DMAT – Assignment 1

Name: Aravindan Srinivasan

Student Number: ­2981707

|  |  |
| --- | --- |
| **Course** | MSCBD-DMAT |
| **Stage / Year** | 1 |
| **Module** | Data Mining Algorithms & Techniques |
| **Semester** | 2 |
| **Assignment** | Assignment 1 |
| **Date of Title Issue** | 20th Feb 2019 |
| **Assignment Deadline** | 2nd April 2019 |
| **Assignment Submission** | Upload to moodle |
| **Assignment Weighting** | 10% of module |

# Objective

To successfully apply a set of data mining skills imparted through lectures and lab session to a previously unseen dataset to achieve knowledge discovery.

# Deliverables

A single zip called firstName\_lastName\_studentNumber.\_ass1.zip to be uploaded to moodle containing the following files

* This file edited to contain the results of your investigation. Each of the **NUMBERED** headings should be expanded to satisfy the requirements of the section.
* A set of supporting files including but not limited to the following, which should be clearly referenced from your documentation.
  + dataset.arff
  + trainigSet.arff
  + testingSet.arff
  + j48tree.arff
  + associationrules.arff
  + kmeans.arff
  + dbscan.arff

# Choosing Your Dataset

1. Your dataset should concern a real-world problem that lends itself to easy understanding by your classmates.
2. It should ideally have >1000 tuples/rows/instances.
3. It should ideally have >=10 attributes
4. It should have attributes which can serve as labels so that the accuracy of your data analysis can be determined.
5. If you cannot find one dataset which is suitable for use with all techniques, then you may choose 2. Please clearly indicate which dataset was used in which case.

Your search for appropriate datasets can begin with <http://www.kdnuggets.com/datasets/index.html> please post to the student discussion forum “Assignment 1 - Dataset Selection” clearly indicating which set you are using so that other students do not select the same dataset.

# Part 1 – Classification

## 1. Description of your dataset and findings – 10%

* **Title**: Brief title to capture the data and objective of your assignment

HIT MOVIES VS FLOP MOVIES

* **Data description:** A description of the data in detail under the following subheadings:
  + The problem domain

Given the list of movies, with 18 attributes to determine the movie is hit or flop. If it is a hit, some attributes support the class attribute and some attribute do not support for flop class attribute. In our dataset, hit or flop is the class attribute.

* + The source of the data

The dataset was downloaded in this link (<https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset>). This dataset contains 5000 rows with list of attributes.

* + The agencies working with the data

This dataset contains the information of IMDB online movie rating. Team of IMDB updates and uses these data to keep track of all the information of the movies and gives rating from 1-10 on the basis of “How good a movie is”.

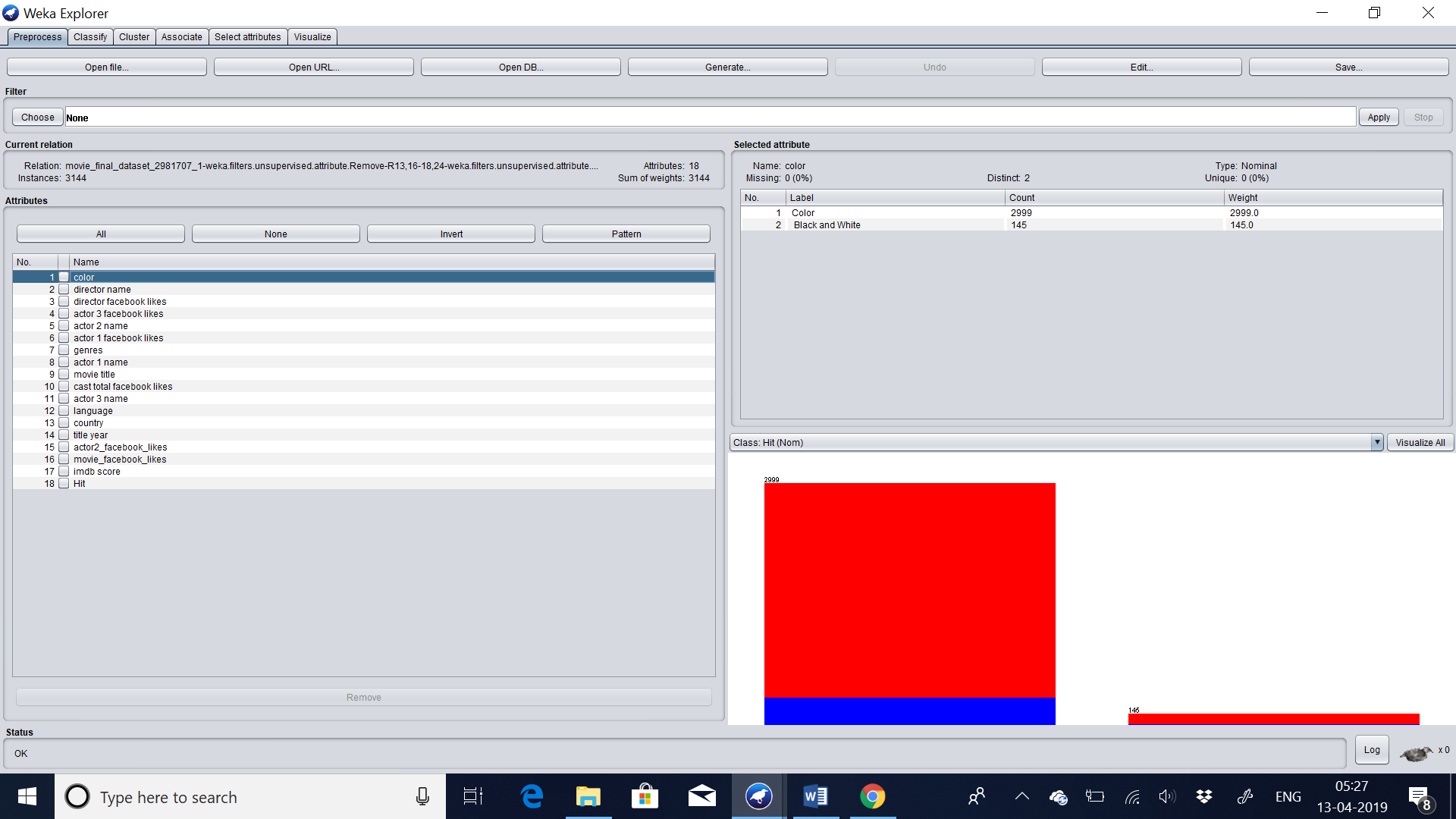
* + The intended use of the data

With this dataset, all of them can able to view information regarding each and every movie. They can find all the information of the movie such as movie name, movie rating, movie language etc.

* + The attribute types of the data

Initially all the attributes in the csv file was numeric. It consists of 18 variables like movie name, movie journal, actor Facebook likes, director Facebook likes movie budget etc. These data are very important to determine a movie is hit or flop or do some analysis.

Please include screen shots (with one or two sentences of summary) of the dataset and also of the data summaries that are available through Weka.



This dataset displays all the attributes name. And it displaces that it has 3144 number of instances. It depicts the relation of the table above the instances. And in the right it displays the insight of all the variables like, displaying the count of occurrence in the dataset.

* **Objective**: Your objective. You can update this as you progress through your project revising it and making it more specific.

We are finding the rows of which movies are hit or flop. We are going to do preprocessing, association, clustering, classification etc.

* **Summary of Findings**: This should be written following the application of your data mining techniques.

With the help of classification, we can able to find the rules of our findings. Total number of leaves is 5 and total size of our tree is 6. Now we can find our values which are allocated to the particular groups and can able to visualize tree.

imdb score = 1.6-3.18: no (42.0)

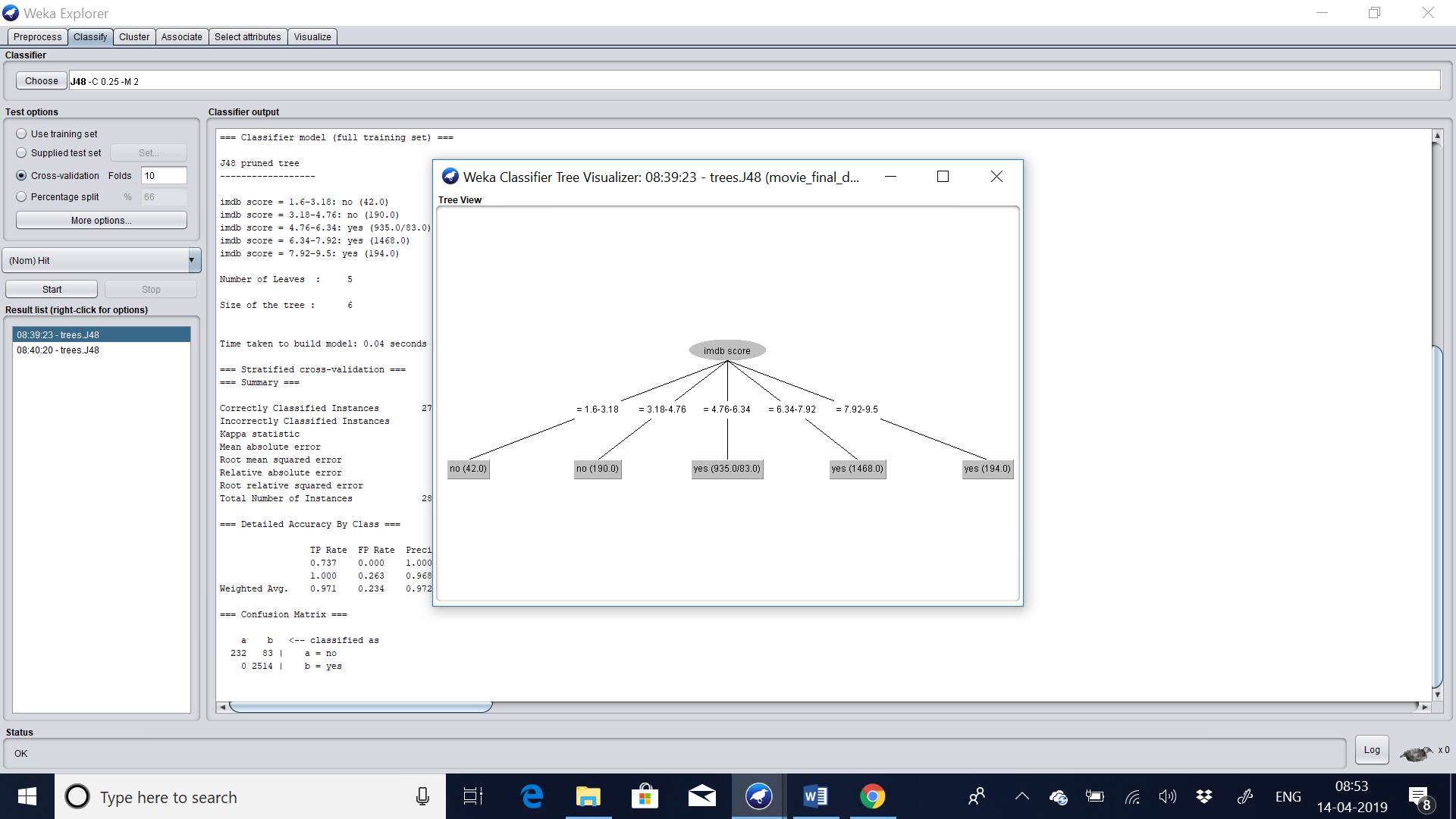
imdb score = 3.18-4.76: no (190.0)

imdb score = 4.76-6.34: yes (935.0/83.0)

imdb score = 6.34-7.92: yes (1468.0)

imdb score = 7.92-9.5: yes (194.0)

With the help of association, we can able to find the best 10 rules. Which will be useful to find the high total transaction values in the above dataset.



## 2. Preprocessing – 7%

In this section you should

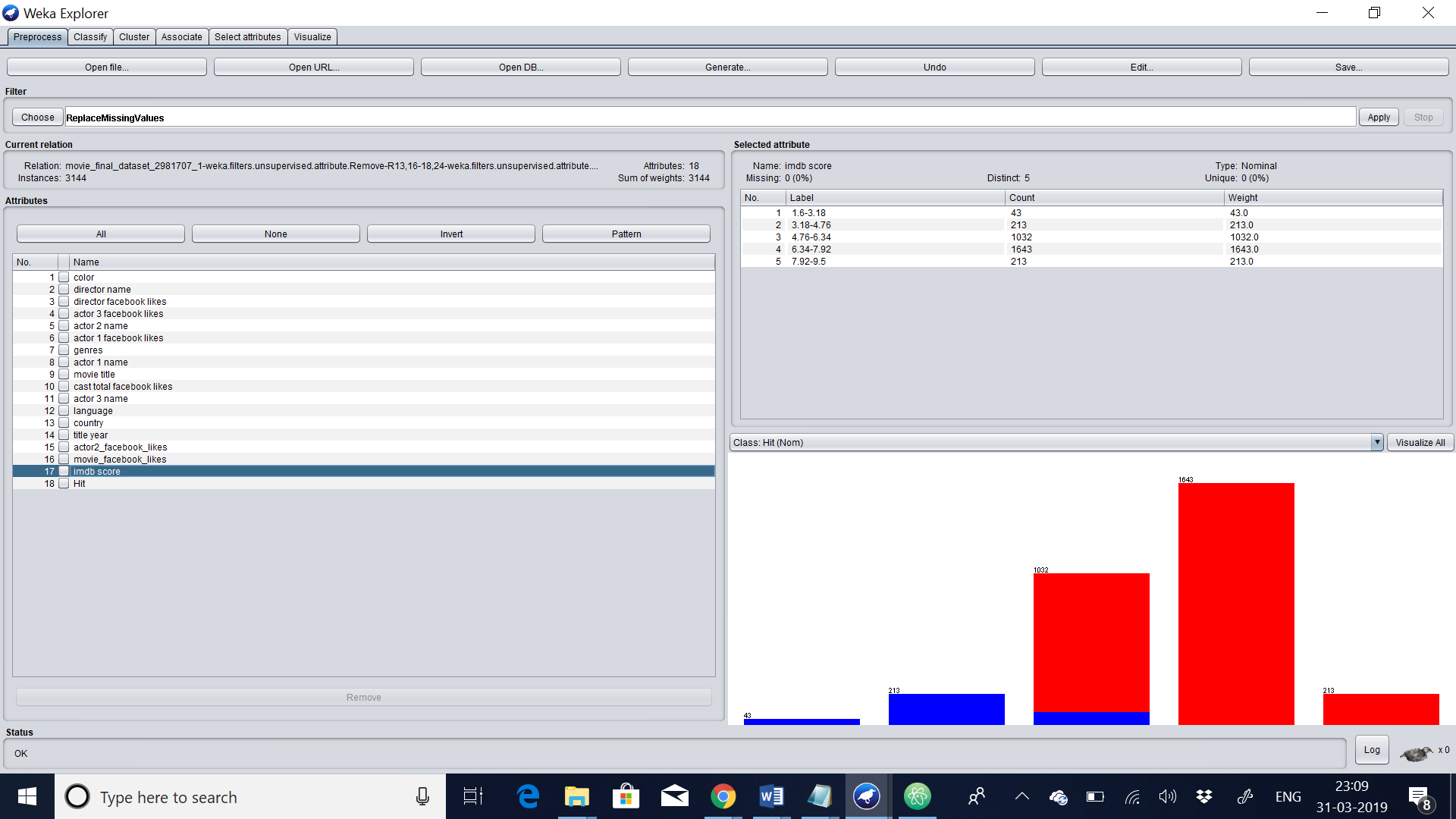
1. Identify the set of preprocessing techniques that can be applied to your data and clearly indicate which techniques are appropriate and which ones are not.

**ANSWER:**

**Initially, I filtered the attributes which are unique and not useful for data mining. Later used the technique of discretization, which helped me to convert numeric data into nominal data. And finally I have used ReplaceMissingValues to replace all the missing values with the mean value.**

1. Provide evidence through screenshot of the effects of preprocessing the data along with a short explanation.

**ANSWER:**



This is the screenshot after applying all the pre processing techniques, Such as removing unique columns such as id. And replace the missing values with median values. And finally converting all the numeric values into nominal values.

1. Generate a file called dataset.arff which is the outcome of the preprocessing.

## 3. Divide your dataset into training and test set – 3%

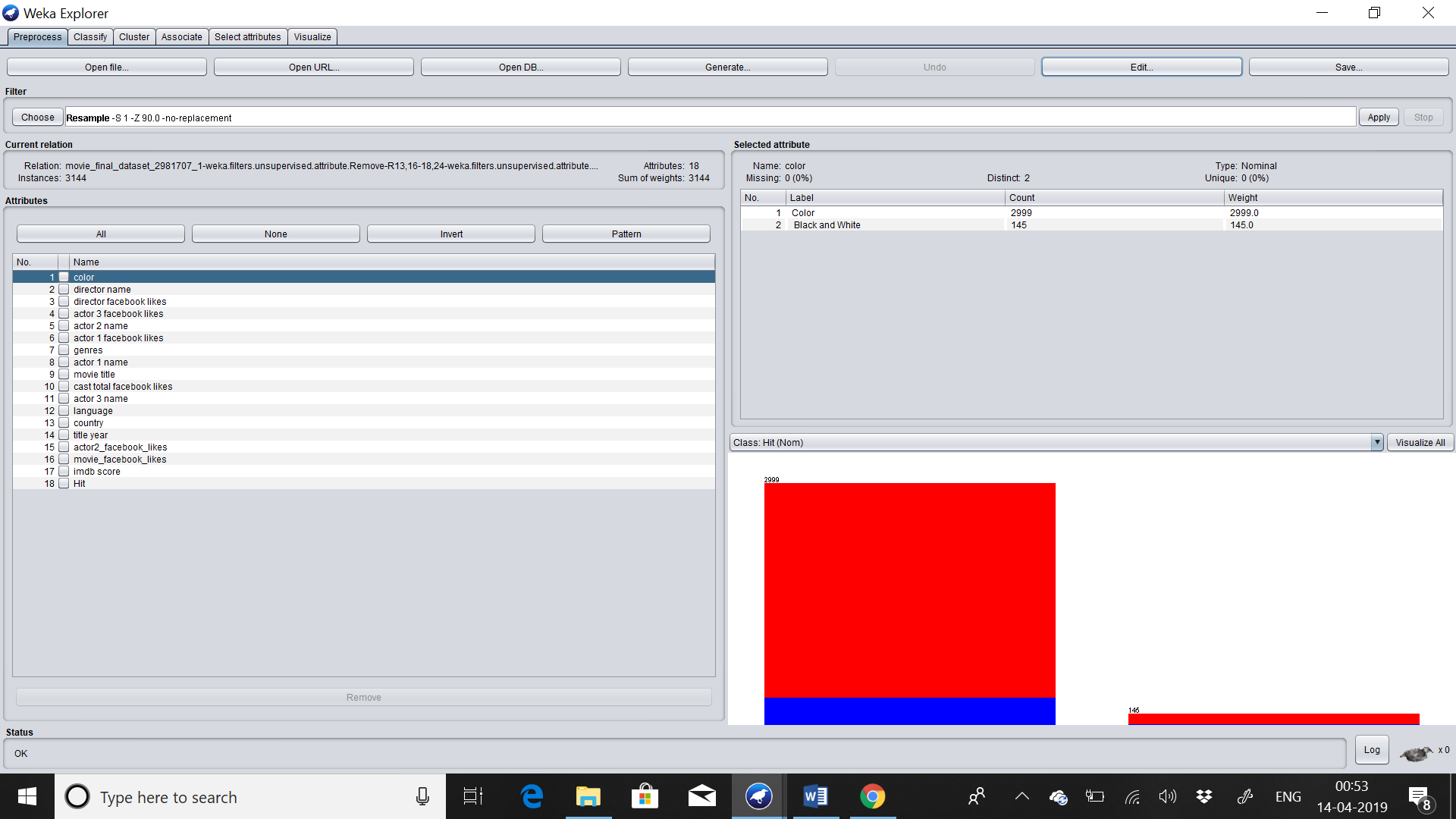
Follow the instructions presented in the link below divide the test into a training and testing set in the ration of (9:1).

<https://weka.wikispaces.com/How+do+I+divide+a+dataset+into+training+and+test+set%3F> The files generated as part of this process should be saved and submitted as the following

**ANSWER:**

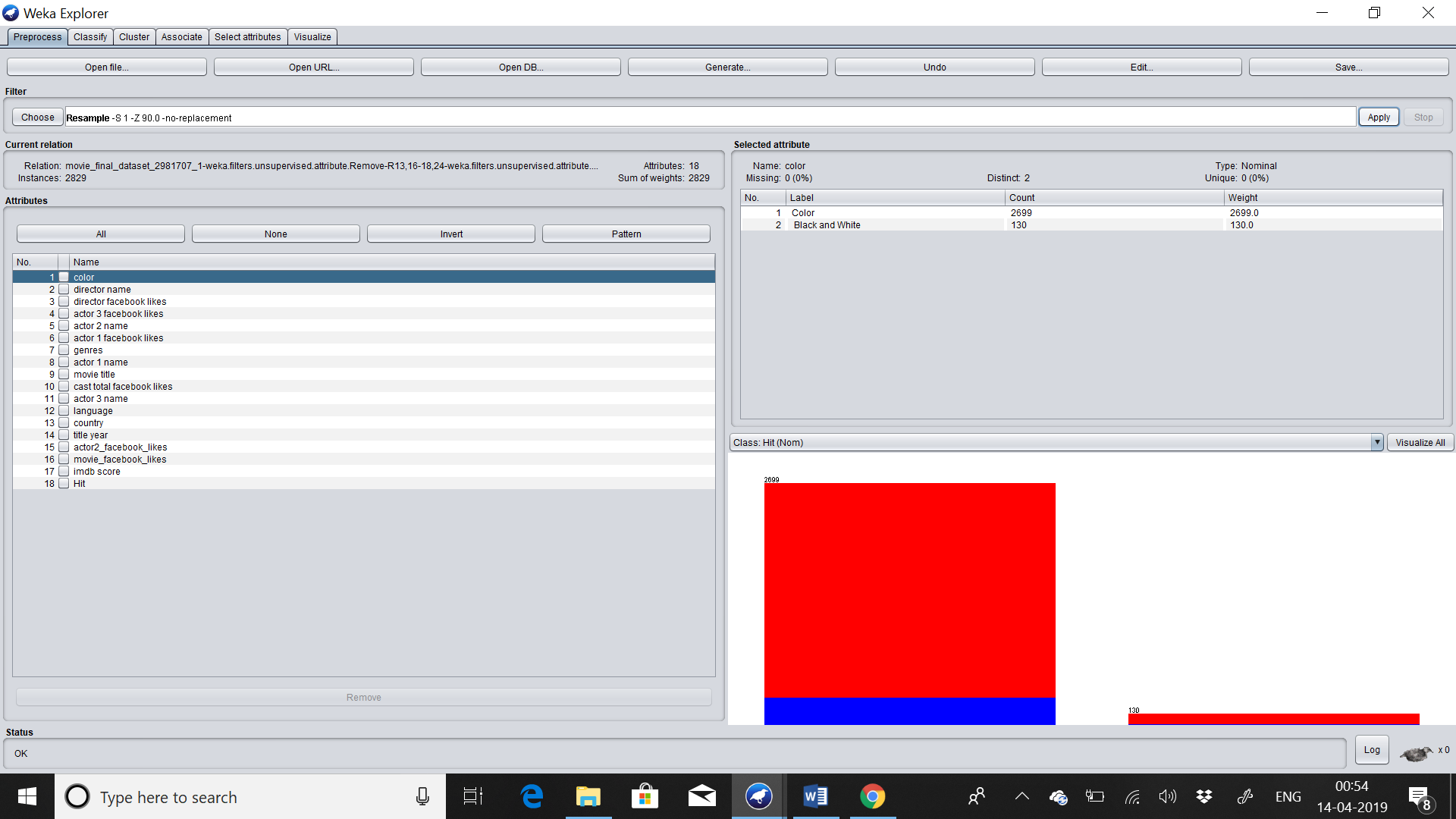
**Now we are going to divide our dataset into training set and testing set. So we have to choose weka filters->unsupervised->instance->Resample. In that we have to choose invert selection false and noReplacement is true in order to avoid duplication. And finally in sample size percent we have to choose 90 to divide our dataset into 9:1 ration.**

**Before training set:**



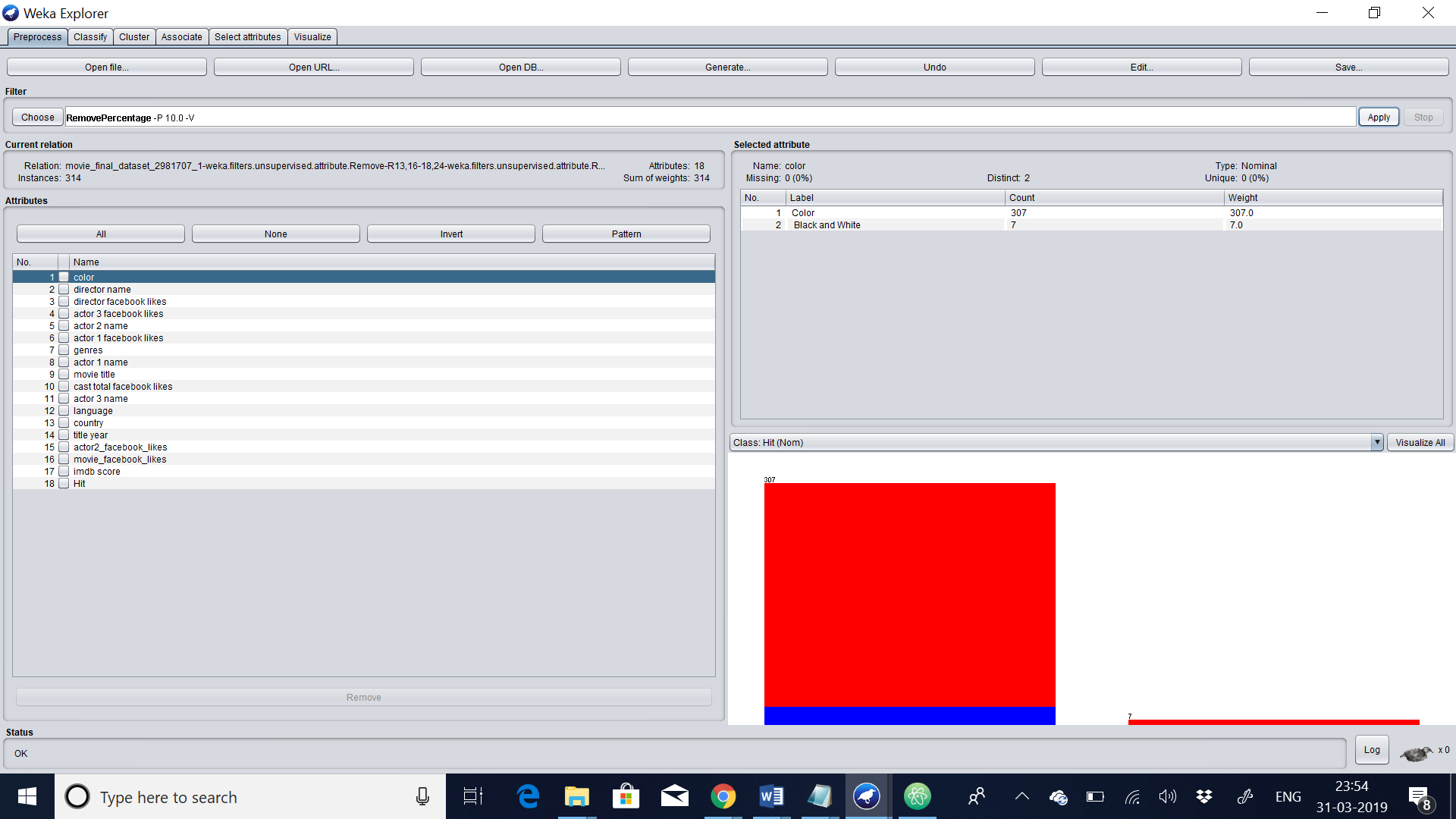
**After training set:**

**trainingSet.arff**



**In testing set we will set invert selection has true in order to fetch the last 10% values from the dataset. And we will choose sample size percent has 10. So this is how we will fetch the values of the remaining 9:1 values from the dataset as the testing set.arff**

**Testingset.arff**



# Experiments

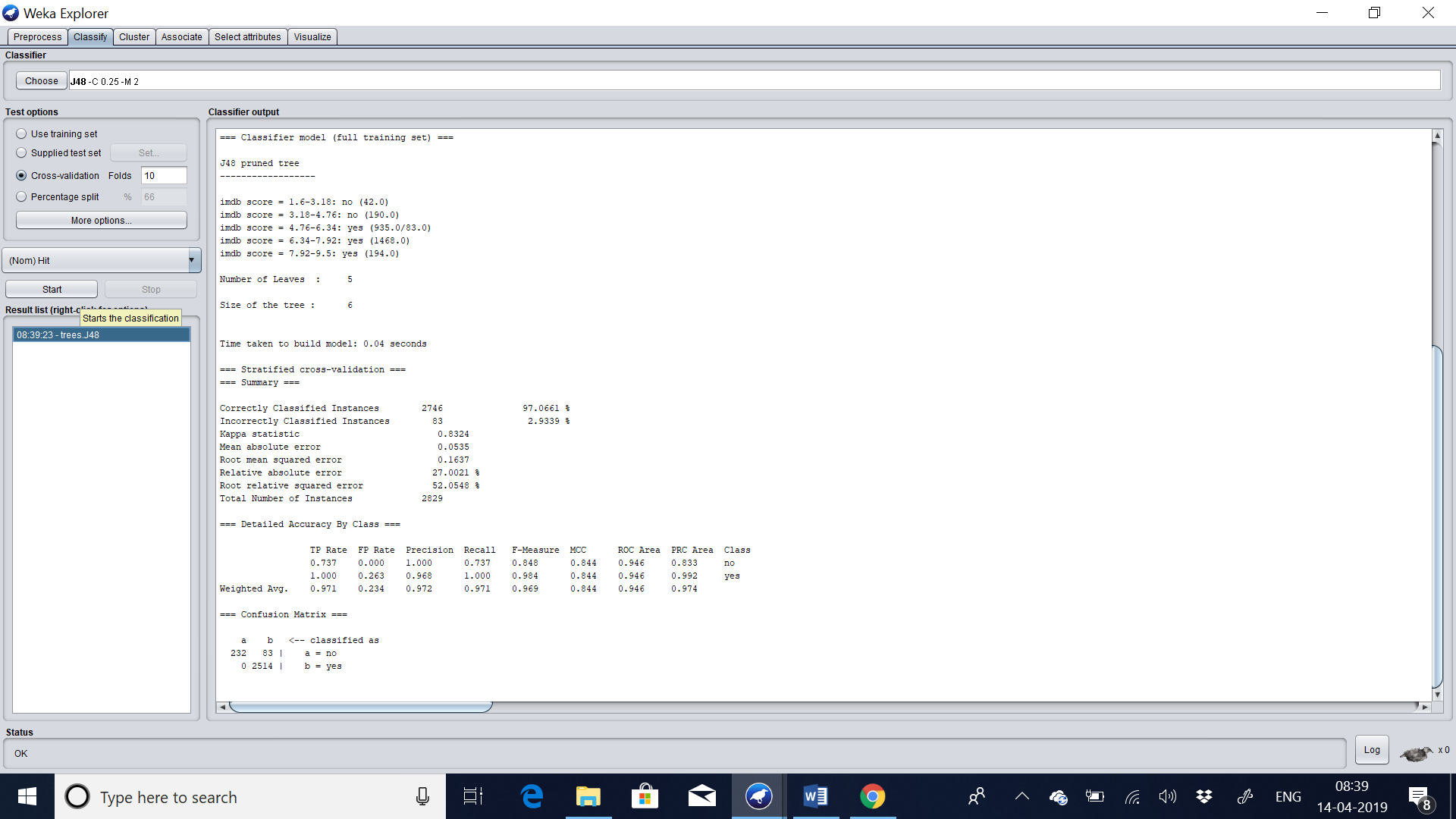
For each of the following classification techniques

1. Train your model using trainingSet.arff
2. Test your model using testingSet.arff
3. Write a few paragraphs analyzing the results. Be sure to vary parameters at least 3 times in each case. Support this analysis with screenshots of the following
   1. The model or a visualization of the model
   2. The results of the model
   3. Any additional output of the model including but not limited to
      1. Rules
      2. Confidence Values
      3. Confusion Matrixes
      4. etc
   4. Simple references to the notes or URL links to online resources complete with a sentence or two of explanation.

## Classification: J48 Tree – 15%

**TRAININGSET:**

**VISUALIZATION OF THE MODEL**



Results:

The algorithm was run with 10-fold cross-validation: this means it was given an opportunity to make a prediction for each instance of the dataset (with different training folds) and the presented result is a summary of those predictions.

**CONFIDENCE:**

Firstly, note the [Classification Accuracy](http://en.wikipedia.org/wiki/Accuracy_and_precision). You can see that the model achieved a result of 2746/2829 correct or 97%

**CONFUSION MATRIX:**

Secondly, look at the [Confusion Matrix](http://en.wikipedia.org/wiki/Confusion_matrix). You can see a table of actual classes compared to predicted classes and you can see that there was 83 error

**RULES:**

imdb score = 1.6-3.18: no (42.0)

imdb score = 3.18-4.76: no (190.0)

imdb score = 4.76-6.34: yes (935.0/83.0)

imdb score = 6.34-7.92: yes (1468.0)

imdb score = 7.92-9.5: yes (194.0)

**NUMBER OF LEAVES:**

It has totally 5 number of leaves

**SIZE OF THE TREE:**

Totally the size of tree is 6

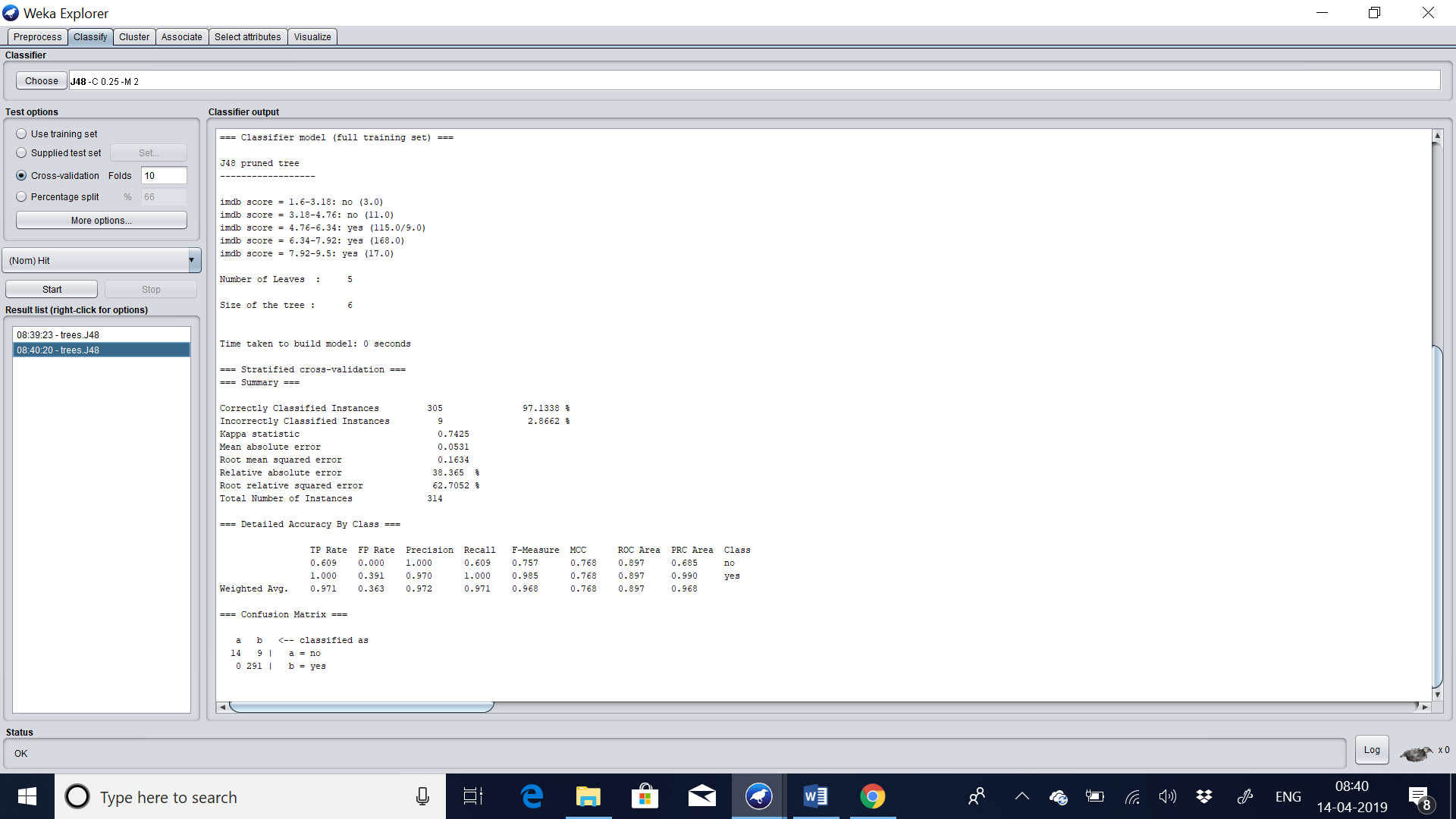
**LINK:**

<https://machinelearningmastery.com/how-to-run-your-first-classifier-in-weka/>

This link helped me to understand the basic working of algorithms such as ZeroR, j48 etc

**TESTING SET:**

**VISUALIZATION OF THE MODEL:**



Results:

The algorithm was run with 10-fold cross-validation: this means it was given an opportunity to make a prediction for each instance of the dataset (with different training folds) and the presented result is a summary of those predictions.

**CONFIDENCE:**

Firstly, note the [Classification Accuracy](http://en.wikipedia.org/wiki/Accuracy_and_precision). You can see that the model achieved a result of 305/314 correct or 97.1338%

**CONFUSION MATRIX:**

Secondly, look at the [Confusion Matrix](http://en.wikipedia.org/wiki/Confusion_matrix). You can see a table of actual classes compared to predicted classes and you can see that there was 9 error

**RULES:**

imdb score = 1.6-3.18: no (3.0)

imdb score = 3.18-4.76: no (11.0)

imdb score = 4.76-6.34: yes (115.0/9.0)

imdb score = 6.34-7.92: yes (168.0)

imdb score = 7.92-9.5: yes (17.0)

**NUMBER OF LEAVES:**

It has totally 5 number of leaves

**SIZE OF THE TREE:**

Totally the size of tree is 6

**LINK:**

<https://machinelearningmastery.com/how-to-run-your-first-classifier-in-weka/>

This link helped me to understand the basic working of algorithms such as ZeroR, j48 etc

## Classification: Association Rules – 15%

APRIORI

Trainingset:

From looking at the “Associator output” window, you can see that the algorithm presented 10 rules learned from the movie dataset. The algorithm is configured to stop at 10 rules by default,

RULES DISCOVERED ARE:

1. cast total facebook likes=0-131346 2823 ==> actor 1 facebook likes=0-128000 2823 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.99)

2. actor 1 facebook likes=0-128000 2823 ==> cast total facebook likes=0-131346 2823 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.99)

3. actor 3 facebook likes=0-3400 cast total facebook likes=0-131346 2800 ==> actor 1 facebook likes=0-128000 2800 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.94)

4. actor 3 facebook likes=0-3400 actor 1 facebook likes=0-128000 2800 ==> cast total facebook likes=0-131346 2800 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.94)

5. director facebook likes=0-4600 cast total facebook likes=0-131346 2743 ==> actor 1 facebook likes=0-128000 2743 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.82)

6. director facebook likes=0-4600 actor 1 facebook likes=0-128000 2743 ==> cast total facebook likes=0-131346 2743 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.82)

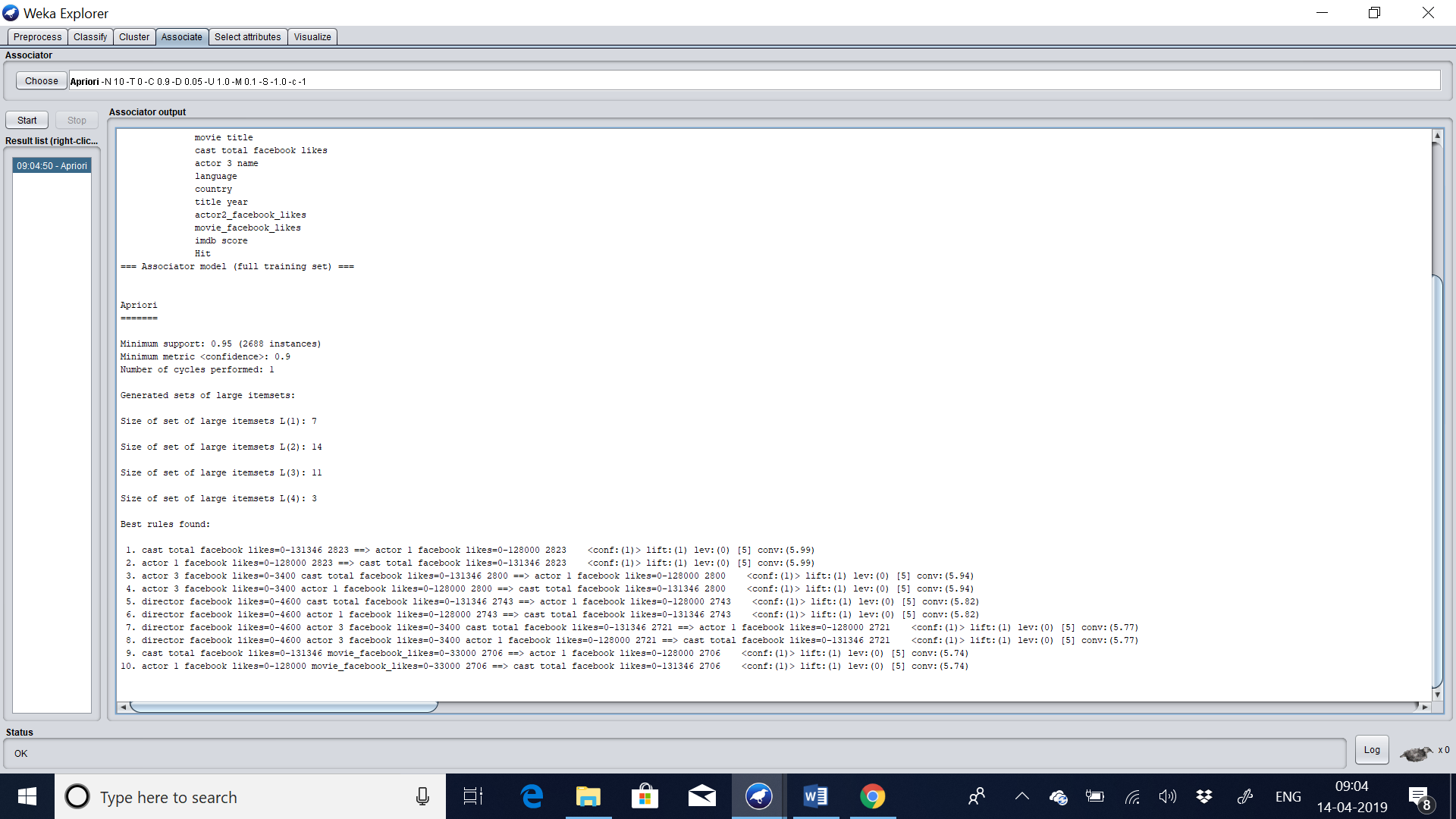
7. director facebook likes=0-4600 actor 3 facebook likes=0-3400 cast total facebook likes=0-131346 2721 ==> actor 1 facebook likes=0-128000 2721 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.77)

8. director facebook likes=0-4600 actor 3 facebook likes=0-3400 actor 1 facebook likes=0-128000 2721 ==> cast total facebook likes=0-131346 2721 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.77)

9. cast total facebook likes=0-131346 movie\_facebook\_likes=0-33000 2706 ==> actor 1 facebook likes=0-128000 2706 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.74)

10. actor 1 facebook likes=0-128000 movie\_facebook\_likes=0-33000 2706 ==> cast total facebook likes=0-131346 2706 <conf:(1)> lift:(1) lev:(0) [5] conv:(5.74)

You can see rules are presented in antecedent => consequent format. The number associated with the antecedent is the absolute coverage in the dataset. The number next to the consequent is the absolute number of instances that match the antecedent and the consequent. The number in brackets on the end is the support for the rule (number of antecedent divided by the number of matching consequents). You can see that a cutoff of 95% was used in selecting rules, mentioned in the “Associator output” window and indicated in that no rule has a coverage less than 0.95.



Testing set:

From looking at the “Associator output” window, you can see that the algorithm presented 10 rules learned from the movie dataset. The algorithm is configured to stop at 10 rules by default,

RULES DISCOVERED ARE:

1. cast total facebook likes=0-131346 314 ==> actor 1 facebook likes=0-128000 314 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

2. actor 1 facebook likes=0-128000 314 ==> cast total facebook likes=0-131346 314 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

3. actor 3 facebook likes=0-3400 308 ==> actor 1 facebook likes=0-128000 308 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

4. actor 3 facebook likes=0-3400 308 ==> cast total facebook likes=0-131346 308 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

5. actor 3 facebook likes=0-3400 cast total facebook likes=0-131346 308 ==> actor 1 facebook likes=0-128000 308 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

6. actor 3 facebook likes=0-3400 actor 1 facebook likes=0-128000 308 ==> cast total facebook likes=0-131346 308 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

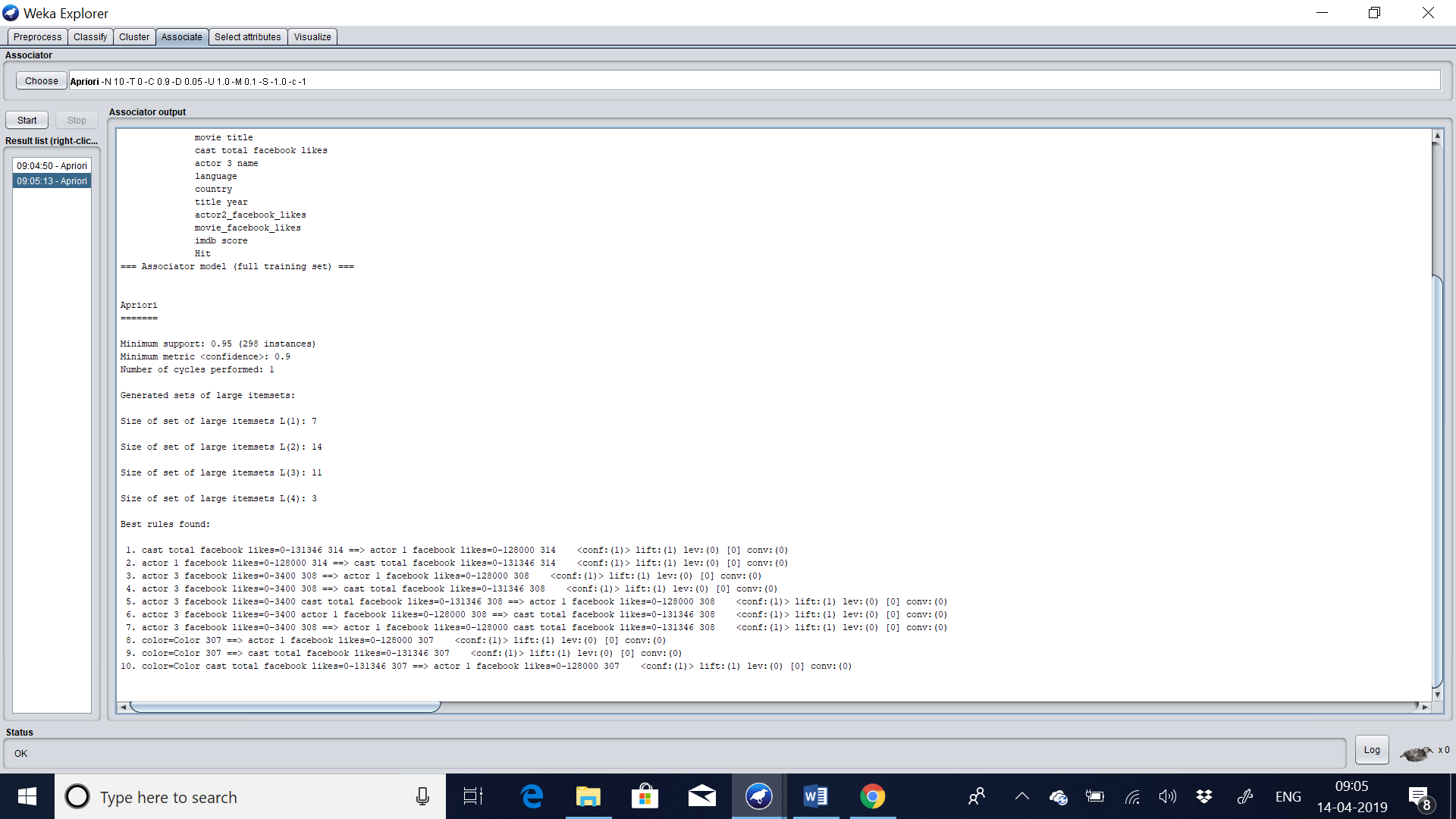
7. actor 3 facebook likes=0-3400 308 ==> actor 1 facebook likes=0-128000 cast total facebook likes=0-131346 308 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

8. color=Color 307 ==> actor 1 facebook likes=0-128000 307 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

9. color=Color 307 ==> cast total facebook likes=0-131346 307 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

10. color=Color cast total facebook likes=0-131346 307 ==> actor 1 facebook likes=0-128000 307 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)

You can see rules are presented in antecedent => consequent format. The number associated with the antecedent is the absolute coverage in the dataset. The number next to the consequent is the absolute number of instances that match the antecedent and the consequent. The number in brackets on the end is the support for the rule (number of antecedent divided by the number of matching consequents). You can see that a cutoff of 95% was used in selecting rules, mentioned in the “Associator output” window and indicated in that no rule has a coverage less than 0.95.



# Part 2 – Clustering

## 6. Description of your dataset and findings – 10%

* **Title**: Brief title to capture the data and objective of your assignment

HOUSES INFORMATION

* **Data description:** A description of the data in detail under the following subheadings:
  + The problem domain

There is list of houses with list of attributes. In this our class variable is based on ratings. And many attributes having the information of each and every information in the dataset. We have to find the x-axis and y0axis clustering of k-means and DBSCAN

* + The source of the data

The dataset was downloaded in this link (<https://www.kaggle.com/carolzhangdc/AirBnB.csv>). This dataset contains information of houses with list of attributes.

* + The agencies working with the data

This dataset contains the information of houses with rating , prices etc. Using these data to keep track of all the information of the houses and gives rating on the basis of “How efficient a house is”.

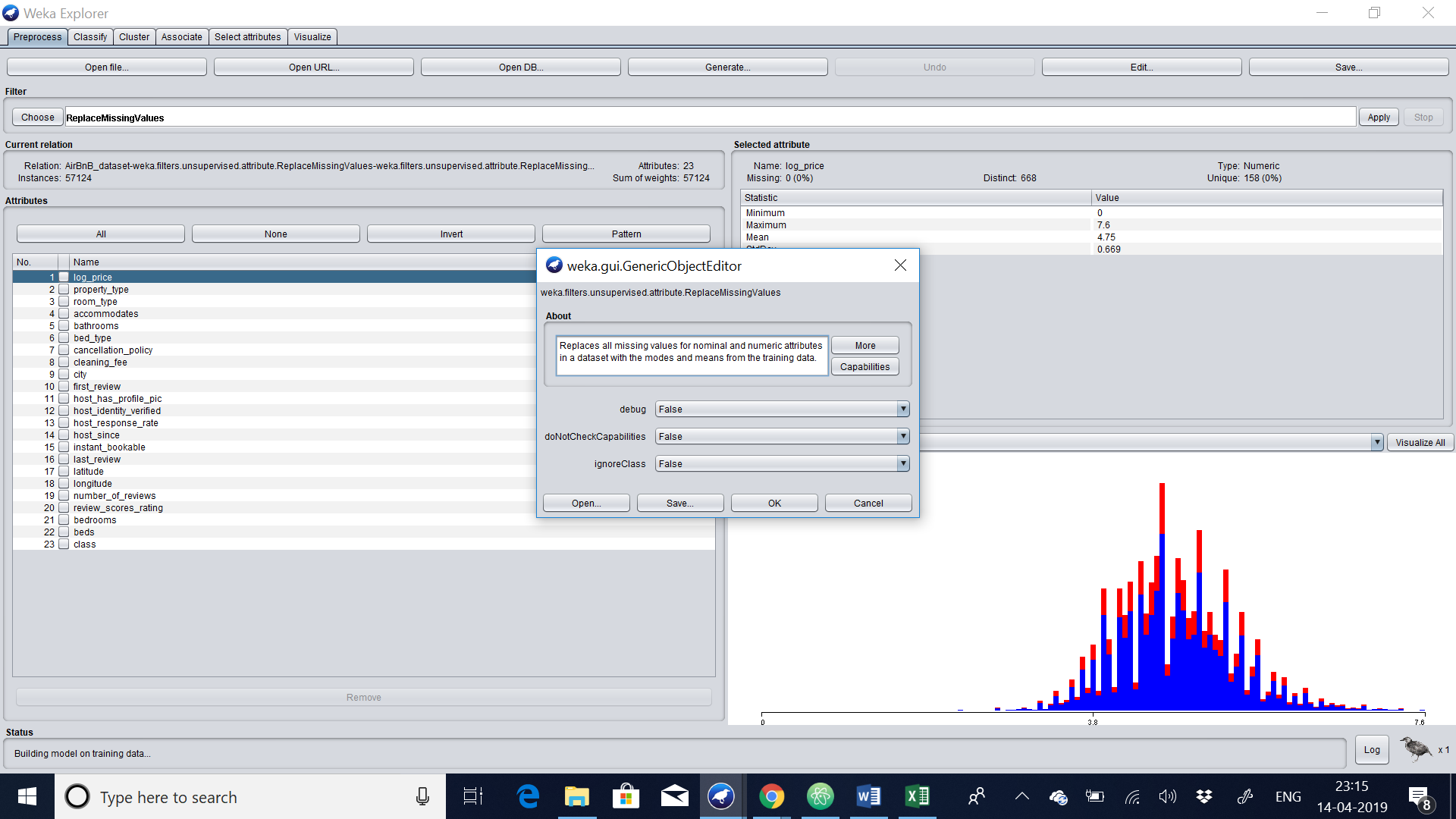
* + The intended use of the data

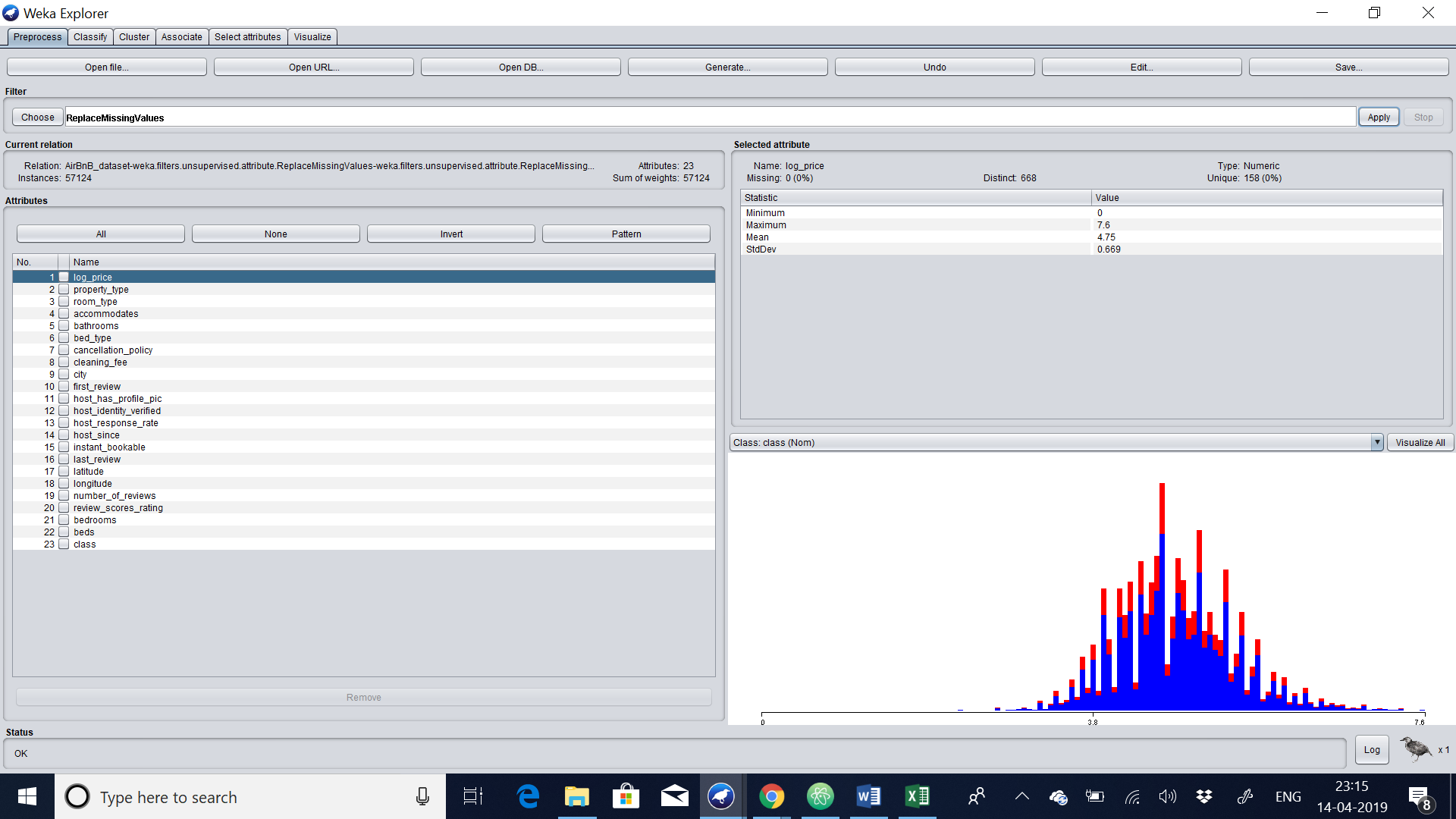
With these data’s we can able to find all the information regarding this houses. And do analytics on finding the centroid and truying to avoid the generalization in the DBSCAN.

* + The attribute types of the data

There are list of attributes such as house price, property type, room type, city etc. There are 23 list of attributes including the class attribute in the dataset.

Please include screen shots (with one or two sentences of summary) of the dataset and also of the data summaries that are available though Weka. \





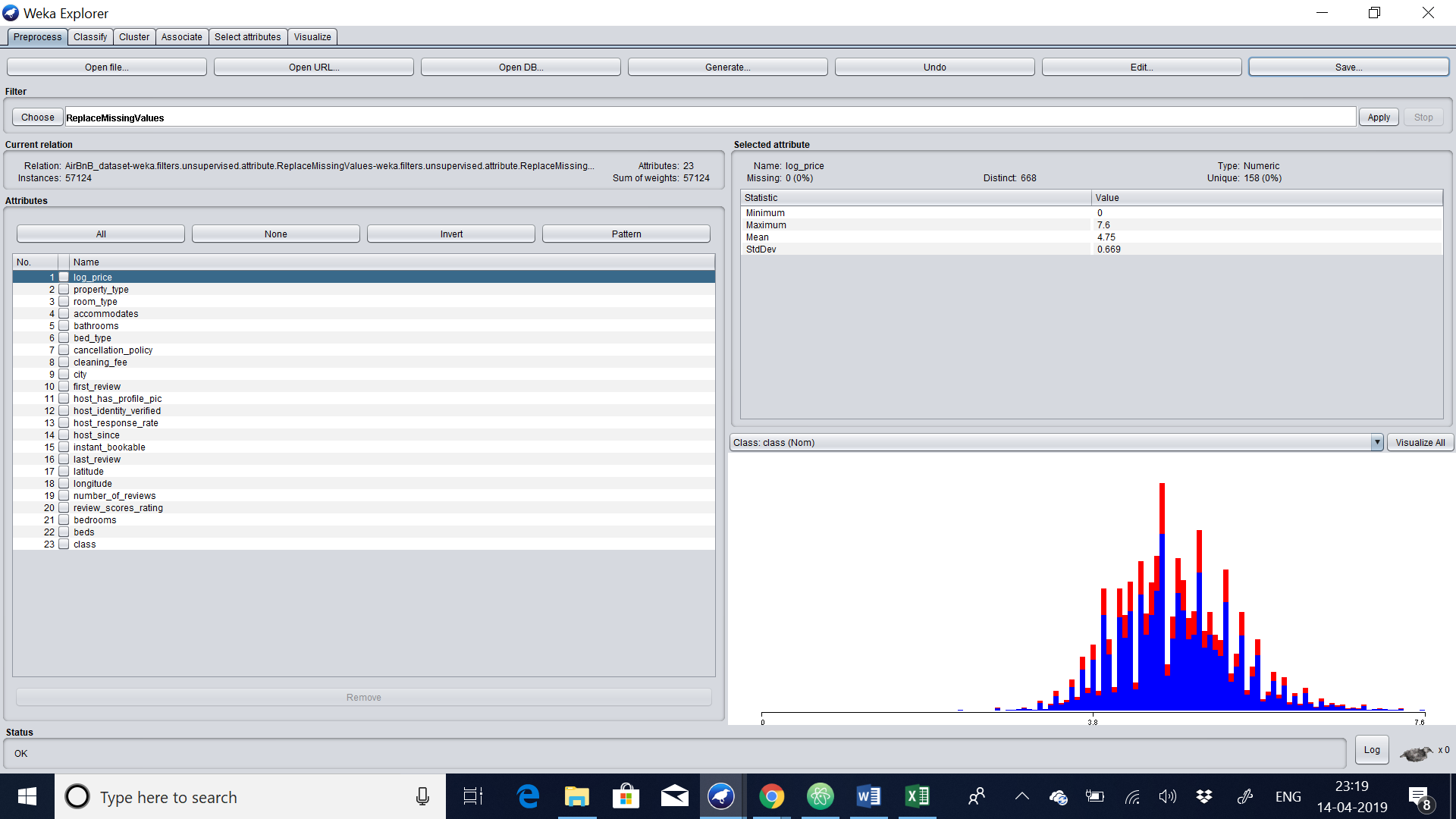
Thus, the preprocessing is done by replacing missing values. Clustering works on numeric values. So, it’s better to make the attributes in the numeric.

* **Objective**: Your objective. You can update this as you progress through your project revising it and making it more specific.

Our objective is to find the cluster which will be grouped together to the groups based on groups or shapes in order to fetch the best clusters in the dataset.

## 7. Preprocessing – 10%

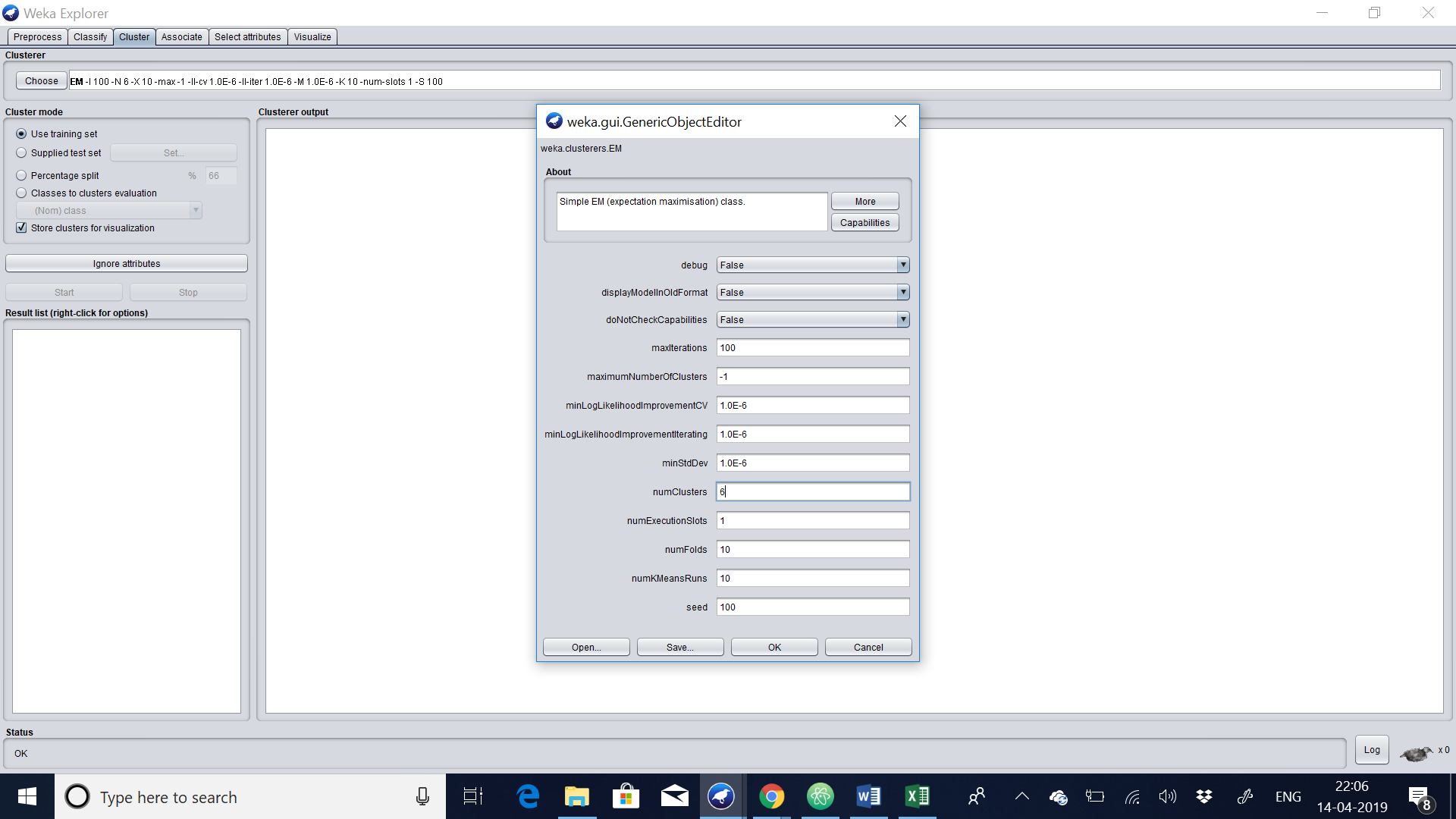
Initially we have to import the csv file. Once the file is imported we have to make sure to replace the missing values. In this dataset, I have replaced the missing values with the average values. So that the data are useful for the association. So I choose the weka->filter->unsupervised->Replacingvalues. This helps me to replace the values of the missing values in the dataset. Once the preprocessing is done. I have saved the file as “clusterdataset.arff”



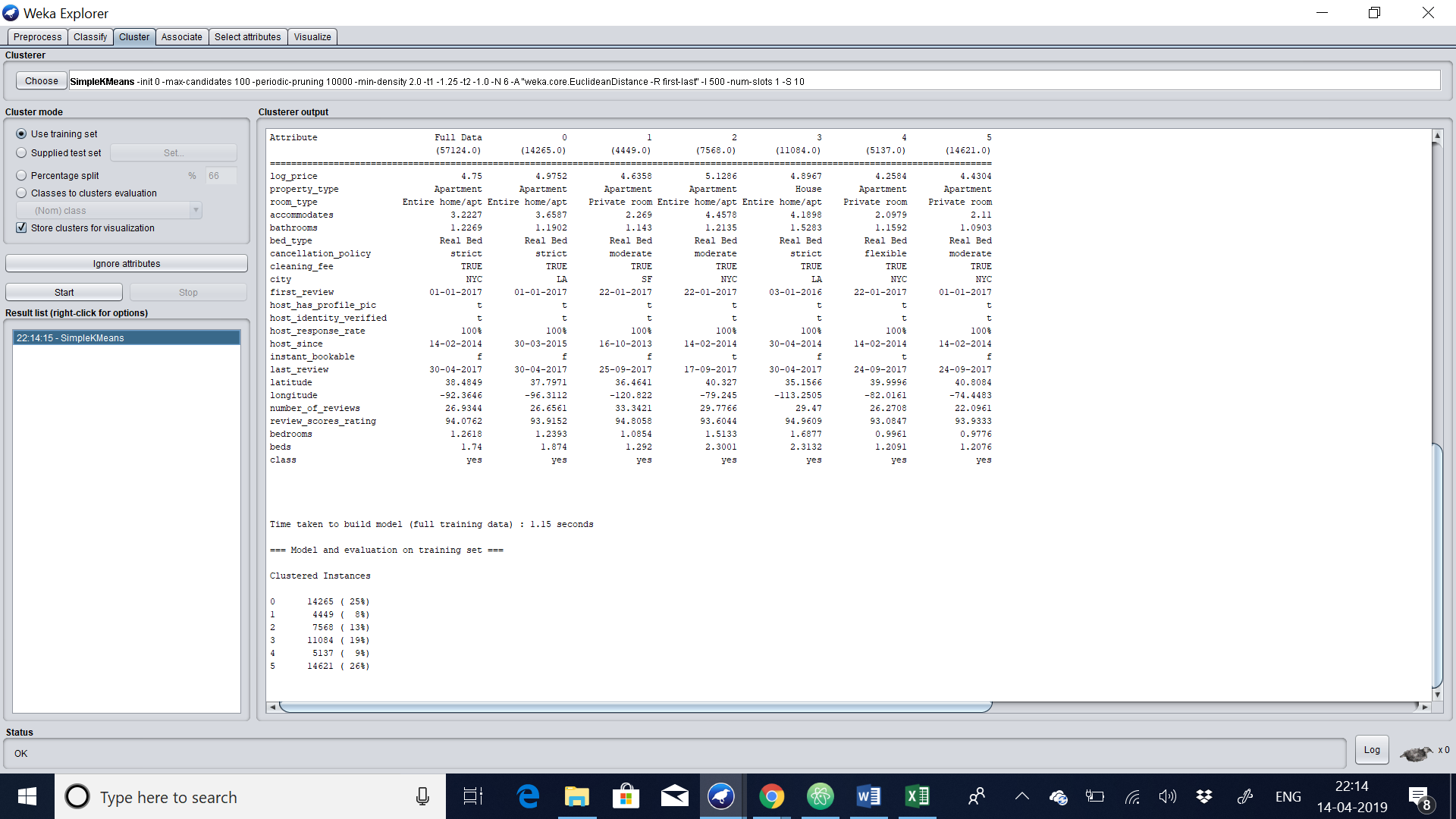
This is the evidence after the preprocessing by maintaining the numeric values and replacing the missing values.

1. **Clustering: kmeans-15%**

**Let us do k-means clustering. The sample data set used for this example is based on the "HOUSES" available in comma-separated format (AirBnB\_dataset.csv). When doing the preprocess we should not remove unique columns and let the values be in numeric itself.**

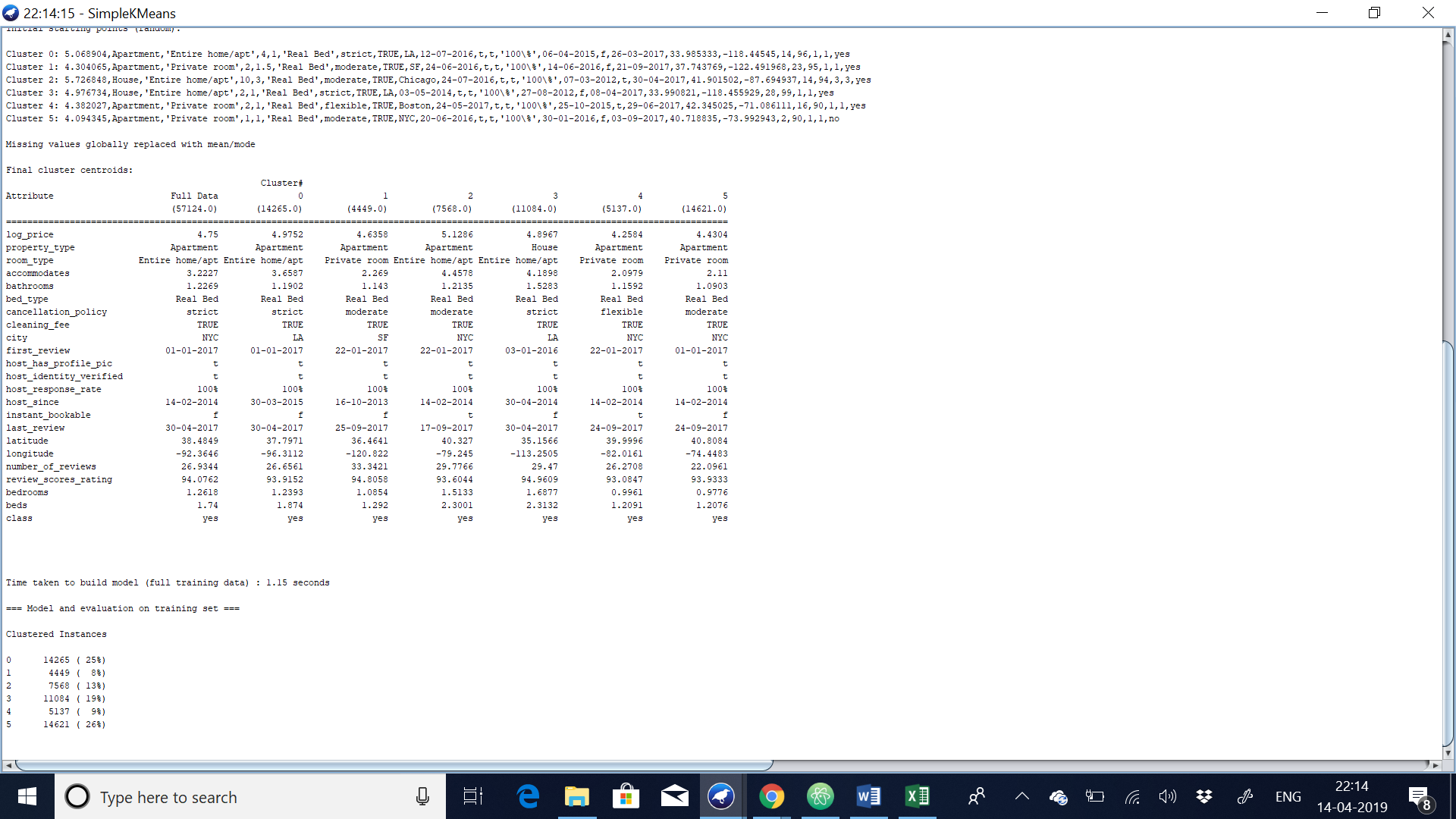
****

**We choose the numclusters 6 and let the seed be default. Note that, in general, K-means is quite sensitive to how clusters are initially assigned. Thus, it is often necessary to try different values and evaluate the results.**

****

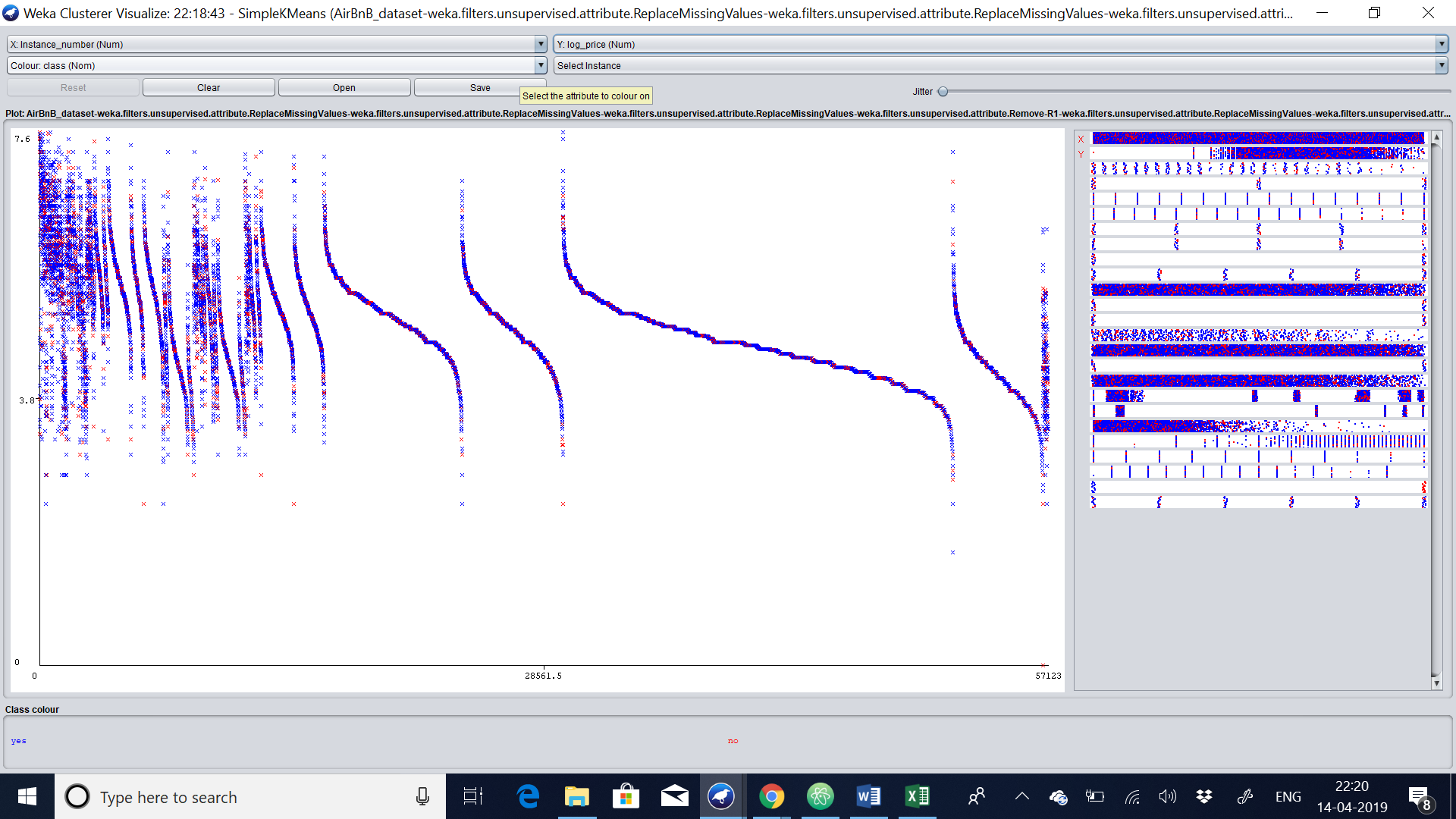
**Let us view the cluster in the separate page.**

**The average values having the class type yes. Because most of the values are average means it has a good review as well as all the room type and most of the houses was in the most occurrence city.**

****

**The result window shows the centroid of each cluster as well as statistics on the number and percentage of instances assigned to different clusters. Cluster centroids are the mean vectors for each cluster. Thus, centroids can be used to characterize the clusters.**

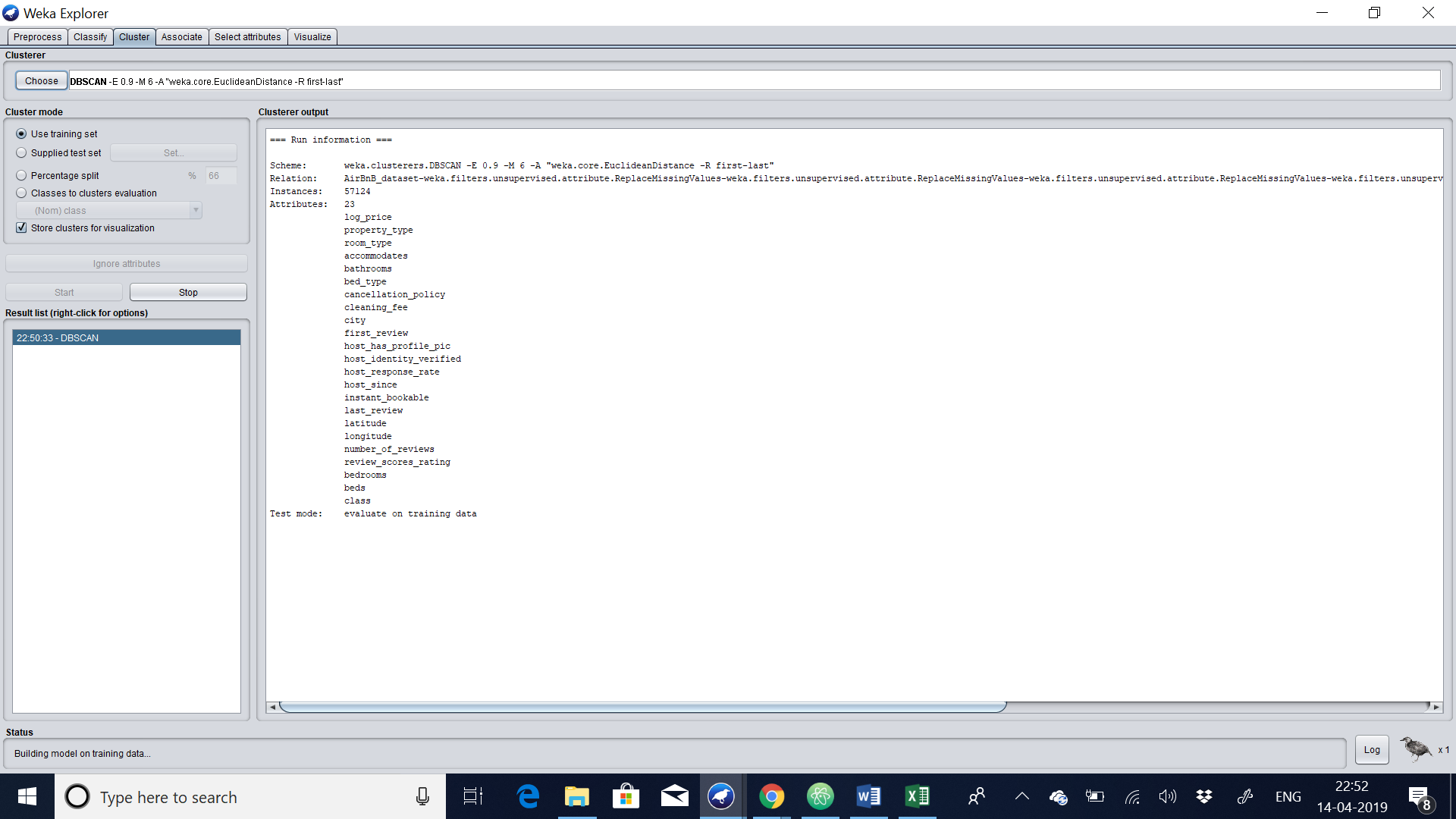
**EXAMPLE OF CENTROID:**



**We choose the x-axis has instance and y-axis has log\_price and the colour has class type in order to fetch the output. All the centroid values chooses the class yes because of its review and rating of each and every house in the dataset.**

## Clustering: DBSCAN – 15%

DBSCAN is the well known density based clustering algorithms. An interesting property of density based clustering is that these algorithms do not assume clusters to have a particular shape. Furthermore, the algorithms allow “noise” objects that do not belong to any of the clusters. Minpts are the important parameter for the dbscan.  Now let us load the dataset in weka and we will choose clustering->dbscan. Let us start the algorithm.



 If you set it too high, at some point there won’t be any clusters anymore, only noise. However, the parameter usually is not hard to choose. If you for example expect clusters to typically have 100 objects, I’d start with a value of 10 or 20. If your clusters are expected to have 10000 objects, then maybe start experimenting with 500. Most of the instance generated are based on cluster with 100% clustered instance. We get 0% clustered from the remaining clusters.

