# Question 1

Data mining methods fall into two categories predictive & descriptive analytics. Predictive uses data to determine the most likely future outcome of an event or a probability of a situation occurring. Descriptive looks at data and analyses past events to understand how to approach the future and understands that performance by mining historical data to look for the reasons behind past successes or failures.

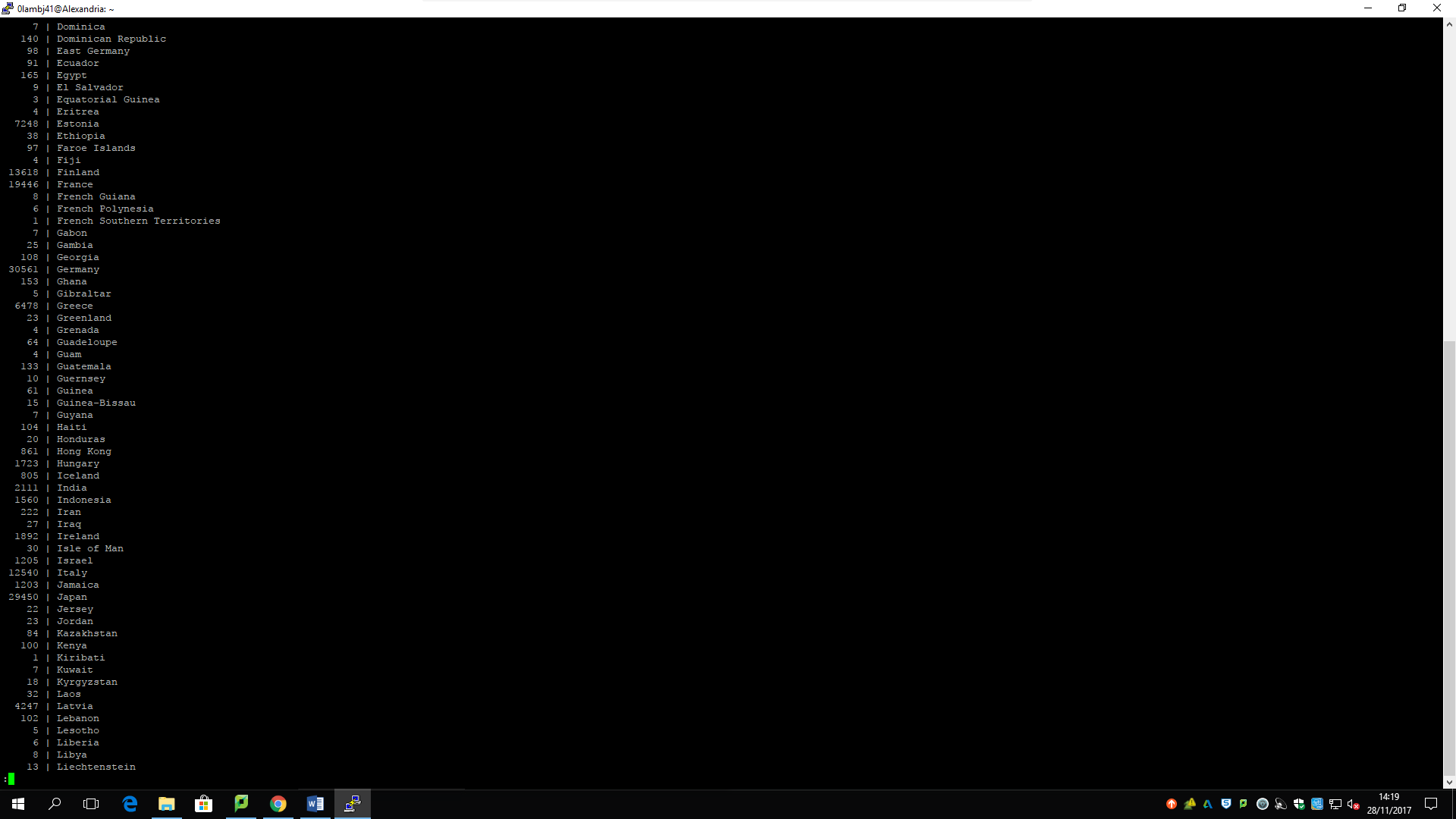
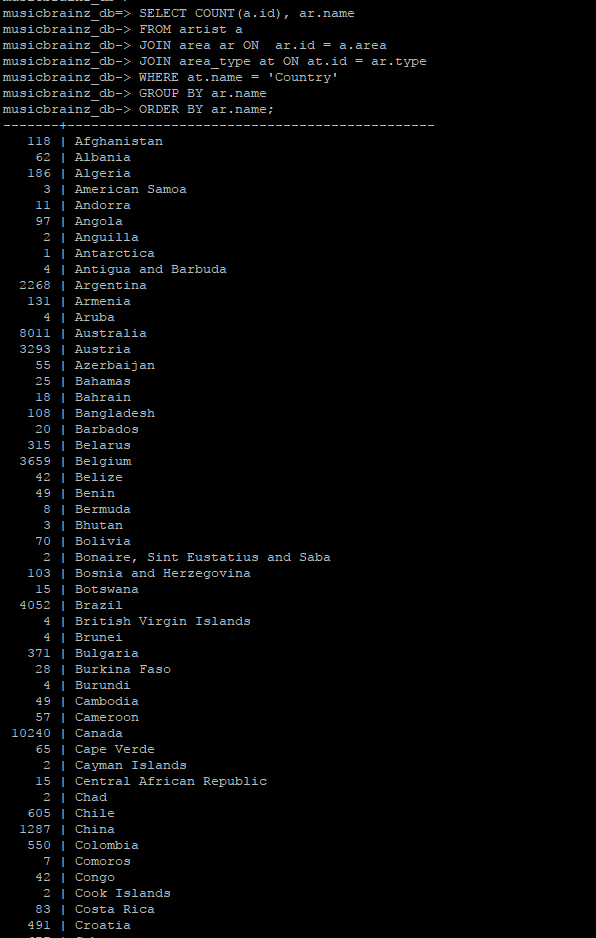
There are three methods of Descriptive analytics one being clustering, this divides the set of data into groups based on data similarity and then assigns labels to the groups. Another Association rules, which are created by analysing data for frequent if/then patterns and using the criteria support & confidence to find the most important relationships. Another statistics, is the science of learning from data and includes everything from collecting & organising to analysing and presenting.

There are three methods of Predictive analytics one being classification, this is a function that assigns items in a collection to target classes or categories, the aim here is to accurately predict the target class for each case in the data. Another regression, it predicts a range of numeric values, given a particular dataset, one type of regression is linear regression used to estimate a relationship between two or more variables. Another time series, an important class of temporal data objects. It is a collection of observations made chronologically.

# Question 2

On Alexandria we are writing an SQL query in Musicbrainz to get the number of Artists and the number of Labels by Country. We need to create to query’s and put them both into one subquery.

Here I am showing the first query I created to get the number of Artists by Country and its results.



SELECT COUNT(a.id), ar.name

FROM artist a

JOIN area ar ON ar.id = a.area

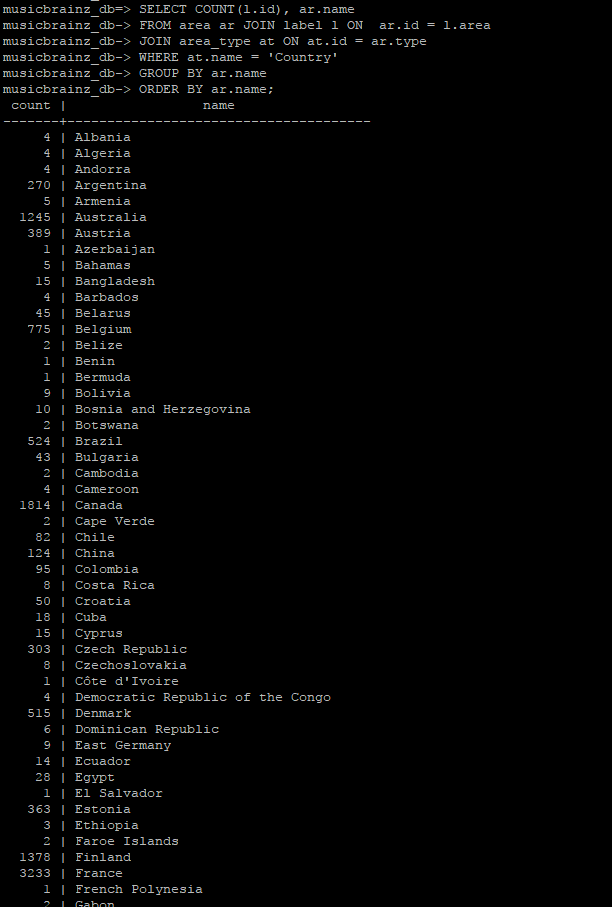
JOIN area\_type at ON at.id = ar.type

WHERE at.name = 'Country'

GROUP BY ar.name

ORDER BY ar.name;

Here I am showing the second query I created to get the number of Labels by Country and its results.



SELECT COUNT(l.id), ar.name

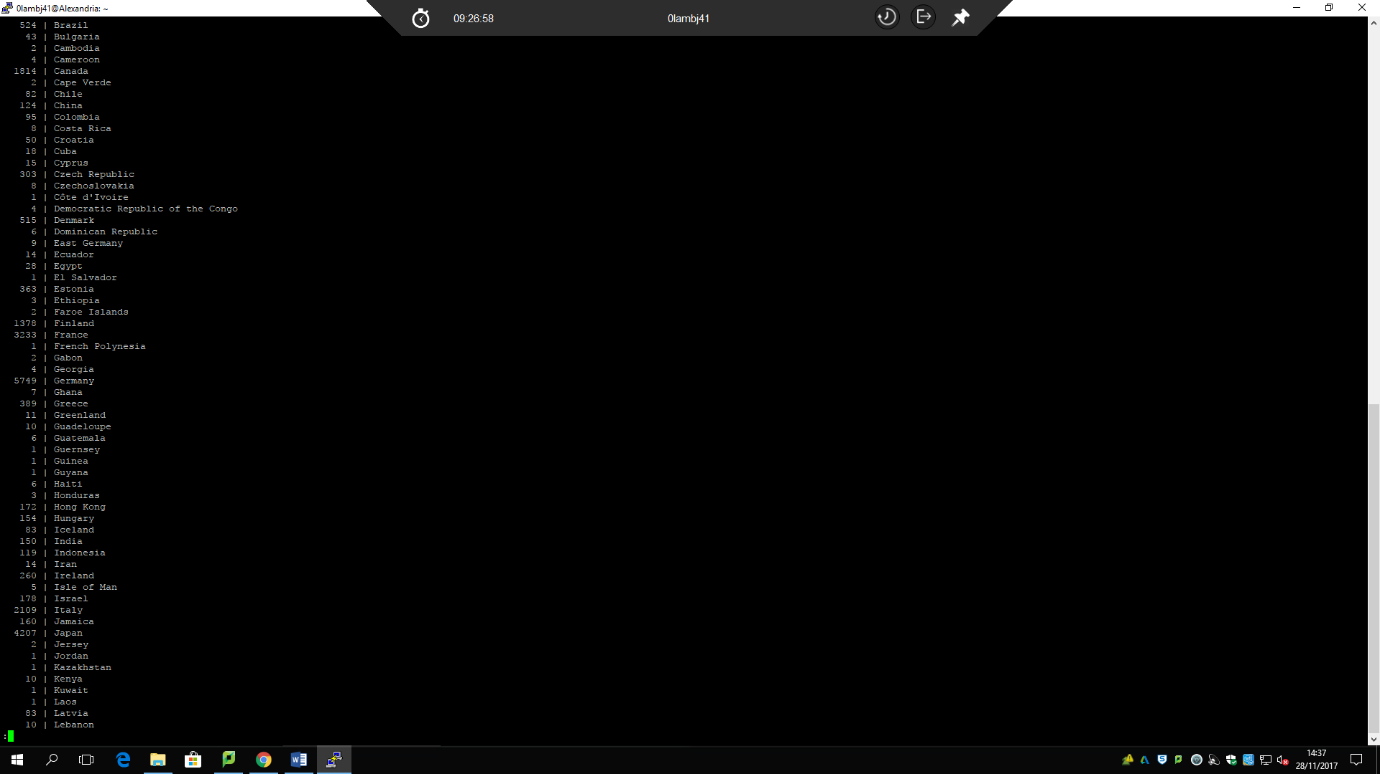
FROM area ar JOIN label l ON ar.id = l.area

JOIN area\_type at ON at.id = ar.type

WHERE at.name = 'Country'

GROUP BY ar.name

ORDER BY ar.name;



Here I am putting the two queries together to form a **subquery** with its results.

SELECT sub1.country "Country", sub1.cnt "Total Artists", sub2.cnt "Total Labels"

FROM(SELECT COUNT(a.id) as cnt, ar.name as country

FROM artist a JOIN area ar ON ar.id = a.area

JOIN area\_type at ON at.id = ar.type

WHERE at.name = 'Country'

GROUP BY ar.name) as sub1,

(SELECT COUNT(l.id) as cnt, ar.name as country

FROM area ar JOIN label l ON ar.id = l.area

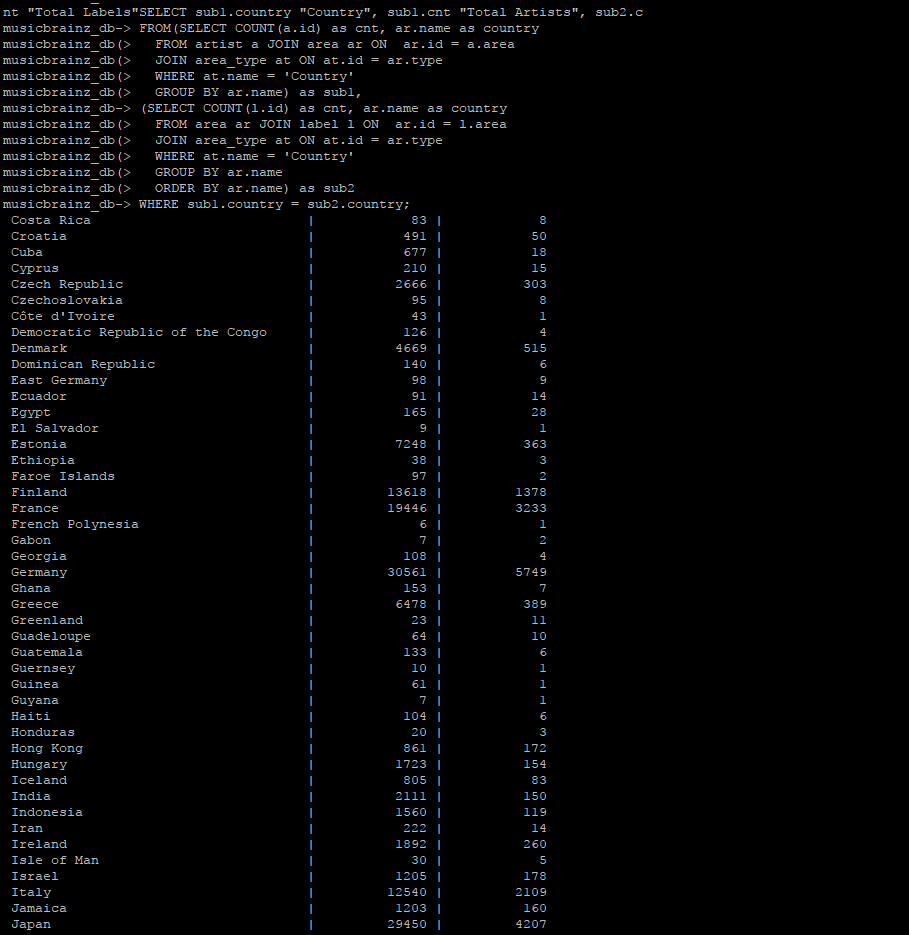
JOIN area\_type at ON at.id = ar.type

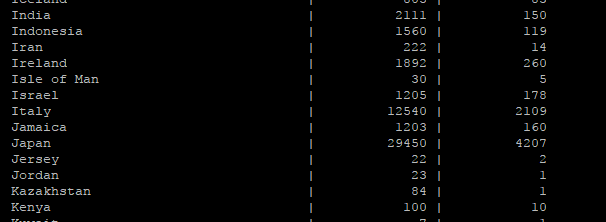
WHERE at.name = 'Country'

GROUP BY ar.name

ORDER BY ar.name) as sub2

WHERE sub1.country = sub2.country;

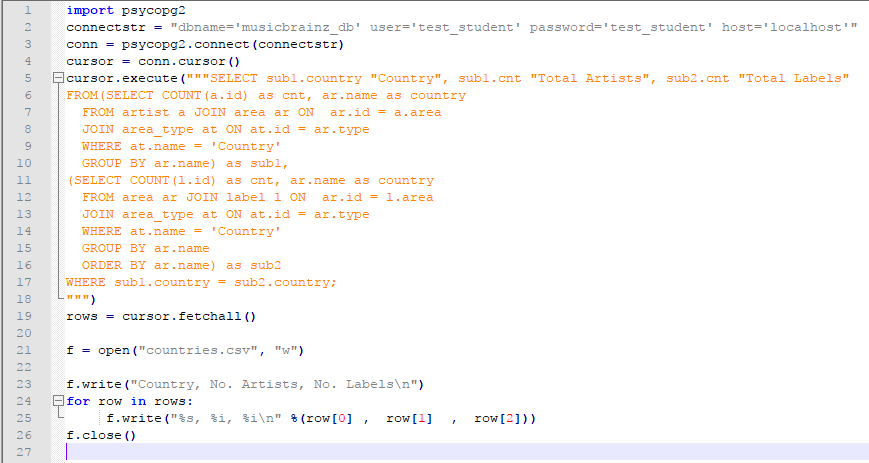




# Question 3

We had to write a Python script with a function which connects to the database and produces CSV data from the results of the query from question 2. Execute the function and print the output in a CSV file (“countries.csv”).

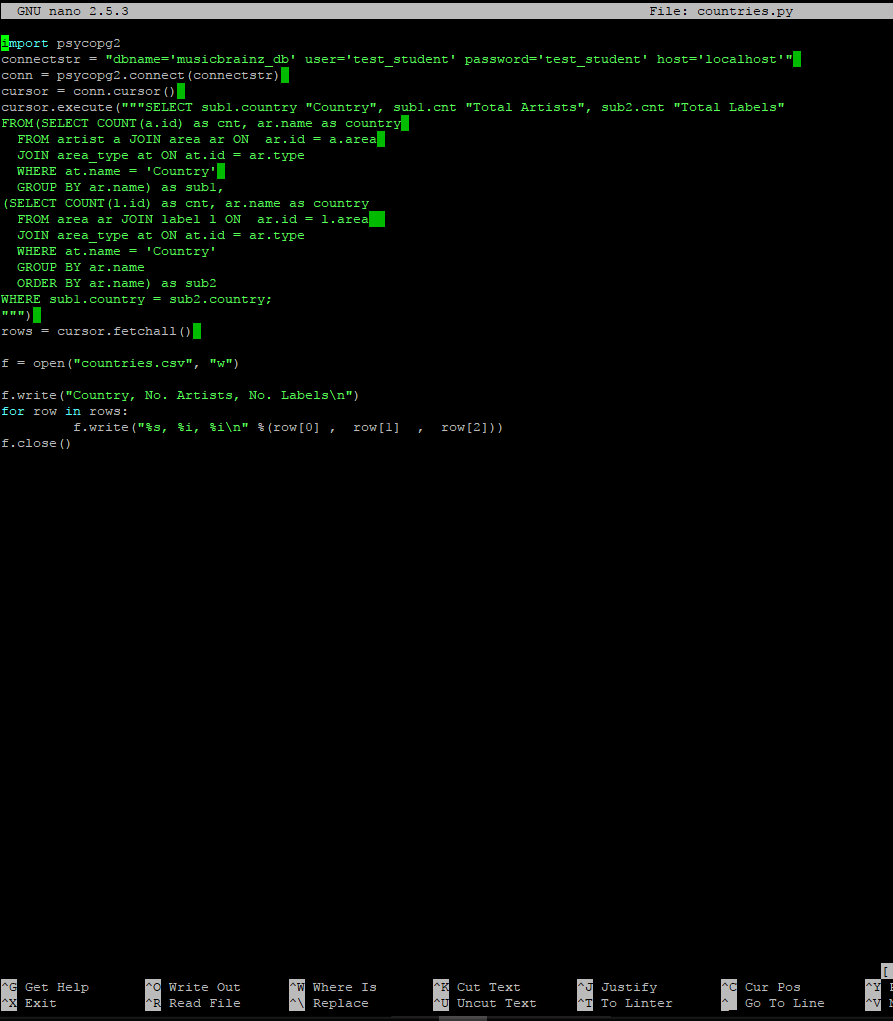
Here is my Python script.



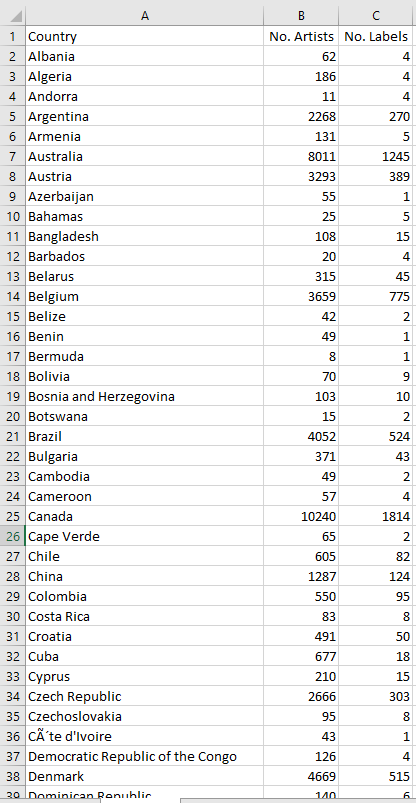
In Alexandria we create a file using **nano countries.py** for our Python script.

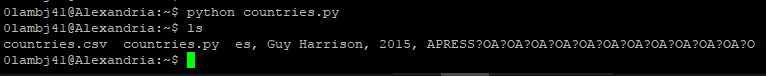


Once this file has been created we can write the function. Our function here connects to the database and produces a CSV file **countries.csv.**



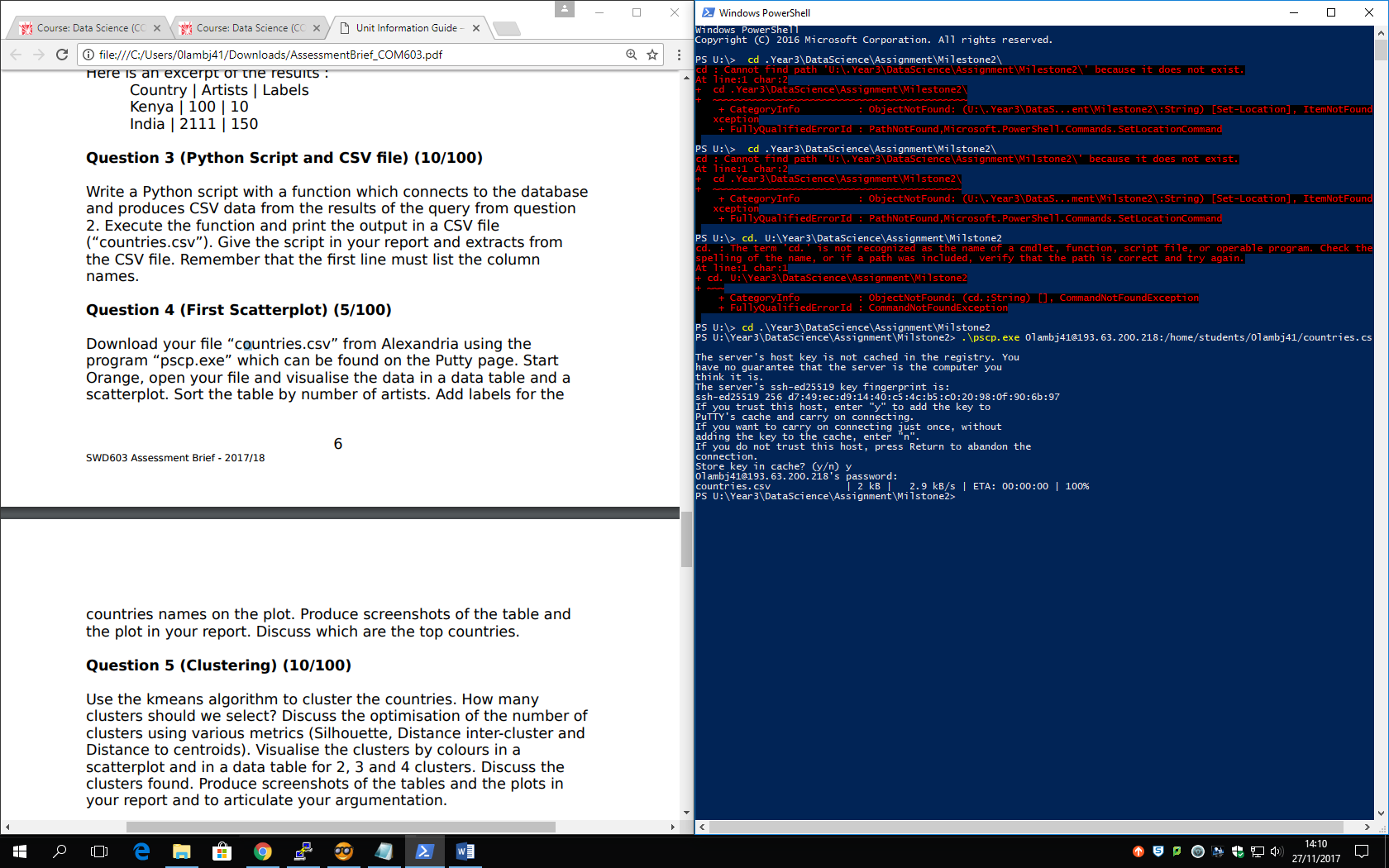
This is the extract from our **countries.csv** file.



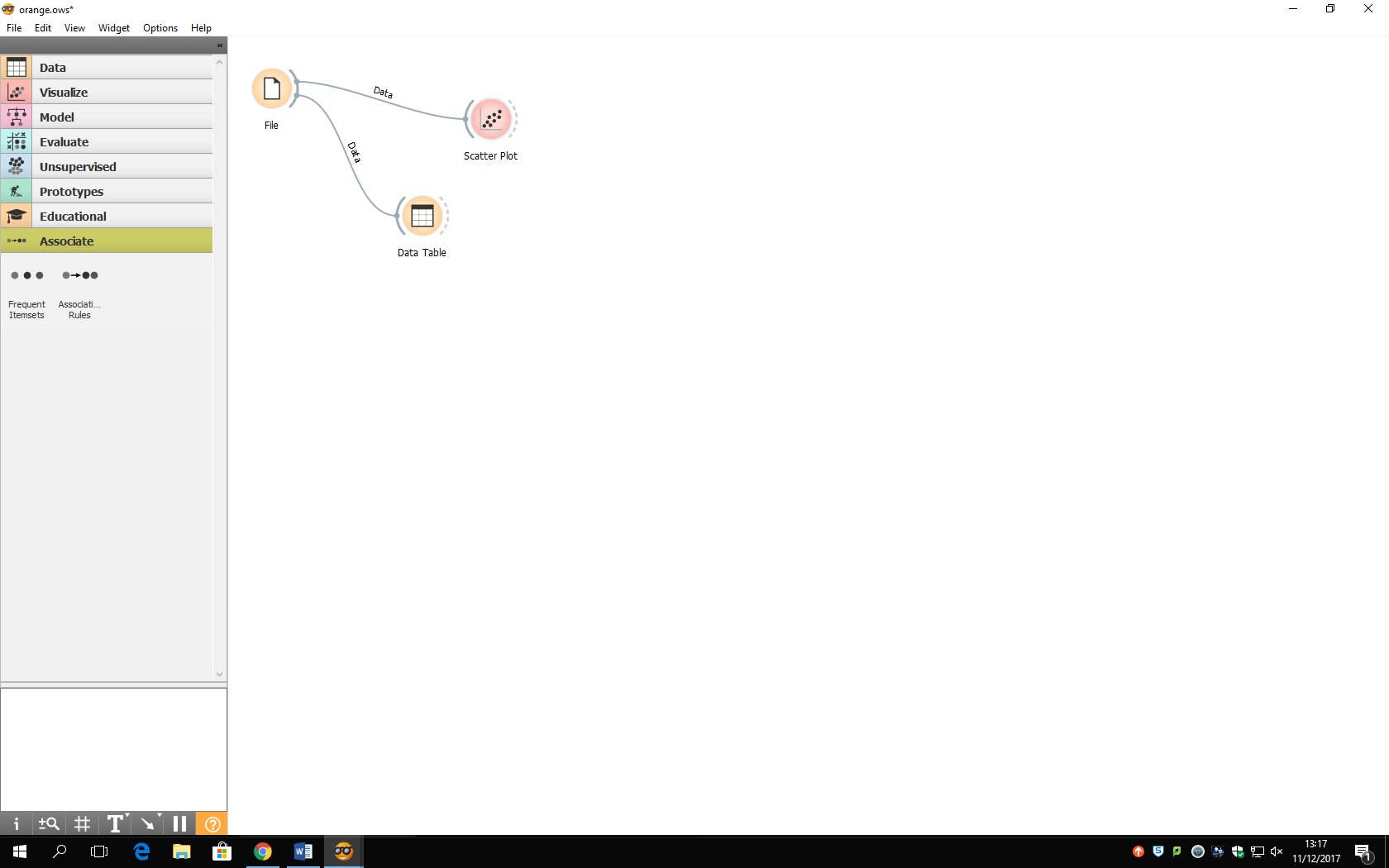
Here we are running the **countries.py** file using Python and showing that it creates the **countries.csv** file.

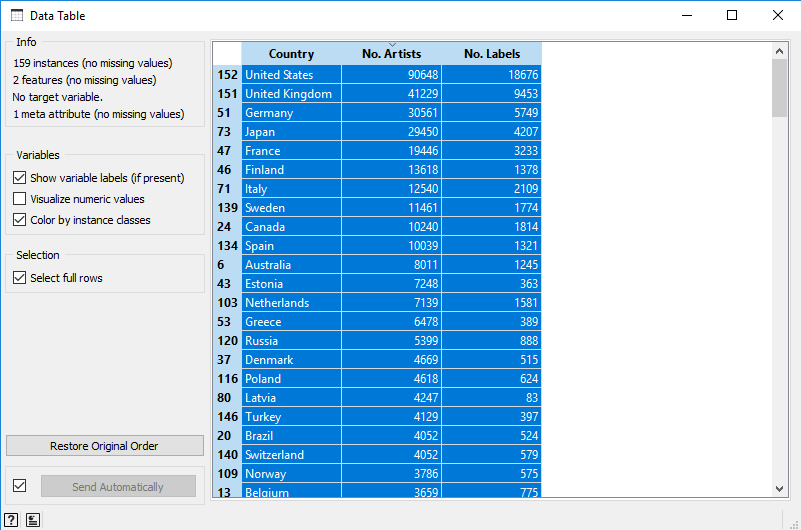
# Question 4

Here we downloaded the file **countries.csv** from Alexandria using the program **pscp.exe.** We opened up a program called Powershell, followed the directory path to the **pscp.exe** program. Then we log into our account and downloaded the **countries.csv** file.

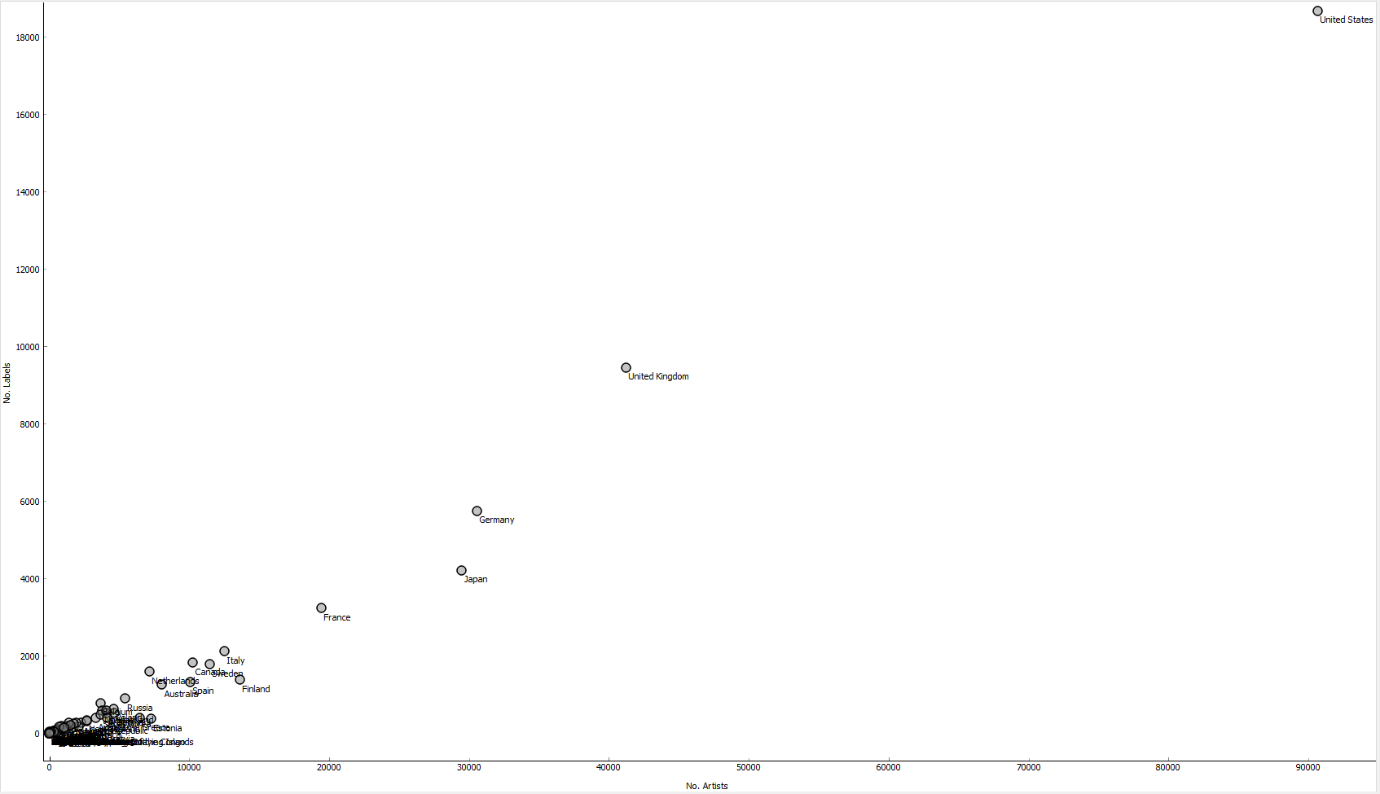


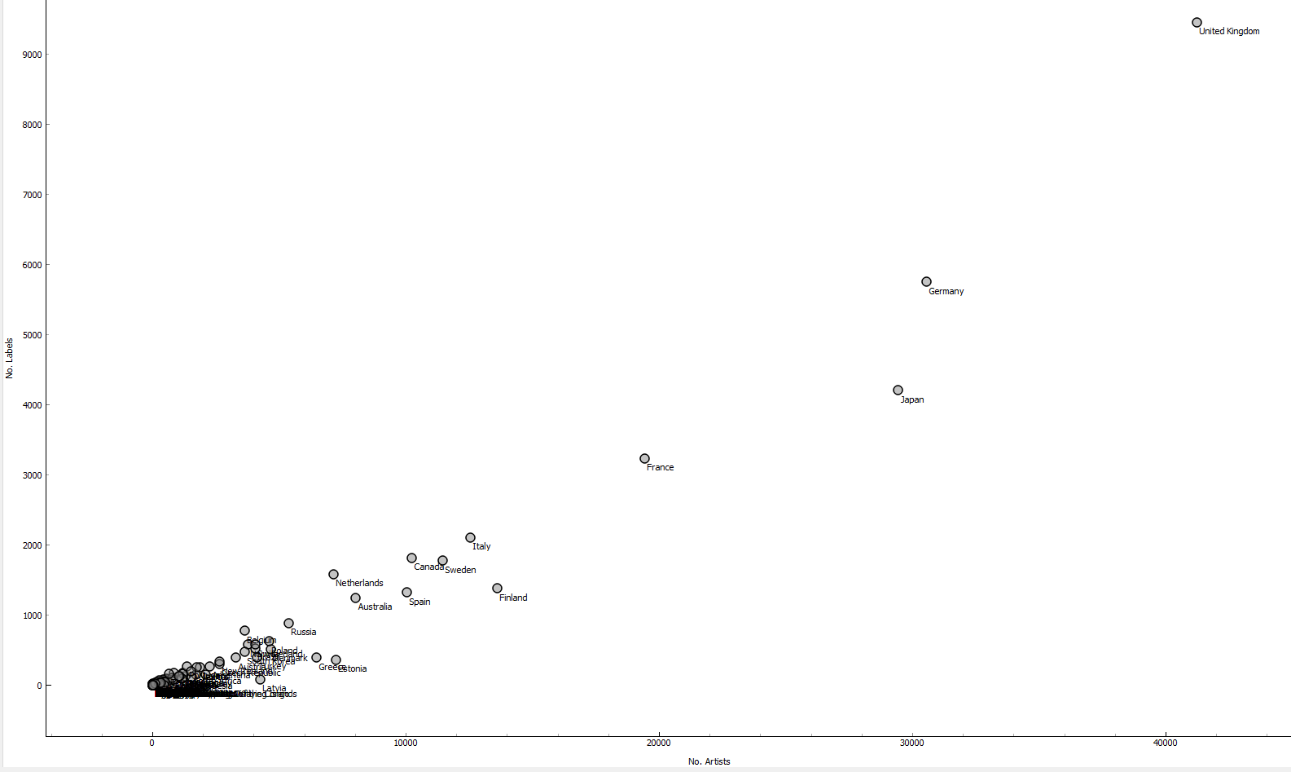
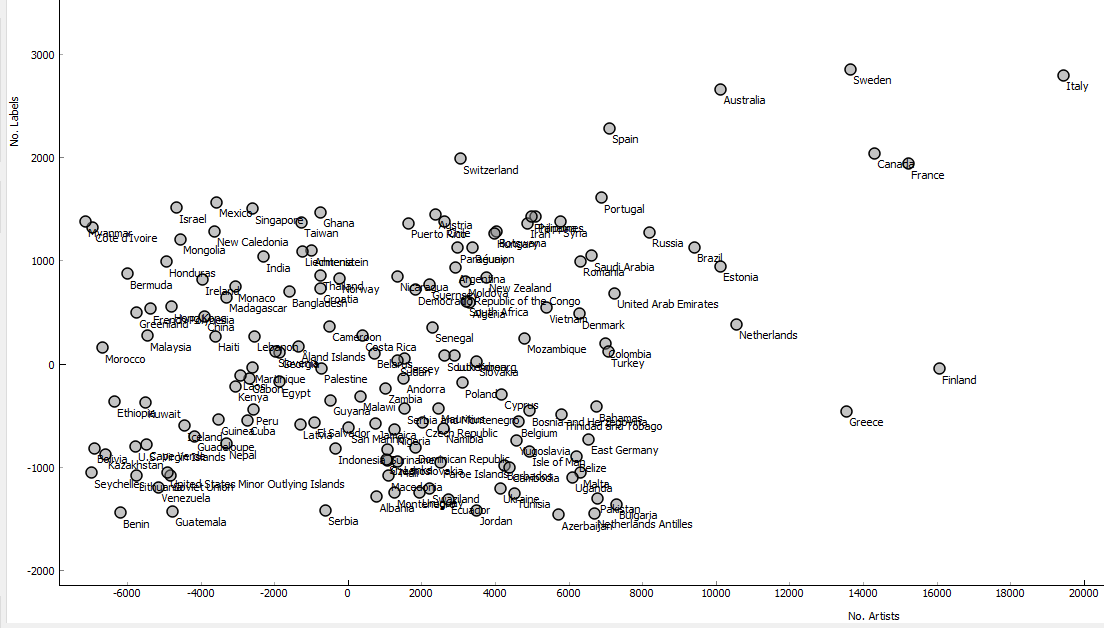
In **Orange.** We have created a **Data Table** and a **Scatter Plot** to visualize the data.



We sorted the table by No. Artists

We added labels for the Country names on the Scatter plot.



This shows the majority of the Countries with 2% jittering due to them being so close together. But I must state that this 2% jittering can cause some of the data to become inaccurate as seen below.

From the screenshots above we can analyse the data and conclude which are the top countries.

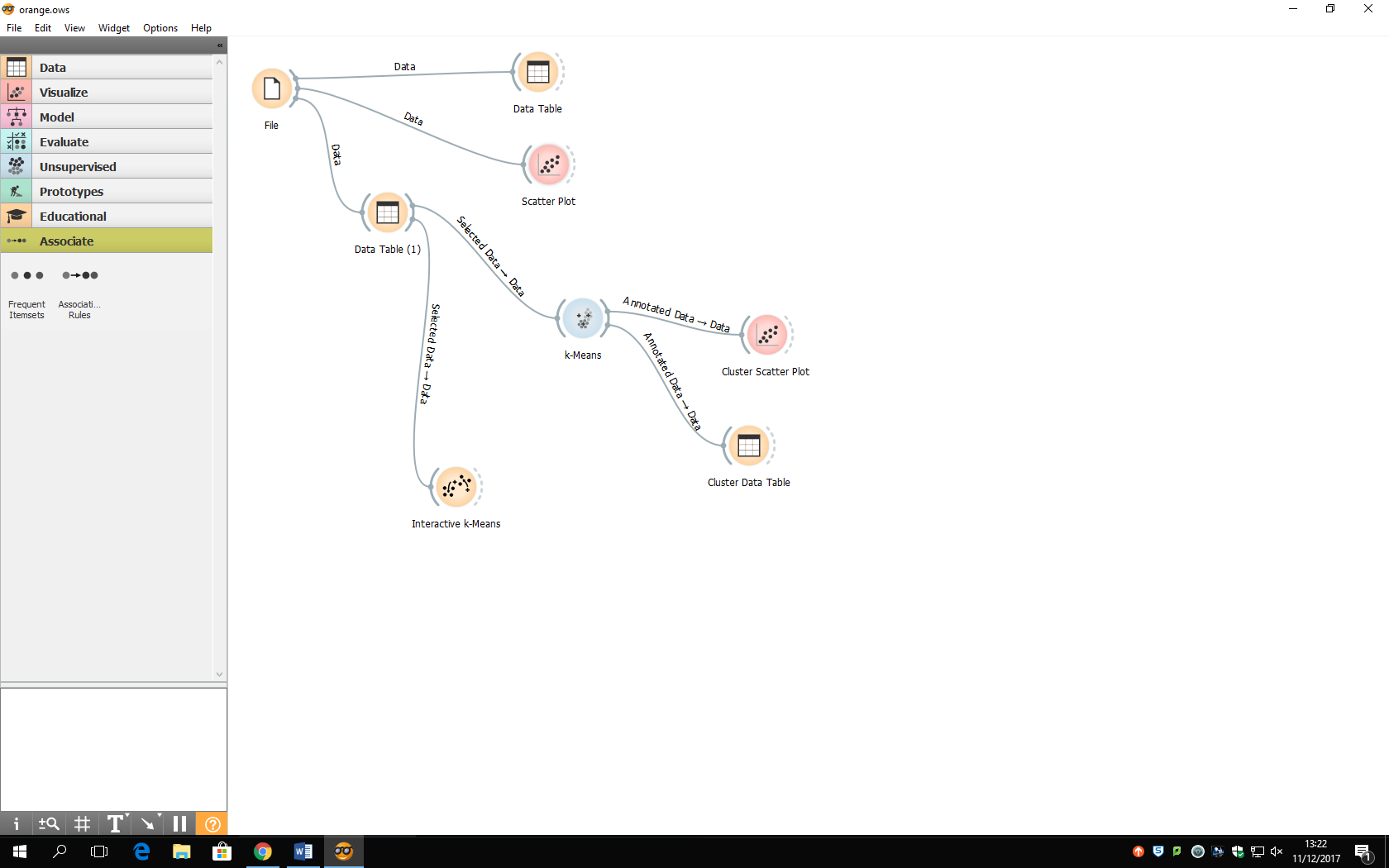
We can see the United States is clearly the single top country with almost *90,000* artists and over *18000* labels, this was predictable as the majority of big artists are worldly known to be from America.

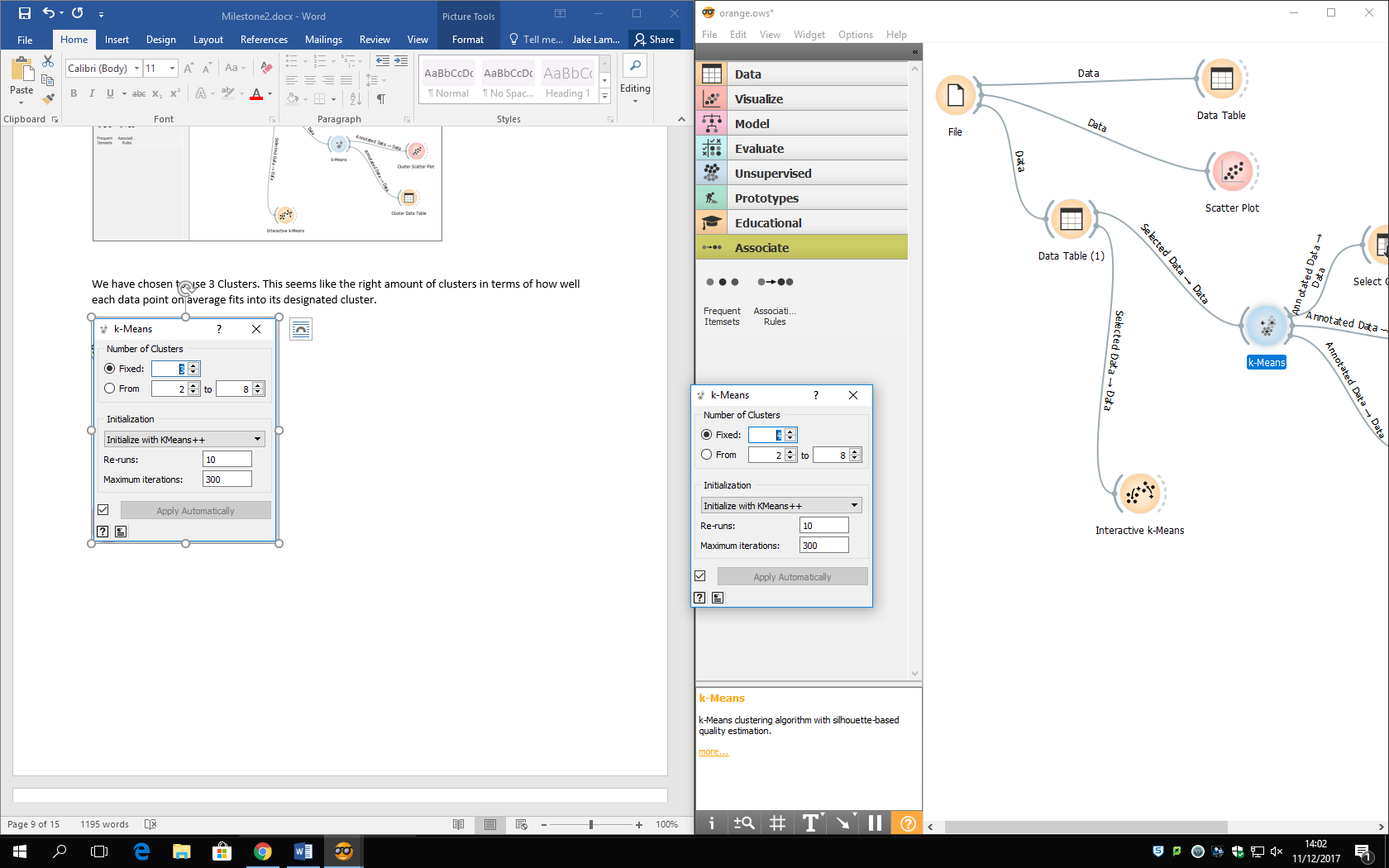
Then the second tier beneath in their own bracket consist of the UK, Germany, Japan & France which range between *3,000 – 10,000* labels and *18,000 – 45,000* artists, as you can see 75% of this group is from Europe.

Then you can argue the rest of the countries are in the same sort of vicinity of each other, and range from all the five continents around the world. But if you look at the screenshot above, the countries with the most artist and labels are clearly again from Europe.

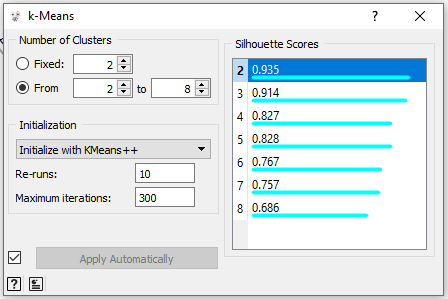
# Question 5

Here we had to use the kmeans algorithm to cluster the countries and decide how many clusters we need to select. Also discuss the optimisation of the number of clusters using various metrics (Silhouette, Distance inter-cluster, and Distance to centroids).

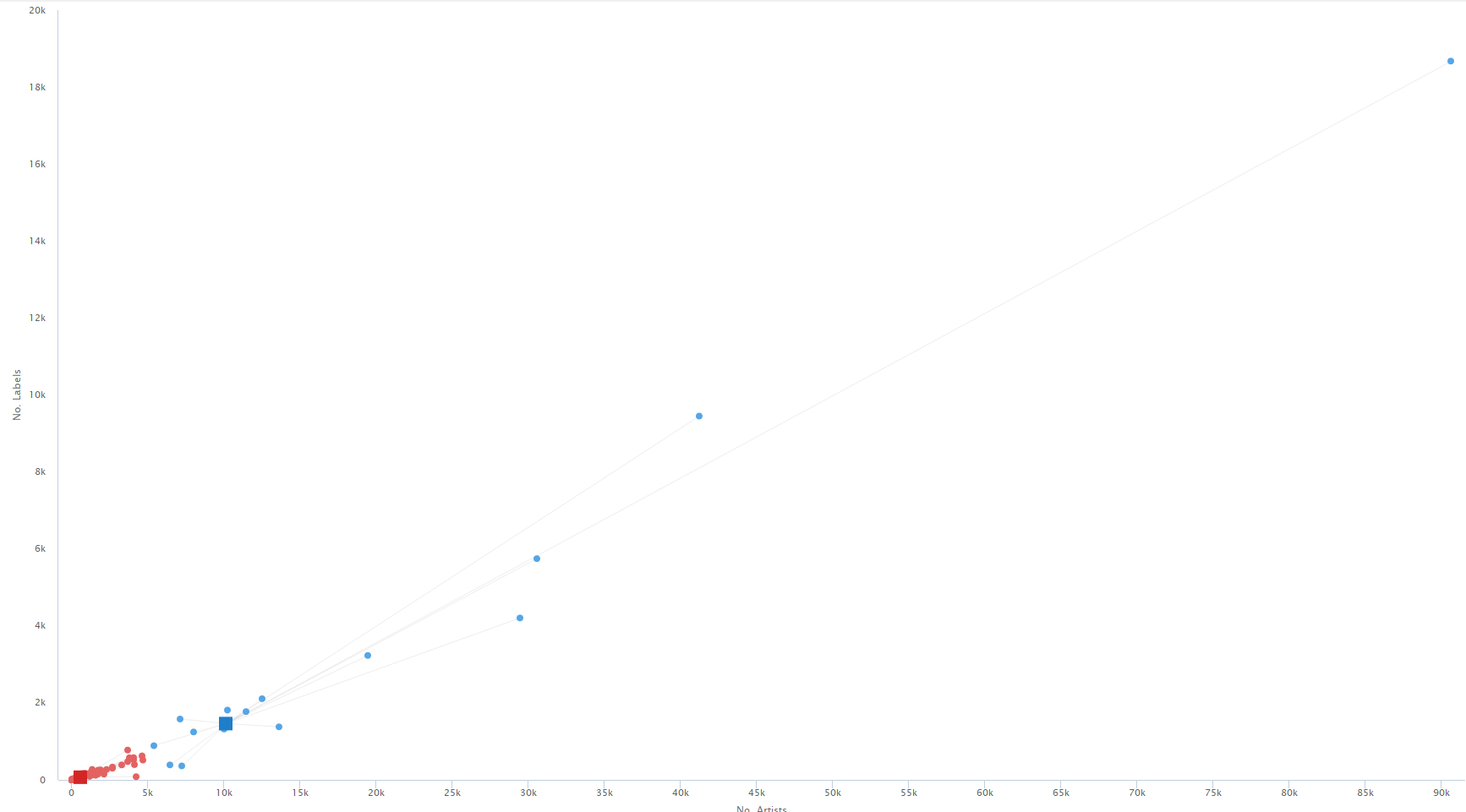


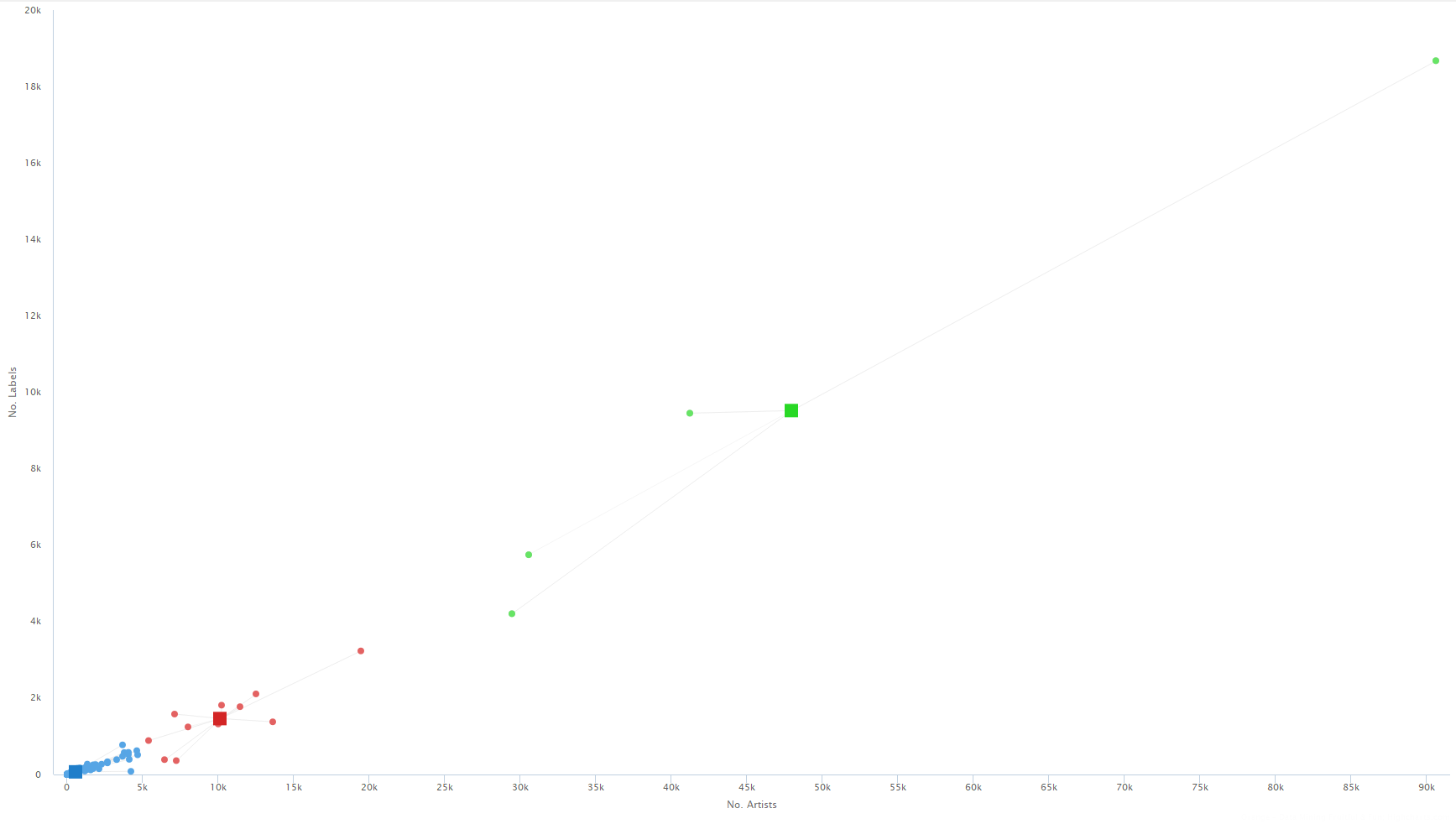
We have chosen to use 4 Clusters. This seems like the right amount of clusters in terms of how well each data point on average fits into its designated cluster.

We can use the Silhouette Score to help us decide the amount of clusters. It shows 2 clusters will be the best to use for our data. One of the drawbacks with Silhouettes in K-means, it works well on compact spherical shape clusters, and fails on shapes of a different kind.



Here we have visualised the clusters by colours in a scatterplot and in a data table for 2, 3 & 4 clusters then discussed the clusters found.





The above 2 screenshots are used by Interactive K-means, they show the distances to centroids with 2 & 3 centroids respectively. The membership lines shows the distances from the data point to the centroid.

This is an effective way to analyse data using centroids to define the clusters. The Recompute Centroids button can move the centroids to the centre of the data points, you can keep repeating this process until the centroids stabilize thus determining the amount of clusters.

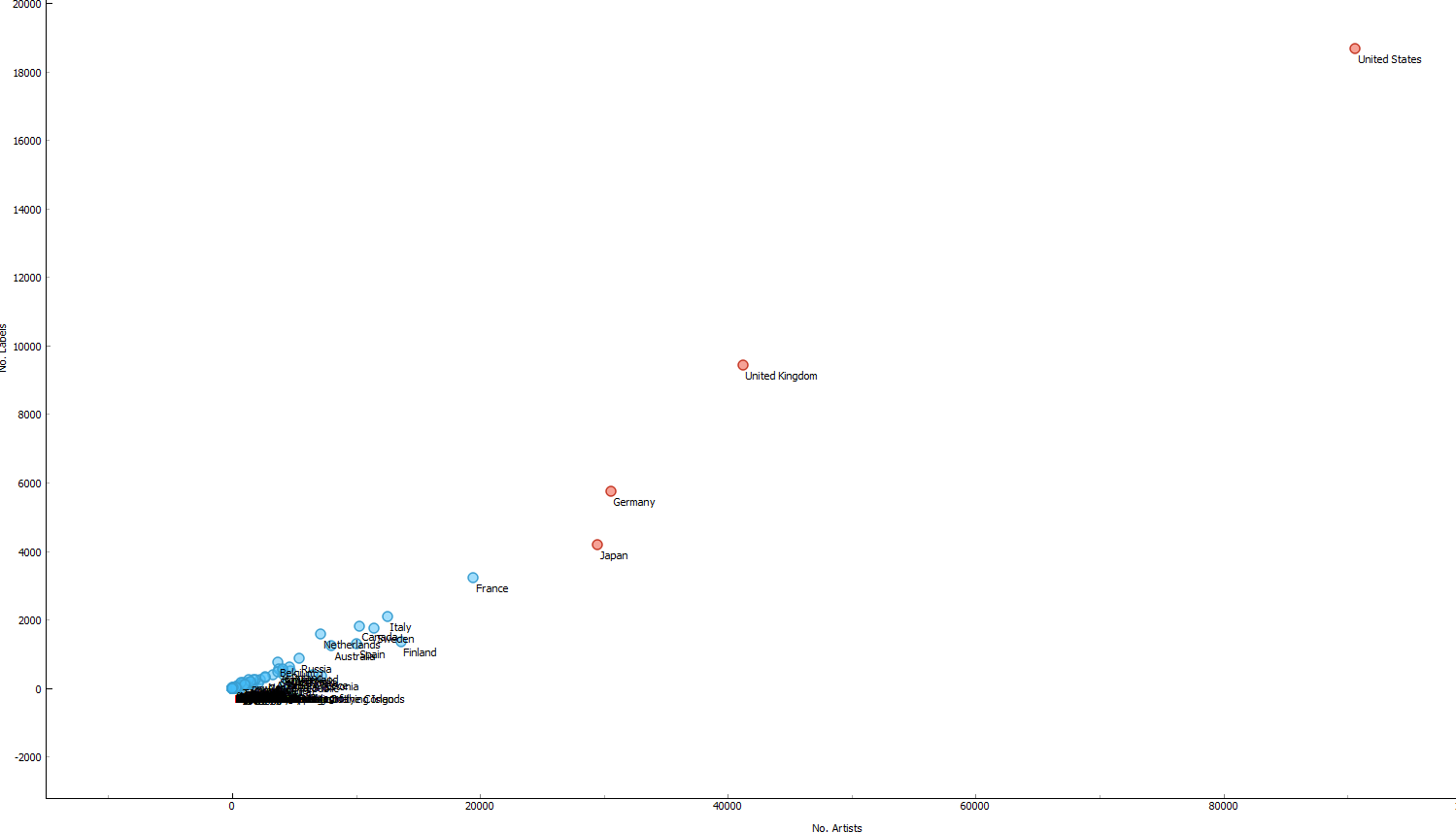
The Inter-cluster Distance between two clusters may be any number of distance measures. For example you can measure the average distance between a data point and the points in its own cluster (distance A), or between the data point and the closest data point in another cluster (distance B).

The Silhouettes score will be determined by the shorter distance A is and the farther distance B is to give the most accurate judgement.

Also for distance inter-cluster you need to maximize the distance between clusters from their data points, this is an effective way to determine how many clusters can be split up.

Visualising clusters in a **Scatter Plot** for **2, 3 & 4** clusters.

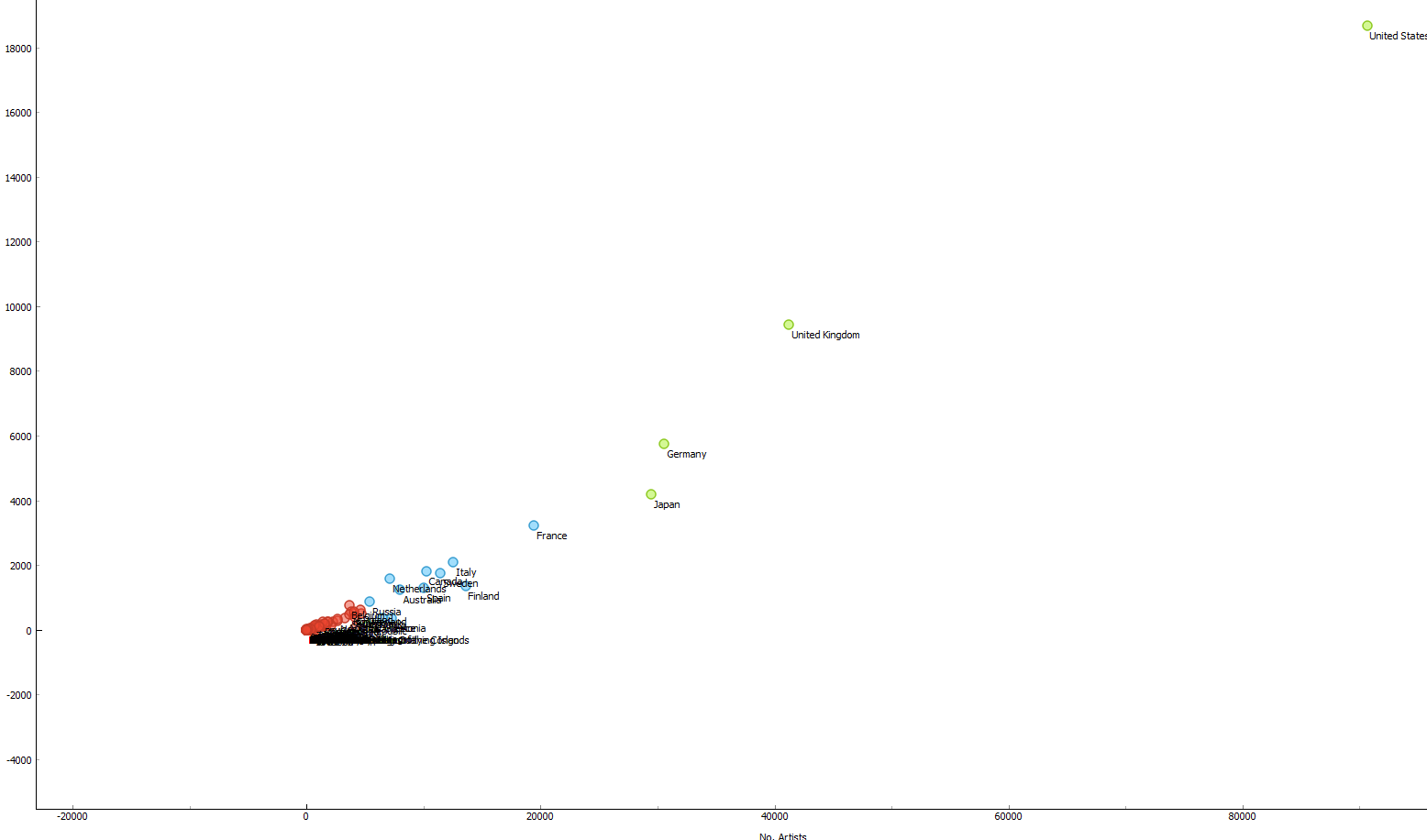
2 Clusters

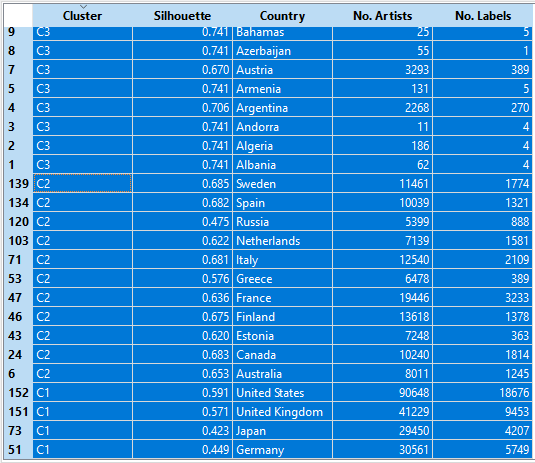




For 2 clusters, it was clear the 4 countries in cluster 2 have far more artist and labels than the rest, in fact the UK and America have double the amount compared to the most highest country in cluster 1, France.

3 Clusters



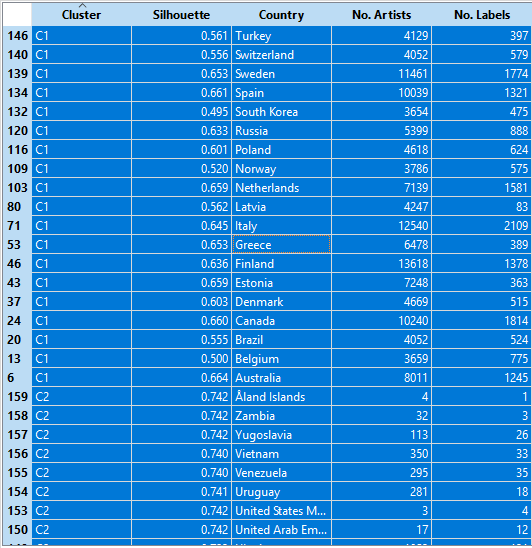
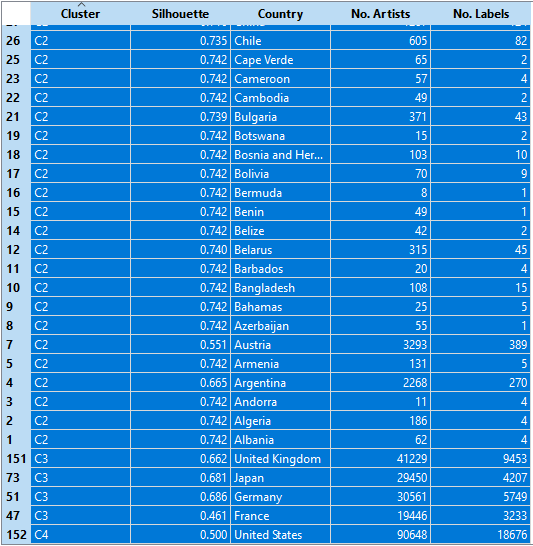


With 3 clusters, you notice there are the same countries still in cluster 1 and the cluster 2 from before has been split into two.

It’s clear there are more countries in cluster 3 than cluster 2, from the data for Artists & Labels cluster 3 countries are very unpopular compared to the two clusters. A clear distinction is starting to appear separating the three clusters.

4 Clusters





I had to use two screenshots to display an equal amount of countries from the four clusters.

Cluster 4 has only the United States of America, this makes sense because it is almost double in terms of Artist & Labels compared to the highest point in cluster 3, the UK.

Cluster 3 is the same from before except it’s minus the US and gained France, this is accurate as these countries in cluster 3 are most similar data points.

Cluster 2 data points are quite distant away from cluster 3 data points and in their own close group away from cluster 1.

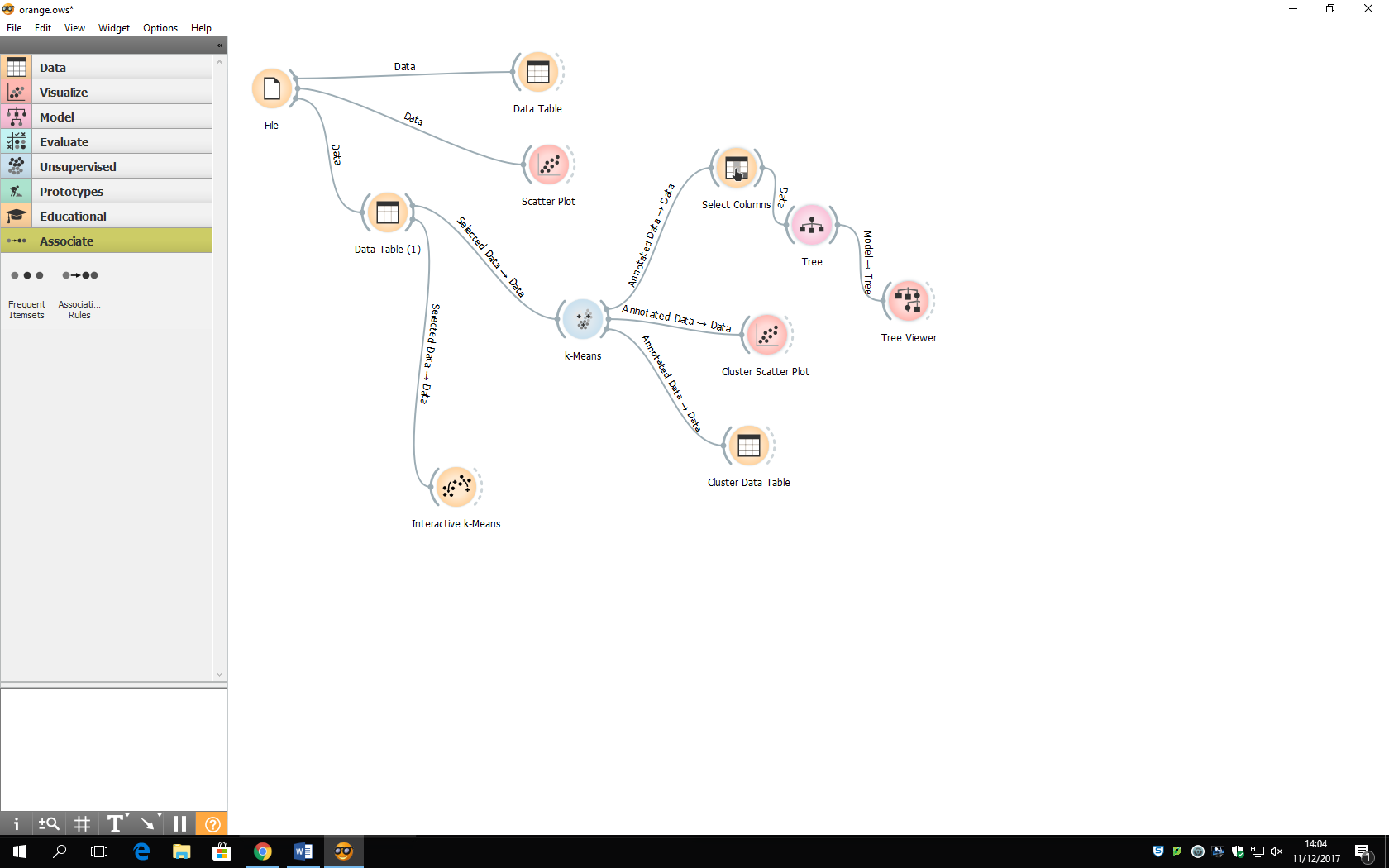
As for cluster 1 the data points seem to be under 400 labels and 4000 artists and are most definitely more grouped together when looking at the scatter plot.

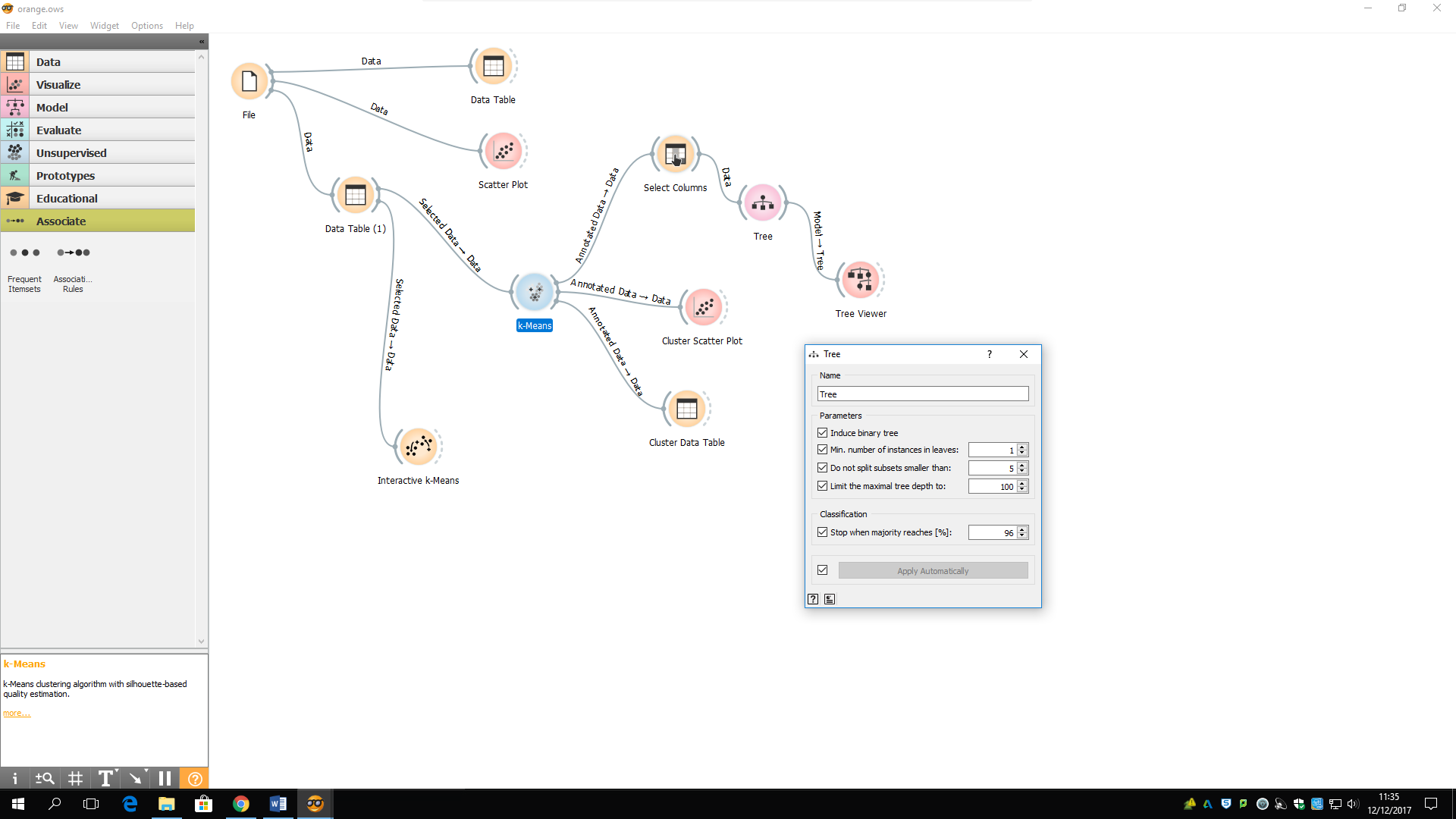
Overall if you look at each cluster in the scatter plots, each cluster has their data points having very similar distances from each other so cluster 3 has long distances where cluster 1 has very short distances.

To compare the 3 clusters we can conclude that determining on how many clusters there are, centroids try to maximize the distance of its farthest data point and same goes for each centroid there is, this then gives the most accurate results when calculating the clusters.

# Question 6

We have to apply a classification tree and a classification tree viewer to the output of kmeans.



We have to change the parameters of the Tree to suit our needs. We set the maximum tree depth to a 100 and the min. number of instances in leaves to 1 so the tree can include clusters that may have only one data point.

By the way C1 should be C2 and C2 should be C1 instead of the awkward order of C2, C1, C3 & C4.

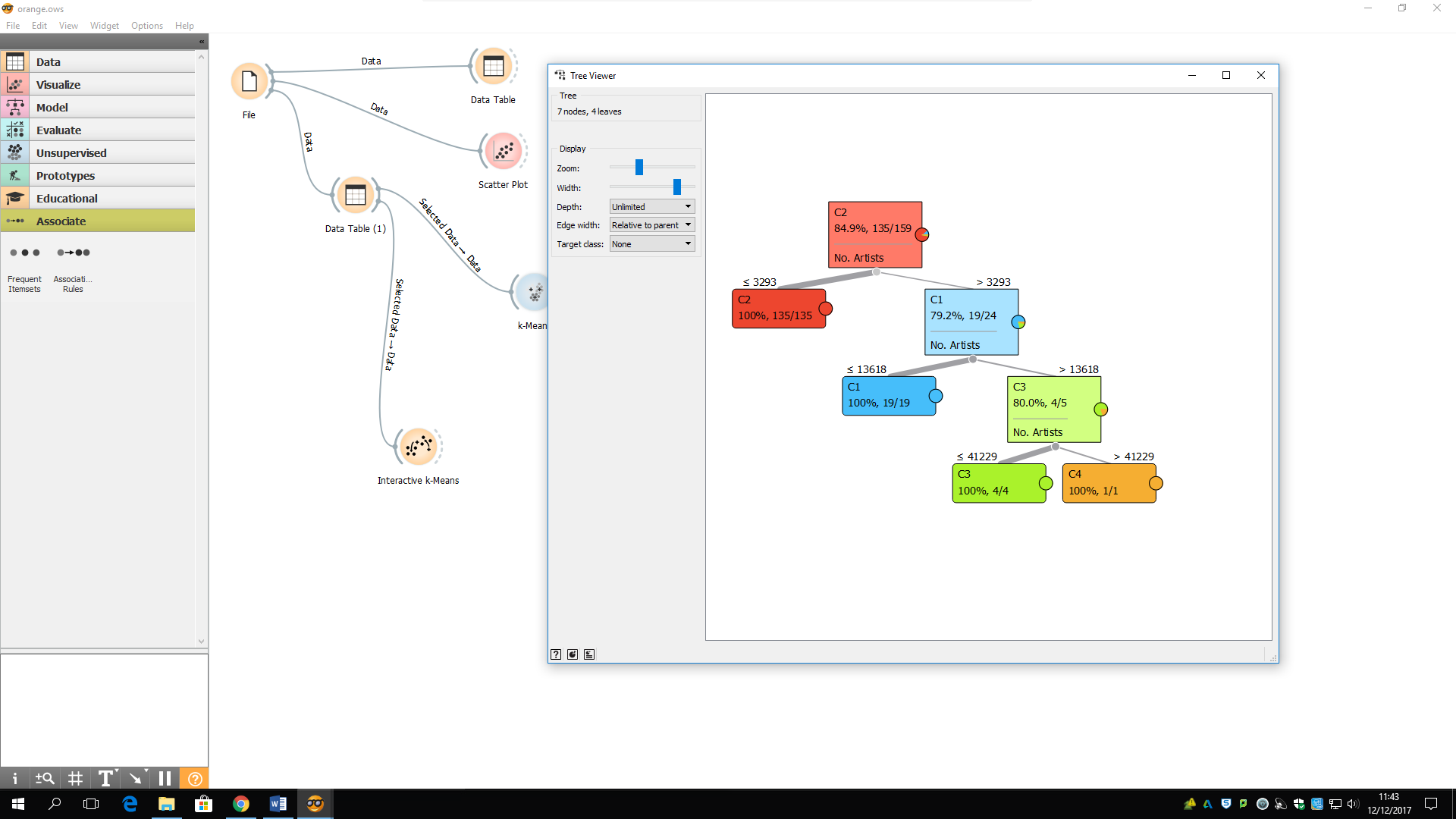
Here is what our Tree Viewer looks like when we have our 4 Clusters. Our tree is formed according to the number of artist and shows how each cluster splits up.

So C2 splits at 3293 as this is the highest data point in its cluster in terms of no. of artist, and it states that C2 has 84.9% (135/159) of data points when you combine it with C1. This determines C2 has the majority of data points over C1.

C1 splits at 13618 its highest point for no. of artist, it states C1 has 79.2% (19/24) when combined with C3 therefore having the majority of data points over C3.

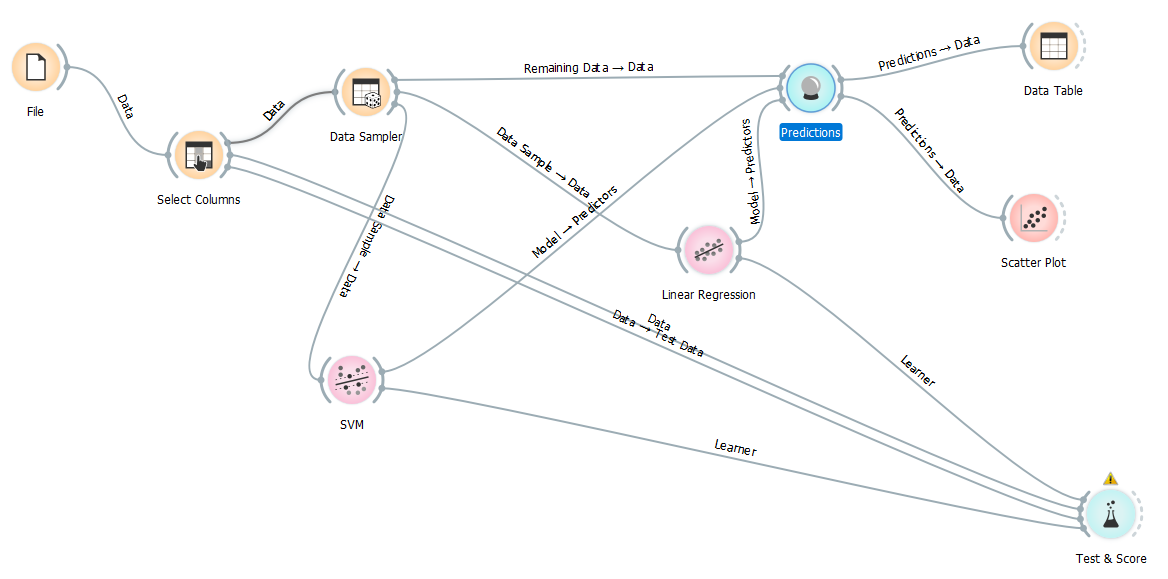
C3 splits at 41229 its highest point for no. of artist, it states C3 has 80% (4/5) when combines with C4, again this has the majority of data point over C4.

We can interpret from C2 to C4 the number of data points decrease from cluster to cluster respectively.

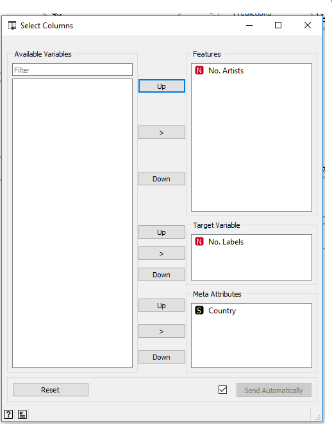


# Question 7

This is the setup to look at predictions. We have connected a select column to choose the number of labels as target variable from the file. We use a data sampler and applied a linear regression to the sample then connected the remaining data and the linear regression coefficients to a prediction and visualised the predicted values in a table and a scatterplot. We also used a test&score widget to evaluate and compare with the SVM regression.



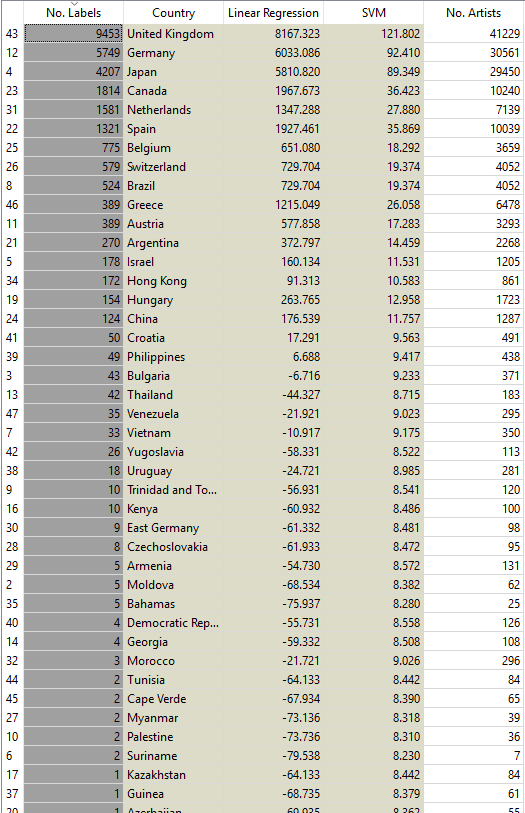
In the Select Column widget we have set the No. of Labels as our target variable and used No. of Artist as our feature.



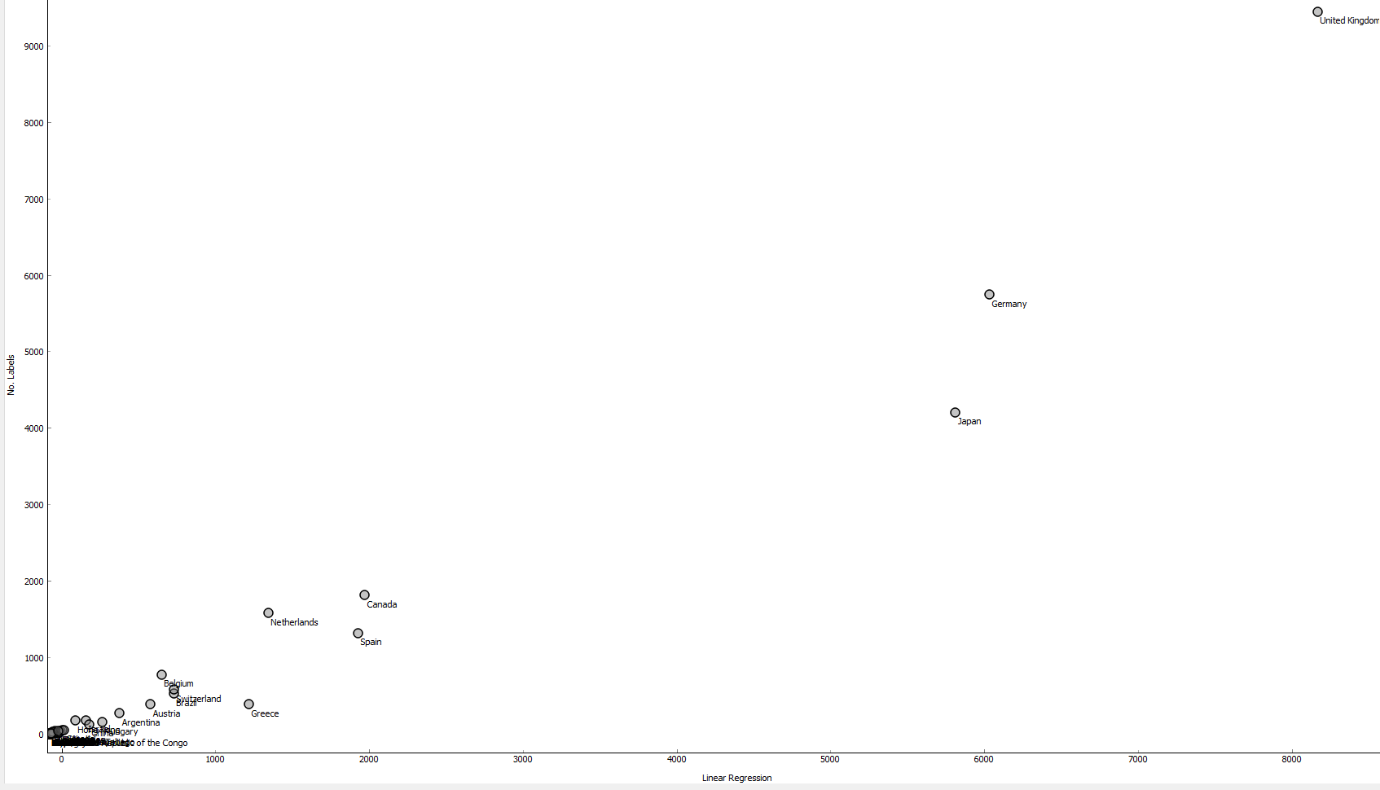
This is a screenshot of the data from the SVM and Linear Regression in the predictions widget. With the No. of labels being the target variable we can compare the SVM and Linear Regression to the original data.



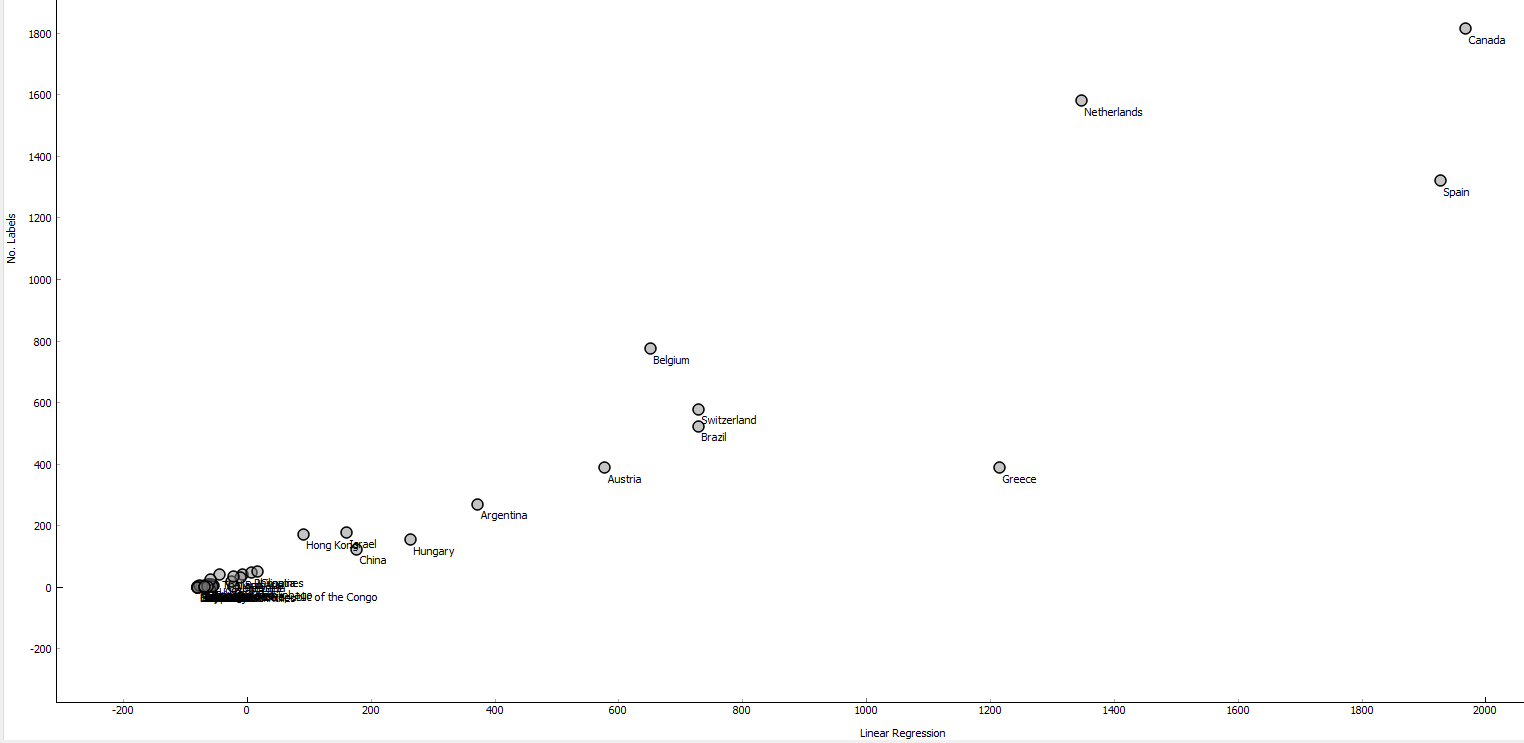
This is a screenshot of the Data Table. Again this is a good way to compare the real data to the predicted data.



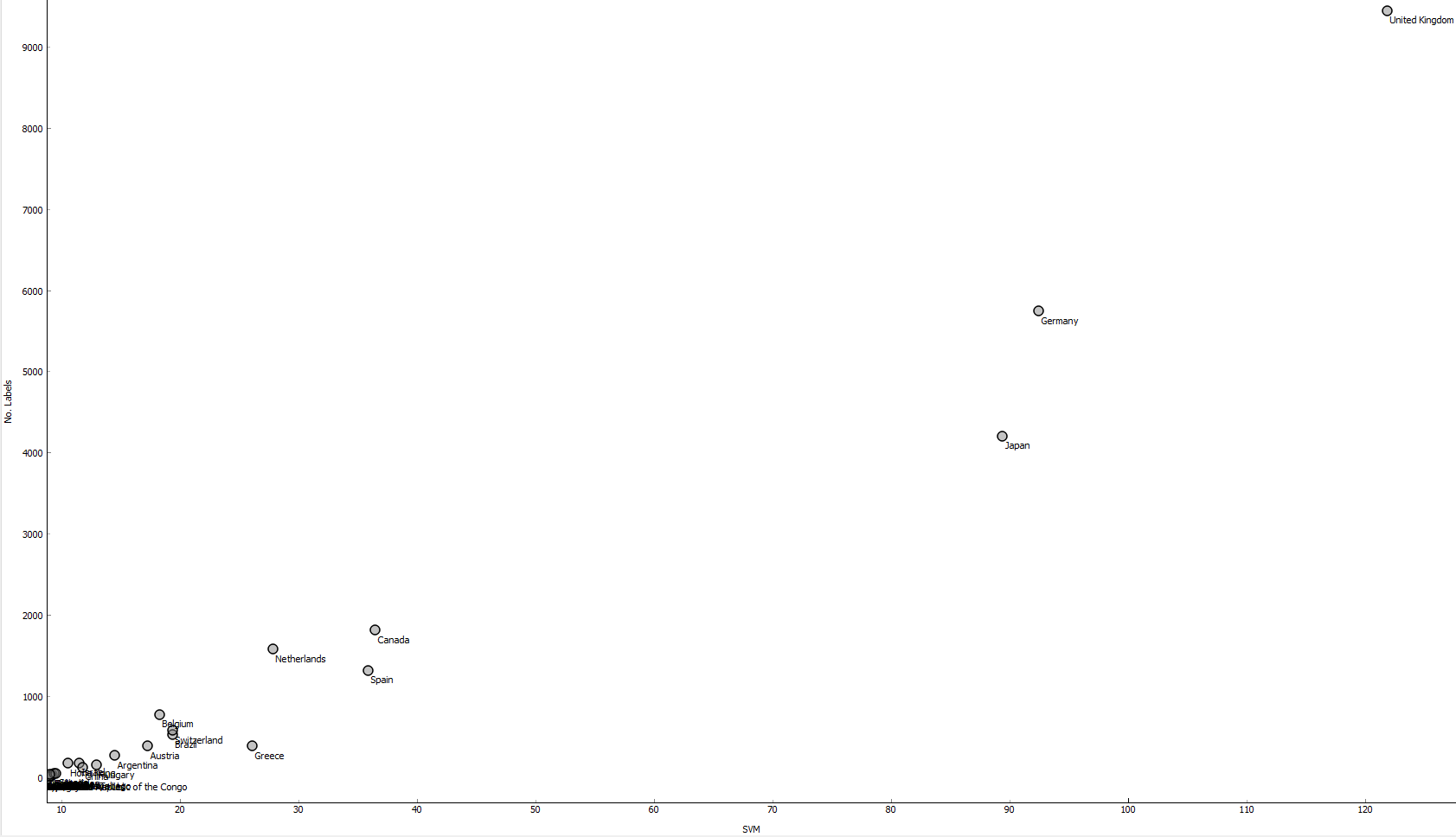
This is a screenshot of the scatter plot with No. of Labels set as the Y axis and the Linear Regression is set as the X axis.



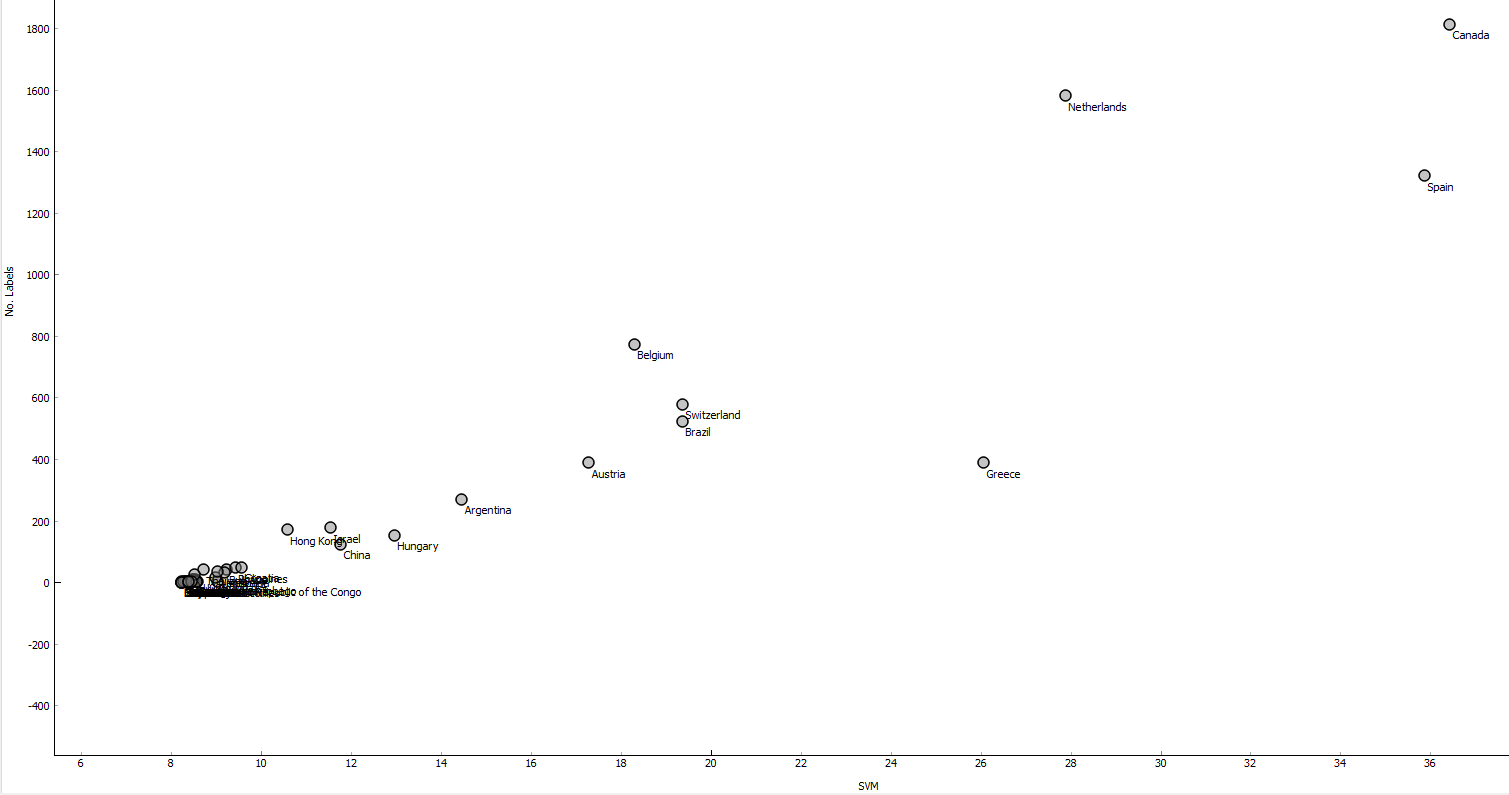
This is a more zoomed in version of the scatter plot above.



This is a screenshot of the scatter plot with No. of Labels set as the Y axis and the Linear Regression is set as the X axis.

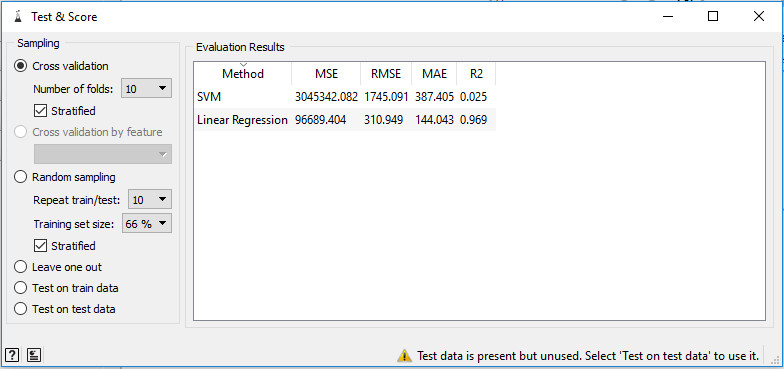


This is a more zoomed in version of the scatter plot above.



Above we have the predicted data and the real data in scatter plots. To compare the quality of the SVM vs the Linear Regression there data points look very identical in their positions on the axis but the difference was the No. of labels were considerably different.

This is a screenshot of the test & score.



The (MSE) measure for the average pf the squares of the errors is very high for SVM and Regression so this means there is a great distance between the estimator and what its estimated.

The (R2) proportion of the variance in the dependant variable that is predictable from the independent variable is pretty low and is at a good level.

# Question 8

The algorithm to compute frequent item sets is called the Apriori algorithm. It is an influential algorithm mining frequent item sets from transactional database for Boolean association rules. In this algorithm, frequent item sets are extended one item at a time (also known as the candidate generation process). Then groups of candidates get are tested against the data. Apriori is designed to operate on databases containing customer transactions, e.g. a collection of items get bought by the customer or details of a website fragmentation.

The complexity of the problem of finding frequent items sets is the counting method iterates through all of the transactions each time, basically the database gets scanned at every level. Also the candidate generation can generate duplicates depending on the implementation, and consistent items make the algorithm a lot heavier.

Techniques used to improve the efficiency:

1. **Sampling**: Mining on a subset of given data with a lower support threshold, this method determines the completeness.
2. **Hash-based itemset counting**: a K-itemset whose corresponding hashing bucket count is below the threshold means it can’t be frequent.
3. **Dynamic itemset counting**: This method will only add the new candidate if all of their subsets are estimated to be frequent.

# Question 9

Using the same methodology as above, we are going to query the database, produce a CSV file, download it and analyse it using Orange and its various widgets for other aspects of the Musicbrainz database.

This is subquery querying the Musicbrainz database we are going to use to analyse. The fist query in the subquery is counting how many artists there are in each genre and the second query is counting how many labels in each genre.

SELECT sub1.genre "Genre", sub1.cnt "Total Artist", sub2.cnt "Total Labels"

FROM(SELECT COUNT(a.id) as cnt, tg.name as genre

FROM artist a

JOIN artist\_tag at ON a.id = at.artist

JOIN tag tg ON tg.id = at.tag

GROUP BY tg.name

ORDER BY cnt DESC

LIMIT 100) as sub1,

(SELECT COUNT(l.id) as cnt, tg.name as genre

FROM label l

JOIN label\_tag lt ON l.id = lt.label

JOIN tag tg ON tg.id = lt.tag

JOIN label\_type la ON la.id = l.type

GROUP BY tg.name

ORDER BY cnt DESC) as sub2

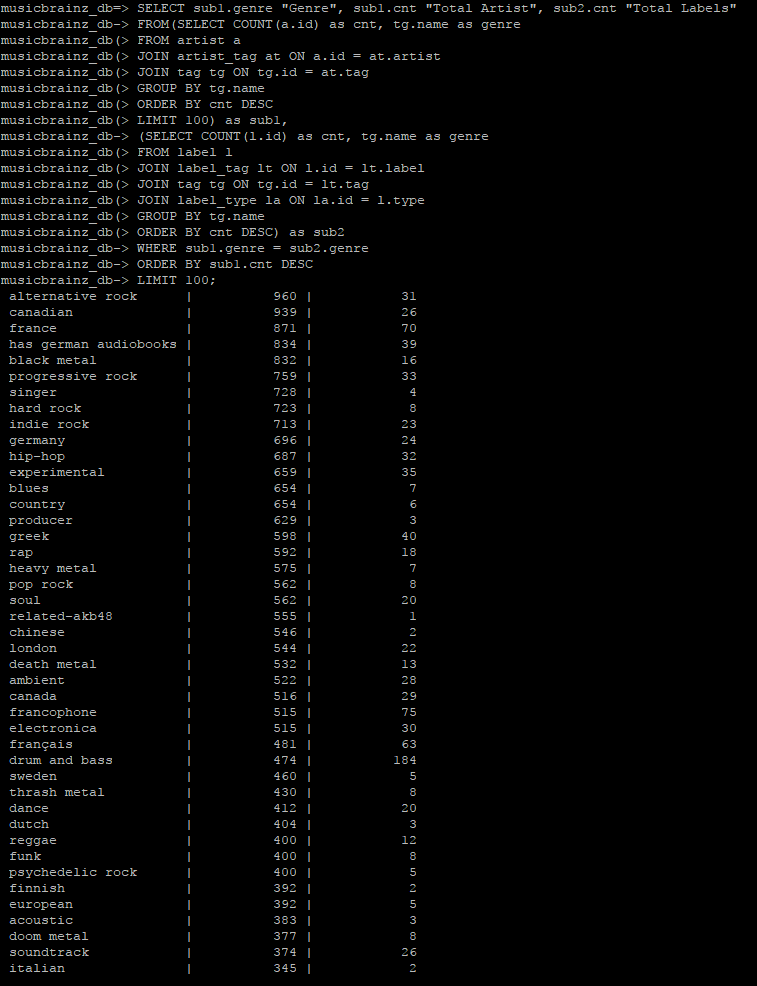
WHERE sub1.genre = sub2.genre

ORDER BY sub1.cnt DESC

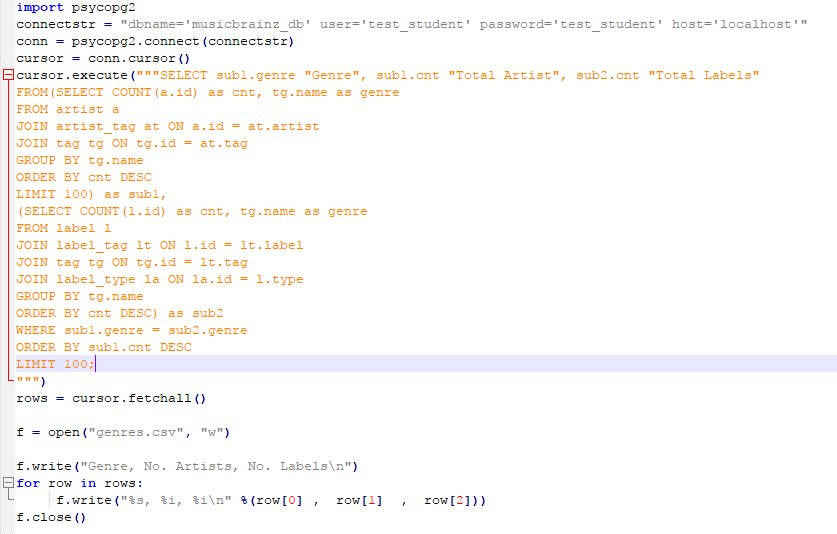
LIMIT 100;

Here we are running the query in the Musicbrainz database with its results. We have limited the artists and labels to 100 each this so we don’t have a surplus of dat when coming to nalysing it in Orange.

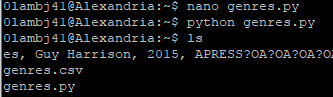
Plus it is ordered by artist and label respectively, in descending order so we have the data with the highest numbers.



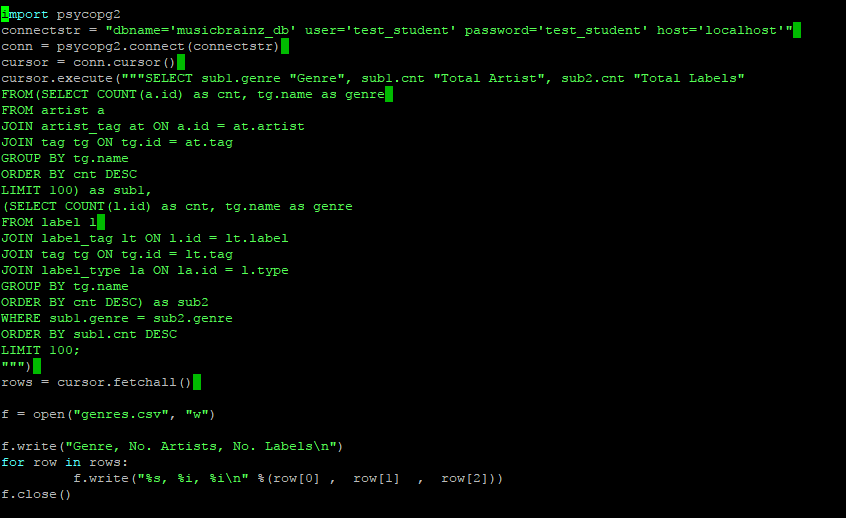
Here we writing a python script to connect to the Musicbrainz database and produce a csv file.



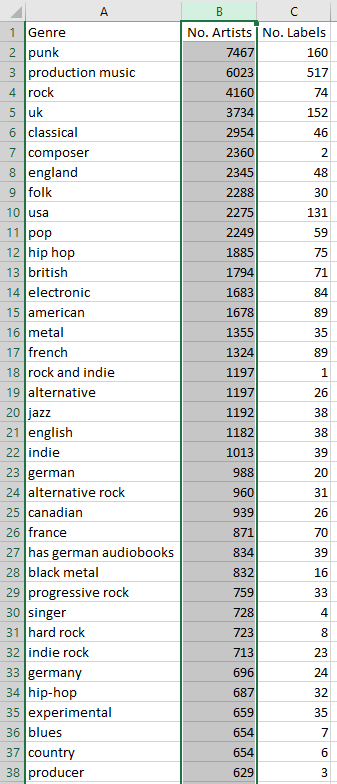
Here we are calling the genres.py python script, in return it creates the csv file with the results from the database.

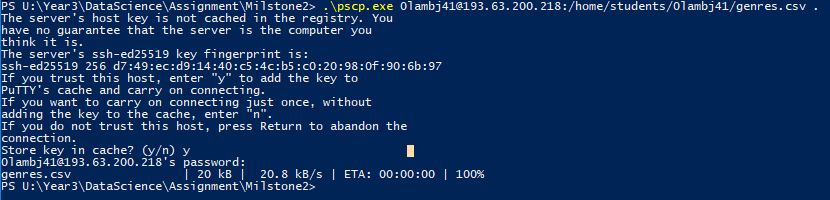


Once this file has been created we can write the function. Our function here connects to the database and produces a CSV file **genres.csv.**

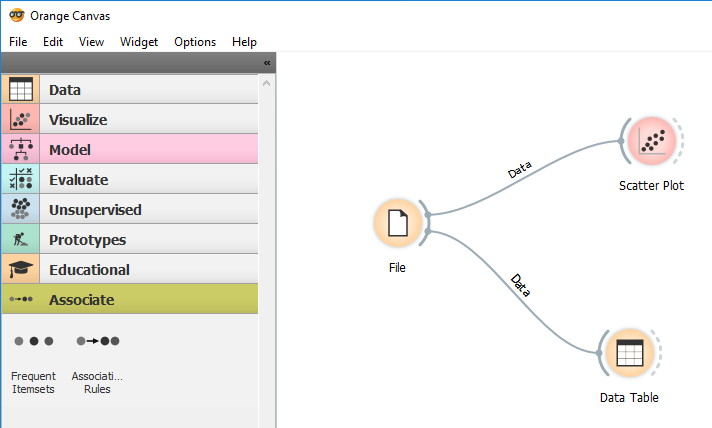


This is the extract from our **genres.csv** file.

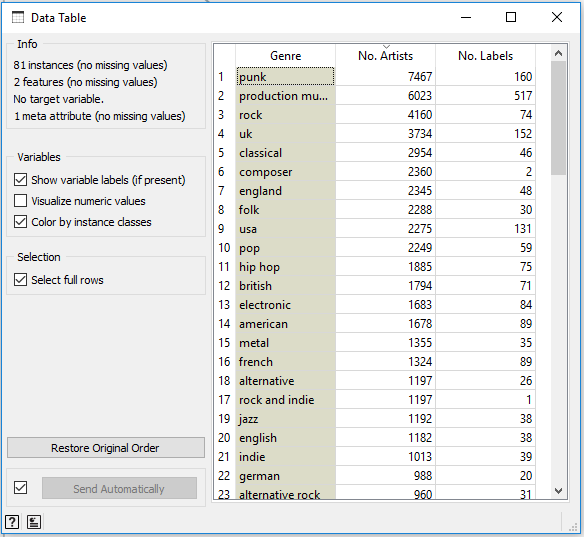


Here we downloaded the file **genres.csv** from Alexandria using the program **pscp.exe.** We opened up a program called Powershell, followed the directory path to the **pscp.exe** program. Then we log into our account and downloaded the **genres.csv** file.

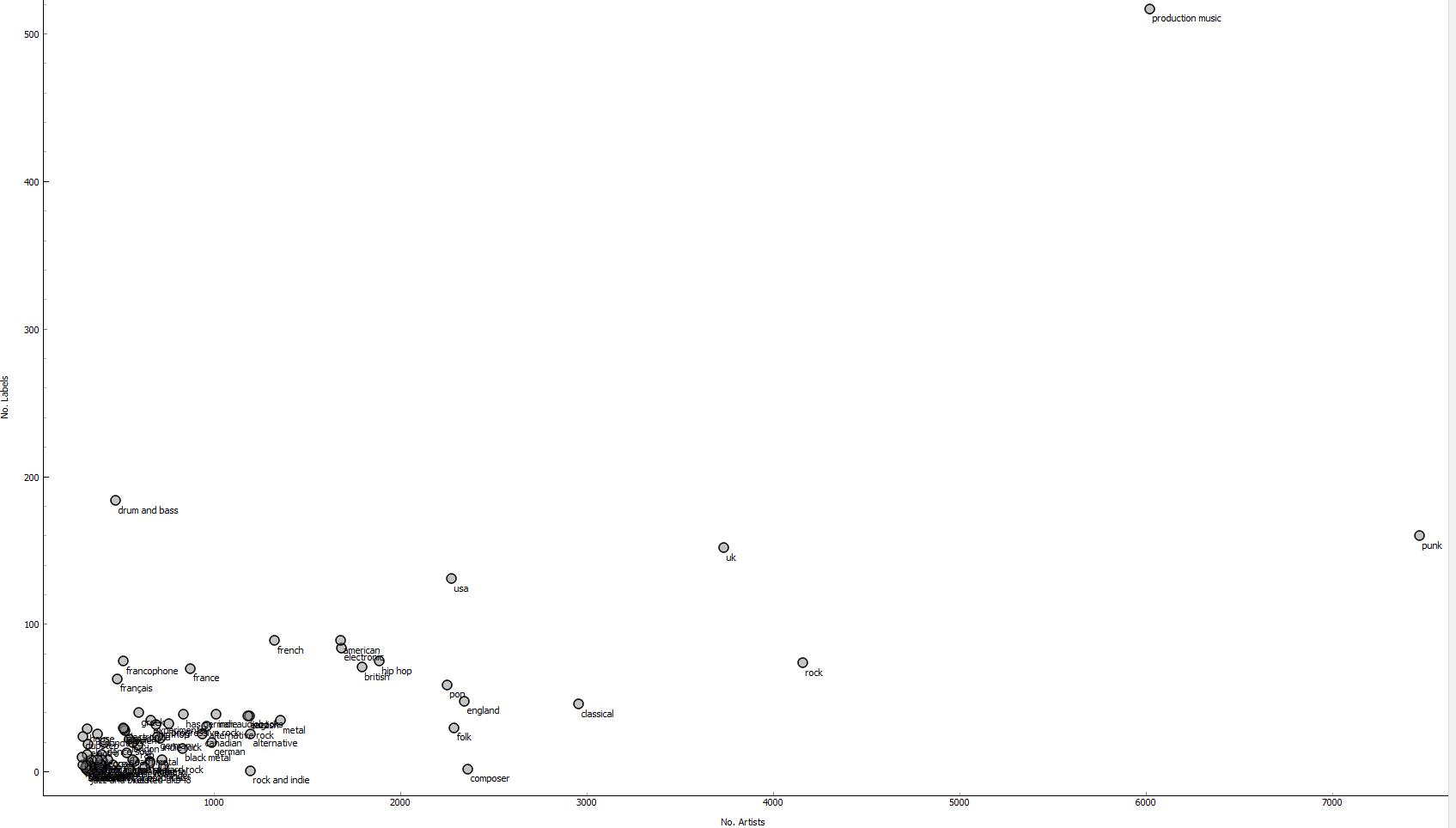
In **Orange.** We have created a **Data Table** and a **Scatter Plot** to visualize the data.

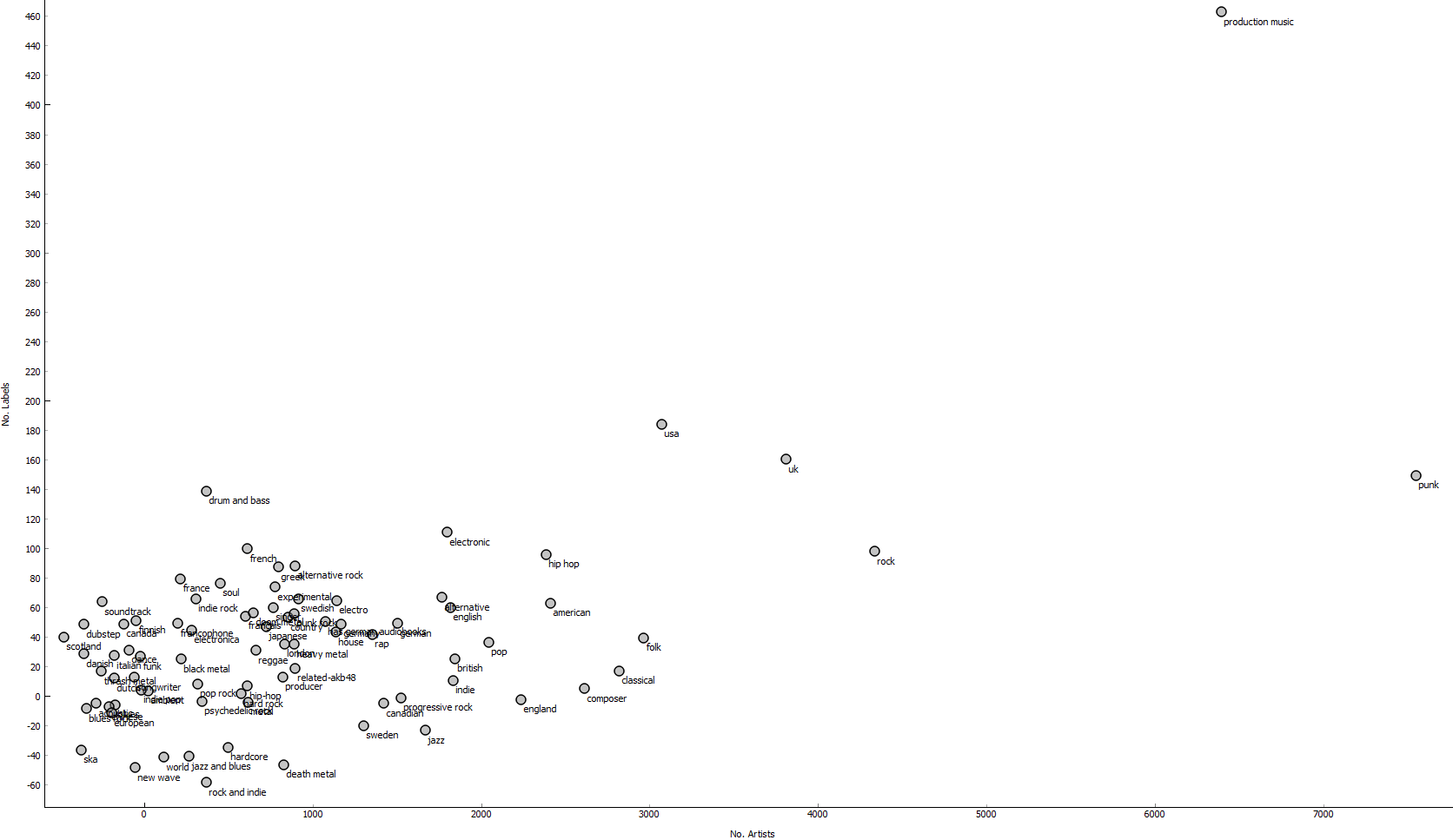


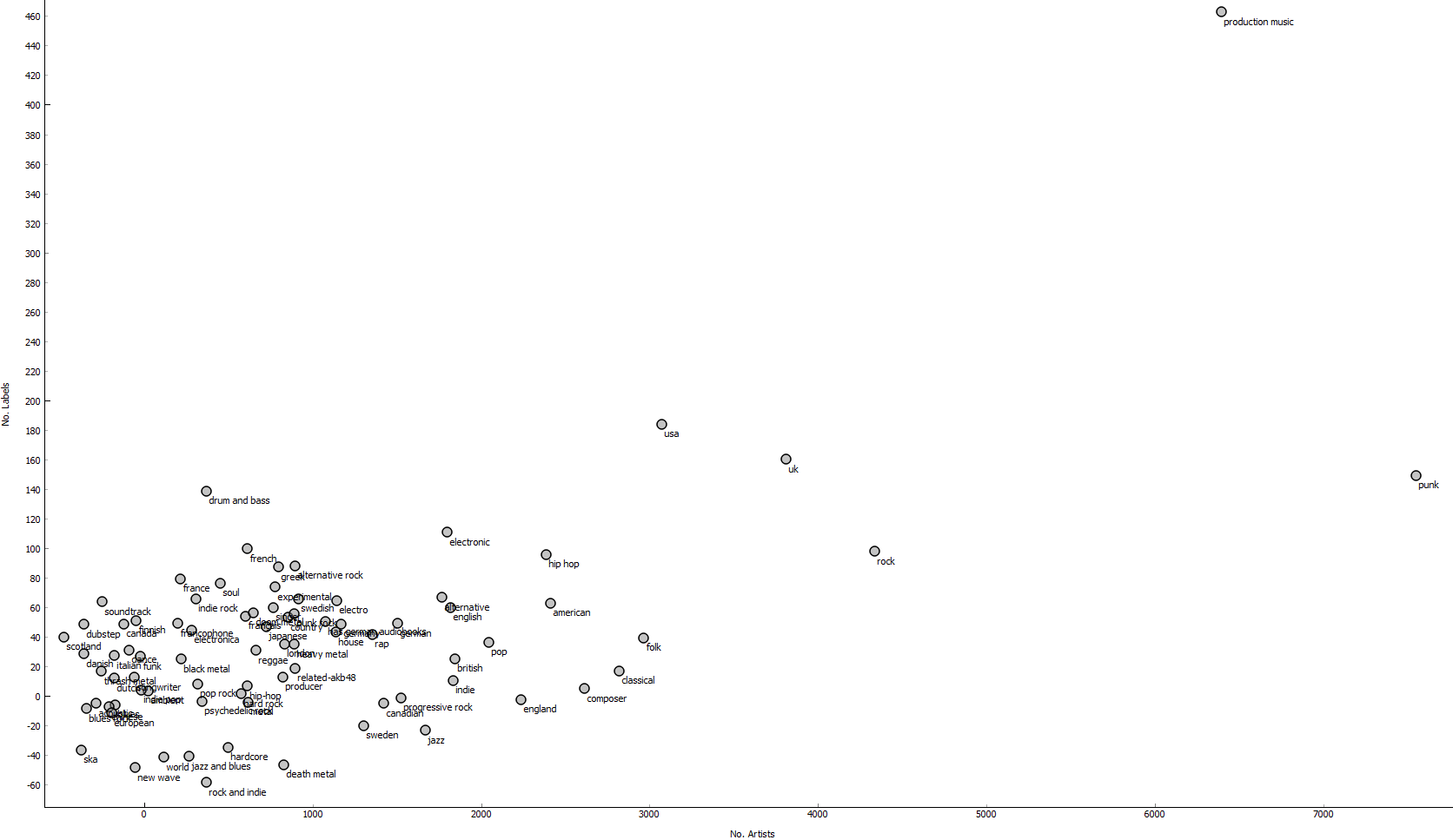
We sorted the table by No. Artists as there are obviously more artists than labels.



We added labels for the genre names on the Scatter plot to make each data point distinct.



This shows the majority of the Countries with 2% jittering due to them being so close together. But I must state that this 2% jittering can cause some of the data to become inaccurate as seen below. 

Slightly more zoomed in.

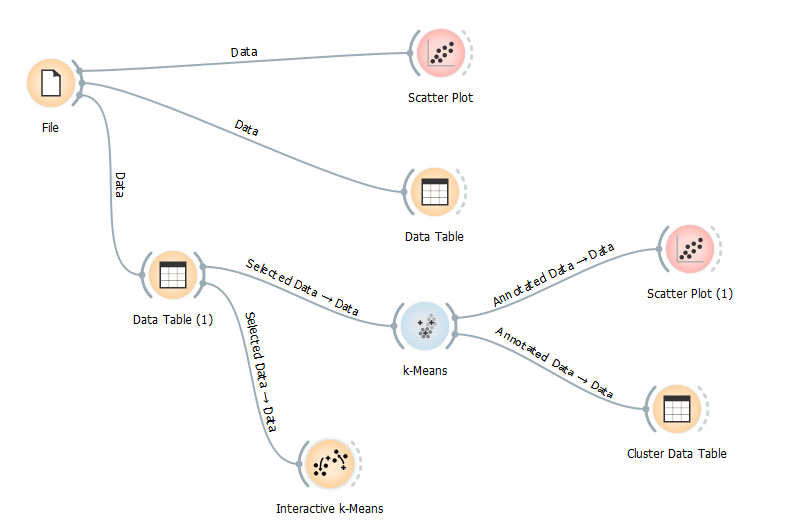
From the screenshots above we can analyse the data and conclude which are the top genres.

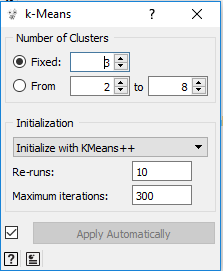
The first thing we notice is some of the genres are country names instead of actual music genre names, but this is what we have in our results so we analyse all data.

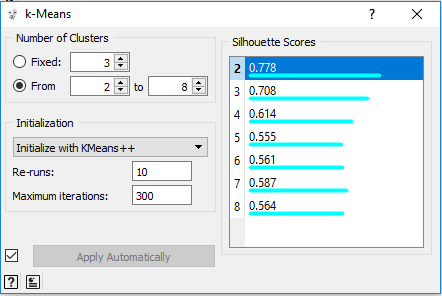
Production music is general recorded music which makes sense to have the highest numbers. The more noticeable genres like Punk, Rock and Drum & Bass are very high in numbers but they do compare against in each other in a different way.

The Punk genres no. of artist are almost 15 times more than Drum & Bass but Drum & Bass has slightly more no. of labels.

Here we had to use the kmeans algorithm to cluster the genres and decide how many clusters we need to select. Also discuss the optimisation of the number of clusters using various metrics.

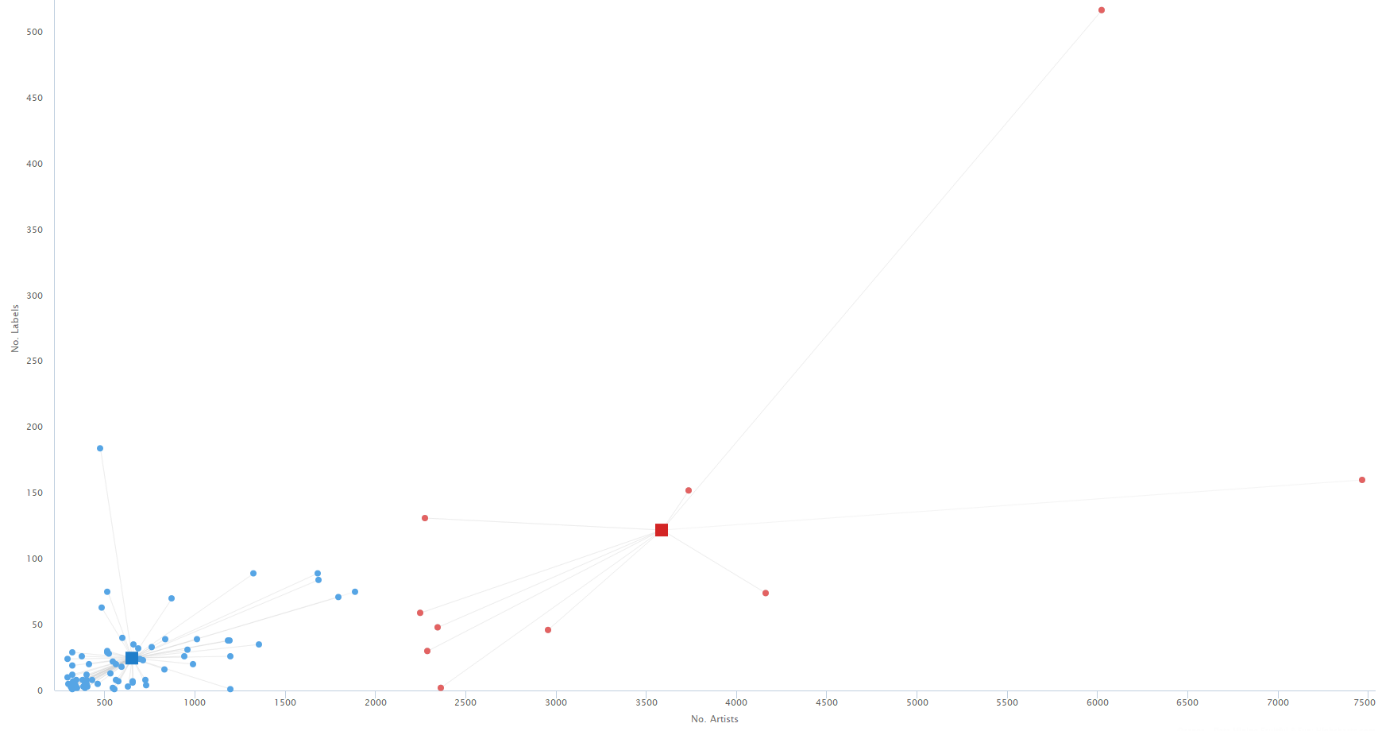


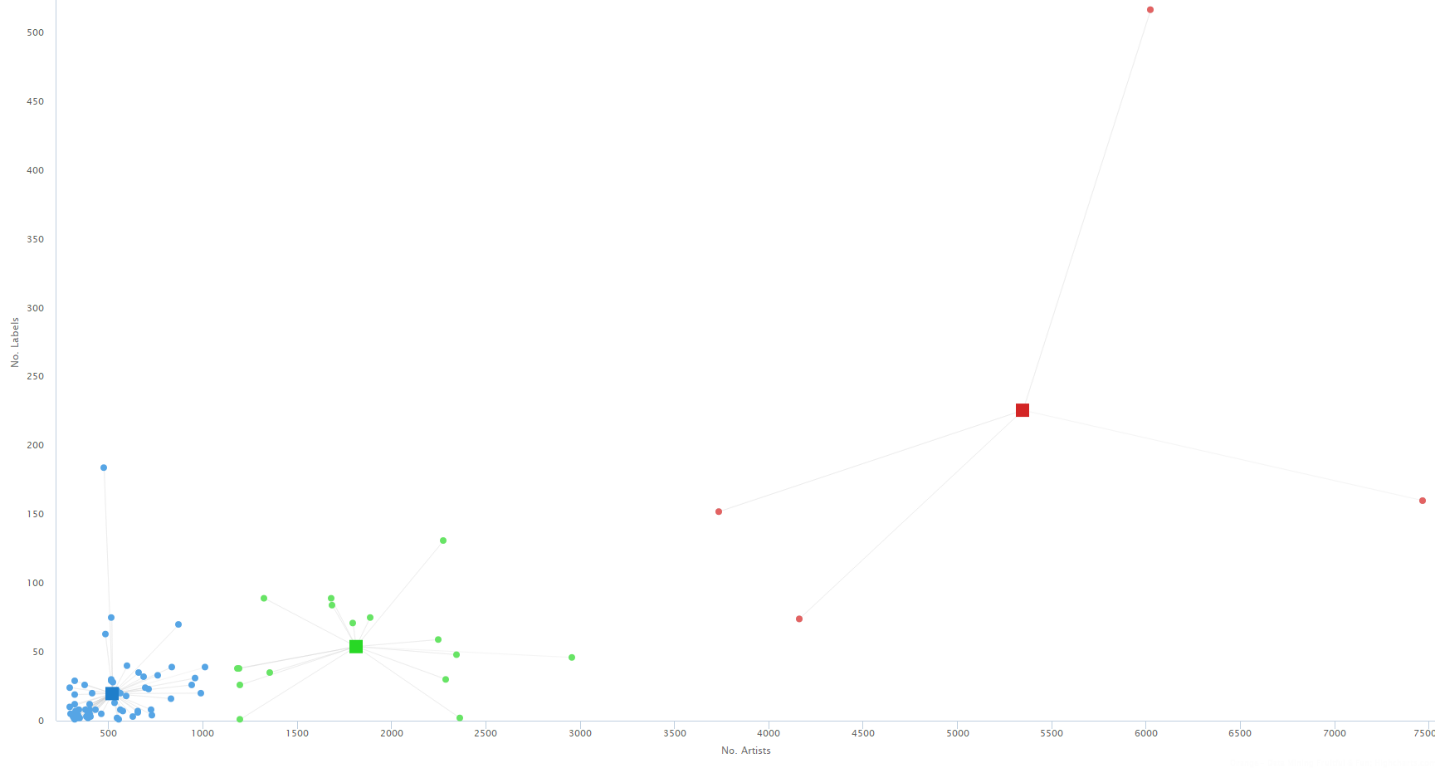
We have chosen to use 3 Clusters to best represent our data.

We can use the Silhouette Score to help us decide the amount of clusters. Again, it shows 2 clusters will be the best to use for our data. However, we have decided on 3 clusters to best display the clusters.

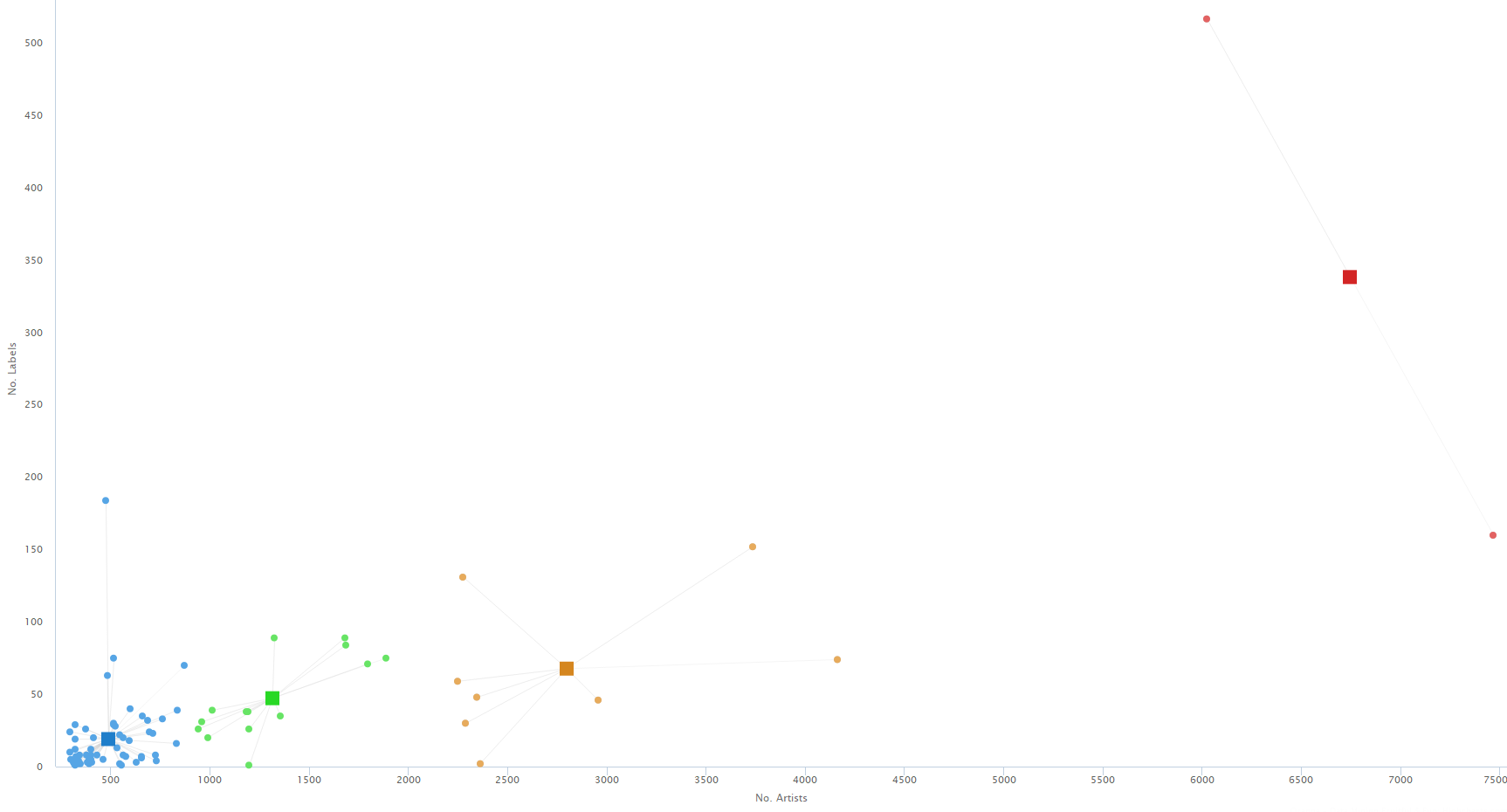
Here we have visualised the clusters by colours in an interactive K-means.

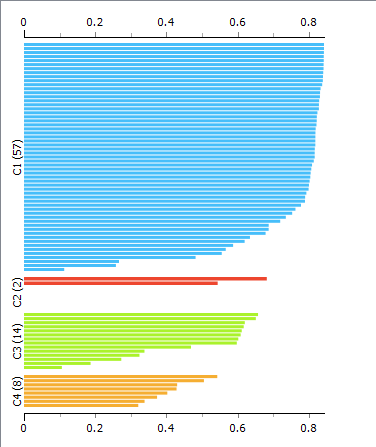
2 clusters



3 clusters

4 clusters



Silhouette plot 

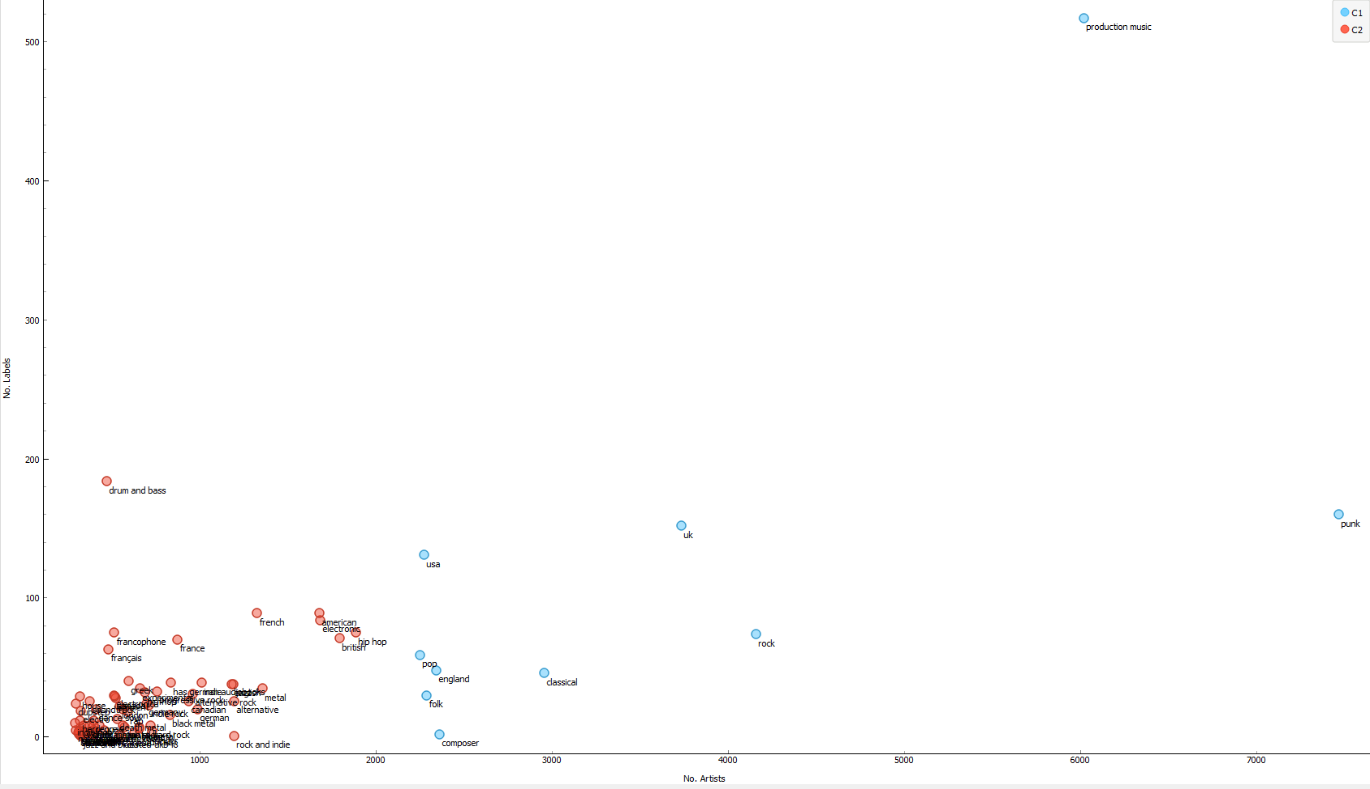
The above 3 screenshots are used by Interactive K-means, they show the distances to centroids with 2 & 3 centroids respectively. The membership lines shows the distances from the data point to the centroid.

This is an effective way to analyse data using centroids to define the clusters. The Recompute Centroids button can move the centroids to the centre of the data points, you can keep repeating this process until the centroids stabilize thus determining the amount of clusters.

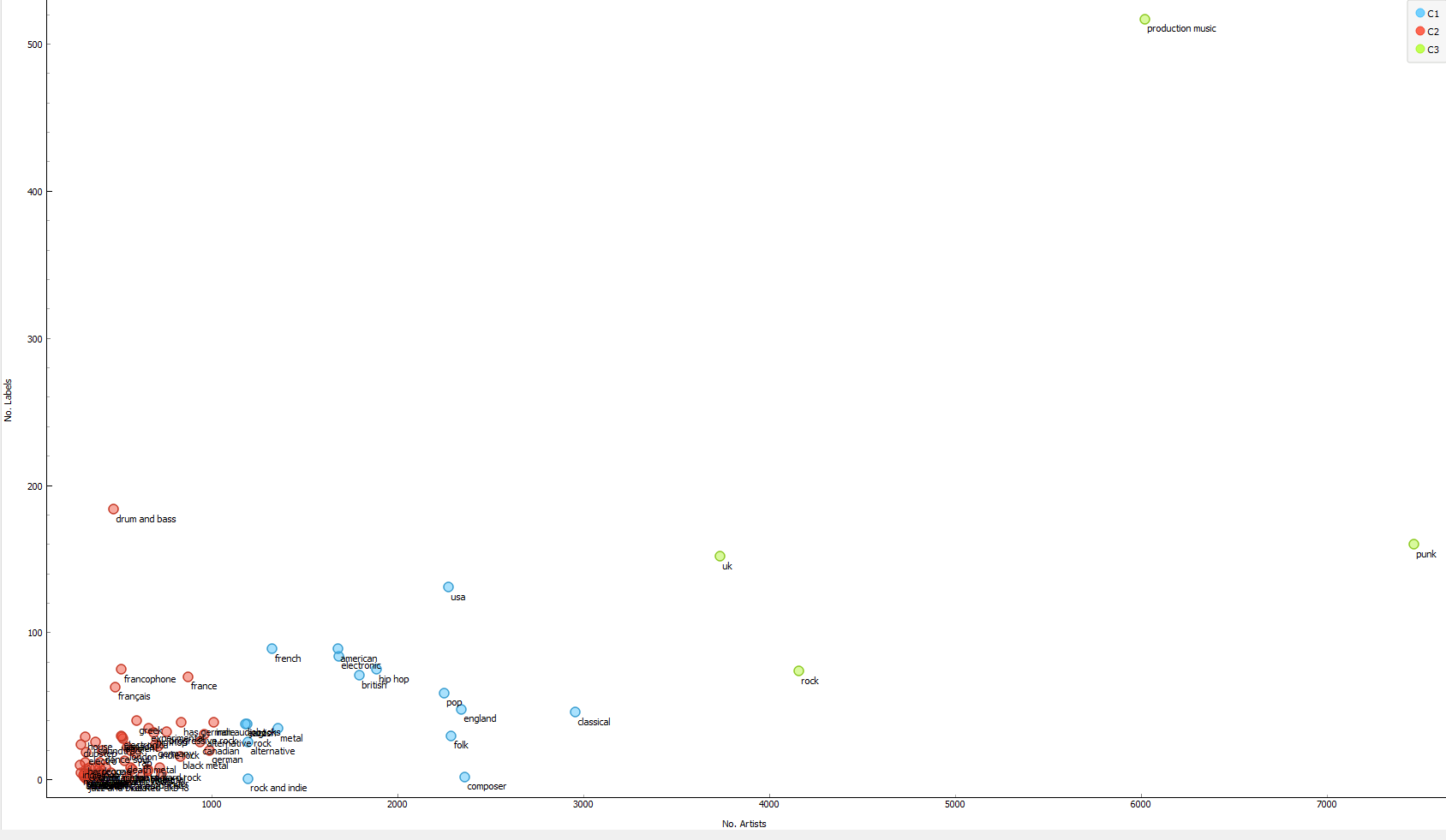
The silhouette plot validates the consistency within clusters of data. It shows a graphical representation of how well each object lies in within its cluster. The high values in C1 indicates that the objects are well matched to its own cluster and poorly matched to the other clusters.

Whereas, if you compare it to C4s low values this clearly indicates the values are slightly matched to the other clusters, this should result in using less clusters.

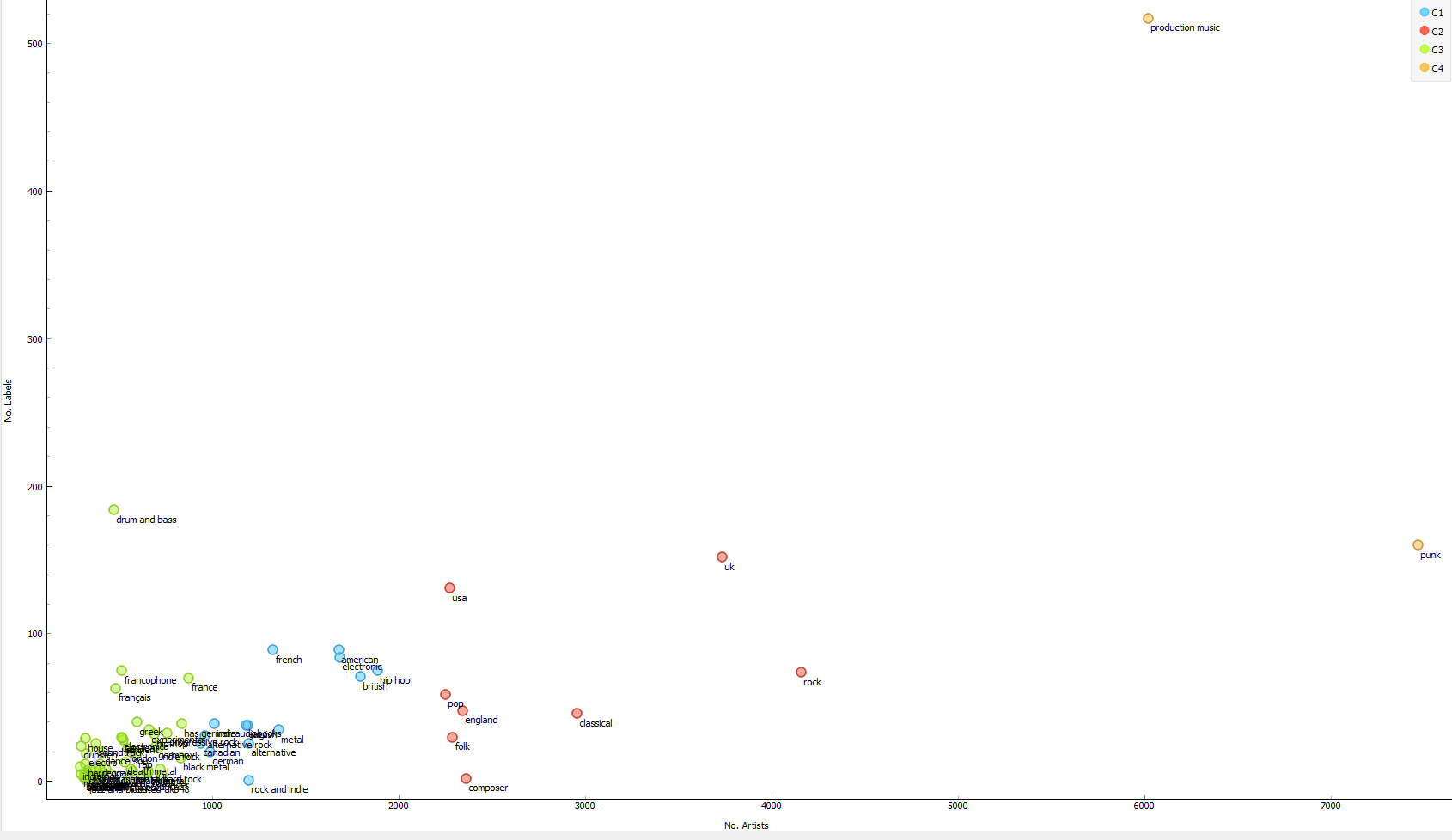
Visualising clusters in a **Scatter Plot** for **2, 3 & 4** clusters.

2 clusters

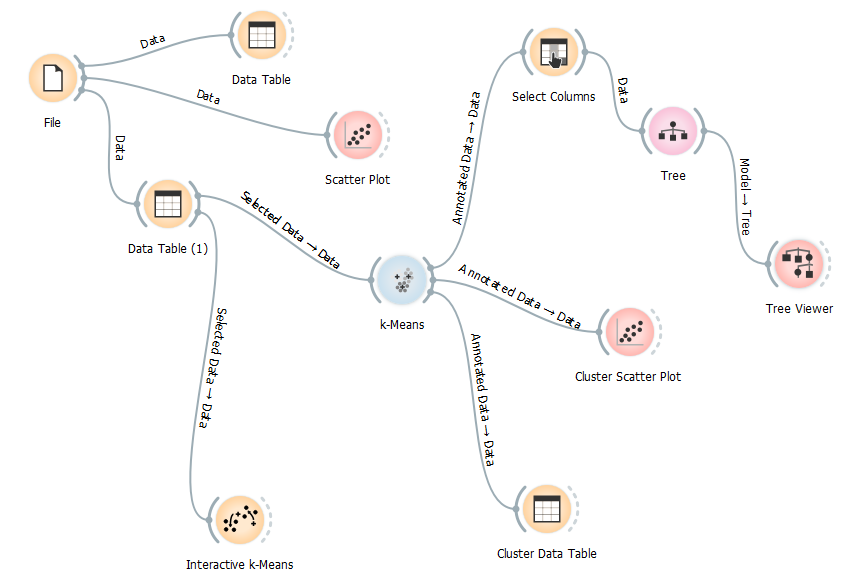
3 clusters

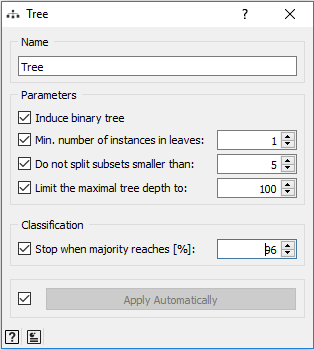
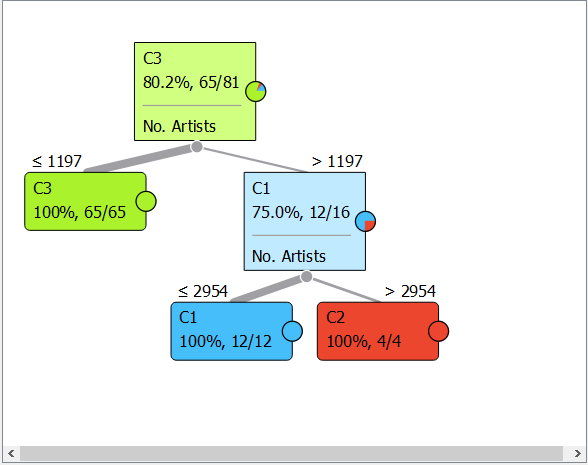


4 clusters



We have applied a classification tree and a classification tree viewer to the output of kmeans.

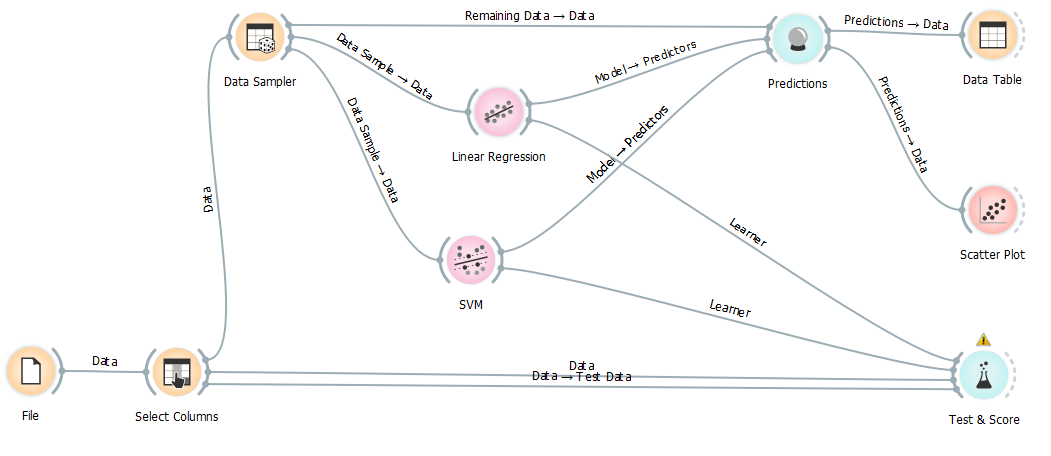


We have to change the parameters of the Tree. We set the maximum tree depth to a 100 and the min. number of instances in leaves to 1 so the tree can include clusters that may have only one data point.

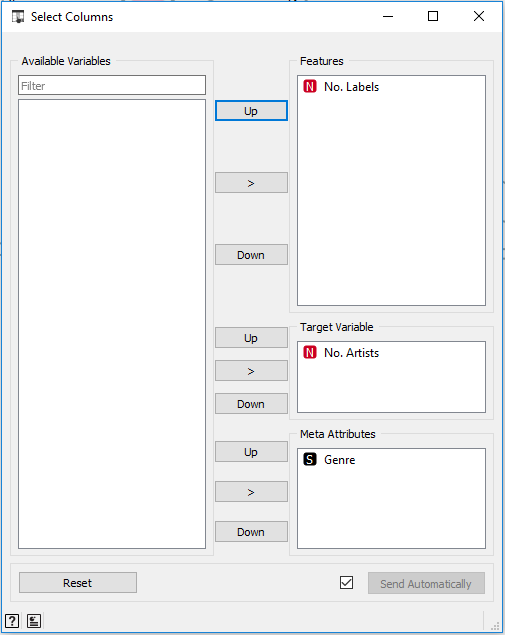
This tree viewer shows the clusters according to number of artists. It shows us C (with the least no. of people) has its data point all below 1197 artists and then it shows us 16 data points make up C1 and C2 where C1 takes 12 of that and C2 takes the remainding 4.

Predictions

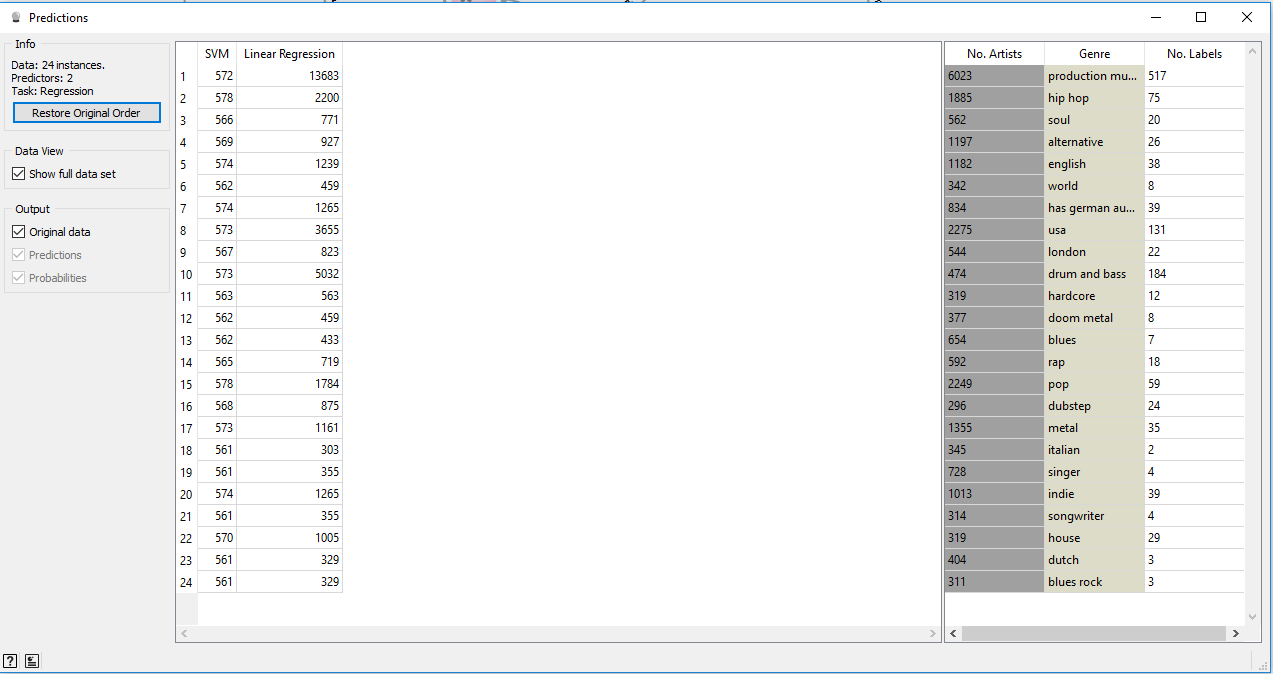
This is a diagram for predictions. We have connected a select column to choose the number of artists as target variable from the file. We use a data sampler and connected it to a linear regression and then connected the remaining data and the linear regression coefficients to a prediction and visualised the predicted values in a scatterplot and table. We also used a test&score widget to evaluate and compare with the SVM regression.



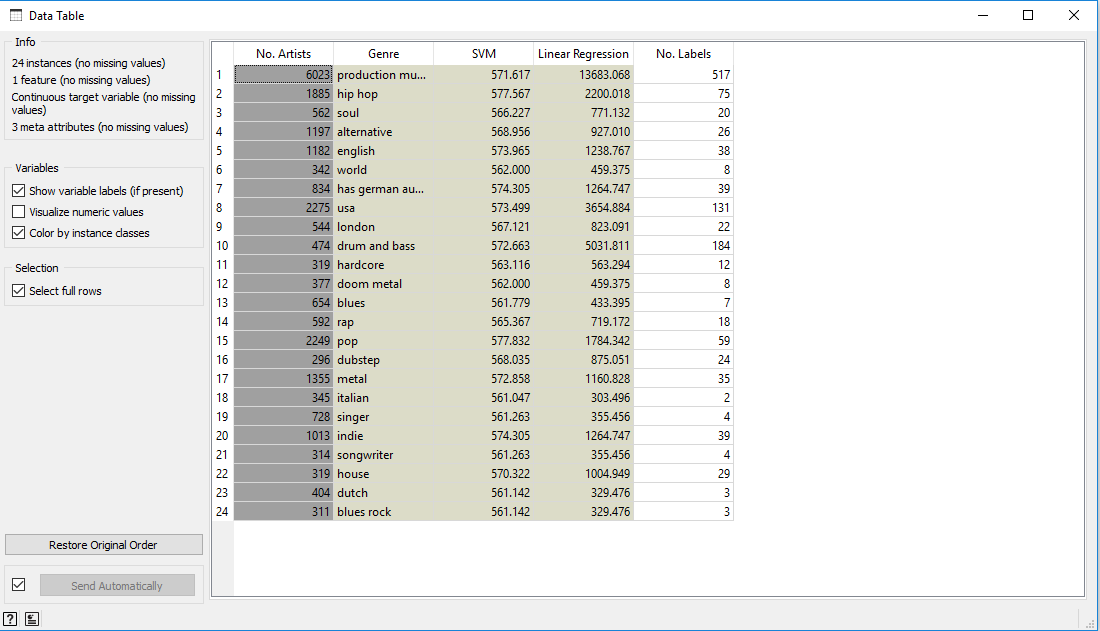
In the Select Column widget we have set the No. of Artists as our target variable and used No. of Artist as our feature. This is so we can test the No. of Artists predictions.



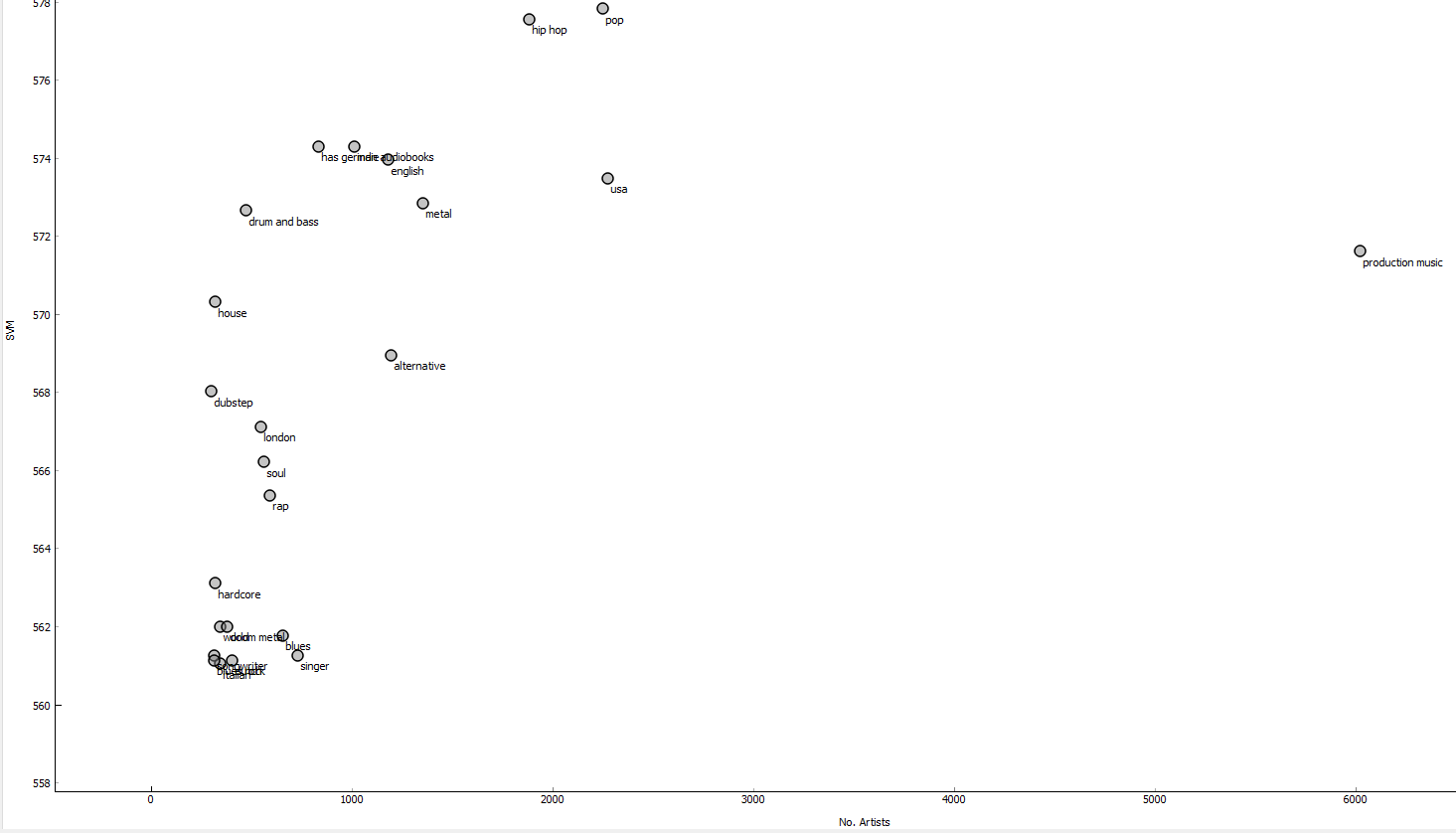
This is a screenshot of the data from the SVM and Linear Regression in the predictions widget. With the No. of artists being the target variable we can compare the SVM and Linear Regression to the original data.



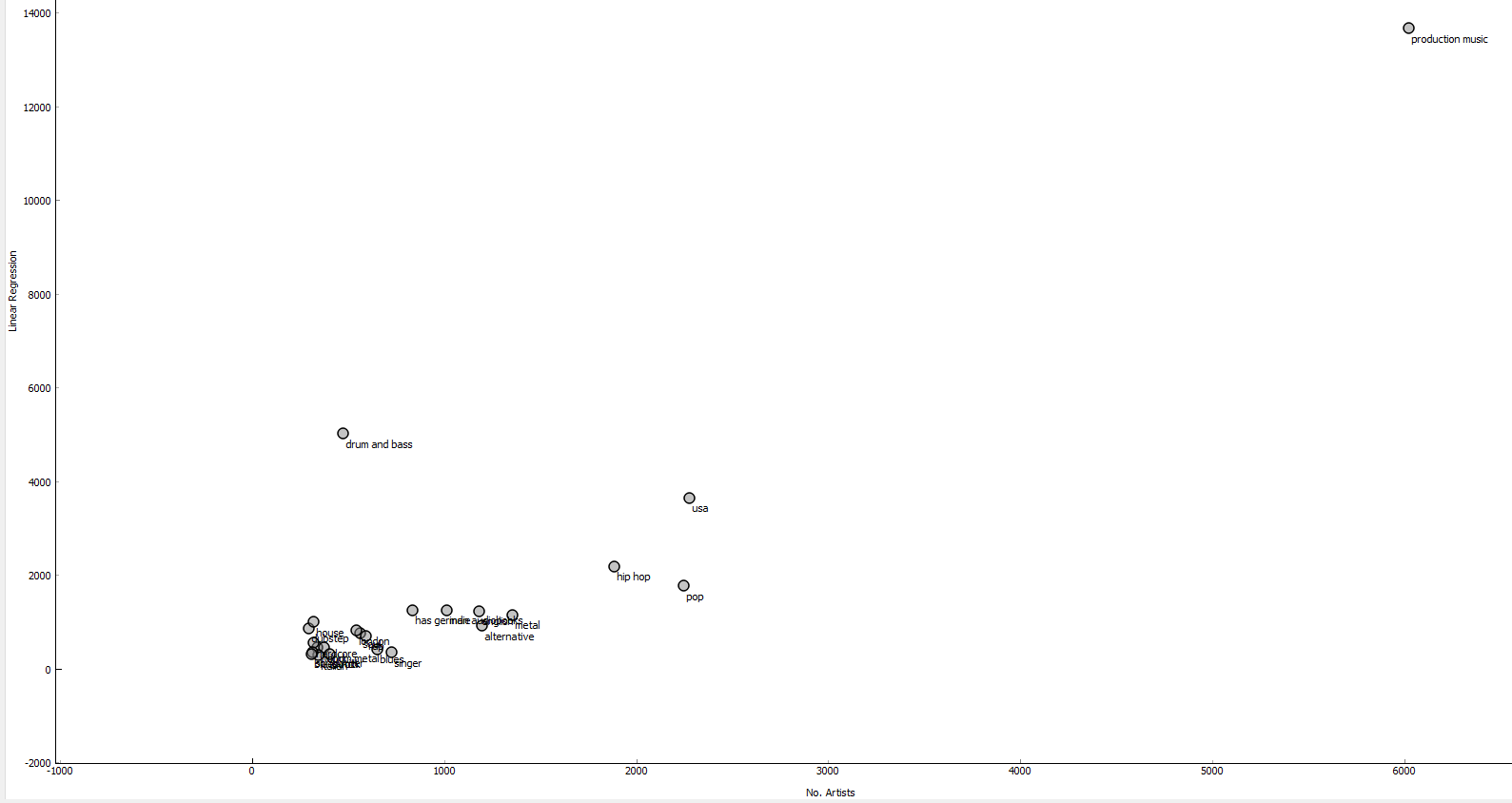
This shows the same data as above but in a data table, we get a view to compare the No. of artists real and predicted data.



This is a screenshot of the scatter plot with No. of Artists set as the X axis and the SVM is set as the Y axis.



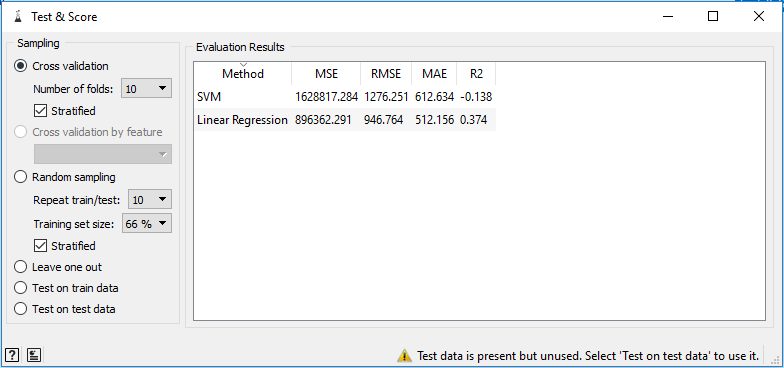
This is a screenshot of the scatter plot with No. of Artists set as the X axis and the SVM is set as the Y axis.



From the data above we acan conclude the quality of the SVM predictions seemed to be way as there were a lot of data points (Genre) that were predicted to surpass production music which was the data point with by far the most N0. Of artists according to the original data.

The quality of the Linear Regression predictions were definitely better as it is not too different comapred to the original data.

This is a screenshot of the test & score.



The (MSE) measure for the average of the squares of the errors are in very high figures which determines there was a few errors.

The (MAE) measure of how close forecasts or predictions are to eventual outcomes shows the predictions were a far way off but this still gave us good data to analyse for our predictions.

The (R2) proportion of the variance in the dependant variable that is predictable from the independent variable are very predictable numbers being very low.

# Question 10

To discuss what kind of applications would be recommended for the Musicbrainz database, firstly we understand its database is very diverse having vast amounts of information on recordings, artists, production etc.

The fact that Musicbrainz has many versions of tracks, tells us it doesn’t just store the most popular music but also a lot of underground music that doesn’t get heard from a lot of people. This leads us to believe an application for music lovers whom have the deepest passion for their types of music, can take a look at all underground music for say the Tech House genre and can explore their content from the artist has added a remix to their favourite song.

If music lovers wanted to find up and coming artist there could be an application where you can search for artist who are becoming popular rapidly, so the algorithm would look at the amount of downloads and determine if there is a rapid increase.

If music lovers really enjoyed certain tracks in a particular genre then there could be an application where the user can find these tracks and then look at the label the track was released from and from this they can save this label and then the app could create a playlist from that particular label