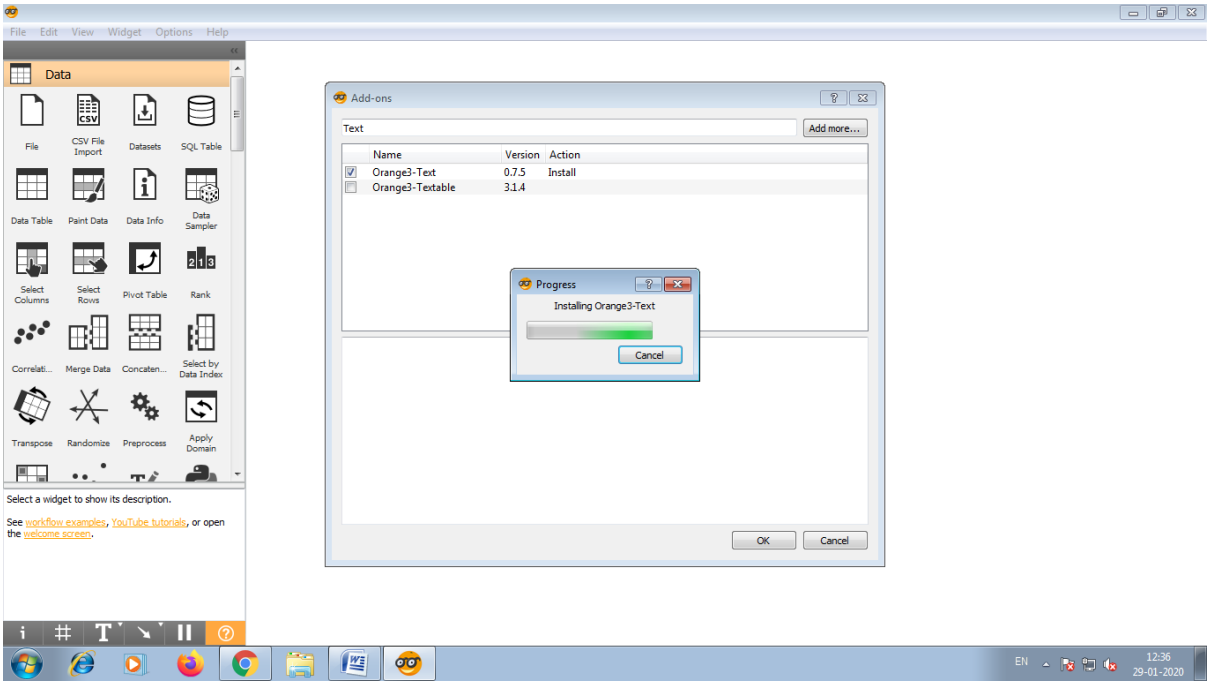
2)select Text-.

MSC CS - I

Name: Merin Kurian

Roll No.: 20



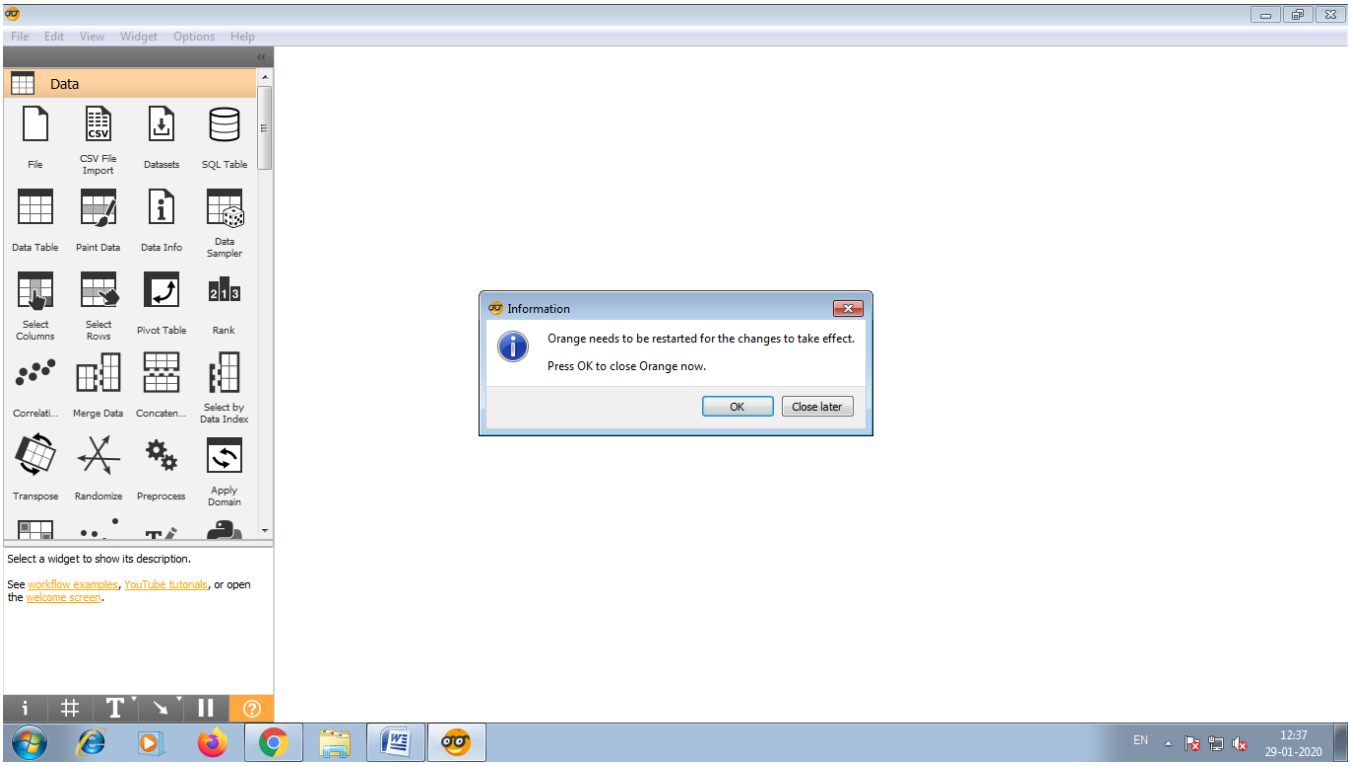


3)Restart orange to for the add-on to appear

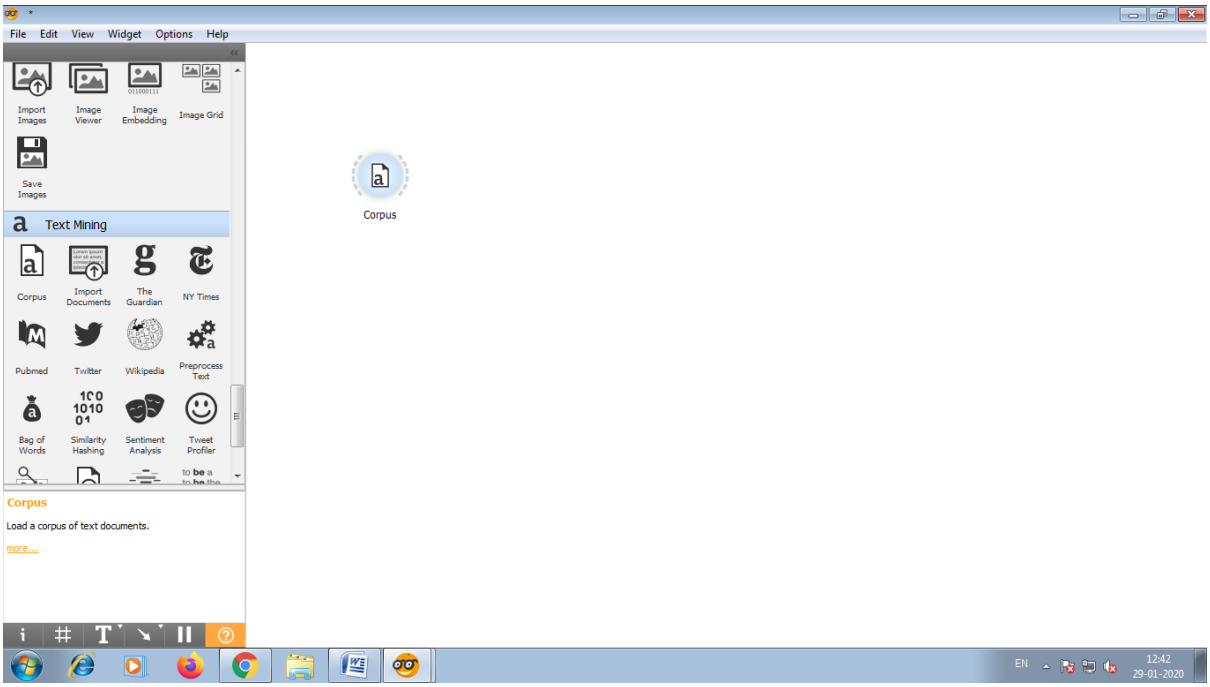
MSC CS - I

Name: Merin Kurian

Roll No.: 20



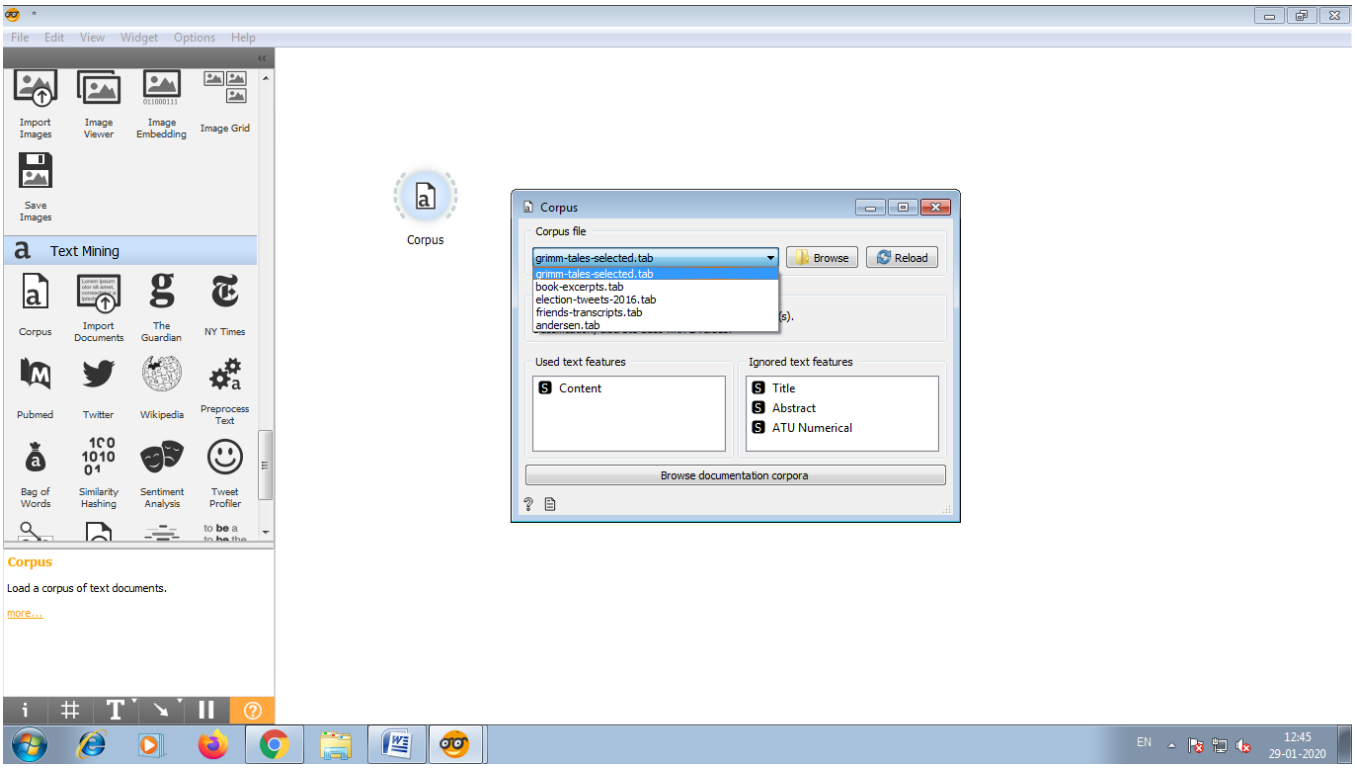
4)Now let us load the data ->Corpus



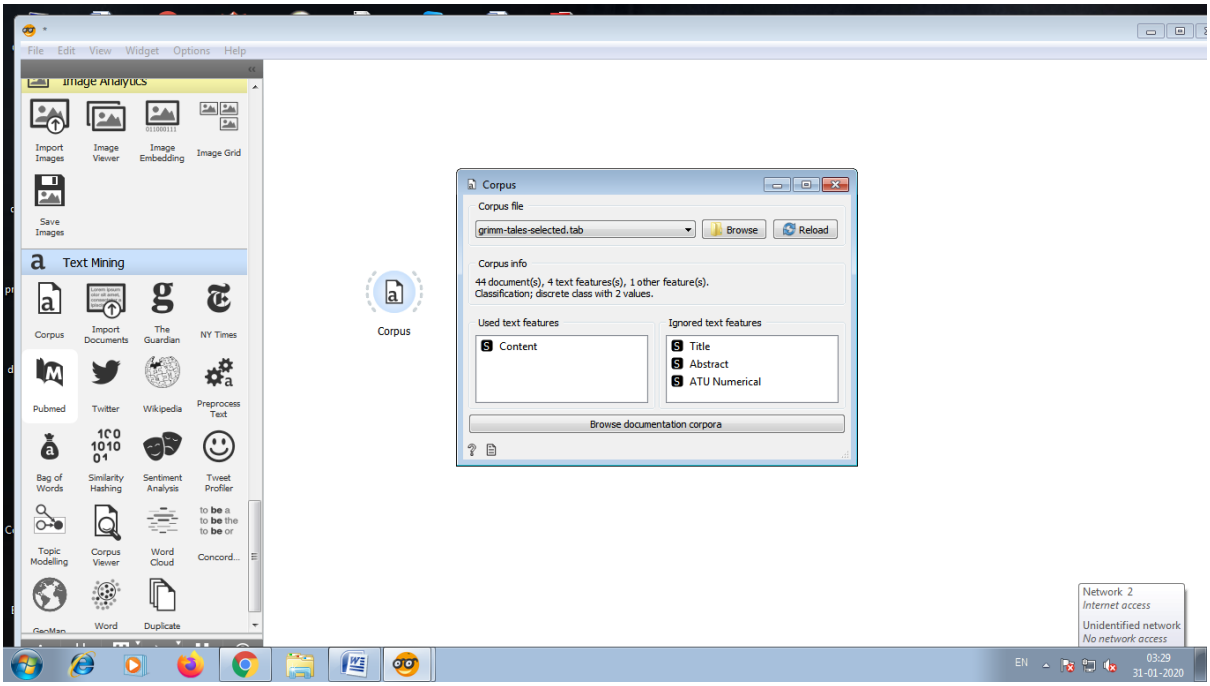
MSC CS - I

Name: Merin Kurian
Roll No.: 20

5)Place Corpus widget on the canvas and open it.Go to browse documentation Corpora ->Select ->and load Grimm-Tales-selected.



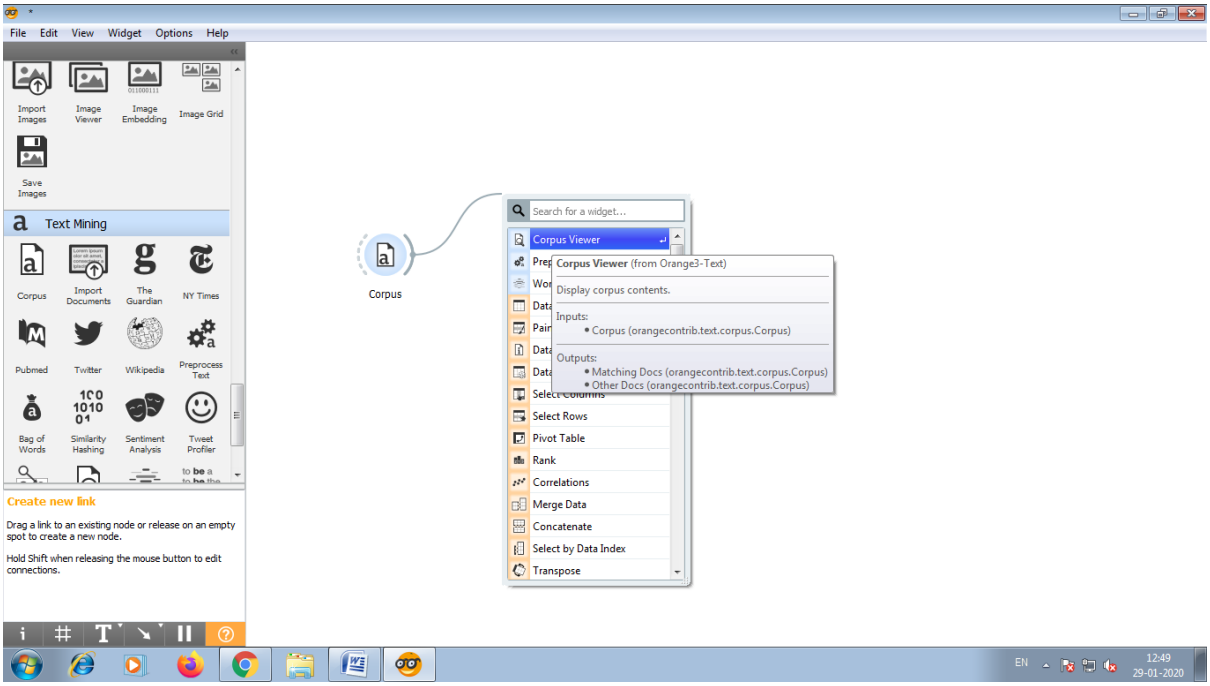
6)We have 44 Grim tales on the output of the widget.



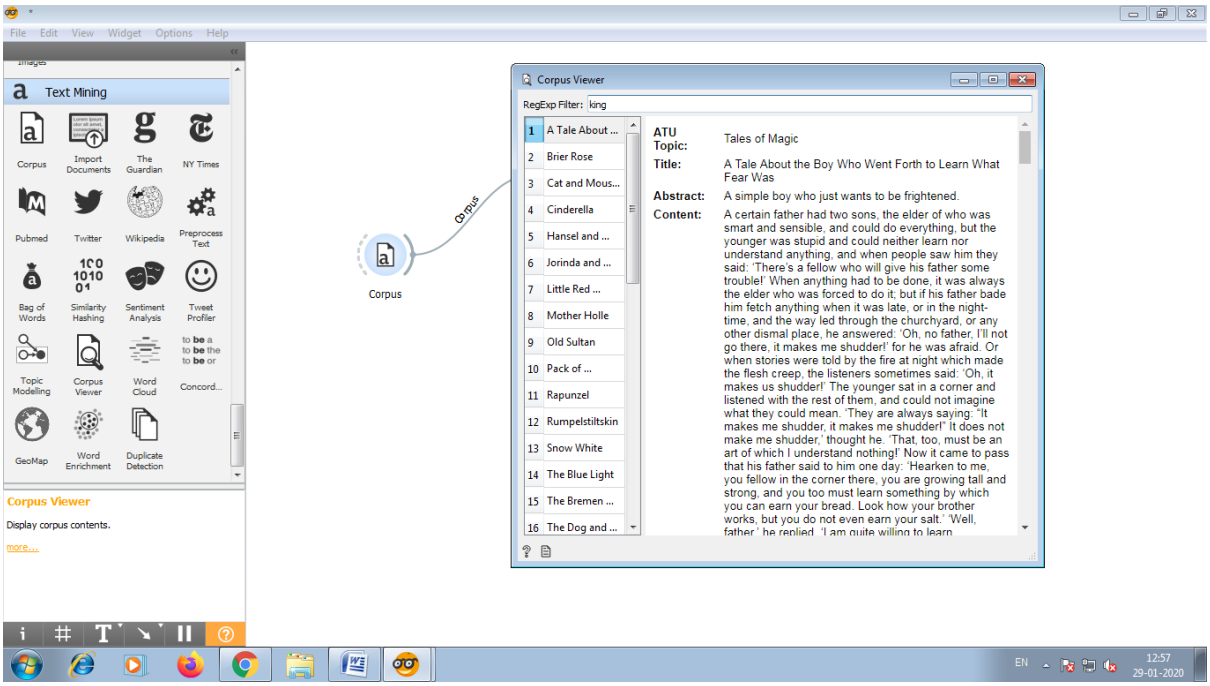
MSC CS - I

Name: Merin Kurian
Roll No.: 20

7)Connect Corpus viewer to corpus.



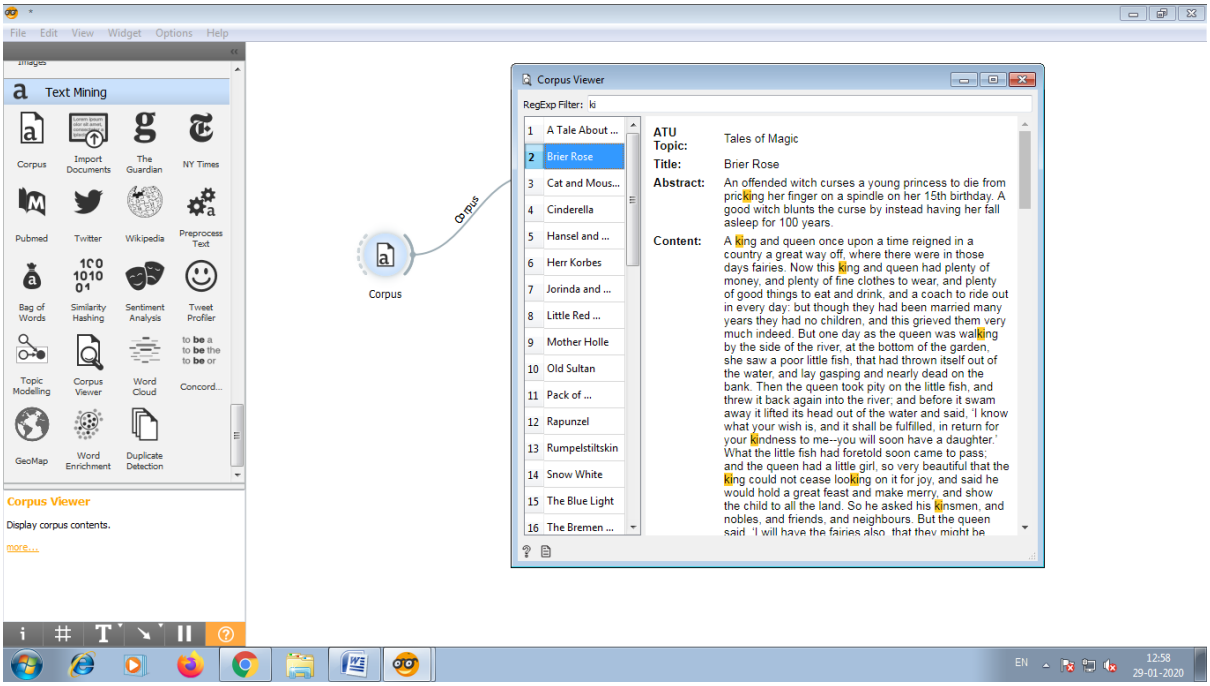
8)Corpus viewer displays text and enable us to browse it.



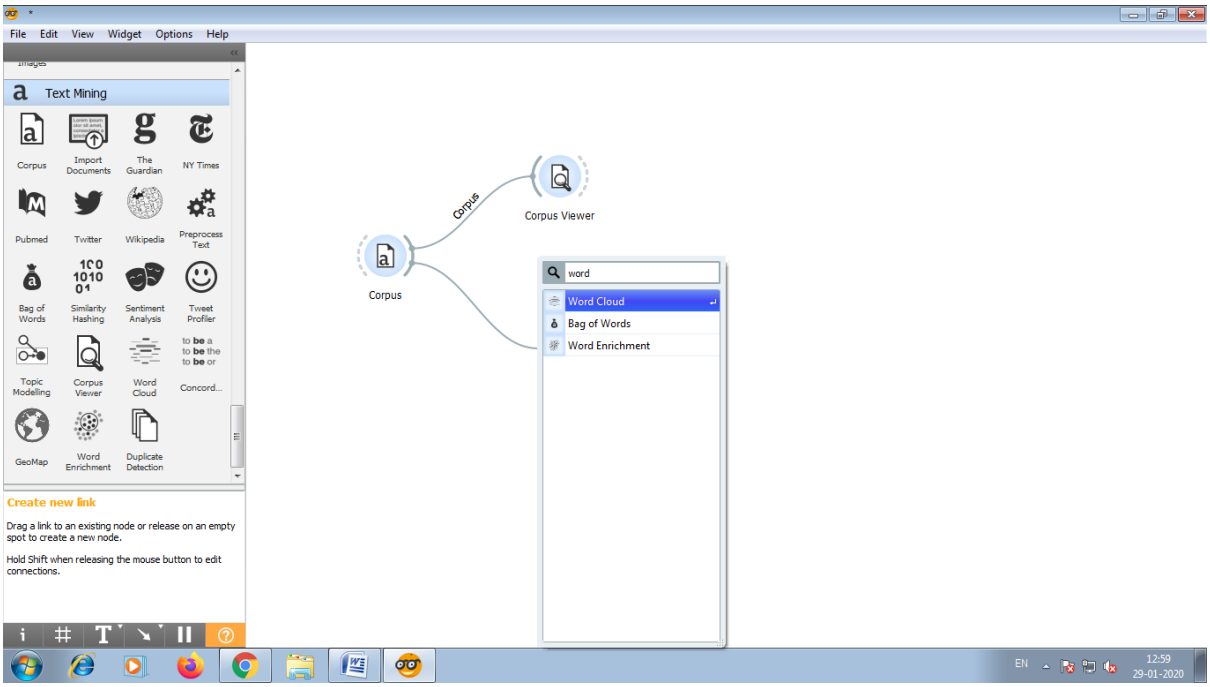
MSC CS - I

Name: Merin Kurian
Roll No.: 20

For example, we can output only those document that contain the word “king”.



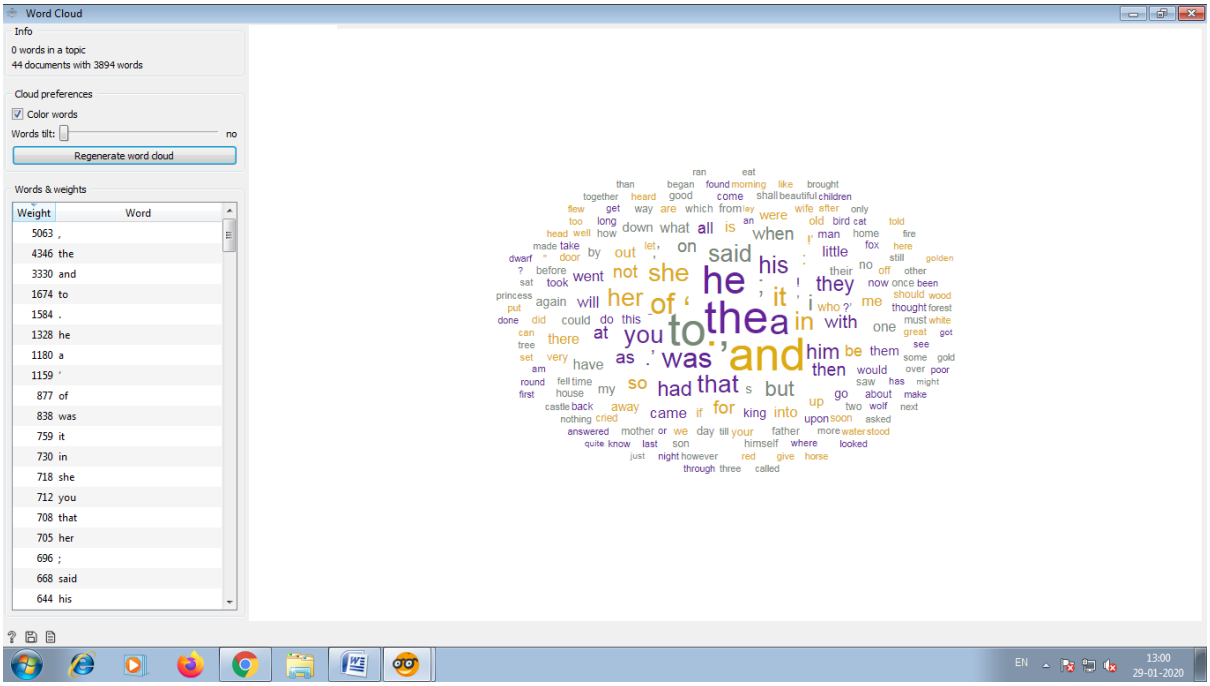
9)Another widget for visualizing the text is word cloud.



MSC CS - I

Name: Merin Kurian
Roll No.: 20

10)The widget displays word frequencies in a cloud.The more frequently the word appears in the text the larger the word will be. Our word cloud show words such as punctuation and uninformative words.

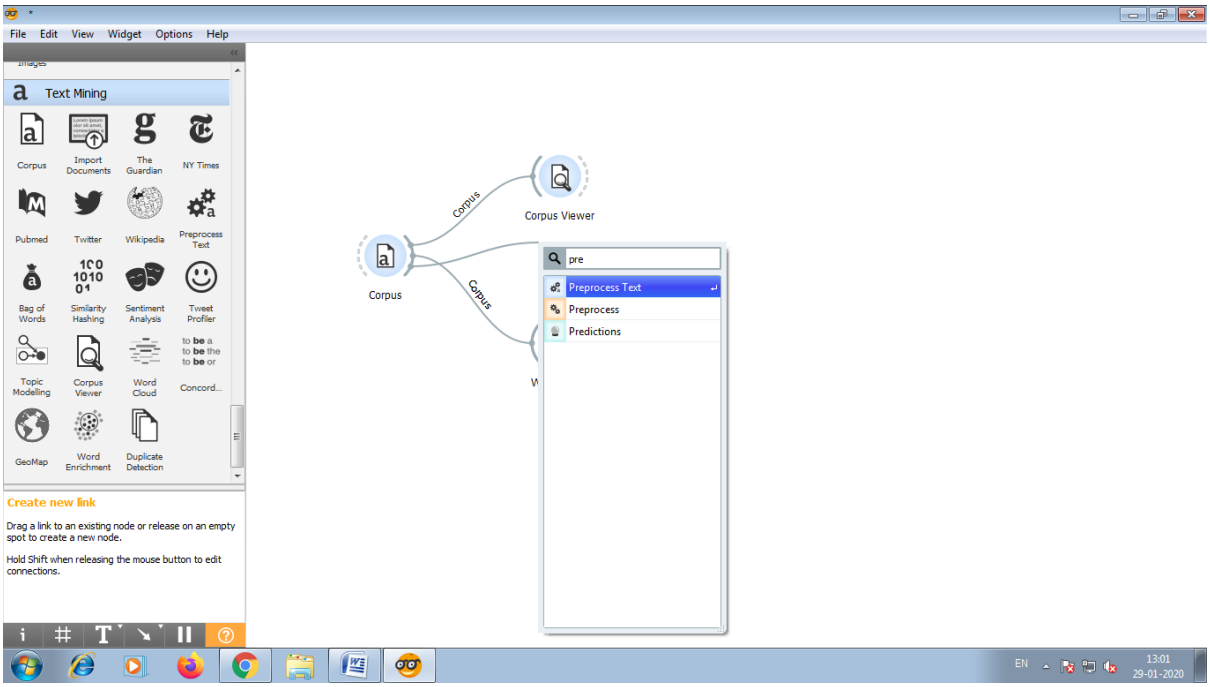


11)To getting rid of this we will now use preprocess text.

MSC CS - I

Name: Merin Kurian

Roll No.: 20

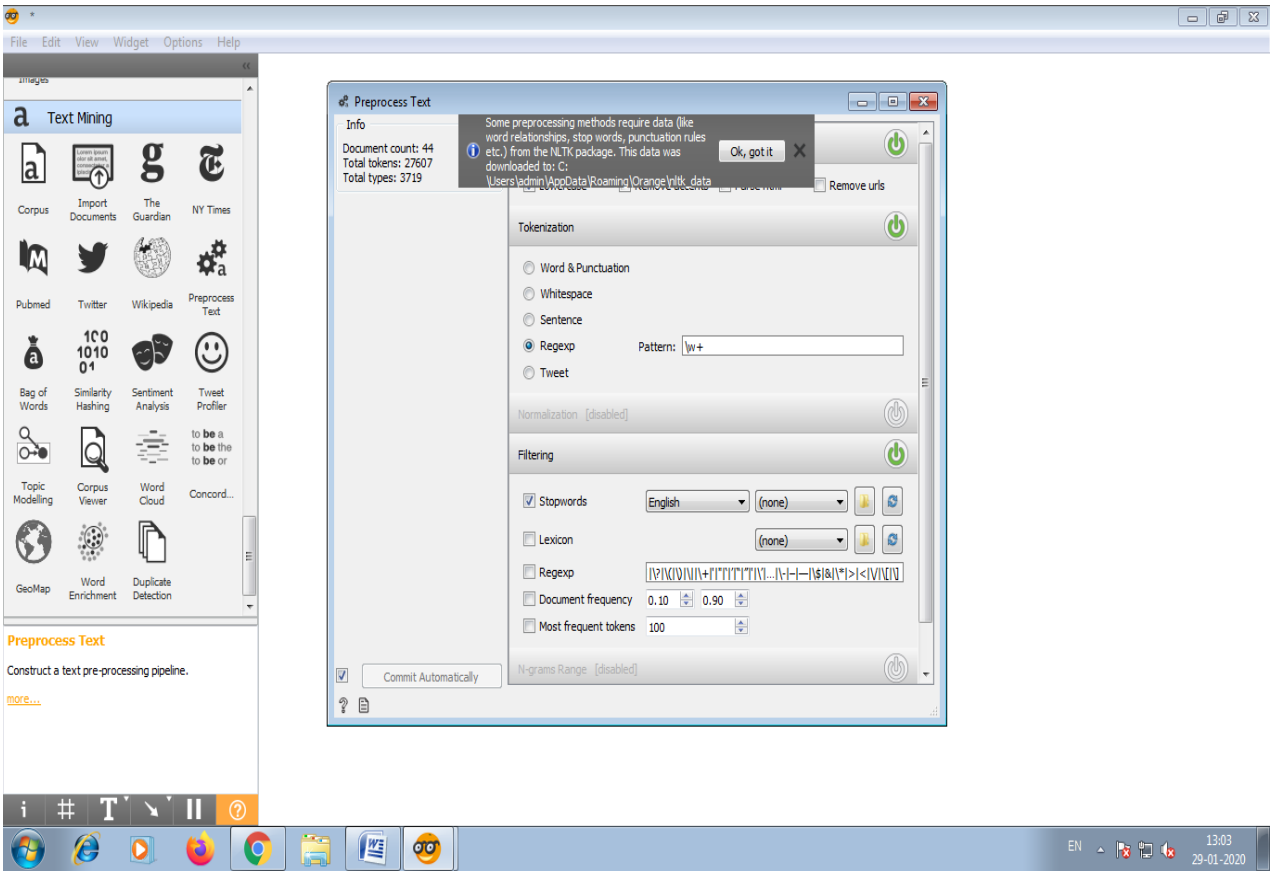


12)This widget will transform all text to lowercase. Next it will convert text into individuals words and omit the punctuation.

Individuals word are called tokens.

Finally it will filter out stopwords.

Name: Merin Kurian
Roll No.: 20



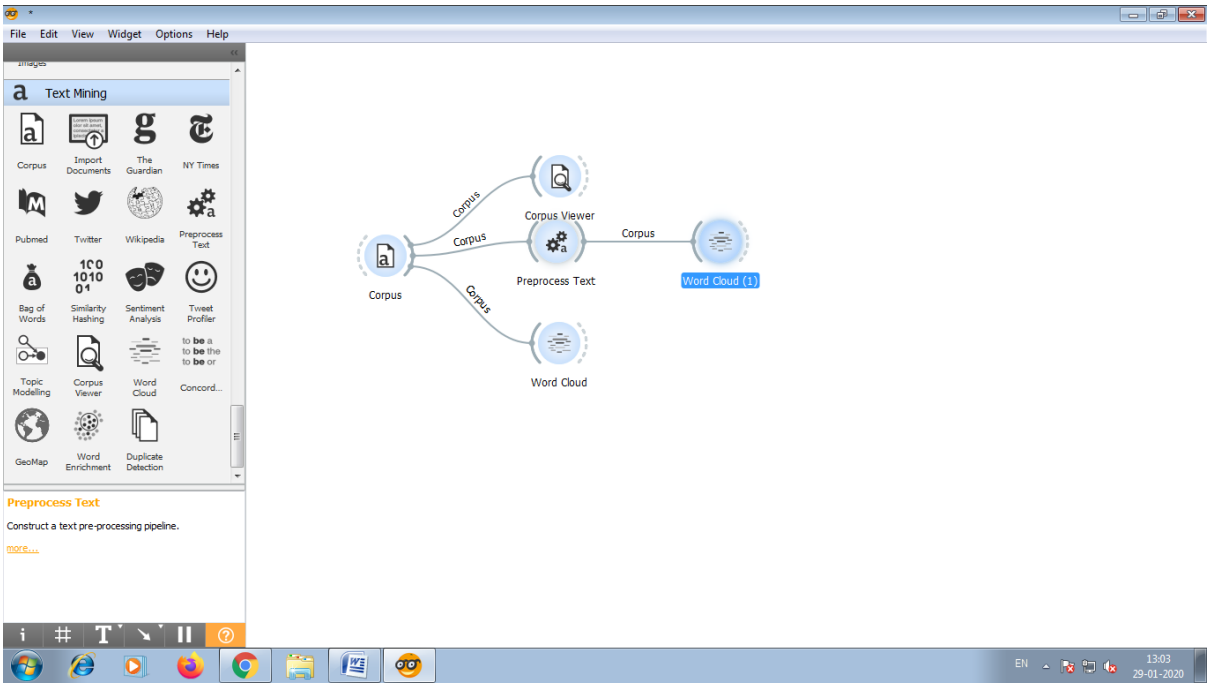
13)The effects of preprocessing can be visually explored in the word cloud.After preprocessing this visualization looks much better.

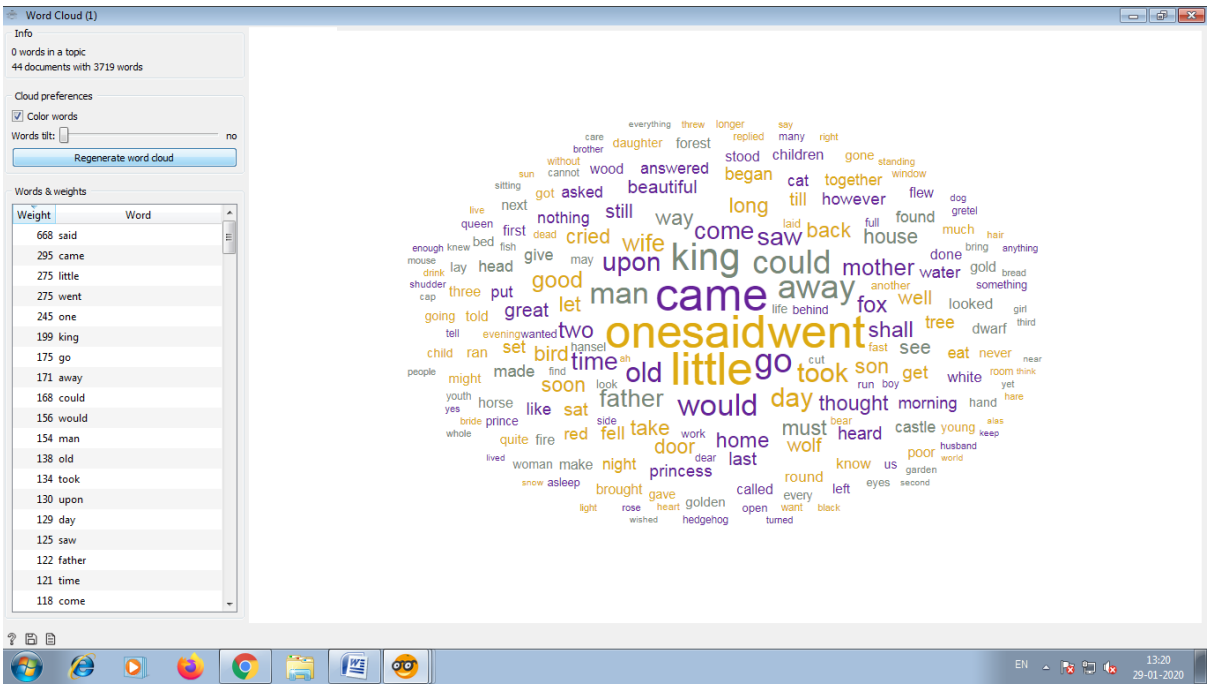
14)Connect another word cloud with preprocess Text. In this we retained only meaningful words.

MSC CS - I

Name: Merin Kurian

Roll No.: 20





MSC CS - I

Name: Merin Kurian
Roll No.: 20

Practical No.: 6

Prediction in orange

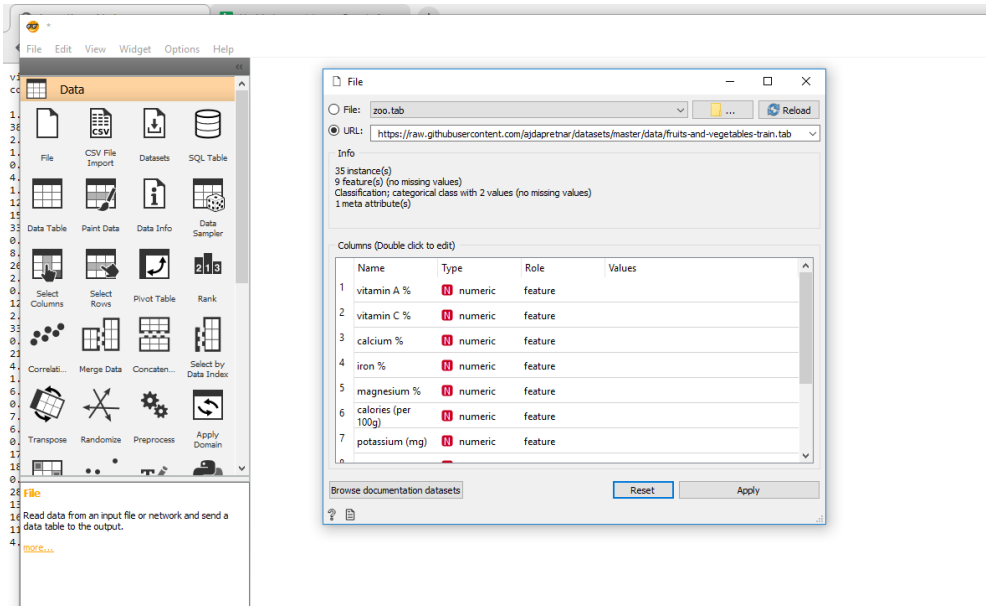
Aim: To predict whether the given dataset is of fruit or vegetable.

Description:

This dataset contain numerous feature which diffrenciate whether the given dataset is of vegetable or fruit. The target variable is classification i.e Is it fruit or vegetable.

Data selection:

We have given the url of the dataset which contain 35 instances and 9 features of fruit and vegetables.

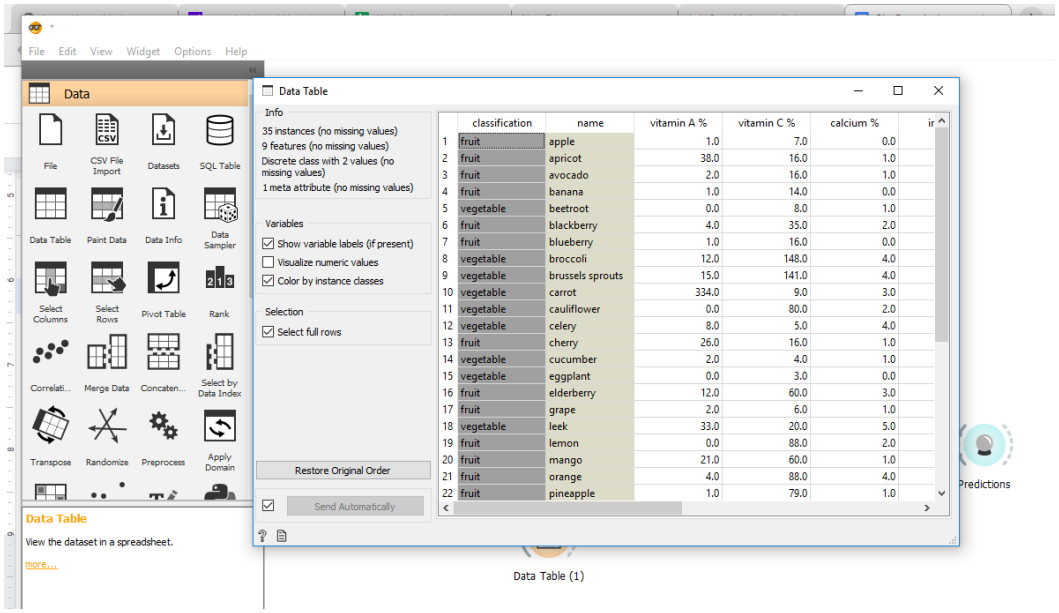


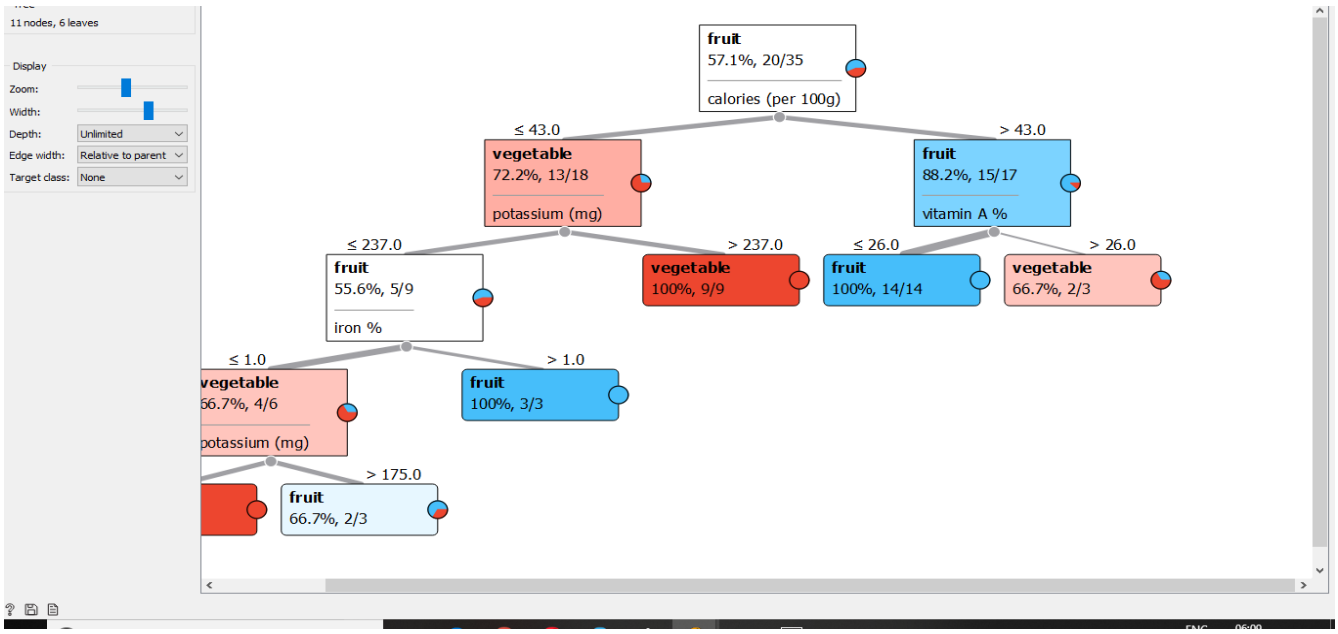
MSC CS - I

Name: Merin Kurian
Roll No.: 20

Data visualization:

We visualize the data by using data table and classification tree.

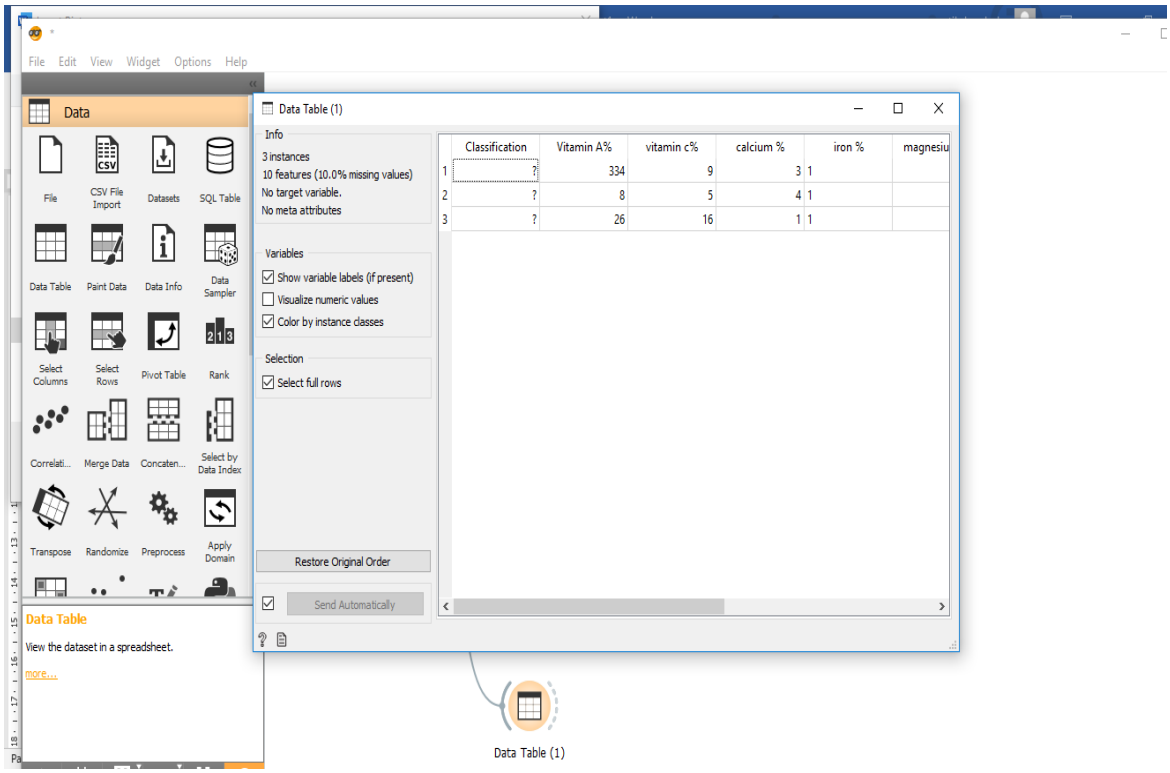




MSC CS - I

Name: Merin Kurian
Roll No.: 20

Again ,You have to create the the dataset of plants which you need to predict and visualize it in data table.

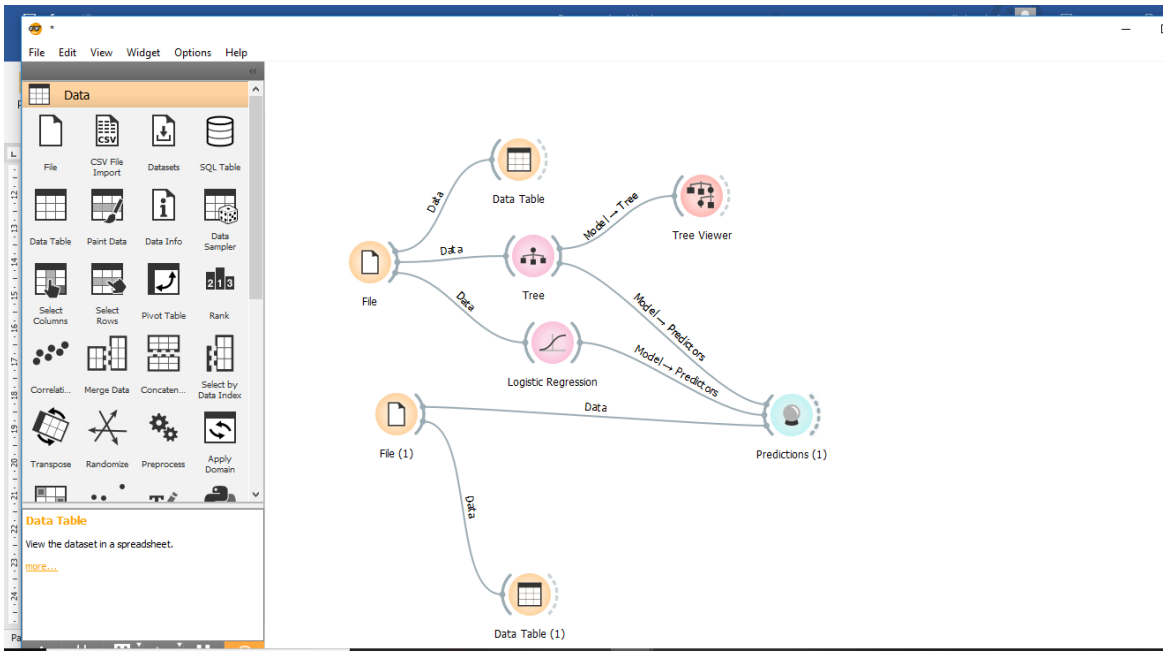


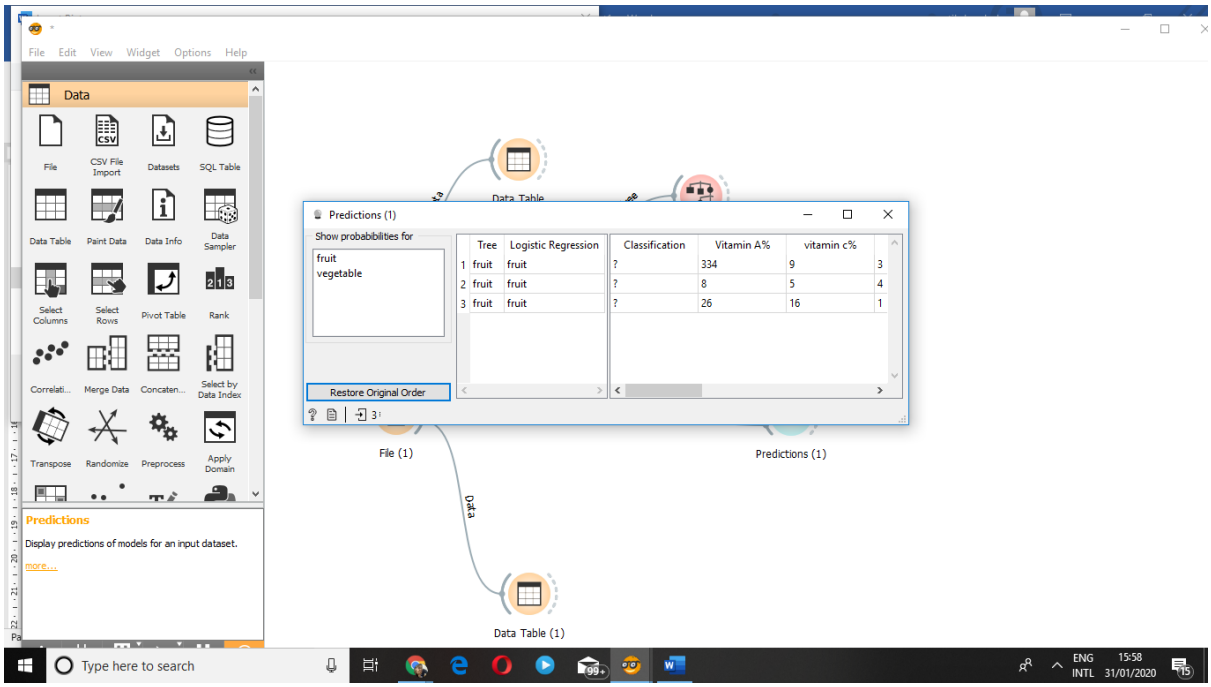
By using prediction widget you need to predict whether the plant is vegetable or fruit .

MSC CS - I

Name: Merin Kurian

Roll No.: 20





The given prediction is again checked by logistic regression.

MSC CS - I

Name: Merin Kurian
Roll No.: 20

Practical No.: 7

Rain Tomorrow Prediction in Orange

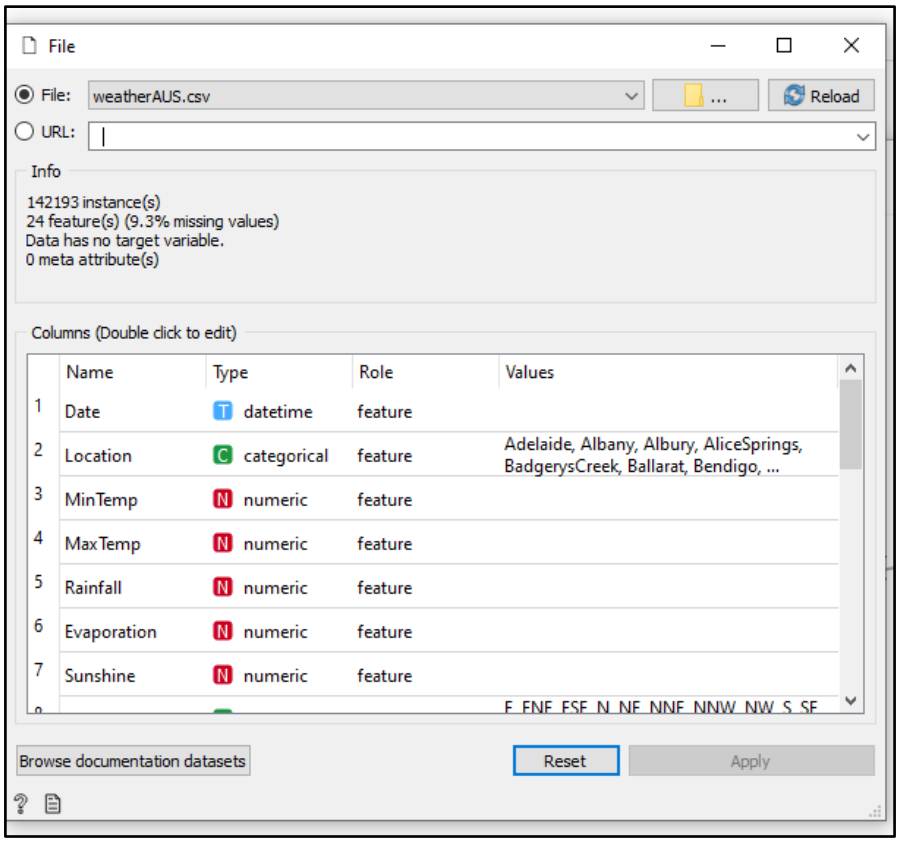
Aim: *Predict whether or not it will rain tomorrow by training a prediction model on target variable RainTomorrow*

Description*: This dataset contains daily weather observations from numerous Australian weather stations.*

The target variable RainTomorrow means: Did it rain the next day? Yes or No.

Data Selection*:*

We have downloaded the dataset (weather dataset) of Australian weather stations which contains about 10 years of daily weather observations from numerous Australian weather stations.

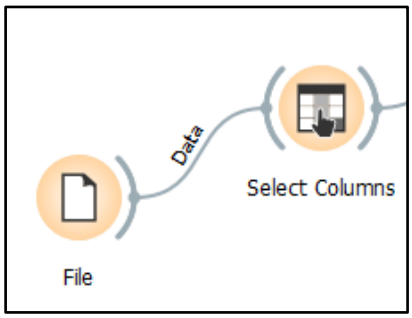


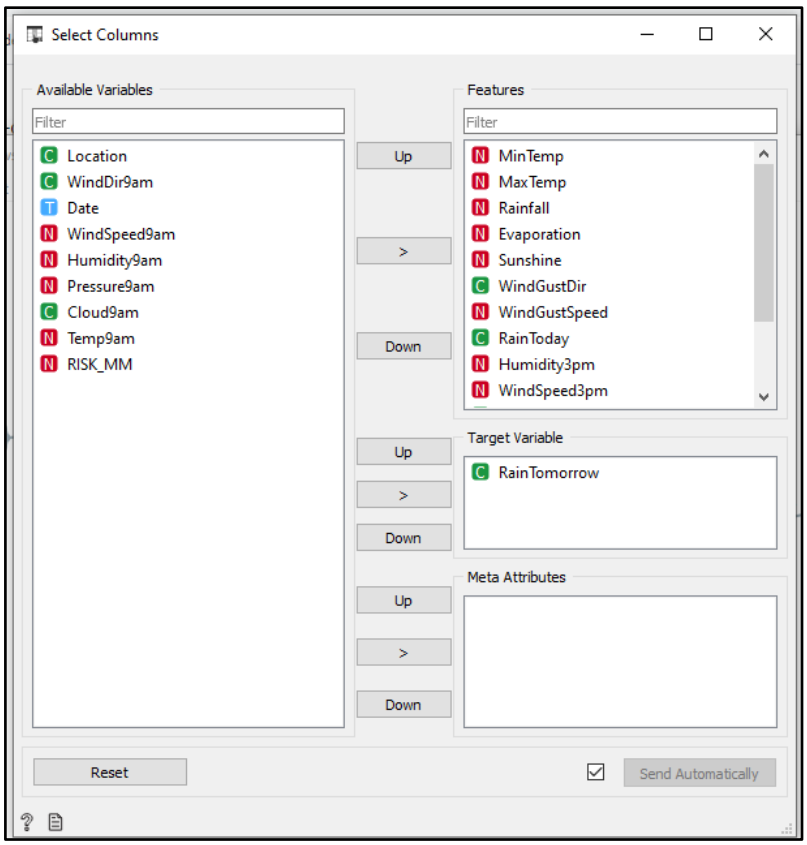
Name: Merin Kurian
Roll No.: 20

Data Cleaning:

In data cleaning, we will select the required column and remove the rows with missing values as shown below:

Column Selection

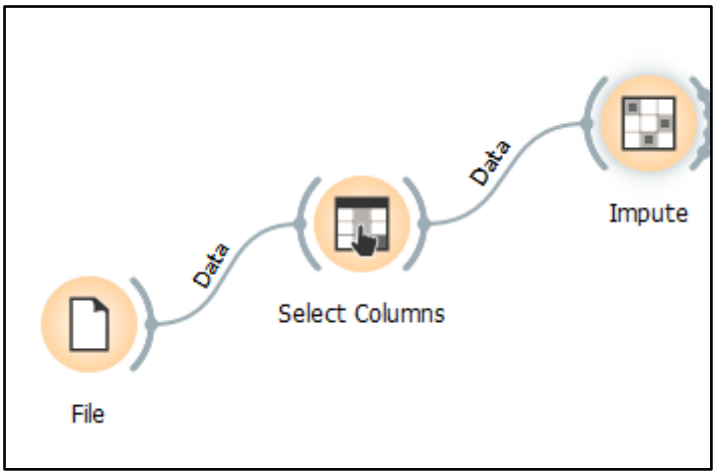


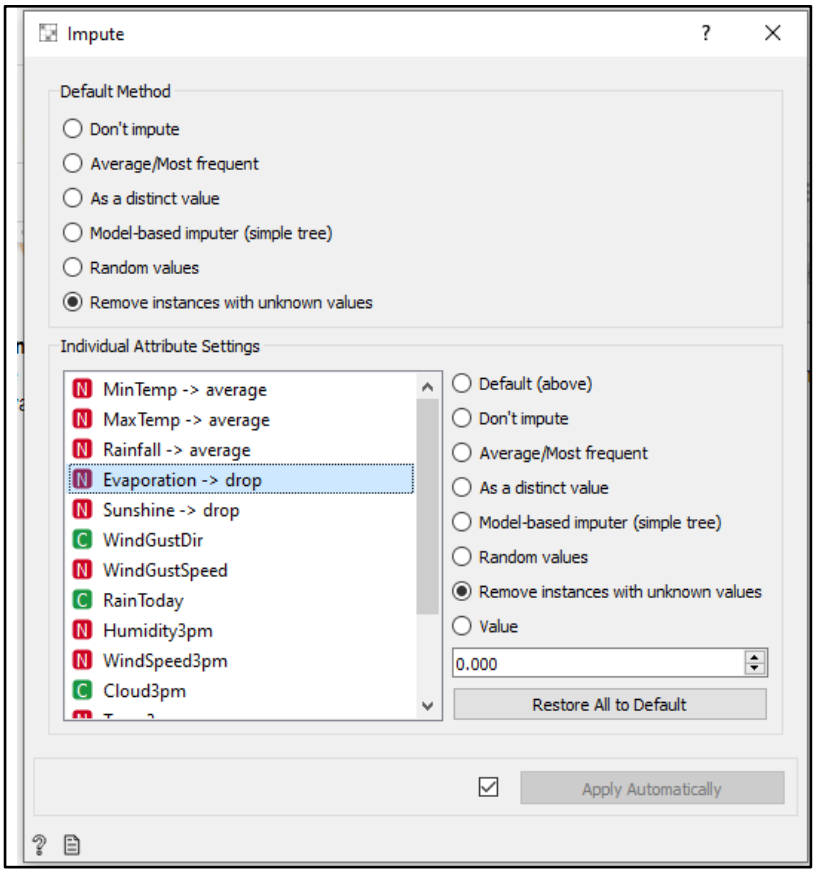


Removing rows with missing values

Name: Merin Kurian
Roll No.: 20

We will use the impute function to remove rows with missing values for columns “Sunshine” and “Evaporation”





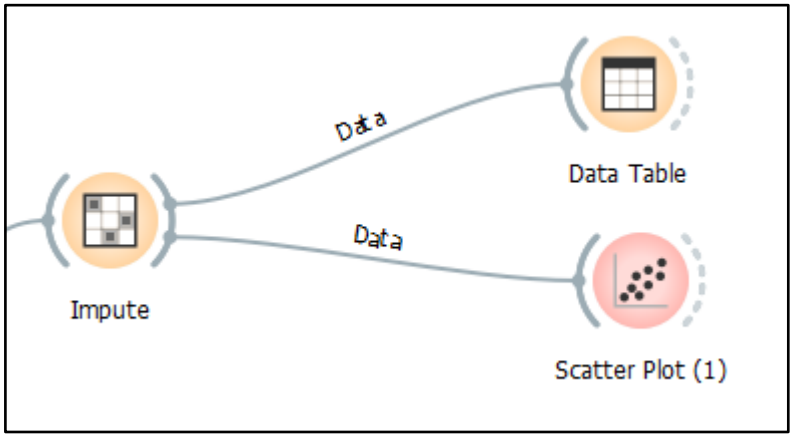
Data Visualization

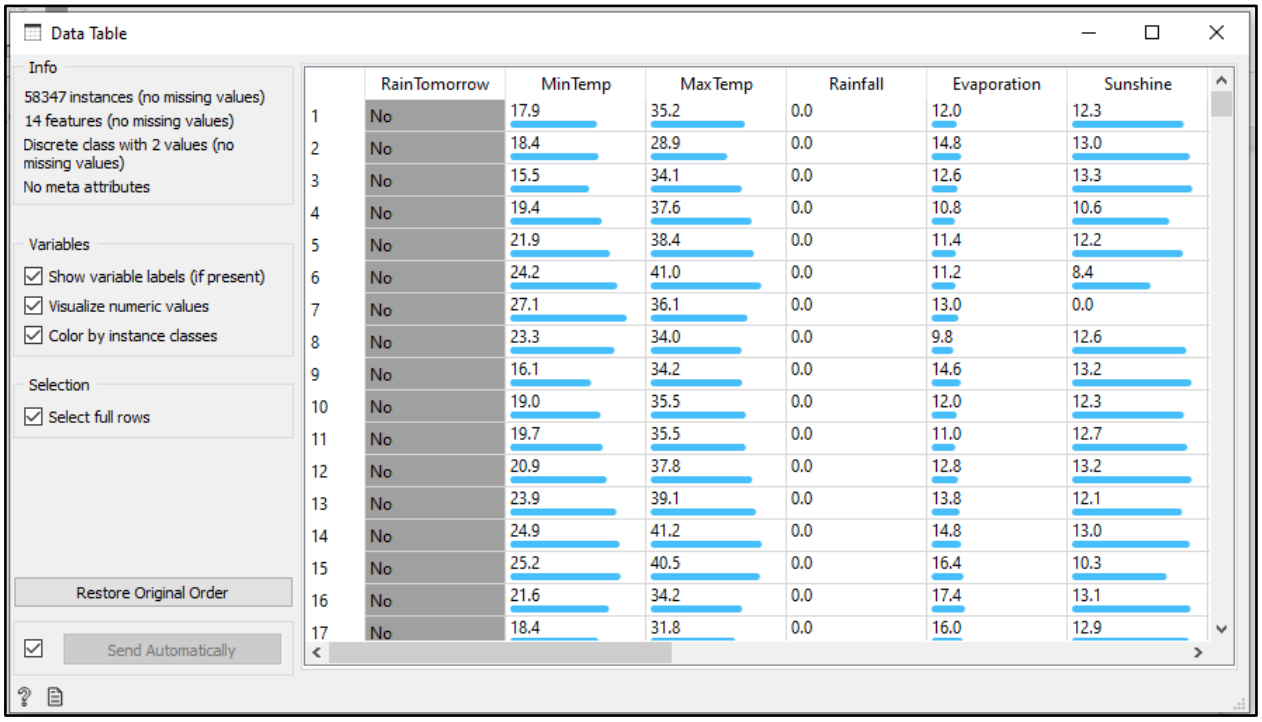
In this step, we see the reflected data selection in data table. Then, we use scatter plot to visualize the data for better understanding of our dataset.

MSC CS - I

Name: Merin Kurian

Roll No.: 20

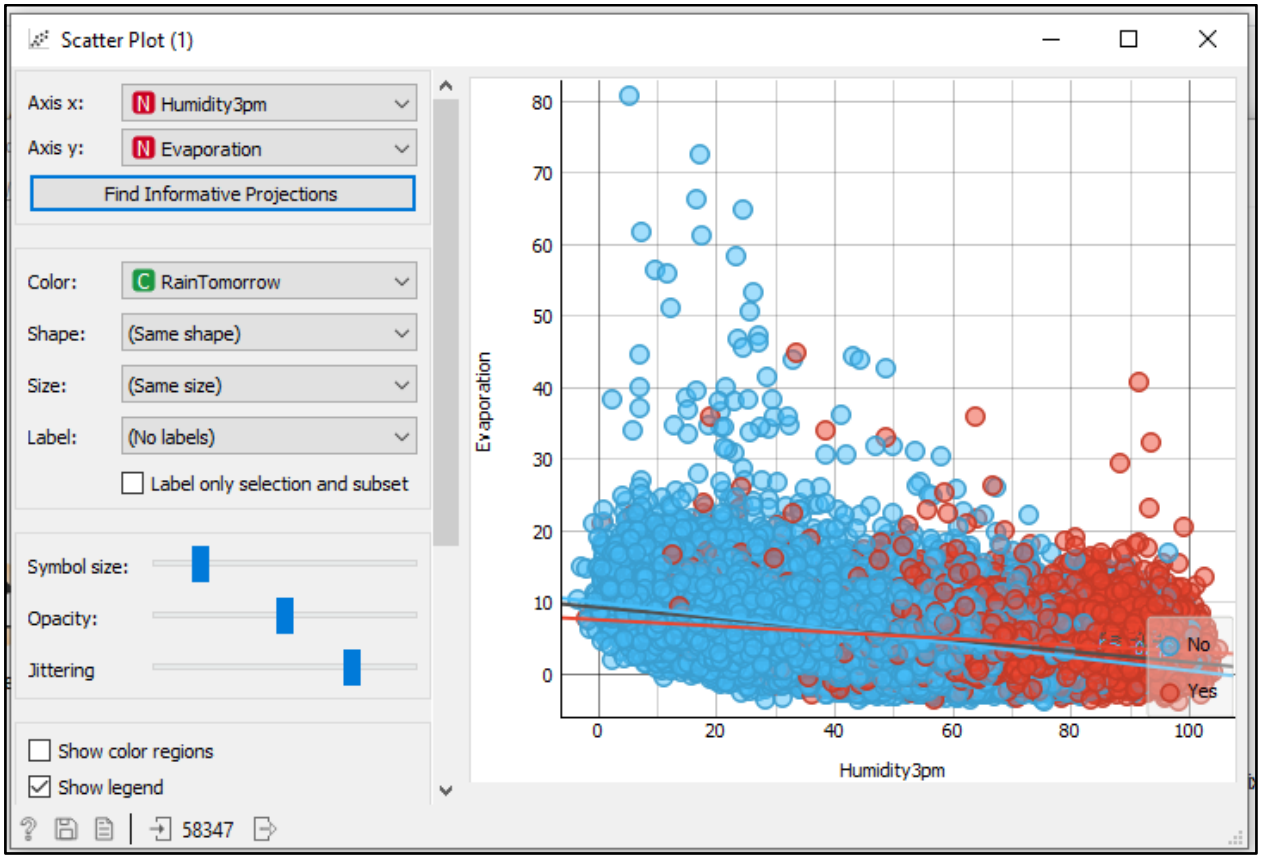




	RainTomorrow	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine
1	No	17.9	35.2	0.0	12.0	12.3
2	No	18.4	28.9	0.0	14.8	13.0
3	No	15.5	34.1	0.0	12.6	13.3
4	No	19.4	37.6	0.0	10.8	10.6
5	No	21.9	38.4	0.0	11.4	12.2
6	No	24.2	41.0	0.0	11.2	8.4
7	No	27.1	36.1	0.0	13.0	0.0
8	No	23.3	34.0	0.0	9.8	12.6
9	No	16.1	34.2	0.0	14.6	13.2
10	No	19.0	35.5	0.0	12.0	12.3
11	No	19.7	35.5	0.0	11.0	12.7
12	No	20.9	37.8	0.0	12.8	13.2
13	No	23.9	39.1	0.0	13.8	12.1
14	No	24.9	41.2	0.0	14.8	13.0
15	No	25.2	40.5	0.0	16.4	10.3
16	No	21.6	34.2	0.0	17.4	13.1
17	No	18.4	31.8	0.0	16.0	12.9

MSC CS - I

Name: Merin Kurian
Roll No.: 20



In the above scatter plot we see, as the humidity level increases and evaporation rates are low the chances of rainfall increases (red dots)

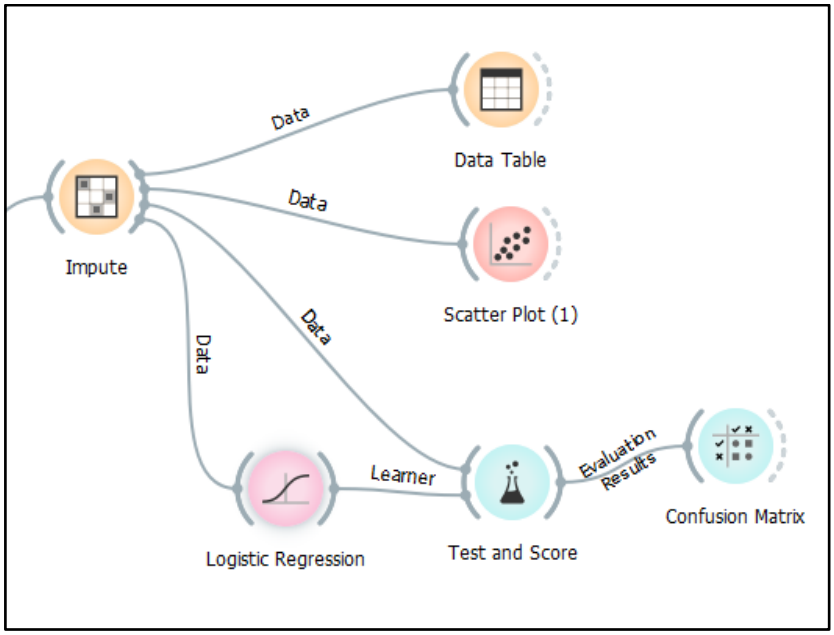
Data Modeling

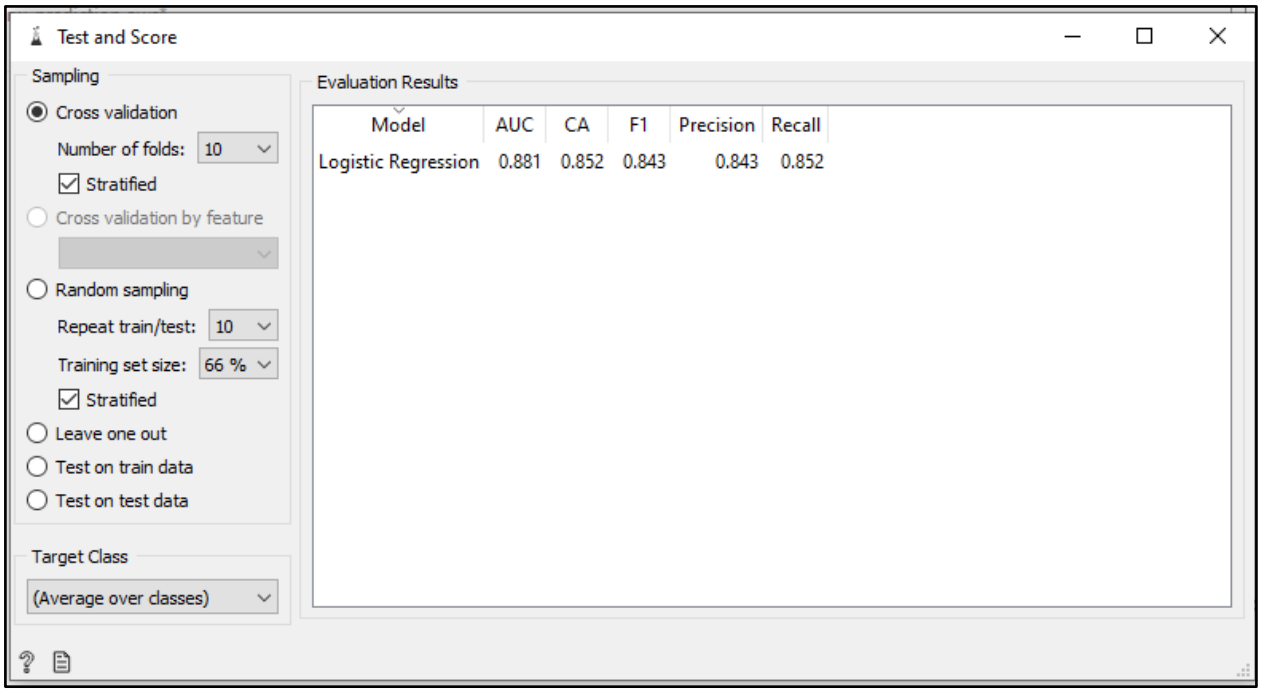
In this step, we use **Logistic regression** to model our data. Then, we use "test and score" function to evaluate the performance of our model. Finally, we use "confusion matrix" to understand the prediction.

MSC CS - I

Name: Merin Kurian

Roll No.: 20

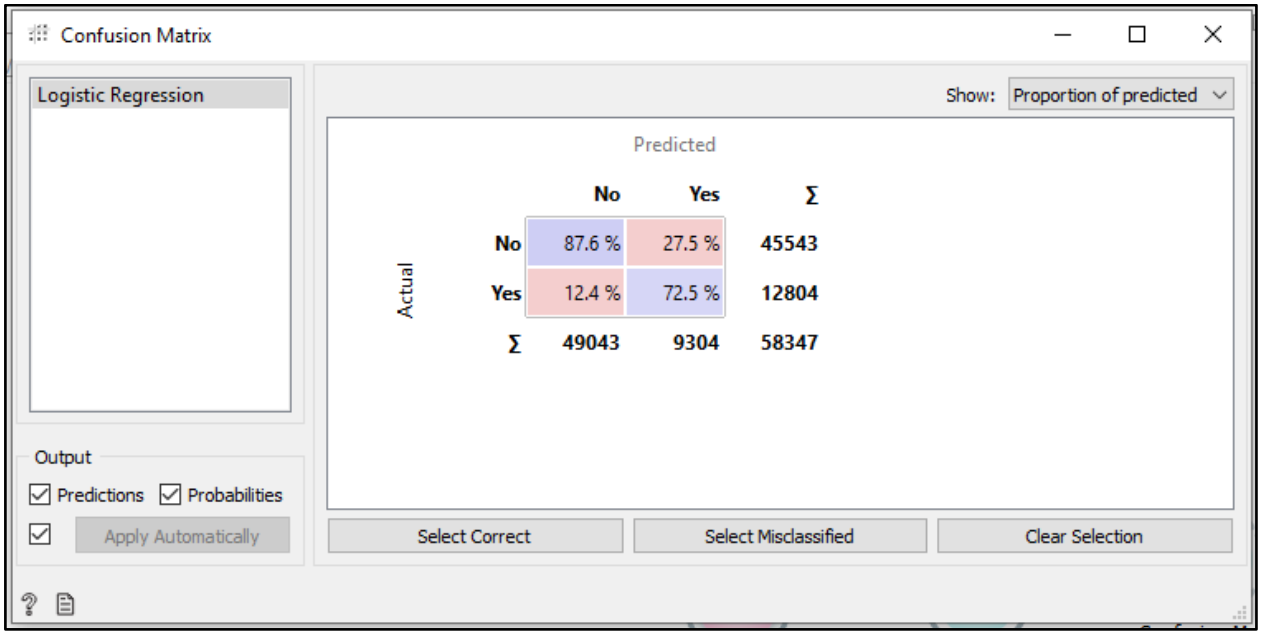




MSC CS - I

Name: Merin Kurian

Roll No.: 20



Name: Merin Kurian
Roll No.: 20

Practical No.: 8

Aim: Write a java program to Calculate the Alon Matias Szegedy Algorithm for given stream.

Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package amsa;
import java.io.*;
import java.util.*;
/**
 *
 * @author Merin
 */
public class AMSA {
   public static int findCharCount(String stream, char XE, int random, int length) {
      int countOccurence = 0;
      for(int i = random; i < length; i++)
      {
         if(stream.charAt(i) == XE)
         {
            countOccurence++;
         }
```

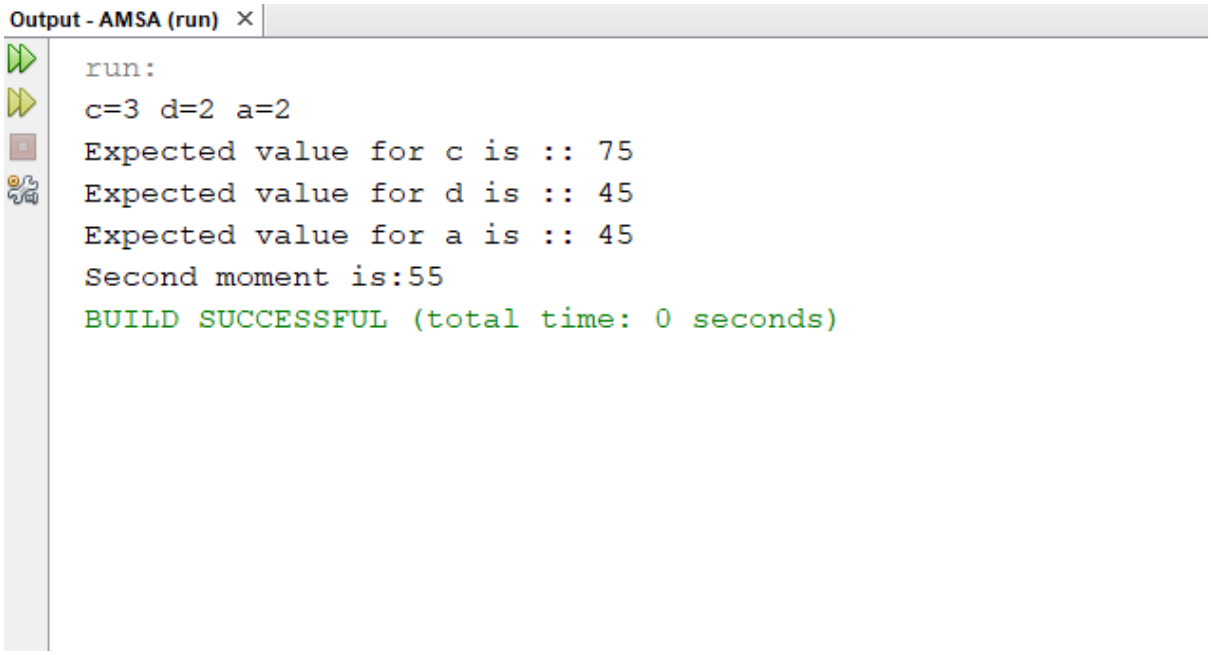
Name: Merin Kurian
Roll No.: 20

```
        }
        return countOccurence;
    }
    public static int estimateValue(int XV1, int n) {
        int expValue;
        expValue = n * (2 * XV1 - 1);
        return expValue;
    }

    public static void main(String[] args) {
        int n = 15;
        String stream = "abcbdacdabdcaab";
        int randomValues[] = {3, 8, 13};
        char[] XE = new char[3];
        int[] XV = new int[3];
        int[] expValue = new int[3];
        int apprSecondMomentValue;
        for(int i = 0; i < randomValues.length; i++)
        {
            XE[i] = stream.charAt(randomValues[i] - 1);
        }
        for(int i = 0; i < randomValues.length; i++)
        {
            XV[i] = findCharCount(stream, XE[i], randomValues[i] - 1, n);
        }
```

Name: Merin Kurian
Roll No.: 20

```
    System.out.println(XE[0] + "=" + XV[0] + " " + XE[1] + "=" + XV[1] + " " + XE[2] + "=" +
XV[2]);

    for(int i = 0; i < randomValues.length; i++)

    {

       expValue[i] = estimateValue(XV[i], n);

    }

    for(int i = 0; i < randomValues.length; i++)

    {

       System.out.println("Expected value for "+XE[i]+" is :: "+expValue[i]);

    }

    apprSecondMomentValue = Arrays.stream(expValue).sum() / 3;

    System.out.println("Second moment is:" + apprSecondMomentValue);

  }

}
```

Output:



MSC CS - I

Name: Merin Kurian
Roll No.: 20

Practical No.: 9

Aim: Write a Program to Construct different types of K-shingles for a given document

Code:

```
require("tm")


kshingle<-function(){
  k<- as.integer(readline("Enter a value for k - 1"))


  u1<- readLines("D:/Git/Sem 2/Practicals/BIBD/Practical 9/BIBD_1.txt")


  shingle<-i<-0
  while(i<nchar(u1)-k+1)
  {
    shingle[i]<-substr(u1,i,i+k)
    print(shingle[i])


    i<-i+1
  }
}


if(interactive())kshingle()
```

Output:

MSC CS - I

Name: Merin Kurian

Roll No.: 20

```
> require("tm")
> kshingle<-function(){
+   k<- as.integer(readline("Enter a value for k - 1"))
+
+   u1<- readLines("D:/Git/Sem 2/Practicals/BIBD/Practical 9/BIBD_1.txt")
+
+   shingle<-i<-0
+   while(i<nchar(u1)-k+1)
+   {
+     shingle[i]<-substr(u1,i,i+k)
+     print(shingle[i])
+
+     i<-i+1
+   }
+ }
> if(interactive())kshingle()
Enter a value for k - 12
character(0)
[1] "Lor"
[1] "ore"
[1] "rem"
[1] "em "
[1] "m i"
[1] " ip"
[1] "ips"
[1] "psu"
[1] "sum"
[1] "um "
[1] "m d"
[1] " do"
[1] "dol"
[1] "olo"
[1] "lor"
[1] "or "
[1] "r s"
```

Name: Merin Kurian
Roll No.: 20

```
[1] "unt"
[1] "nt "
[1] "t m"
[1] " mo"
[1] "mol"
[1] "oll"
[1] "lli"
[1] "lit"
[1] "it "
[1] "t a"
[1] " an"
[1] "ani"
[1] "nim"
[1] "im "
[1] "m i"
[1] " id"
[1] "id "
[1] "d e"
[1] " es"
[1] "est"
[1] "st "
[1] "t l"
[1] " la"
[1] "lab"
[1] "abo"
[1] "bor"
[1] "oru"
[1] "rum"
[1] "um."
Warning message:
In readLines("D:/Git/Sem 2/Practicals/BIBD/Practical 9/BIBD_1.txt") :
  incomplete final line found on 'D:/Git/Sem 2/Practicals/BIBD/Practical 9/BIBD_
1.txt'
```

MSC CS - I

Name: Merin Kurian
Roll No.: 20

Practical 10

Aim: Write a program for measuring similarity among documents and detecting passages which have been reused.

Code:

```
install.packages("Corpus")

install.packages("ggplot2")

install.packages("textreuse")

install.packages("devtools")


install.packages("tm")

require("tm")

install.packages("corpus")

install.packages("ggplot2")

install.packages("textreuse")

install.packages("devtools")


my.corpus<-Corpus(DirSource("Text Files"))

my.corpus<-tm_map(my.corpus,removeWords,stopwords("english"))

my.tdm<-TermDocumentMatrix(my.corpus)

#inspect(my.tdm)

my.dtm<-
DocumentTermMatrix(my.corpus,control=list(weighting=weightTfldf,stopwords=TRUE))

#inspect(my.dtm)

my.df<-as.data.frame(inspect(my.tdm))

my.df.scale<-scale(my.df)
```

MSC CS - I

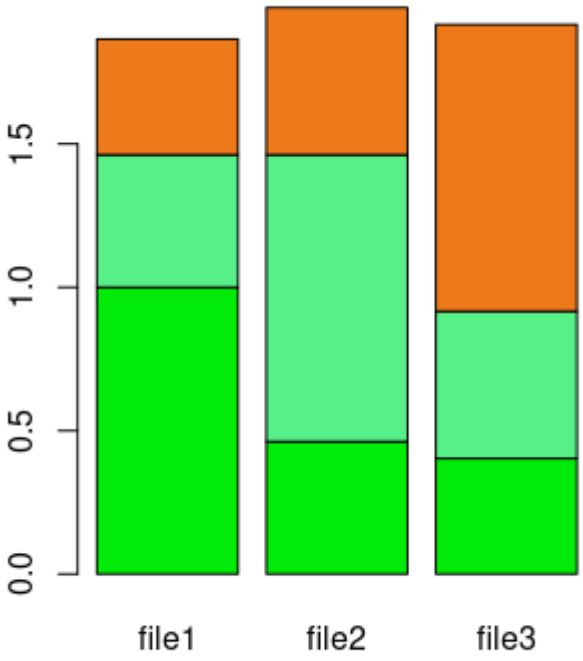
Name: Merin Kurian
Roll No.: 20

```
d<-dist(my.df.scale,method = "euclidean")

fit<-hclust(d,method = "ward.D")

plot(fit)


my.corpus<-Corpus(DirSource("/cloud/project/Text Files"))

my.corpus<-tm_map(my.corpus,removeWords,stopwords("english"))

my.tdm<-TermDocumentMatrix(my.corpus)

inspect(my.tdm)

my.df<-as.data.frame(inspect(my.tdm))

barplot(as.matrix(my.tdm))

#barplot(as.matrix(my.tdm),col=color)

barplot(as.matrix(my.tdm),col= c("Red","Green","Blue"))


library("textreuse")


minhash <- minhash_generator(200, seed = 235)

ats <- TextReuseCorpus(dir = "files", tokenizer = tokenize_ngrams, n = 5, minhash_func =
minhash)


buckets <- lsh(ats, bands = 50, progress = interactive())

candidates <- lsh_candidates(buckets)

scores <- lsh_compare(candidates, ats, jaccard_similarity, progress = F)

scores


barplot(as.matrix(scores), col = c("#00eb07", "#57ef87", "#ed791a", "#5e5fff", "#1cf1c6",
"#5e035b"))
```

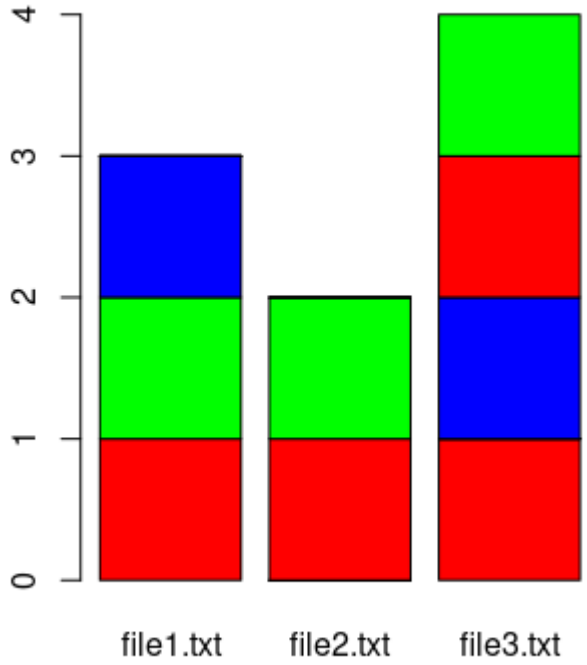
MSC CS - I

Name: Merin Kurian
Roll No.: 20

Output:

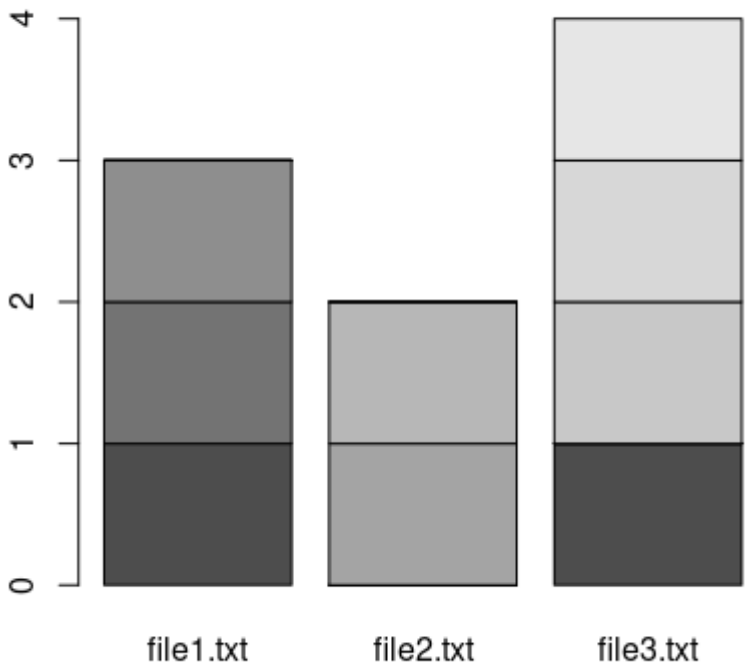



```
# A tibble: 3 × 3
  a     b     score
  <chr> <chr> <dbl>
1 file1 file2 0.462
2 file1 file3 0.403
3 file2 file3 0.513
```

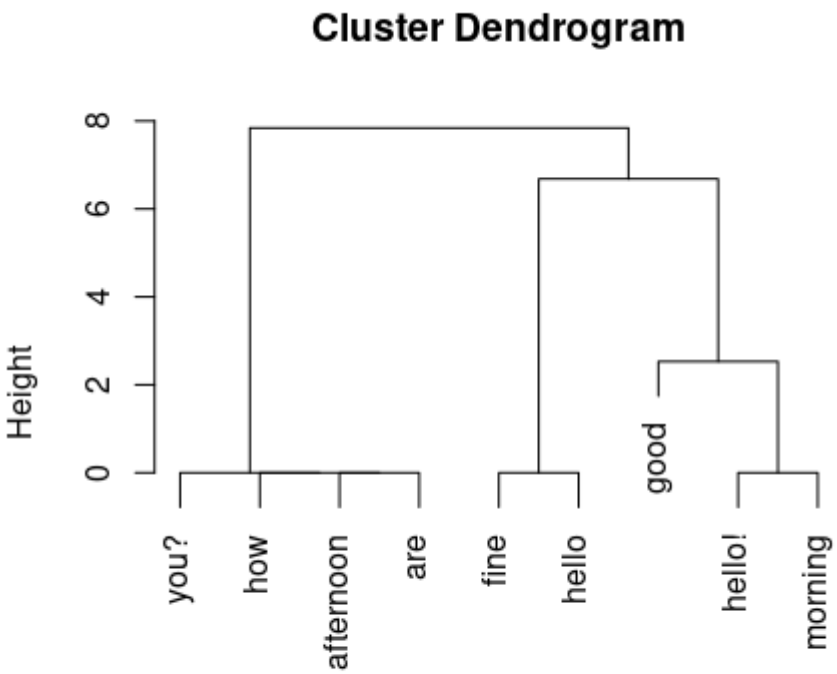


MSC CS - I

Name: Merin Kurian
Roll No.: 20



Name: Merin Kurian
Roll No.: 20



d
hclust (*, "ward.D")

```
<<TermDocumentMatrix (terms: 9, documents: 3)>>
Non-/sparse entries: 10/17
Sparsity           : 63%
Maximal term length: 9
Weighting          : term frequency (tf)
Sample             :
           Docs
Terms       file1.txt file2.txt file3.txt
  afternoon         0         0         1
  are               0         0         1
  fine              0         1         0
  good              1         0         1
  hello             0         1         0
  hello!            1         0         0
  how               0         0         1
  morning           1         0         0
  you?              0         0         1
> my.df.scale<-scale(my.df)
> d<-dist(my.df.scale,method = "euclidean")
> fit<-hclust(d,method = "ward.D")
> plot(fit)
>
```

Name: Merin Kurian
Roll No.: 20

Practical No.: 11

Aim: Write a java Program to demonstrate the k-moments for zeroth, first and second moments.

Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package kmoments;
import java.io.*;
import java.util.*;

/**
 *
 * @author Merin
 */
public class KMoments {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int n = 15;
        String stream[] = {"a","b","c","b","d","a","c","d","a","b","d","c","a","a","b"};
```

Name: Merin Kurian
Roll No.: 20

```
    int zerothMoment = 0, firstMoment = 0, secondMoment = 0, count = 1, flag = 0;
    ArrayList<Integer> arrayList = new ArrayList();
    for(String character : stream){
       System.out.print(character + "\t\t\t");
    }
    System.out.println();
    Arrays.sort(stream);
    for(int i = 1; i < n; i++)
  {
  if(stream[i] == stream[i - 1]){
    count++;
  }
  else
  {
    arrayList.add(count);
    count = 1;
  }
}
  arrayList.add(count);
  System.out.println("Zeroth moment:\t\t\t" + zerothMoment);
  for(int i : arrayList){
    firstMoment += i;
    secondMoment += i * i;
  }
  System.out.println("First Moment:\t\t\t" + firstMoment);
```

Name: Merin Kurian
Roll No.: 20

```
    System.out.println("Second Moment:\t\t\t" + secondMoment);

}



}
```

Output: