Table of Contents

[Twitter-assignment 2](#_Toc532889613)

[KKBOX-assignment 3](#_Toc532889614)

[Appendix Twitter 4](#_Toc532889615)

[Appendix KKBOX 6](#_Toc532889616)

# **Twitter-assignment**

Denmark has 9 major political parties, each with different political leaders. Lars Løkke Rasmussen is the prime minster of Denmark and comes from the party, Left. Anders Samuelssen is the foreign minister, and comes from party, Liberal Alliance. In this practice I received a dataset with the Twitter accounts of the political party leaders. My assignment was to answer the following question: “Are the major players on the political scene in Denmark who we think they are?”. To provide an explanation of this questions, I will use and interpret the scraped dataset, and explain which techniques/algorithms I used for the analysis. Hence I will present a description of the results in terms of quantitive data methods and lastly I will conclude.

I started by importing Networkx, which is a standard programming interface and graph implementation, helping to receive an overview of the Twitter-network. Thus, I could figure out the relationship between the most influential individuals in the scraped dataset. One of the algorithms I used were centrality-measures, to find the most important nodes in the network. Moreover I used the Pagerank-algorithm related to the concept of eigenvector centrality. It was used to rate webpages objectively and effectively measure the attention devoted to them.

From the analysis, I found out that we have a directed graph, with degree centrality measures of Inflow and outflow centrality. Looking at figure 1, which shows the number of edges connected to a node, we have 1.637.533 amount of nodes, and 2.014.920 number of edges. The average in-degree is 1.2305, which likewise apply to the out-degree. This means that the average amounts of edges incoming and outgoing from the vertices are 1.2305.

Furthermore, Lars Løkke has the highest degree centrality of almost 1, placing him at the top. Anders Samuelssen comes at second with 0,1789 and Morten Oestergaard at third. Figure 2 shows the Closeness-centrality, which explains the average length of the shortest path from the node to all other nodes. Similarly, Rasmussen tops the board, with Samuelssen at second place and Oestergaard at third. Figure 3 portrays the Betweenness-centrality, which is the number of times a node is present in the shortest path between other nodes. Once more, Rasmussen tops the board by a score of 0.385, nonetheless, with oestergaard and the other politicians at the bottom low. Finally figure 4, shows the Pagerank, where Rasmussen lastly tops the table again with a score of 0.19, Oestergaard comes secondly with half the amount and Samuelssen at sixth place.

What the above mentioned concludes, is that Rasmussen is the most influential individual, both in terms of power and dependence. He has the highest degree of centrality, as well with regard to betweenness and closeness. Degree-centrality shows that he holds great amount of information and referencing betweenness-centrality, obviously confirms his power, since he influence the flow of other individuals dependence. Finally, looking at pagerank, it display that he is highly ranked concerning incoming links. On the basis of this, it clearly proves that the major players on the political scene in Denmark are who we think they are. Rasmussen is the prime minster of Denmark, and Samuelsson is the foreign minister which in the most essential way, place them at the top

# **KKBOX-assignment**

The next assignment in this project was about KKBOX. KKBOX is Asia’s leading music streaming service, holding the world’s most comprehensive Asia-Pop music library with over 30 million tracks. They offer a generous, unlimited version of their service to millions of people, supported

by advertising. My assignment was to build an algorithm that predicts whether an user will churn after their subscription.

This project was mostly about cleaning datasets and machine-learning. I firstly imported the user\_logs\_v2.csv and cleaned the file. Subsequently I made a dictionary, with users, where every user was a key of 1. I used NumPy array with all data from user logs for every key, while I added them together. Secondly I used the train\_v2.csv file, which supported the binary values of 1 and 0. I added this to my dictionary, by concatenating with panda functions. In addition, I also used pandas, to import it into a CSV-file. Regarding the perception part, I used the slides from the lecture weeks, copied directly. Here I changed most of the values to floats, since they were imported as strings, objects and so on. In addition, I removed username and date from the perceptron part, and computed the accuracy, precession and F1-score.

From my results in the merged files I found out that the total amount of users were 754.551. Figure 5 points out, the total of 21 Cities encoded, with no city "2" in the data set. Figure 6 provides us with the classes of "3", "4", "7", "9", "13" listed as registration method, where registration method “7” was the most frequent . There are almost equal percentage of males labeled as (“0.0”) and females labelled as(“0.1”), as figure 7 portrays. Thus some of data is missing in gender field, which is why we filled the missing entries with (“nan”). The birth date is mostly 27-53 as seen in figure 8. Proceeding with the algorithms, I also checked theaccuracy**,** which is a good intuitive performance measure. From figure 9 My result was an average accuracy of 0.87. Furthermore I also looked at precision, where I got 0.841, which is the ratio of correctly predicted positive observations. Additionally I looked at the sensitivity, where I got a value of 0.908. lastly I checked for the F1-score, which is the weighted average of precision and recall, which In our case, was a score 0.8678.

To conclude one may think that, if we have high accuracy then our model is perfect. In some degree, that is true, but not when we have a dataset with missing values. Therefore, we look at other parameters like precision, sensitivity and F1-score. Precision gives use the following answer: Of all users who will churn after their subscription, how many will? A High precision of 0.841 indicates low false positive rates as figure 10 shows. This means, in a similar way, that there is a great chance of users more likely to decide churning after their subscription. To back this argument up, our recall score of 0.908, shows that we label many users, which likewise is good for this model. In addition our recall score, is also above 0.5, that similarly is sufficient. Lastly the F1-score takes both false positives and false negatives into account. F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Our F1-score landed on 0.8678, which overall makes it clear, that we should trust in the accuracy of this predictor.

# **Appendix Twitter**

Figure 1: Degree-centrality

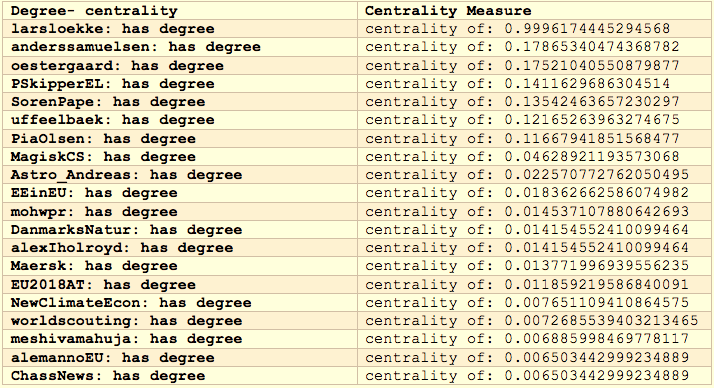


Figure 2: Closeness-centrality

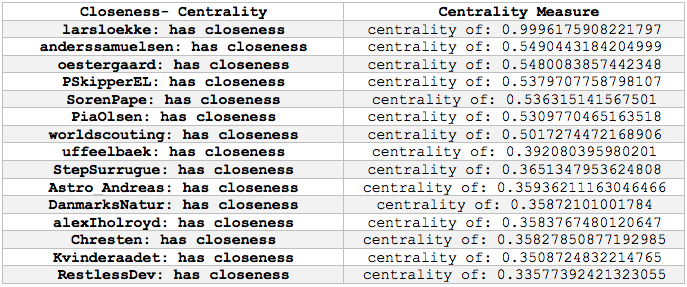


Figure 3: Betweeness-centrality

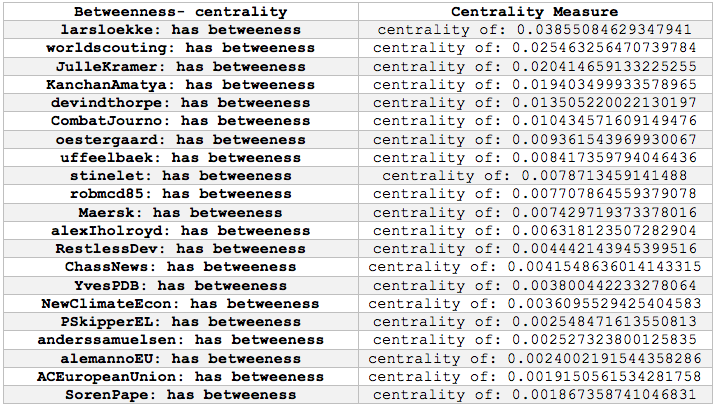
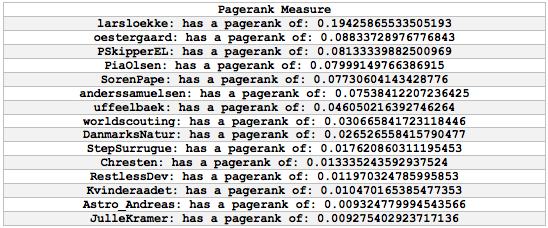


Figure 4: Pagerank Measure



# **Appendix KKBOX**

Figure 5 – Frequence of City count

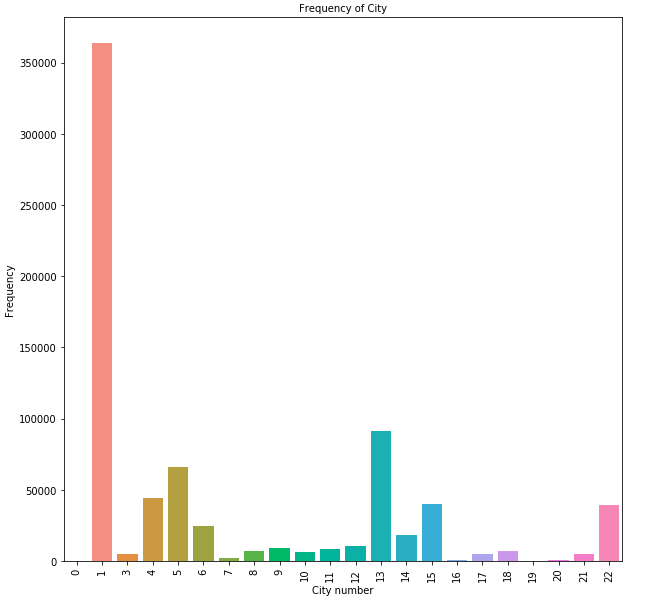


Figure 6 – Frequence of Registered via method

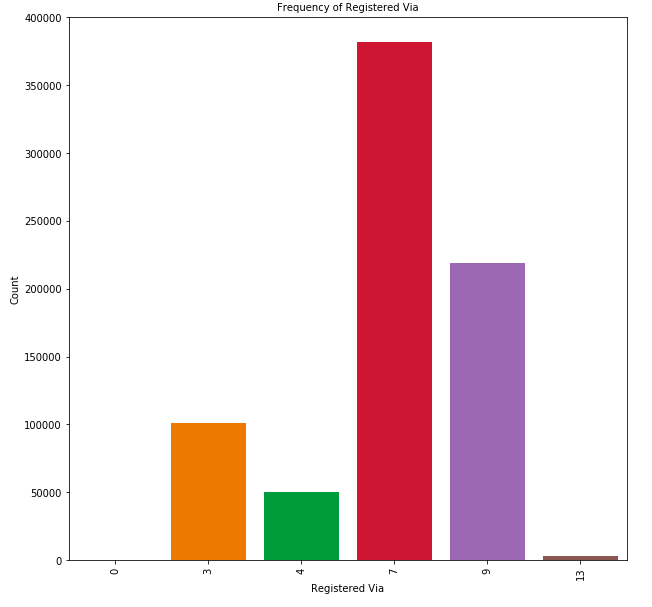


Figure 7 – Frequency of gender

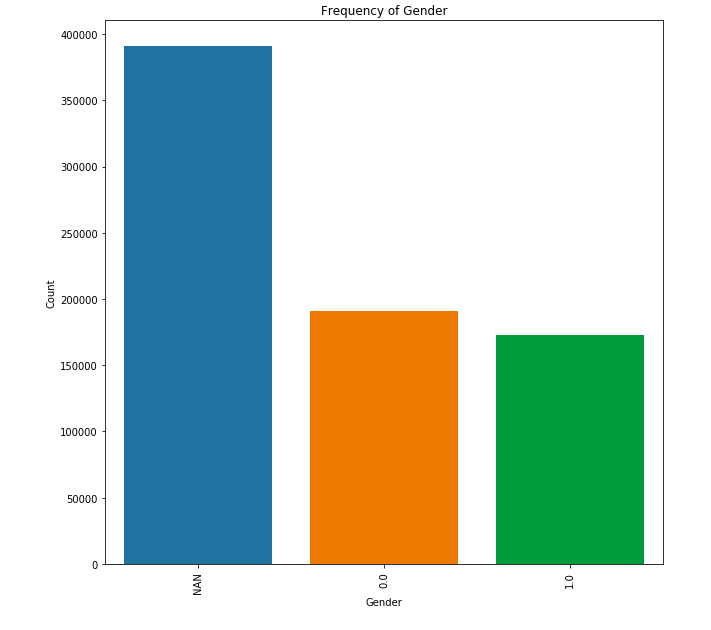


Figure 8 – Frequency of BD count

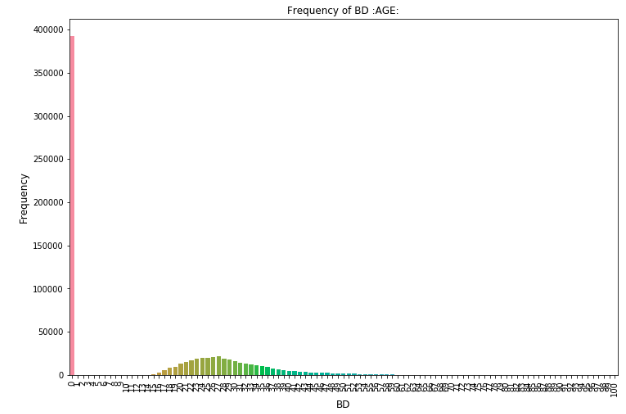


Figure 9 – Frequency of BD count

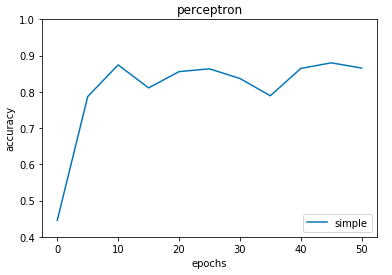


Figure 10 – TP,FN,FP,TN

