Used command: ./clustering_s2956586.py BINNENLAND ECONOMIE RECENSIE

Task 1:

Contingency matrix

```
[[ 119   14  609]
 [  56  225  368]
 [1615    0   48]]
```

Purity 0.802
Adjusted rand-index: 0.634

Task 2:

Cluster 1:	Cluster 2:	Cluster 3:
de	de	de
,	.	.
.	procent	van
van	van	het
het	miljard	,
een	miljoen	in
en	kortom	een
in	het	en
is	,	dat
zijn	In	is

Task 3:

Contingency matrix

```
[[ 121  609   12]
 [  57  371  221]
 [1619   44    0]]
```

Purity 0.802

Adjusted rand-index: 0.636

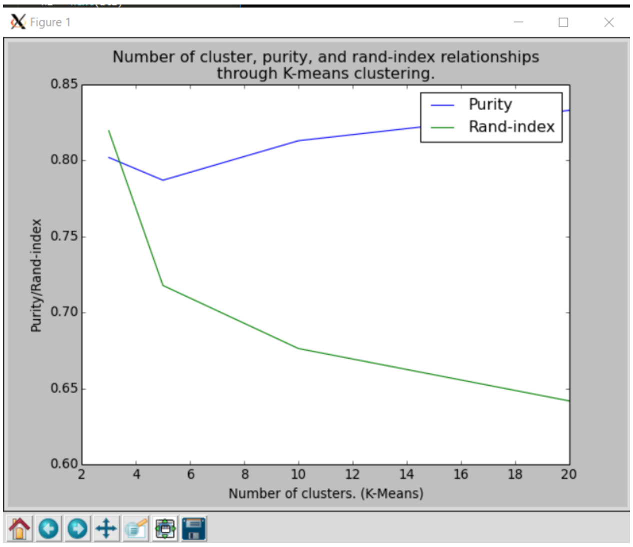
Rand-index: 0.8206301208662248

There is a difference between the adjusted rand-index and the rand-index. This is because the rand-index compares pairs throughout all clusters. This means it will count a success if a pair of elements are either in the same cluster of each partition or in an other cluster of each partition. However, the adjusted rand-index considers the chances of overlap with different clusters whereas rand-index does not. Therefor the adjusted rand-index outcome is lower than that of the rand-index. This is because the rand-index finds more possibilities for success.

Task 4:

```
sudo apt-get install python3-matplotlib
```

K-means: 3	K-means: 5	K-means: 10	K-means: 20
Purity: 0.802	Purity: 0.787	Purity: 0.813	Purity: 0.833
Rand-index:	Rand-index:	Rand-index:	Rand-index:
0.8196232419570345	0.7177300564937577	0.6762154995430005	0.6417119858702327



To see plot in the program please remove '#' in front of 'show_plot()'.

The purity goes a bit down and then up again when the number of clusters increases. The rand-index goes down when the number of clusters increases. The purity goes up because the data used is for the most part correctly placed. By increasing clusters, it is easier for the datapoints to be in the correct label. Because each label now has more clusters. But every cluster has its own purity which is used in the overall calculation. In the micro-average purity score large clusters carry more weight because the number of datapoints are used. In the macro-average purity score a more relative approach is used. By using percentage instead of the number of datapoints the score is steadier than with the micro-average purity score.

By increasing the clusters, the chance for the rand-index to find incorrect data like false positive and false negative also increases. This means that their must also be a decrease in true positives and true negatives. This leads to a lower outcome.

Task 5:

```
Contingency matrix
[[ 741    1    0]
 [ 646    0    3]
 [1663    0    0]]
Purity 0.546
Adjusted rand-index: 0.002
Rand-index:      0.4019761339239041
```

Task 6:

Outcome task 1

```
Contingency matrix
[[ 119   14  609]
 [  56  225  368]
 [1615    0   48]]
Purity 0.802
Adjusted rand-index: 0.634
```

Outcome task 5:

```
Contingency matrix
[[ 741    1    0]
 [ 646    0    3]
 [1663    0    0]]
Purity 0.546
Adjusted rand-index: 0.002
Rand-index:      0.4019761339239041
```

The difference in purity is 0.256. The K-means clustering approach has the highest purity.
The difference in adjusted rand-index is 0.8. The K-means clustering approach has the highest adjusted rand-index. The K-means clustering contingency matrix spreads more over the whole matrix whereas the hierarchical clustering has almost all its datapoints within the first column of the matrix.

Task 7:

OUTCOME TASK 1

Contingency matrix
```
[[ 119   14  609]
 [  56  225  368]
 [1615    0   48]]
```
Purity: 0.802
Adjusted rand-index: 0.634
Rand-index: 0.8191551955616675

Added ‘precompute_distances=True’. This increases the speed of the program. It does however use more memory. I also used the parameter ‘n_jobs=-1’ parameter to Kmeans. This increases the number of processors it uses for its calculation and therefor increases speed. -1 takes all processors.

Lowered, stripped from whitespace, removed punctuation, removed stop words, tokenised and stemmed the text with the Snowball Stemmer for Dutch words.

Contingency matrix
```
[[  29   47  666]
 [ 306    9  334]
 [   1 1604   58]]
```
Purity 0.843
Adjusted rand-index: 0.720
Rand-index: 0.8638896628886185

Cluster 1:
procent
miljoen
miljard
guld
jar
dollar
vorig
winst
omzet
aandel

Cluster 2:
boek
jar
the
war
wel
grot
lev
werk
eerst
verhal

Cluster 3:
jar
volgen
nederland
amsterdam
politie
minister
onz
gemeent
nieuw
gul

The top term words do differ than those of task 2. This is because the top term words in task two contained punctuations and stop words. In the top terms words of task 7 are pre-processed. In the pre-processing punctuation and pre-processing were removed allowing new words to do their entry. However, because of stemming and lowering we can see that some words were altered.

Task 8:

I would choose for clustering the data 10 times. This is because the RSS is lower than with the clustering of two for example. The lower the RSS the better the model fits to the data. However, you can see that from 10 to 12 stagnation occurs. This might lead to overfitting. That’s why 10 clusters should be the perfect number between under- and overfitting.

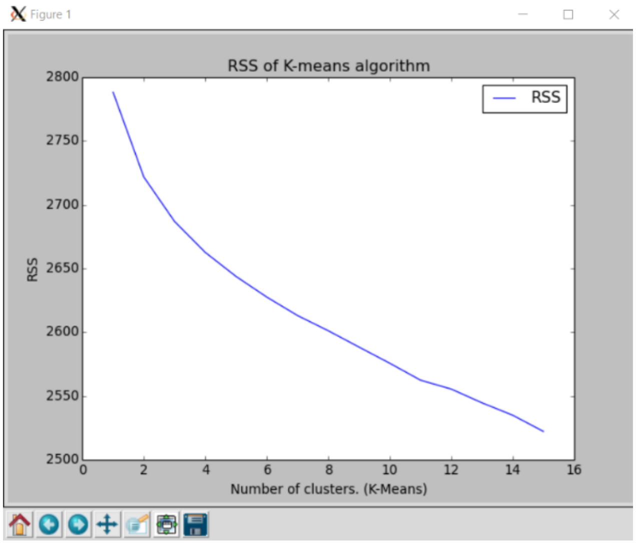


Figure 1
RSS of K-means algorithm
RSS
Number of clusters. (K-Means)
RSS