FACULTY OF ECONOMICS AND
BUSINESS ADMINISTRATION

Social Media and Web Analytics: Assignment

Technical report

Course: Social Media and Web Analytics

Teacher: Prof. Dr. Matthias Bogaert

Students: Team 14

Viktor Vandenbulcke	01803499	viktor.vandenbulcke@ugent.be
Wouter Dewitte	01806106	wodwitte.dewitte@ugent.be
Bert Jonckheere	01706446	bert.jonckheere@ugent.be
Konstantin Lazarov	01700453	konstantin.lazarov@ugent.be
Guillaume Lambert	01710087	guillaume.lambert@ugent.be
Artur Tyvaert	01706871	artur.tyvaert@ugent.be

Academic year: 2021–2022

GHENT
UNIVERSITY

Contents

1 Targeting

1.1 Introduction

The goal of this part is to find out on how many users we should build our adjacency matrix, by limiting the number of followers that we extract, while achieving a similar performance. We solved this assignment in four parts. First, we extracted all the followers and followers of followers of the profile. Next, we computed the ground truth based on the adjacency matrix. In the third part, we looked at the correlation between the rank in the ground truth and the rank of different subsets with an increasing number of data. Finally, we determined an optimal cut off point to indicate on how many users you should build your adjacency matrix.

1.2 Extracting the followers and their followers

We chose to analyze the twitter network of team member Viktor Vandenbulcke (screen name: *viktor_ vdb2*). To analyze his network, we scraped his 144 followers. Next, we got all the followers of Viktor's followers. This data was stored in the `R`-file `followers_viktor_vdb2.RData`. However, for some of his followers, we were unable to scrape their data. We deleted to 3 followers as we were unable to scrape their full network due to their number of followers being above 4998 (twitter API scraping limits). Besides, we also removed 21 followers due to privacy settings. This leaves us a document by term matrix with 120 followers (Viktor included) and 13,720 followers of followers. This full network is represented in **appendix 1**.

1.3 Computing the ground truth

The goal of our analysis is to determine on how many users we should build our adjacency matrix. Thus, this means we are interested in the first degree followers, which are present in the rows of our document by term matrix. We decided we want to create the adjacency matrix in 3 different ways to see the effect of different approaches. Once by using the dot product, once by using the Euclidean distance and once by using the Cosine distance. As we are interested in the first degree followers, these should be present in our adjacency matrix. When we calculate the adjacency matrix with the dot product, we determine A as follows: $A = X \cdot X^T$ with X the document by term matrix. This results in an adjacency matrix with dimension $[120 \times 120]$, where the rows and columns contain the sources of interest (the followers). After deleting Viktor from the matrix, we can rank all our followers based on their degree in this network and consider it the ground truth. The highest degree has the most connections, hence the highest rank. An overview of the ground truth for dot product can be found in **appendix 2** and **3**. Second, we used the euclidean distance to calculate the adjacency matrix. This value represents the distance between two users in the network and can be computed with the distance function in `R`. The lower the distance, the closer the people are to each other in the network. With these distances, we created a new ground truth, ranking the lower distances higher as they are closer to the other users in the network. The ground truth top 10 and its evolution for this method can be found in **appendix 4** and **5**. Finally, we also used the cosine similarity to compute the adjacency and the ground truth. The cosine similarity indicates the similarity between users in a network. With these similarities, we created a new ground truth, ranking the followers based on similarities from high to low. The ground truth top 10 and its evolution for this method can be found in **appendix 6** and **7**.

1.4 Ground truth and rank in different subsets

Our goal is to find out on how many users we should build our adjacency matrix. Therefore, we made a loop in which we calculate the adjacency matrix with an increasing number of columns and calculate the rank of the users based on these different matrices with increasing number of information. These ranks are than compared to the original ground truth ranks with the spearman correlation coefficient. We computed these loops in two different ways. In the first method, the number of columns was increased from the second column up until the 5000th column. For each adjacency matrix method, the correlation and the network size was stored and this is plotted in **appendix 8**, **9**, and **10**. However, there is a major issue with this method. First, only the first 5000 columns could be chosen. This could create a huge bias. To tackle this problem, we also performed a second method. In this method, we randomly draw from 2 up until 5000 columns out of our entire matrix. In order to improve the robustness of our method, we performed this loop 5 times and took the average for each network size. This method gives much more reliable results and we used this data to perform the rest of our analysis. A graph with the correlation per network size for the 2nd method can be found in **appendix 11**, **12**, and **13** (one for each method).

1.5 Determining a cut off point for the adjacency matrix

In order to indicate on how many users you should build your adjacency matrix, we looked at the learning rate of the average correlation for an increasing number of network size. To smooth our average correlation, we used the smooth spline method. An example of the smoothed correlation graph can be found in **appendix 14**. For each way of calculating the adjacency matrix, we determined an optimal cut off point for a 0.001 and a 0.0001 learning rate (see **appendix 15**). For a learning rate of 0.001 we can see that the euclidean distance method, while for 0.0001 the dot product is superior. Further, we can compare the different methods based on the graphs with the ground truth approximation in **appendix 16**, **17**, and **18**. In these plots, we see that the average correlation for dot product and cosine similarity is significantly higher than for the euclidean distance. Besides, the cosine similarity is computationally expensive. In general, the dot product method seems the most suited for this analysis. To minimize the number of followers to extract, we refer to **appendix 15** as it depends on your learning rate and the application.

2 Performance

2.1 Introduction

In this part of the assignment we take a look a the *Starbucks Corporation* tweets and if they have is a relation with the *Starbucks Corporation* stock (SBUX). More concrete, we used the hourly change of the stock price of *Starbucks Corporation* as dependent variable and the Starbucks tweets of the previous hour as the independent variable.

2.2 Part 1: Descriptive

2.2.1 Wordclouds

We began our descriptive part by making a wordcloud. The tweets were converted to text and cleaned. We decided to only include terms that have a frequency higher than 30. By looking at the cloud in **appendix 19**, we notice that the most important topics are either very actual topics, the war in Ukraine, or very relevant topics when it comes to *Starbucks Corporation*, like the coffee.

2.2.2 Topic modelling

We applied the latent Dirichlet allocation to discover common topics in our collection of tweets. In order to do this we created a corpus and applied the following prepossessing: remove punctuation's, remove numbers, remove white-spaces and remove stop-words after which we created a document by term matrix and removed all terms who have a sparsity higher than 95%. In order to find the optimal number of topics we used a grid search with initial values for values of the LDA parameters (alpha and beta) set to [0.2, 0.4, 0.6, 0.8]. We used the Umass coherence metric to find the number of topics such that co-occurring terms will be penalised. After deploying LDA on the full dataset we obtained a Umass score of -286.3039 for $\alpha = 0.4$ and $\beta = 0.6$. However, we wanted to used these topics as features in our predictive model. For this reason we chose only 2 topics and selected the 5 most important terms (see below). These topics were added as features by counting how often these words occurred in a tweet. We can clearly see that the first topics regards the coffee experience, while the second topic is linked to the war in Ukraine. This topic table can be found in the **appendix 21**.

2.2.3 Word embeddings

We did create word embeddings both using `word2vec` and `GloVe`. Interesting things can be noted from these word embeddings. For instance, the cosine similarity can be used to measure the correlation between words occurring in the same tweets. As an example we looked at the words most with the highest cosine similarity with "russia": `cocacola` (0.8065444), `business` (0.8042734), `mcdonalds` (0.7958537), `operations` (0.7529080), `companies` (0.7246531), `pepsi` (0.7050898), and `ukraine` (0.6946635). These all prove that Russia was primarily tweeted about in combination with "*#Starbucks*" regarding the situation of certain companies only boycotting Russia after heavy opposition.

2.2.4 Word networks

To have a better overview of some network characteristics of our subject we created a word network. To make our code more fluent, we only used a subset of all the tweets available. More precisely, we only used 1000 tweets. We cleaned our tweets in the same way we did with our wordcloud. While looking at our first networks, we saw that a bot was present. The bot tweeted the same tweets about *Poshmark* and these tweets created a cluster of their own in the network. We eliminated this bot, because we consider this as noise. The word network can be seen in **appendix 22**. When studying this figure, we can see a clear green cluster about the current troubles happening between Ukraine and Russia and the companies quitting their actions in Russia. Next to that, we see the red cluster. This cluster is more focused on the company *Starbucks Corporation.* The other cluster is less clear.

2.2.5 Sentiment Analysis

One of the features in our final model is the aggregated sentiment of all tweets for a specific time period. This will give us an insight into the general opinion of the tweets. We applied 2 lexicon based approaches.The first uses a dictionary to get the valence for each word in the tweet, after which the average is taken of these scores in order to get a sentiment score for the tweet (these values range from -4 to 4) The second approach uses the `sentimentr` package. We first do some basic preprocessing: replace emoticons and replace word elongations after which we get a polarity score for the tweet. Additionally we also obtained a polarity score by standardizing the mean average. These sentiment values range from -2 (negative) to 1 (positive).

2.3 Part 2: Predictive

2.3.1 Data collection

We had two major data parts that we needed to collect: financial data on SBUX and tweets with *Starbucks.* The financial data was used from an external source. We scraped tweets from 02/03/2022 until 24/03/2021. We only scraped tweets that were in English because this seemed the most interesting for our analysis. We could only collect about 2000 tweets each day, which limited us in our analysis. We faced the issue of having very poor tweets with a lot of spelling mistakes and automated bots, which we removed. We tried to predict the movement of the stock (upwards or downwards) 1 hour ahead. We faced the issue that the stock market is only open from 9:30 AM till 4:30 PM. We solved this problem by aggregating the tweets that occurred outside of this period to match the next period in which the stock market was open. We illustrate this with an example: all tweets that were tweeted after 4:30 PM till 9:30 AM the following day are averaged and used to predict the first hour of the following day. During the weekend all data is aggregated to predict the first hour on Monday. Additionally, we also had to take the time difference effect into account: the scraped tweets are in UTC time, while the financial data is quoted on the New York stock exchange. In order to match time zones we added 4 hours to the financial data.

2.3.2 Data preprocessing

Before linking the financial data and the Twitter data, we first cleaned the data. Multiple tweets were posted during the same hour, so we had to group them by hour. Before performing this, we off course have to split the data into test and train set. After that, we scaled the data using the mean and standard deviation of the training set. Then we aggregated the tweets by time and we used the mean to calculate the sentiment values. Finally, we merged the tweet data and the financial data to create our training and test basetable.

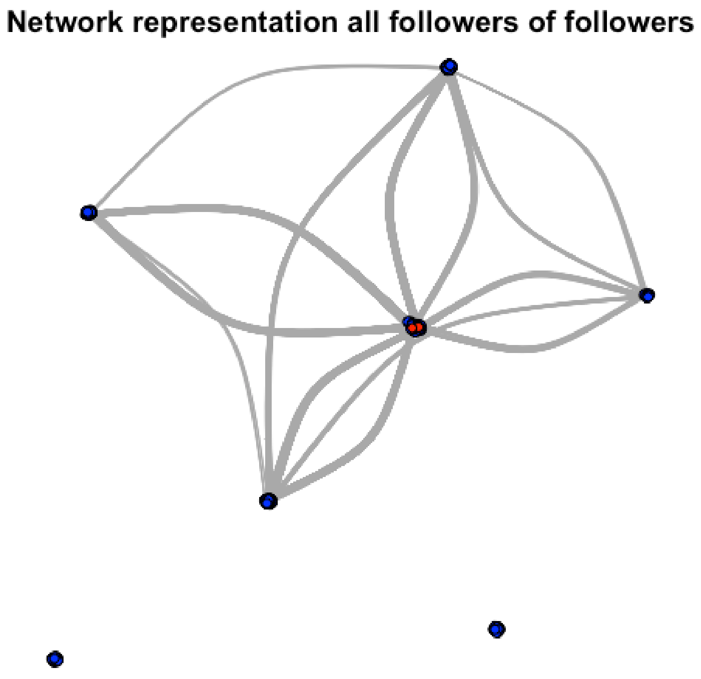
2.3.3 Modeling

The first model that we created was Logistic Regression. We built a model by only using the Twitter data and we also created a model by using the whole dataset. This way we could investigate the Twitter data separately. Next to that, we also took a look at the variable importance. This can be seen in **appendix 26** and **27**. this table shows the order of explainability of the variables. But by looking at the summary of both models in **appendix 28** we see that only the *rsi* feature is significant. So, our features do not have any explanatory power over the changes of the Starbucks stock. Looking at the bigger picture, this was to be expected: stock volatility is not only driven by individuals tweeting positive or negative comments of the company. We also performed RandomForest and LightGBM but this lead to the same conclusion.Furthermore, although we would have obtained far better results with more data, a simple neural network was trained on the data extracted from the tweets. We used two hidden layers, the first having 8 nodes, the second 4. Both layers used the Leaky ReLU activation function and had their weight limited by a maxnorm constraint.

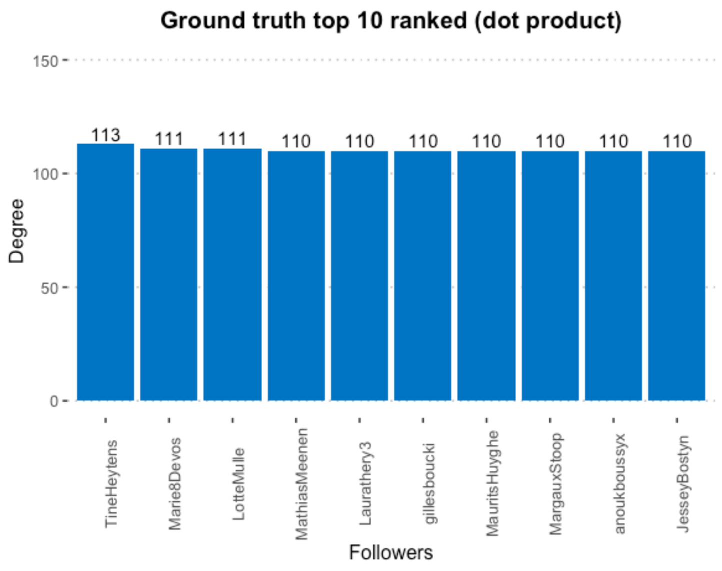
2.3.4 Evaluation and Deployment

For the evaluation we looked at the Logistic Regression model. We used AUC and accuracy of the confusion matrix as evaluation metrics. The confusion matrix of the Twitter model can be found in **appendix 30**. The AUC of the Twitter model is 0.606 and the AUC of the full model is 0.5, which means that the model is not better than random. The ROC-curve with corresponding AUC of the Twitter model can be found in the **appendix 31**. The learning curves of the neural network can be seen in **appendix 32**. The ROC curve (with a corresponding AUC equal to 0.692) can be seen in **appendix 33**.

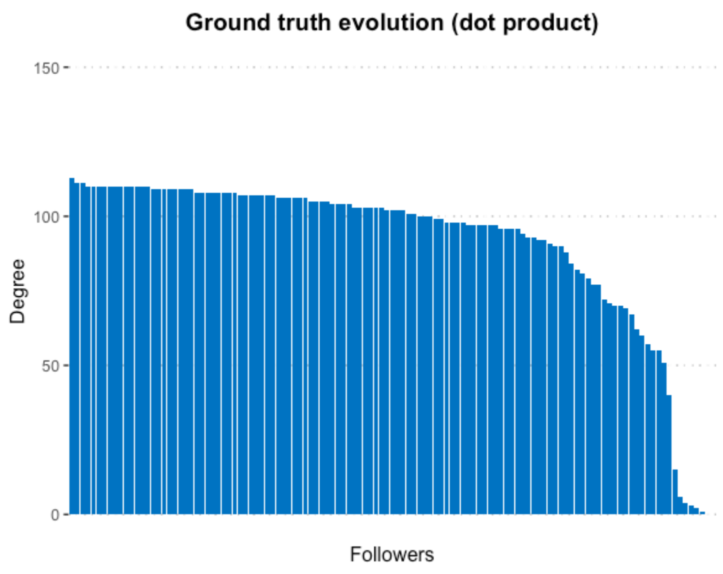
Appendices



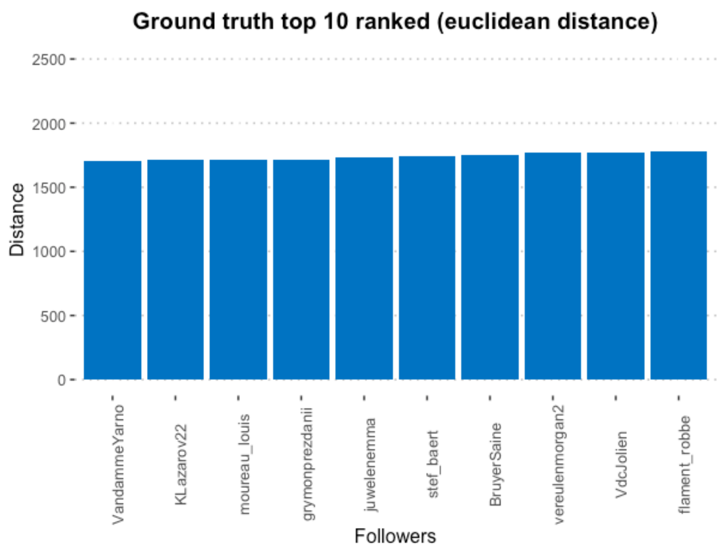
Appendix 1: Network representation



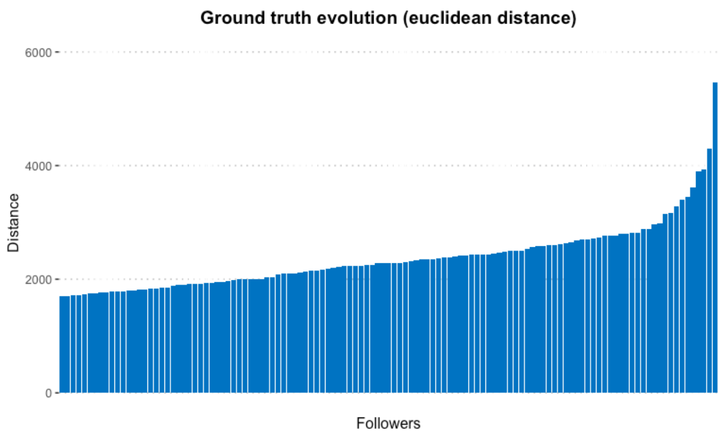
Appendix 2: Ground truth top 10 (dot product)



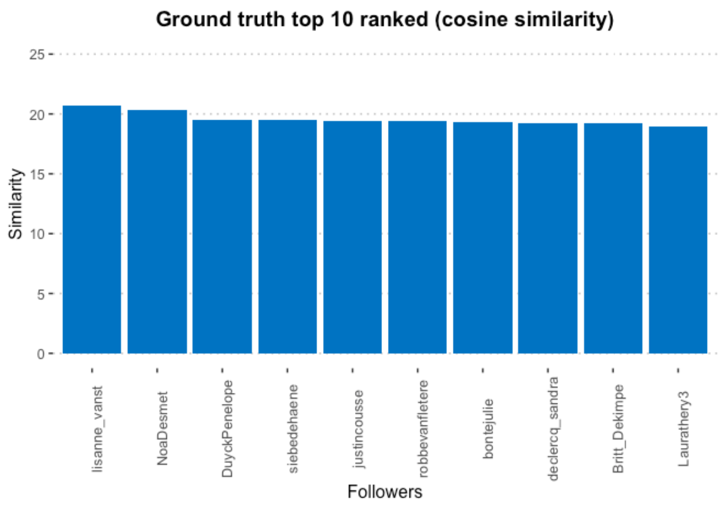
Appendix 3: Ground truth evolution (dot product)



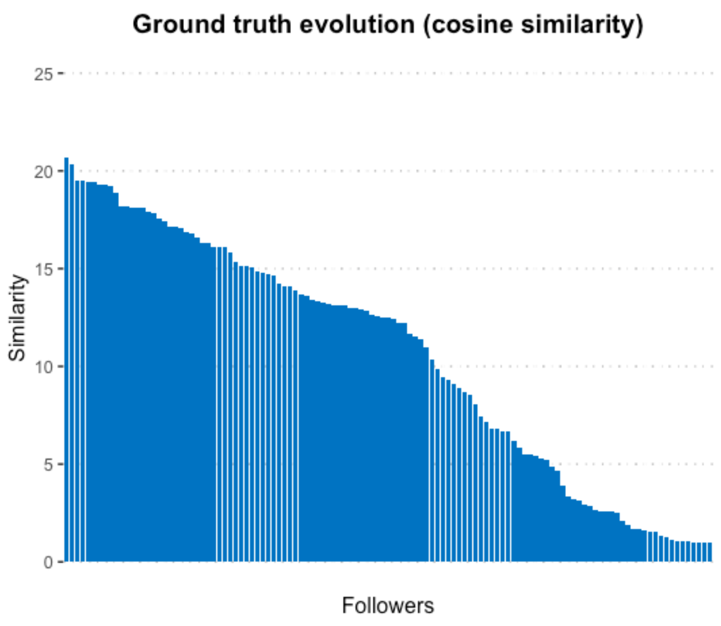
Appendix 4: Ground truth top 10 (euclidean distance))



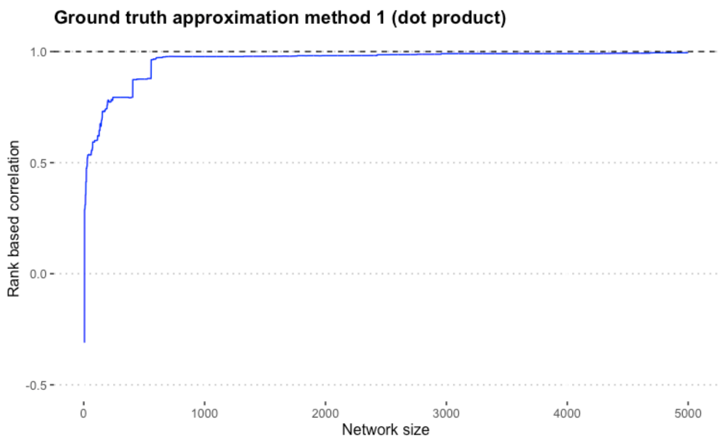
Appendix 5: Ground truth evolution (euclidean distance)



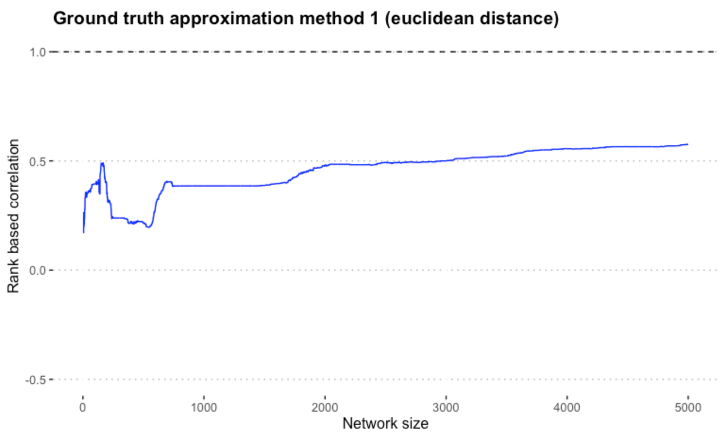
Appendix 6: Ground truth top 10 (cosine similarity)



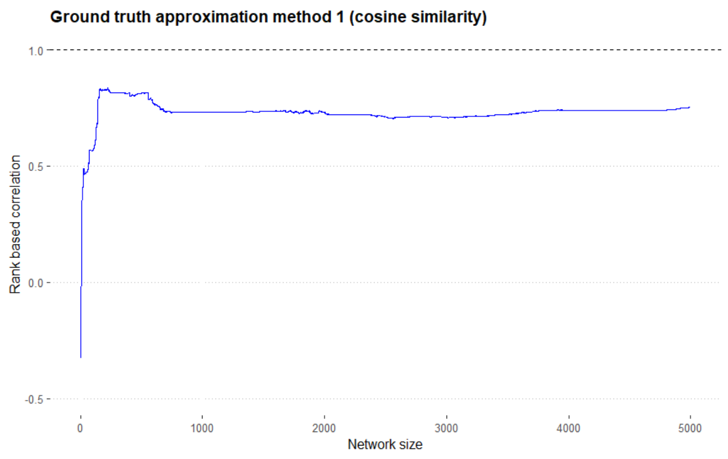
Appendix 7: Ground truth evolution (cosine similarity)



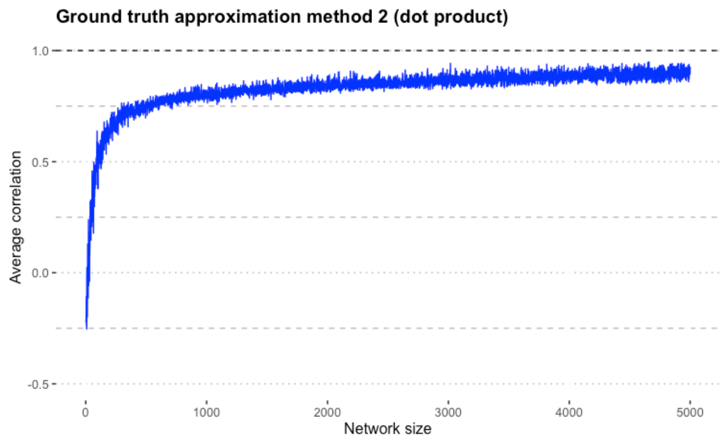
Appendix 8: Ground truth approximation method 1 (dot product)



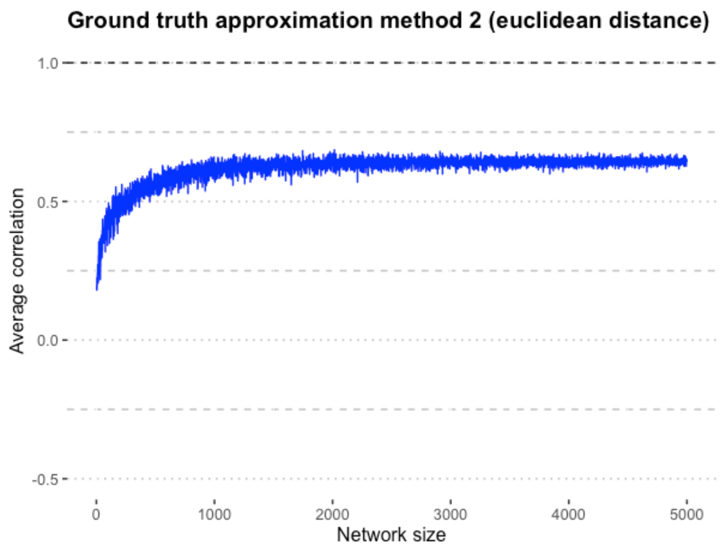
Appendix 9: Ground truth approximation method 1 (euclidean distance)



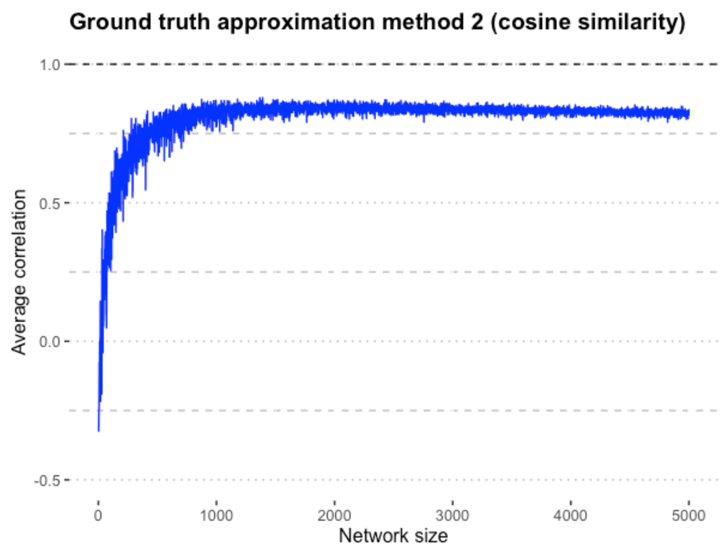
Appendix 10: Ground truth approximation method 1 (cosine similarity)



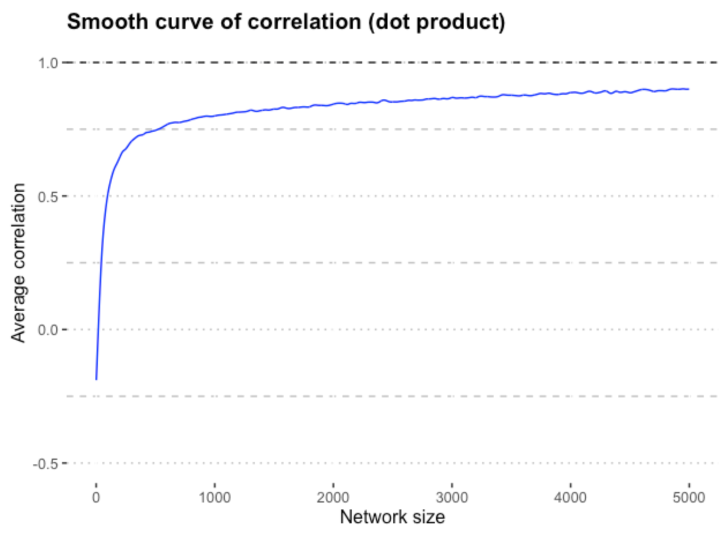
Appendix 11: Ground truth approximation method 2 (dot product)



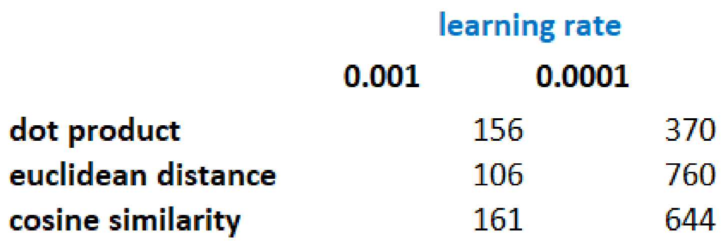
Appendix 12: Ground truth approximation method 2 (euclidean distance)



Appendix 13: Ground truth approximation method 2 (cosine similarity)

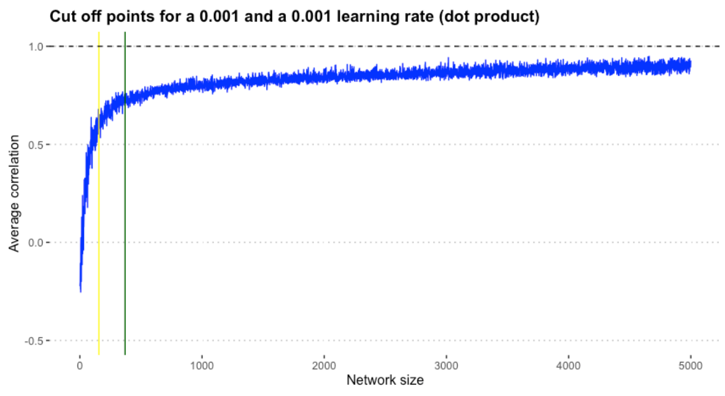


Appendix 14: Smooth curve of correlation (dot product)

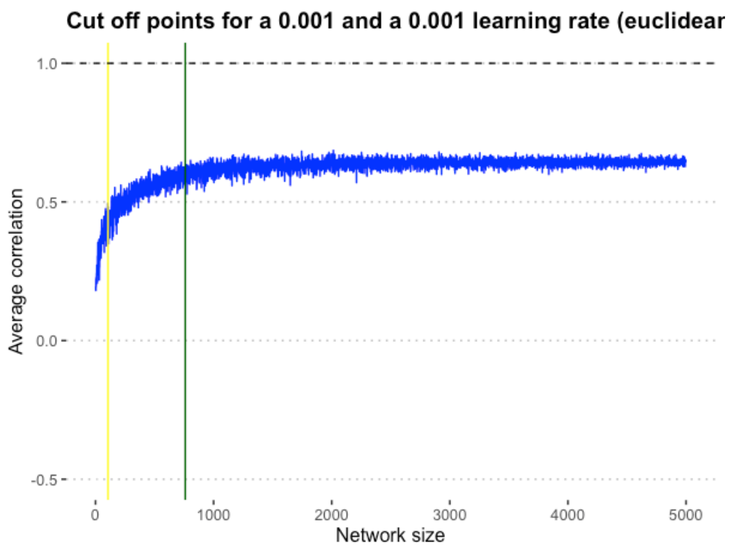


	learning rate	
	0.001	0.0001
dot product	156	370
euclidean distance	106	760
cosine similarity	161	644

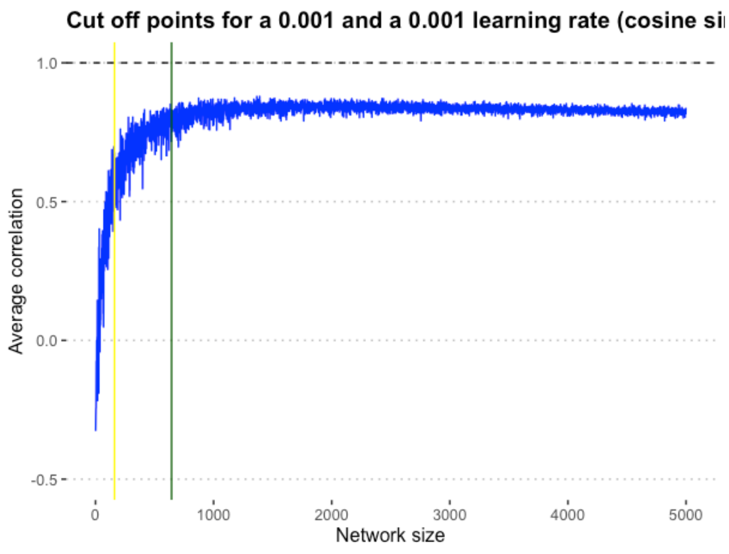
Appendix 15: Cut off points



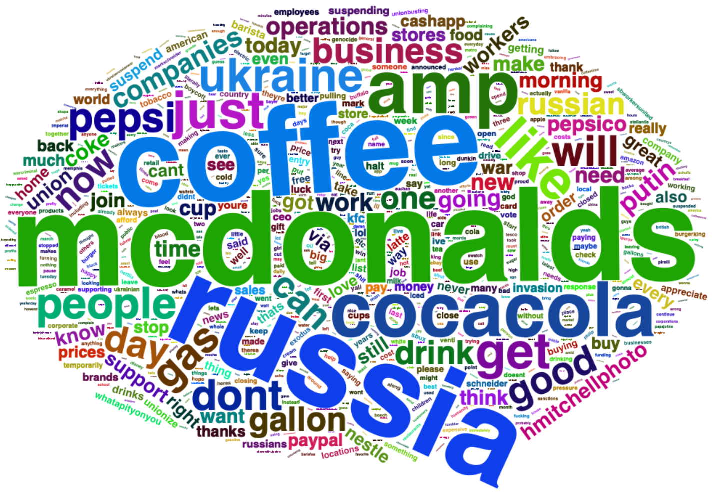
Appendix 16: Cut off points for a 0.001 and 0.001 learning rate (dot product)



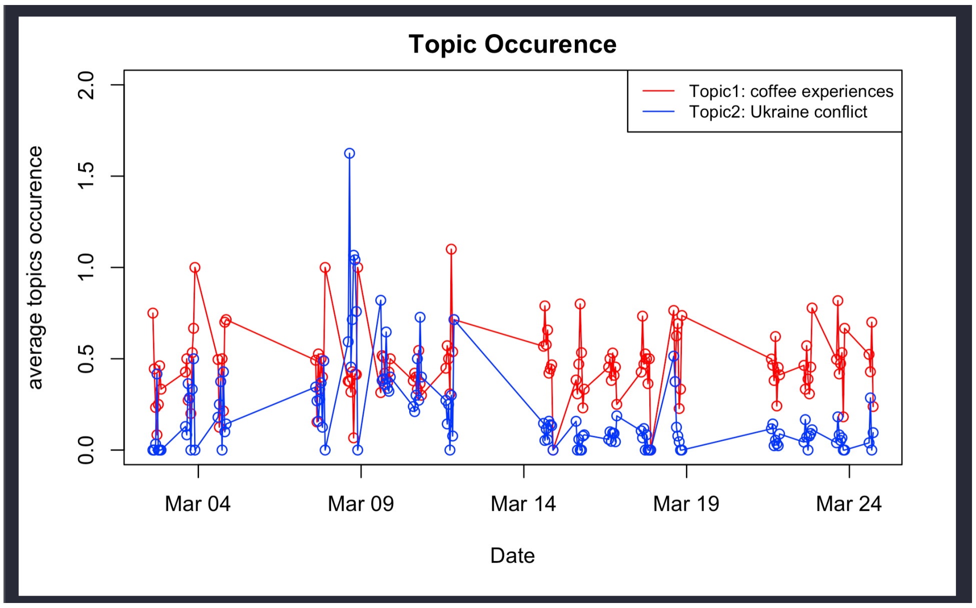
Appendix 17: Cut off points for a 0.001 and 0.001 learning rate (euclidean distance)



Appendix 18: Cut off points for a 0.001 and 0.001 learning rate (cosine similarity)



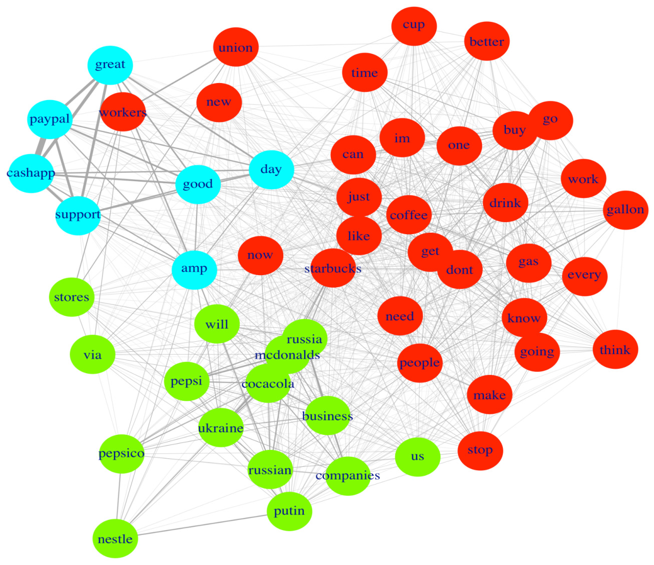
Appendix 19: Wordcloud



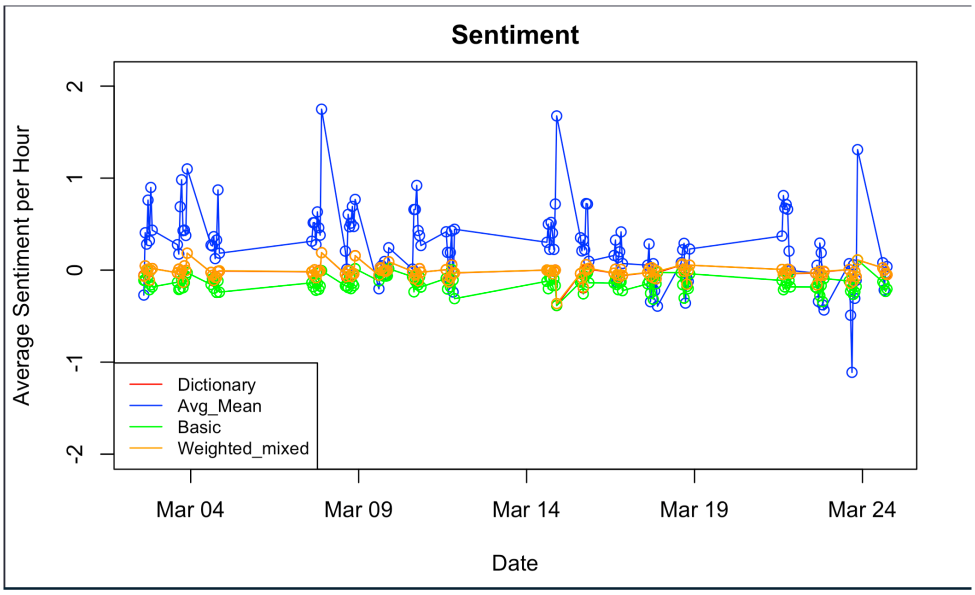
Appendix 20: Topic counts aggregated per day

Topic 1	Topic 2
coffee	russia
amp	mcdonalds
good	cocacola
like	get
just	ukraine

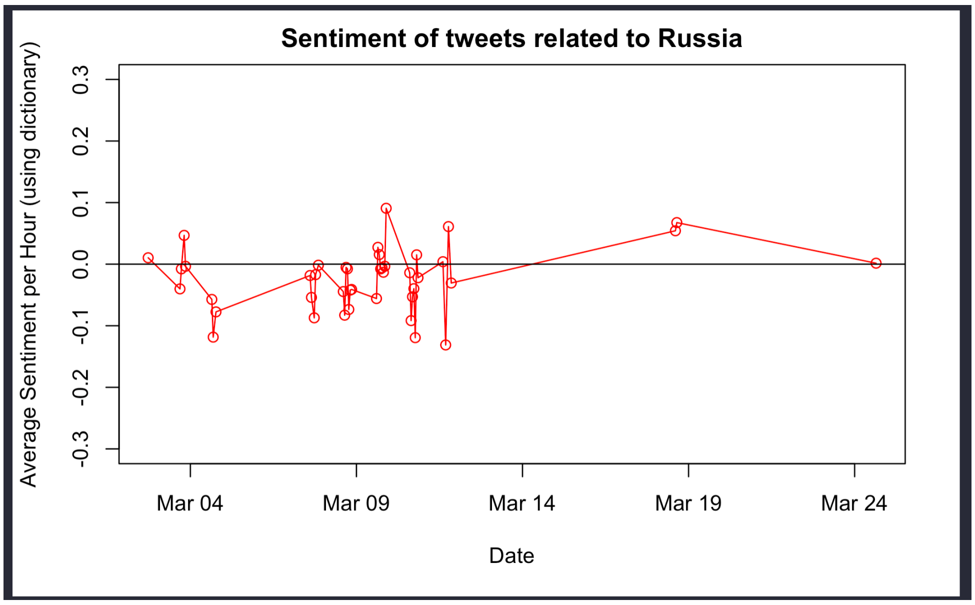
Appendix 21: Topic Table



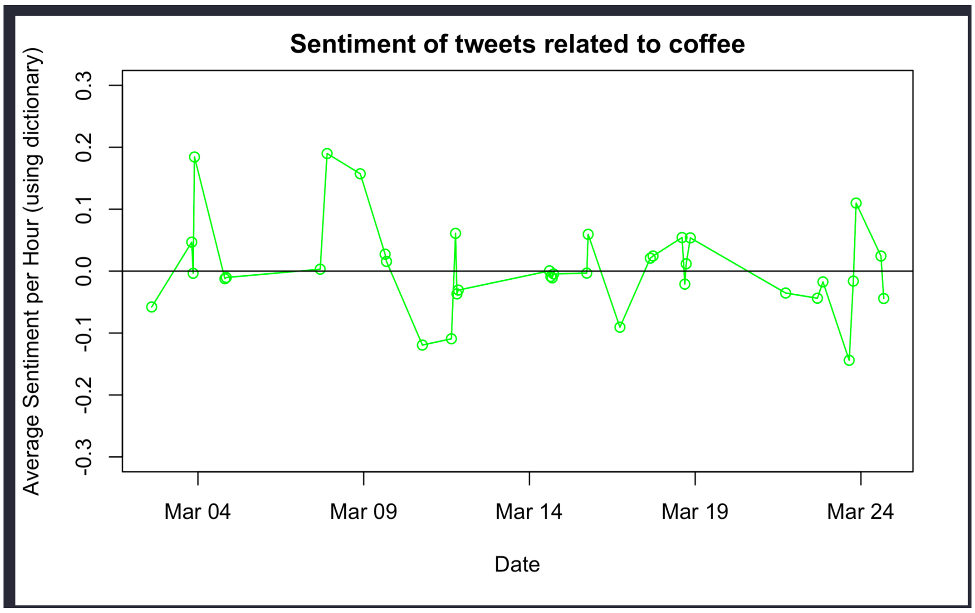
Appendix 22: Word network



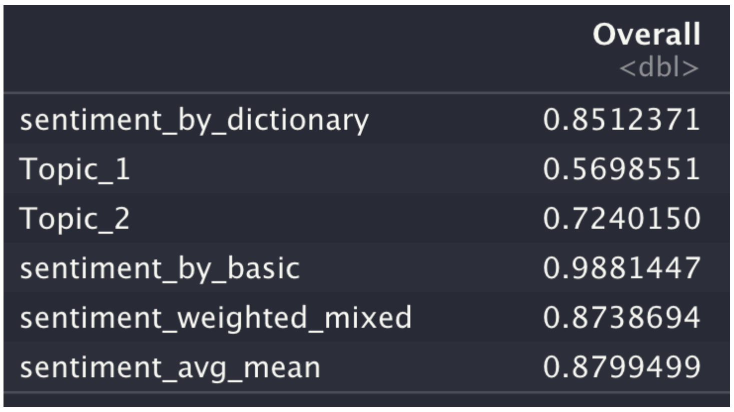
Appendix 23: Sentiment values aggregated per day before scaling



Appendix 24: Sentiment of tweets related to Russia

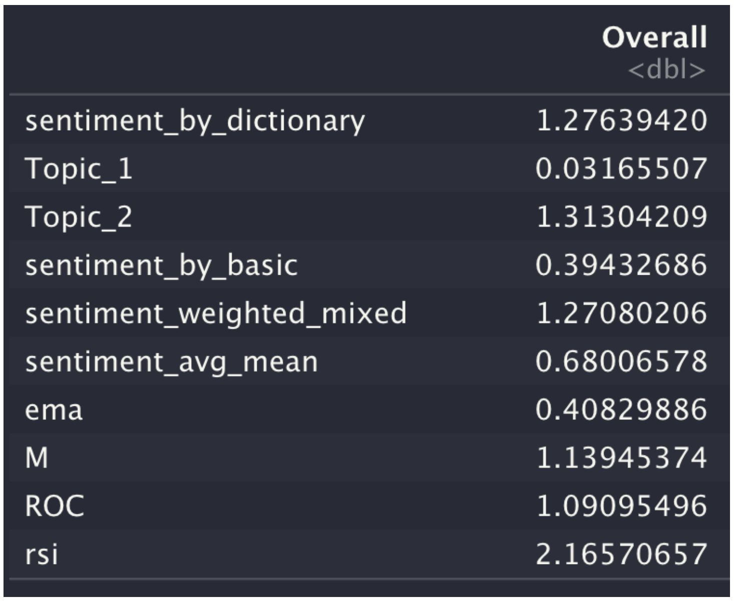


Appendix 25: Sentiment of tweets related to coffee



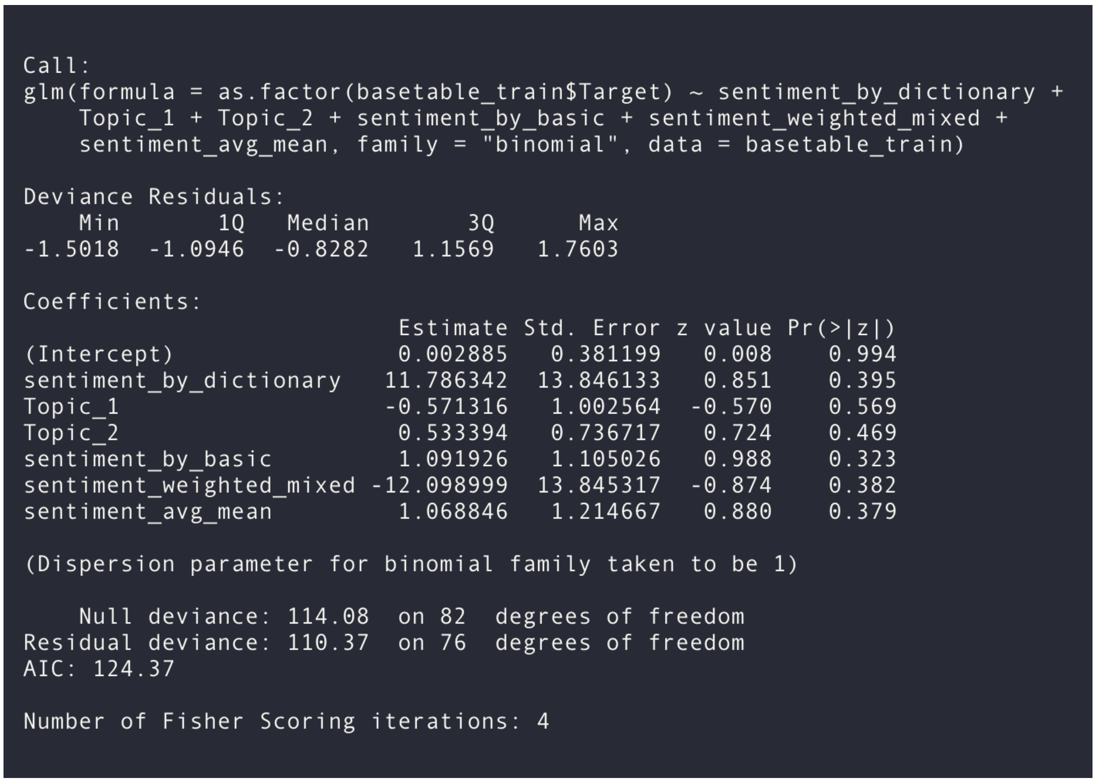
	Overall <dbl>
sentiment_by_dictionary	0.8512371
Topic_1	0.5698551
Topic_2	0.7240150
sentiment_by_basic	0.9881447
sentiment_weighted_mixed	0.8738694
sentiment_avg_mean	0.8799499

Appendix 26: Variable importance Twitter model



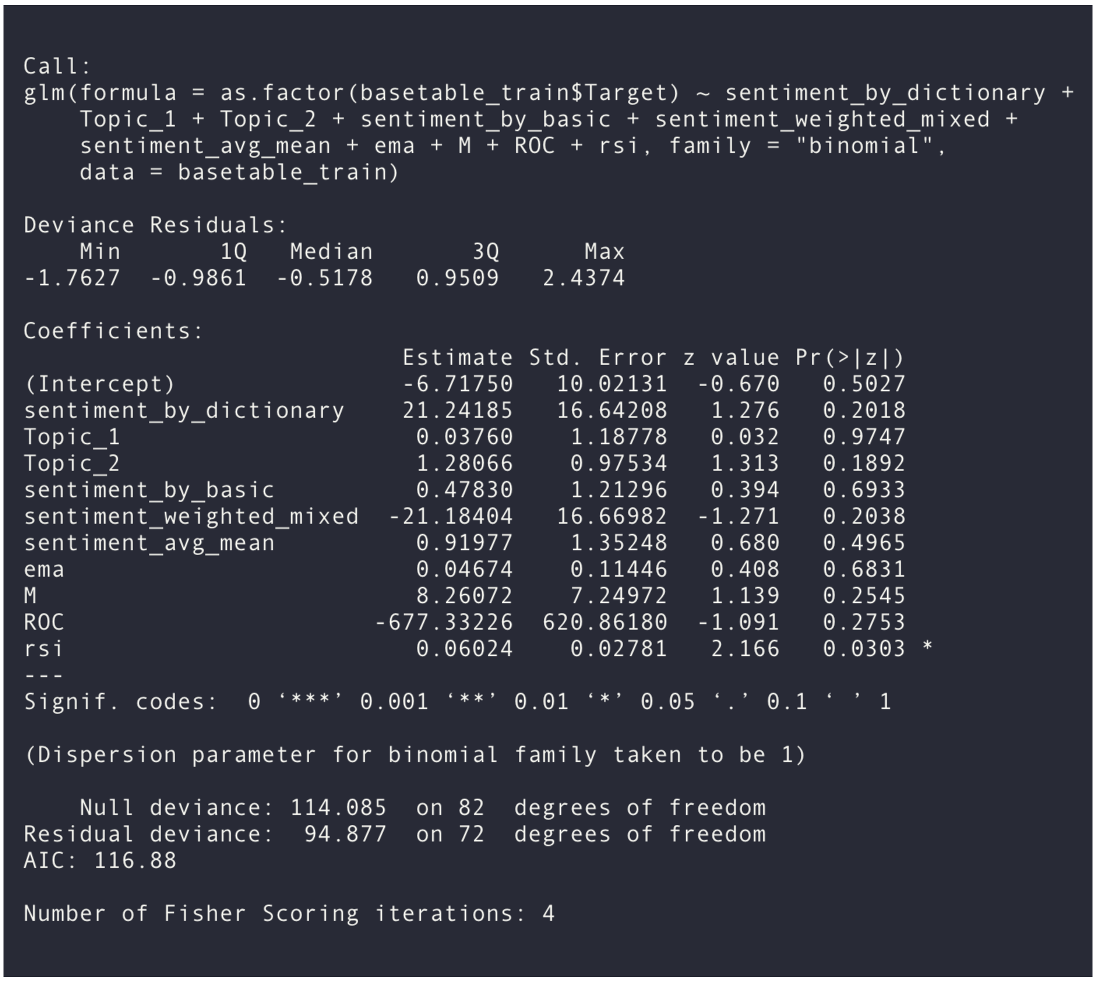
	Overall <dbl>
sentiment_by_dictionary	1.27639420
Topic_1	0.03165507
Topic_2	1.31304209
sentiment_by_basic	0.39432686
sentiment_weighted_mixed	1.27080206
sentiment_avg_mean	0.68006578
ema	0.40829886
M	1.13945374
ROC	1.09095496
rsi	2.16570657

Appendix 27: Variable importance full model



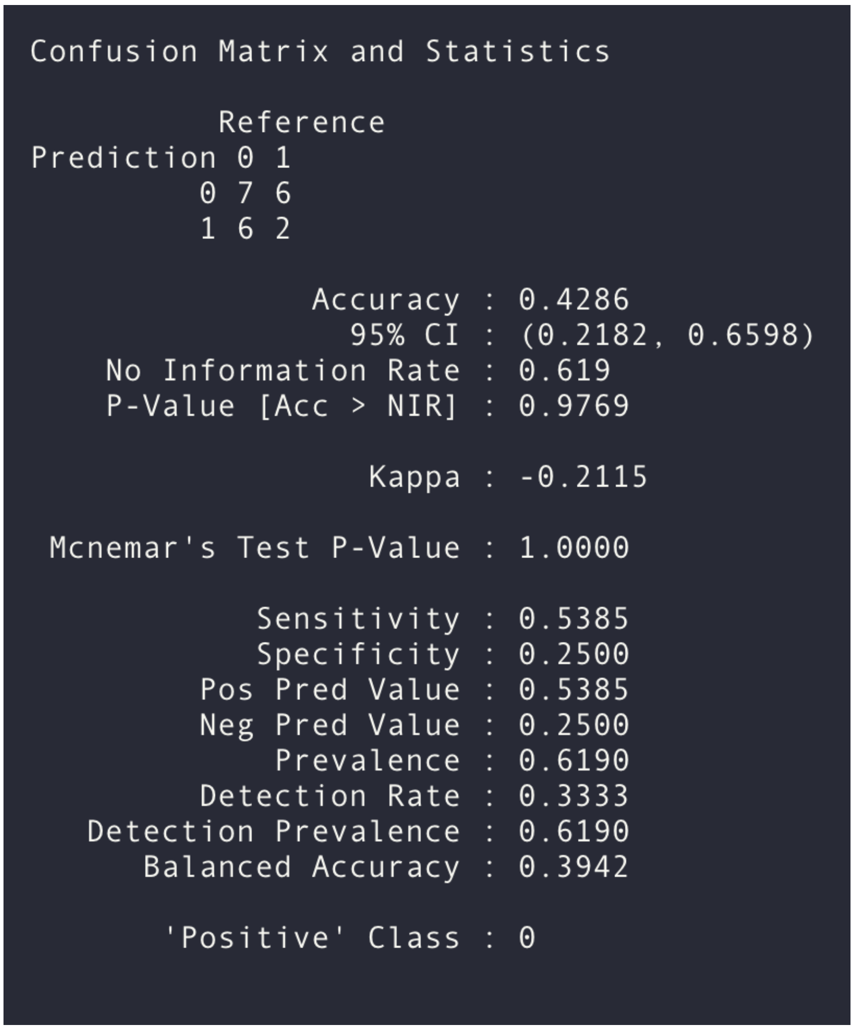
```
Call:
glm(formula = as.factor(basetable_train$Target) ~ sentiment_by_dictionary +
    Topic_1 + Topic_2 + sentiment_by_basic + sentiment_weighted_mixed +
    sentiment_avg_mean, family = "binomial", data = basetable_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5018  -1.0946  -0.8282   1.1569   1.7603

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)               0.002885   0.381199   0.008    0.994
sentiment_by_dictionary  11.786342  13.846133   0.851    0.395
Topic_1                  -0.571316   1.002564  -0.570    0.569
Topic_2                   0.533394   0.736717   0.724    0.469
sentiment_by_basic        1.091926   1.105026   0.988    0.323
sentiment_weighted_mixed -12.098999  13.845317  -0.874    0.382
sentiment_avg_mean        1.068846   1.214667   0.880    0.379

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 114.08  on 82  degrees of freedom
Residual deviance: 110.37  on 76  degrees of freedom
AIC: 124.37

Number of Fisher Scoring iterations: 4
```

Appendix 28: Summary Twitter model



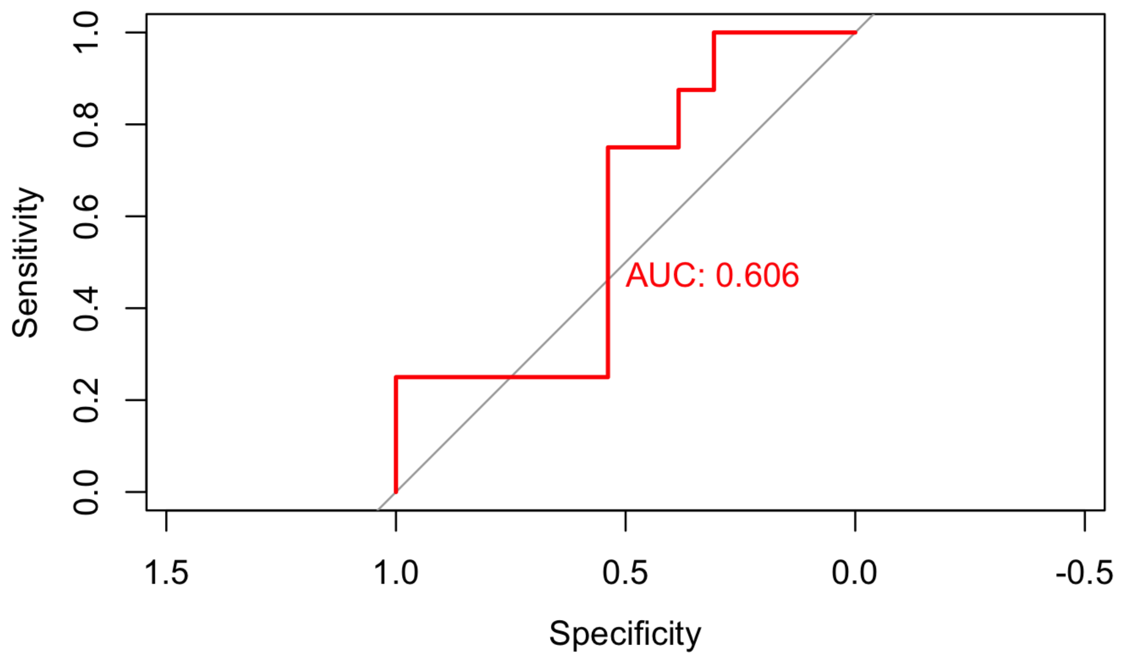
```
Call:
glm(formula = as.factor(basetable_train$Target) ~ sentiment_by_dictionary +
    Topic_1 + Topic_2 + sentiment_by_basic + sentiment_weighted_mixed +
    sentiment_avg_mean + ema + M + ROC + rsi, family = "binomial",
    data = basetable_train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7627  -0.9861  -0.5178   0.9509   2.4374

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                -6.71750   10.02131  -0.670   0.5027
sentiment_by_dictionary    21.24185   16.64208   1.276   0.2018
Topic_1                     0.03760    1.18778   0.032   0.9747
Topic_2                     1.28066    0.97534   1.313   0.1892
sentiment_by_basic          0.47830    1.21296   0.394   0.6933
sentiment_weighted_mixed  -21.18404   16.66982  -1.271   0.2038
sentiment_avg_mean          0.91977    1.35248   0.680   0.4965
ema                         0.04674    0.11446   0.408   0.6831
M                           8.26072    7.24972   1.139   0.2545
ROC                      -677.33226  620.86180  -1.091   0.2753
rsi                         0.06024    0.02781   2.166   0.0303 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 114.085  on 82  degrees of freedom
Residual deviance:  94.877  on 72  degrees of freedom
AIC: 116.88

Number of Fisher Scoring iterations: 4
```

Appendix 29: Summary full model

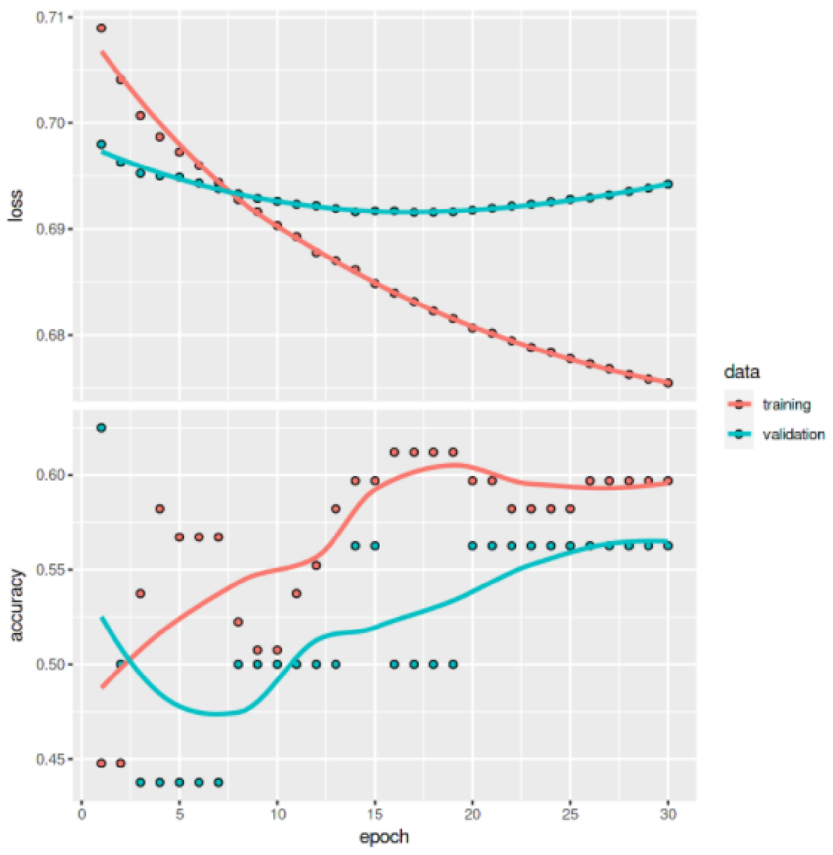


```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 7 6
         1 6 2

               Accuracy : 0.4286
                 95% CI : (0.2182, 0.6598)
    No Information Rate : 0.619
    P-Value [Acc > NIR] : 0.9769

                  Kappa : -0.2115

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.5385
            Specificity : 0.2500
         Pos Pred Value : 0.5385
         Neg Pred Value : 0.2500
             Prevalence : 0.6190
         Detection Rate : 0.3333
   Detection Prevalence : 0.6190
      Balanced Accuracy : 0.3942

       'Positive' Class : 0
```

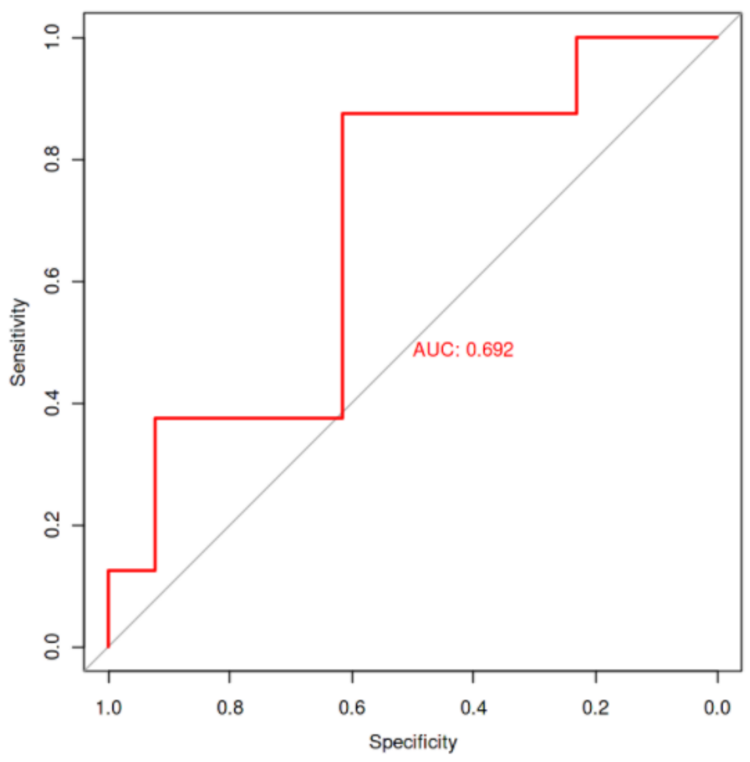
Appendix 30: Confusion matrix Twitter model



Appendix 31: ROC-curve Twitter model: Logistic Regression



Appendix 32: Learning curves neural network



Appendix 33: ROC-curve of the neural network