Text Analysis for Financial News

Gleb Tcypin, 4IZ172

Link, referring to dataset: https://www.kaggle.com/datasets/ankurzing/sentiment-analysis-for-financial-news

or *all-data.csv* file.

1. Basic understanding of data:

Data description: dataset contains 2 variables **- sentiment** and **text**.

Observations: **4846** (according to Kaggle, 4837 of these are unique)

sentiment variable: our target variable, contains 3 values – **negative, neutral** and **positive**.

text variable: each row(observation) corresponds to post about financial event

	sentiment	text
0	neutral	According to Gran , the company has no plans t...
1	neutral	Technopolis plans to develop in stages an area...
2	negative	The international electronic industry company ...
3	positive	With the new production plant the company woul...
4	positive	According to the company 's updated strategy f...

Pic. 1: df.head() of initial dataset

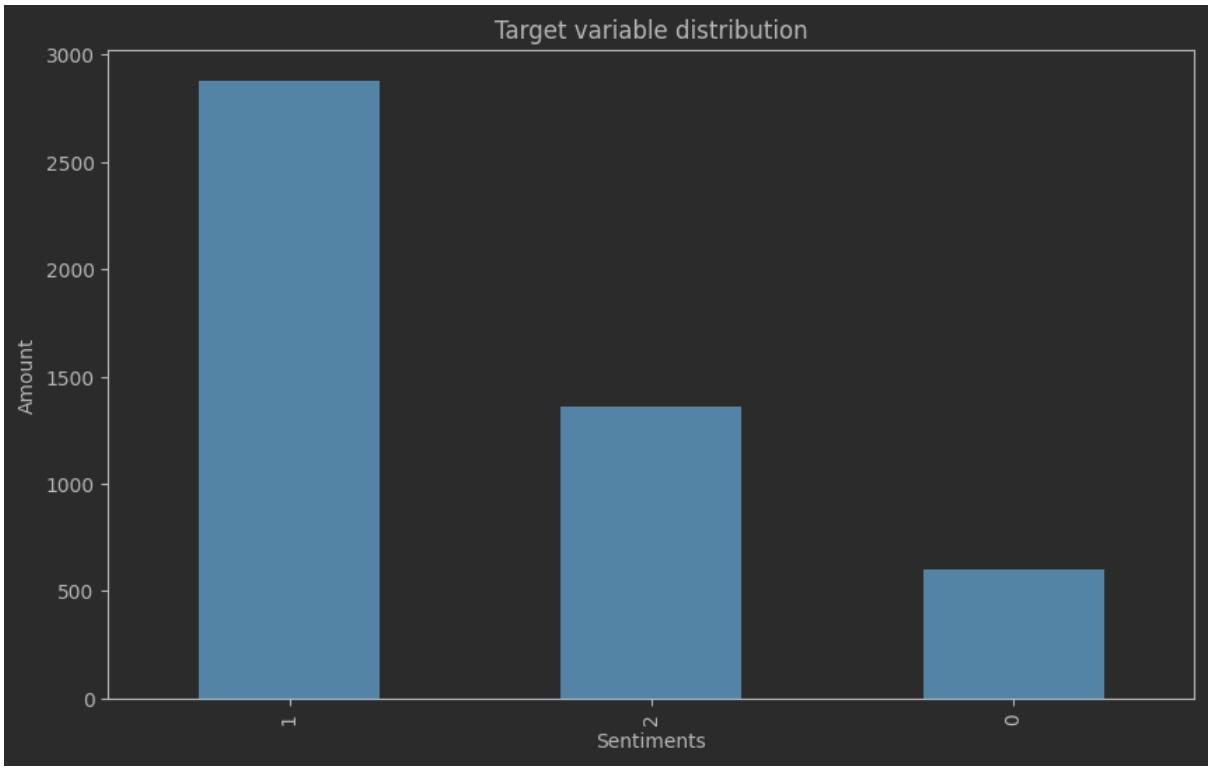
2. Basic setup:

Libraries used: refers to *SolutionCode.ipynb* file.

Target variable encoding: sentiment variable was encoded by LabelEncoder() function -> 0 – negative, 1 – neutral, 2 – positive

3. Quick EDA:

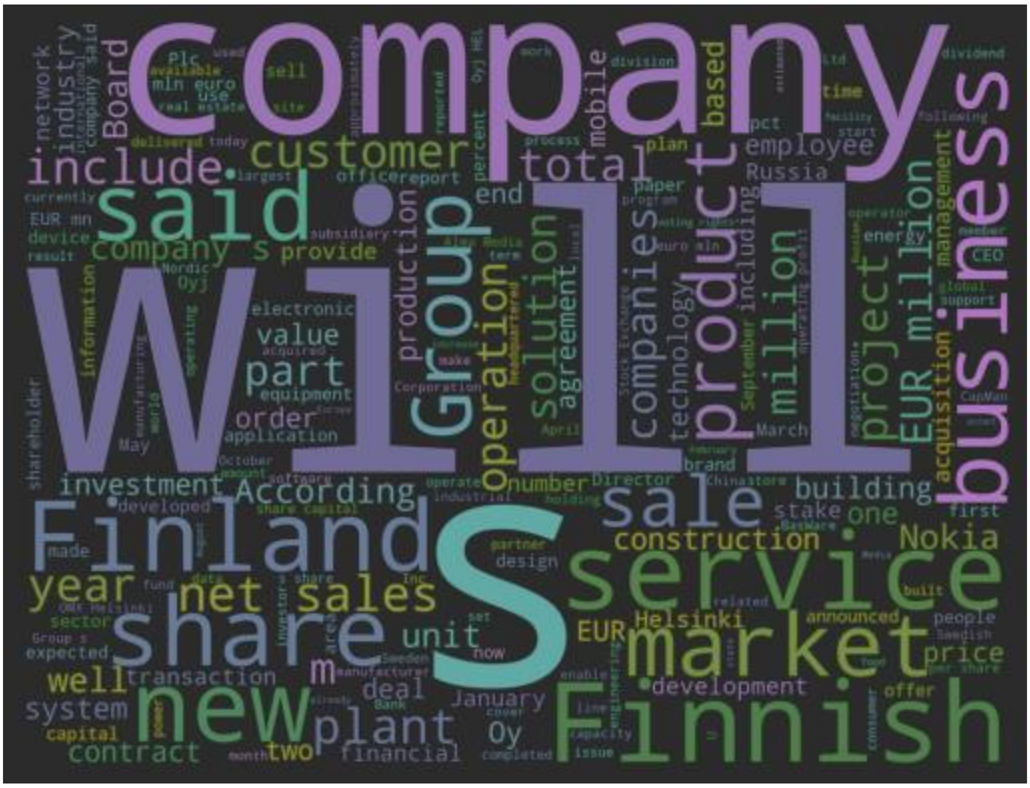
Here is introduced the target variable distribution. As we can see **neutral** class is overbalancing the rest. The **negative** class is the lowest one. We can conclude that target variable is imbalance.



Pic. 2: Target variable distribution

Here are the most frequent words for each class:

Pic. 3: The most frequent words of negative sentiment



Pic. 4: The most frequent words of neutral sentiment

Pic. 5: The most frequent words of positive sentiment

4. Basic preprocessing:

The **text** variable was straightly preprocessed for model by custom function *preprocess(text)*. Inputted text was switched to lowercase, tokenized, lemmatized and joined with respect of English stopwords and punctuation, such both of these things were deleted from dataset to make our models produce more meaningful results.

4.1. Additional preprocessing:

For Clustering and Collocation parts digits were replaced by empty strings and only words were matched for further procedures.

5. Models (classification procedure)

First of our models will be Multinomial Logistic Regression (because of having multiple classes in target variable) with 5000 of maximum iterations which help us to achieve the most optimal values.

	precision	recall	f1-score	support
0	0.77	0.42	0.54	110
1	0.73	0.95	0.83	571
2	0.77	0.45	0.56	289
accuracy			0.74	970
macro avg	0.75	0.60	0.64	970
weighted avg	0.75	0.74	0.72	970

Pic. 6: Metrics and corresponding values of MultiNom LogReg

These values can be considered as good, because it provides us 75% of precision, 60 % of recall (in case of equally importance of classes) and 74% (in case of imbalanced class distribution, which is our case). The same principle works for F1-score. Overall, considering that F1 is more descriptive metric than others and that our dataset is not balanced, 74% can be considered as good model.

Secondly, SVM model was introduced.

	precision	recall	f1-score	support
0	0.74	0.35	0.48	110
1	0.71	0.97	0.82	571
2	0.79	0.37	0.51	289
accuracy			0.72	970
macro avg	0.75	0.57	0.60	970
weighted avg	0.74	0.72	0.69	970

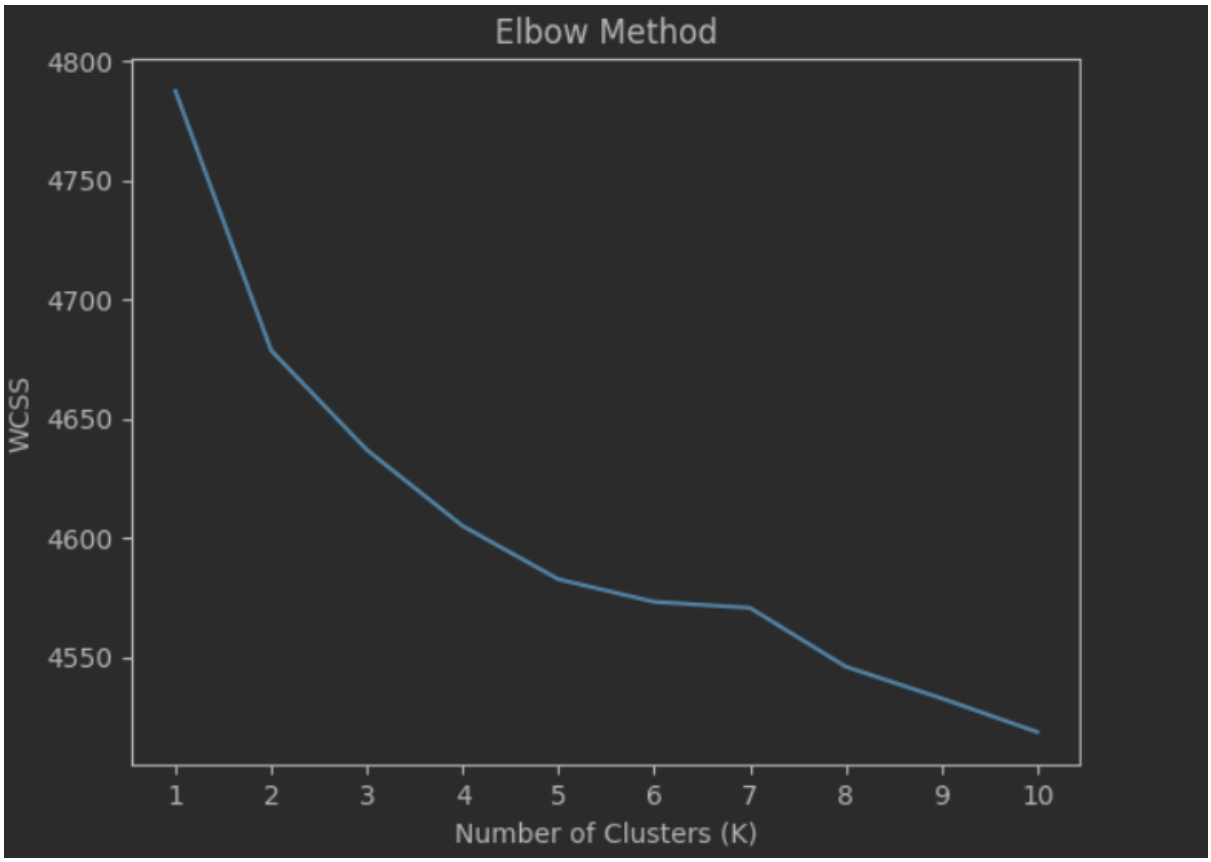
Pic. 7: Metrics and corresponding values of SVM (One-vs-One approach)

Results of this model is overall worse than Multinomial LogReg, despite of some better individual performances.

6. Clustering:

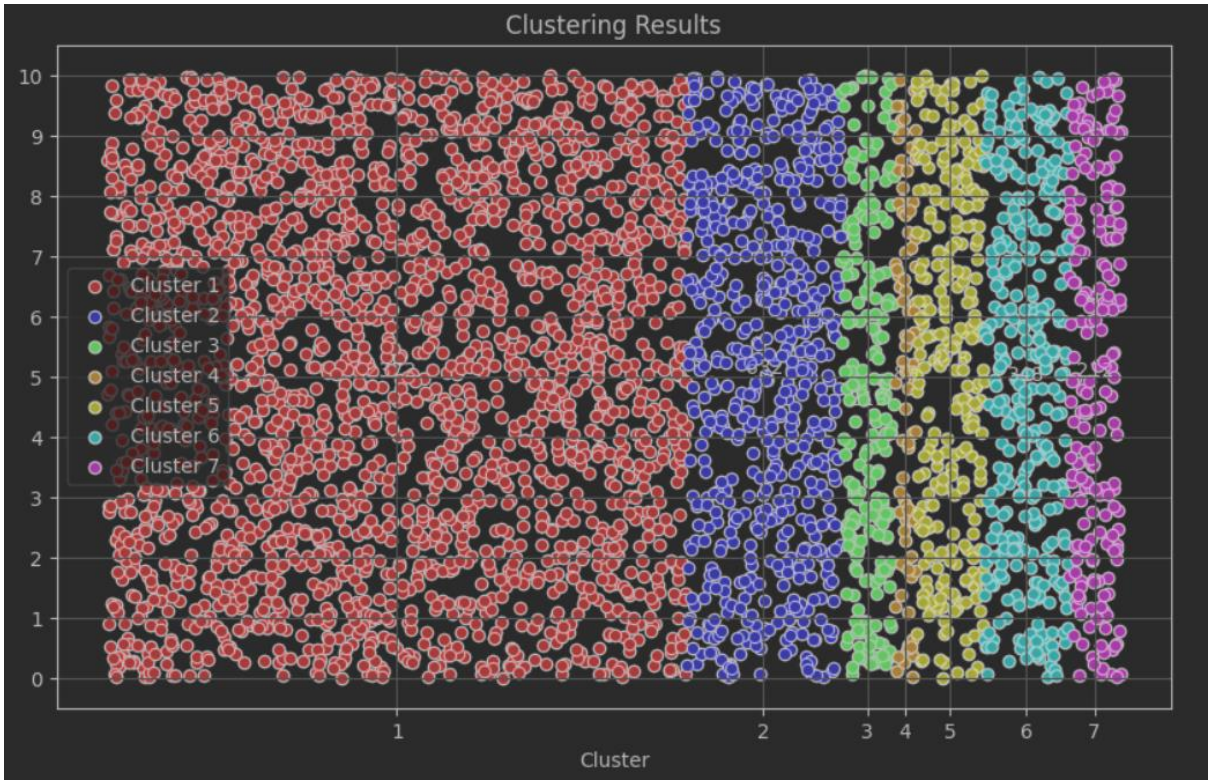
K-means algorithm was used for solving this particular task.

Optimal number, which is 7 ,was found by Elbow method.



Pic. 8: Inertia – K curve

At 7 number of WCSS starts to decrease more slowly, which can be considered as suitable point for balance the WCSS (inertia) – K (number of clusters) tradeoff.



Pic. 9: Clustering results

Number of documents in each cluster:

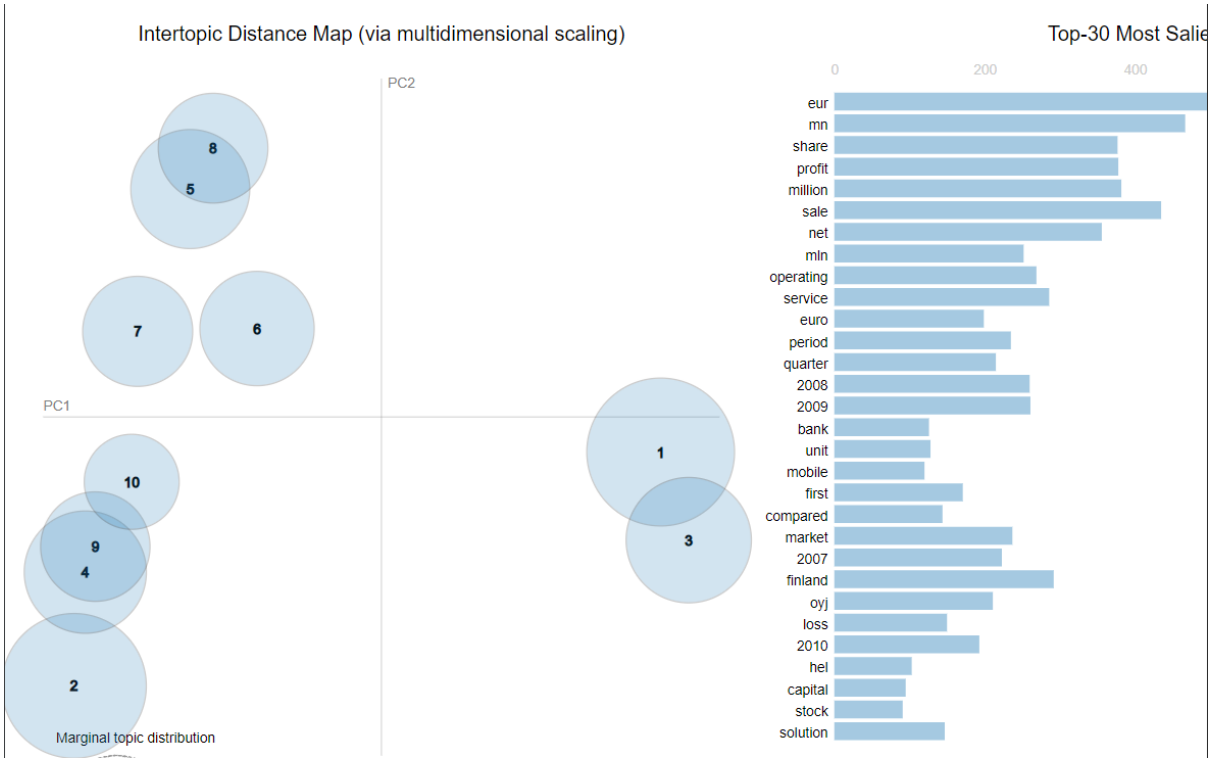
- Cluster 1 – 2372
- Cluster 2 – 632
- Cluster 3 – 219
- Cluster 4 – 89
- Cluster 5 – 278
- Cluster 6 – 348
- Cluster 7 – 212

More detailed description of clustering located in code.

7. Topic modelling:

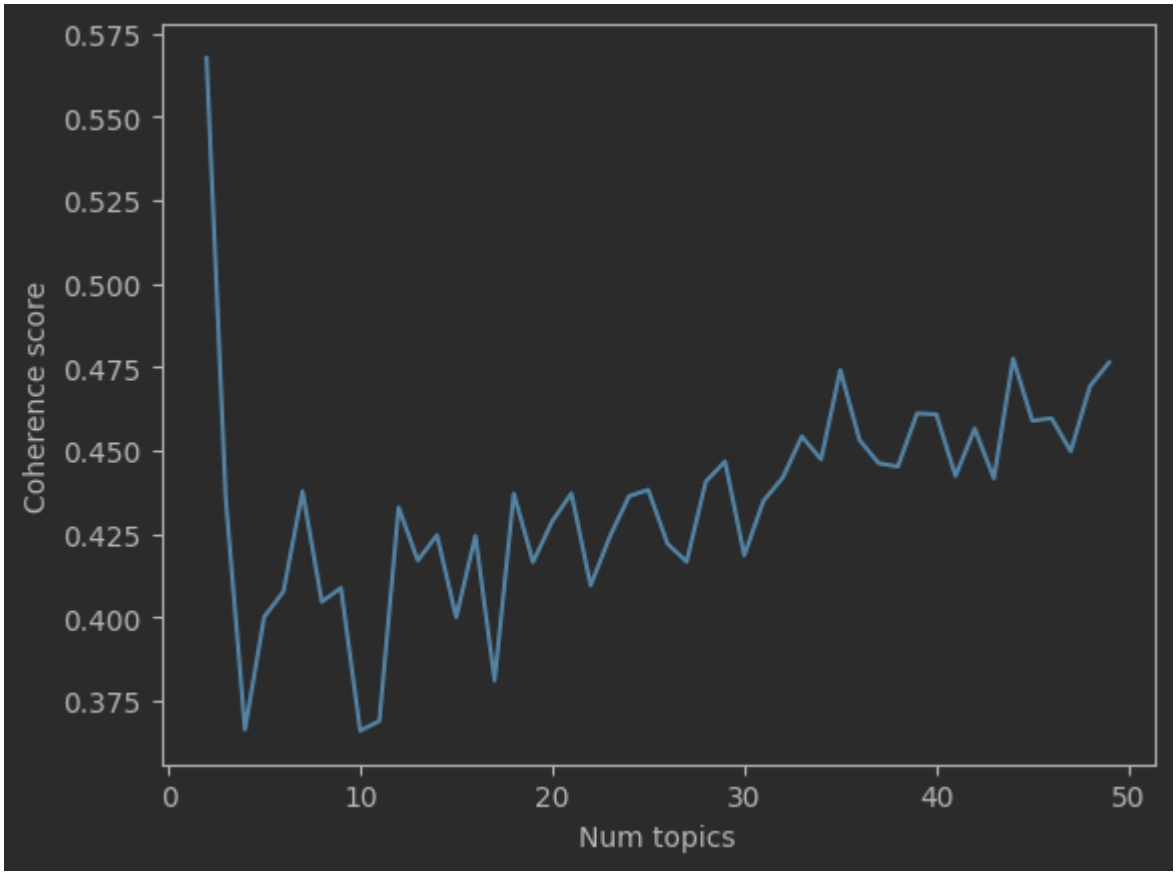
This task was solved with LDA model.

First visualization of LDA with default 10 topics:



Pic. 10: Topic modelling visualization with default (10) topics

Then, it becomes clear, that we need to find optimal number of topics, despite our default model showing relatively good performance: -8.2325 is Perplexity score and 0.4093 is Coherence score.

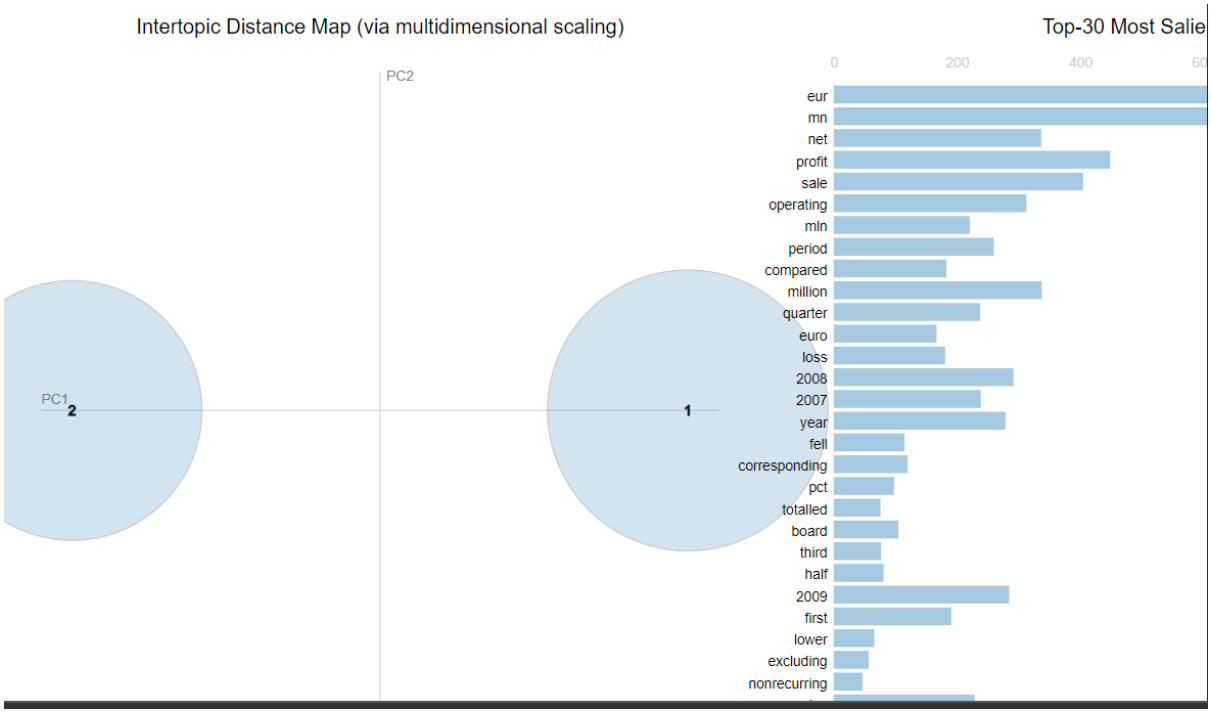


Pic. 11: Finding the optimal number of topics

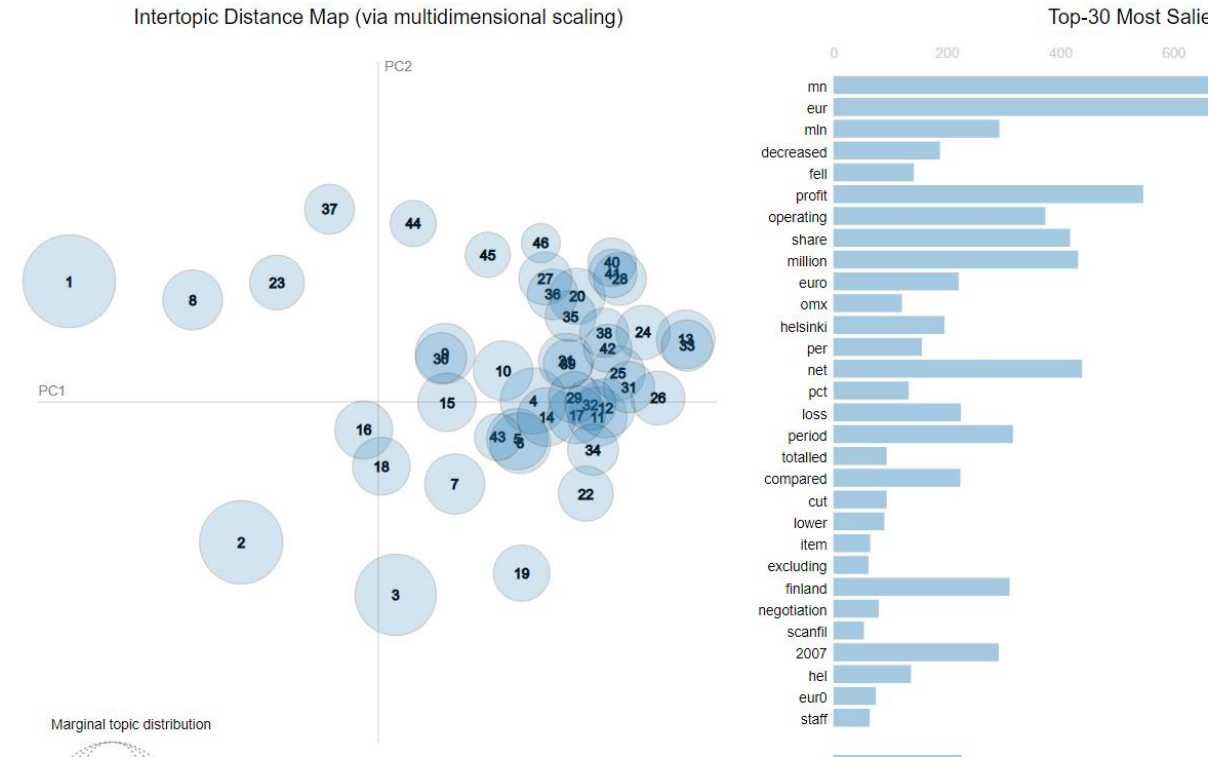
The task was as follows: find the number of topics, whose are not overlapping much with each other, meanwhile, to have enough topics.

However, this requirement was not satisfied.

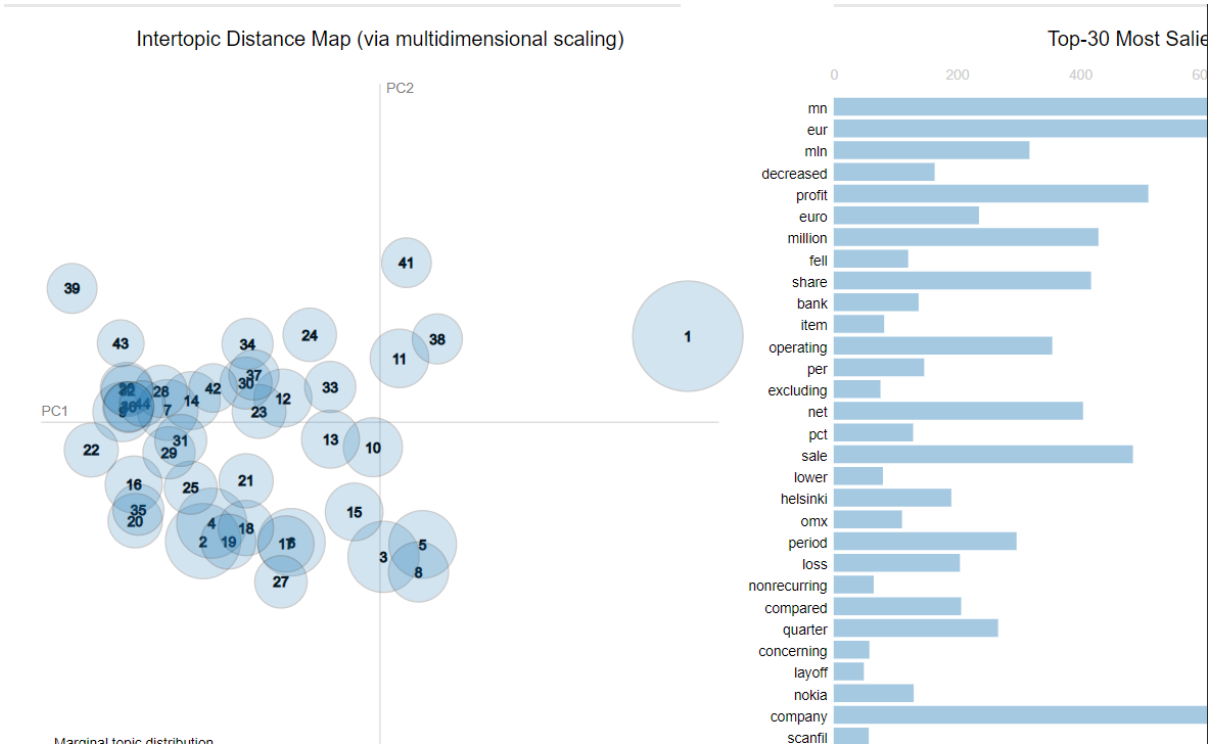
The maximum Coherence score was at 2 topics which is very small amount, followed by 44 and 46 number of topics, whose are kind of excessively high numbers.



Pic. 12: LDA model visualization with 2 topics



Pic.13: LDA model visualization with 46 topics



Pic. 14 LDA model visualization with 44 topics

As we can see, most satisfied model from topic homogeneity perspective is model with 2 topics. Both 44 and 46 topics else showing us some homogeneity of topics, but number of topics is much bigger. From the other side, model with 2 topics doesn't tell us much about variety of topics in the corpus.

8. Analysis of mutual similarity:

Cosine similarity approach was used to solve this task. Cosine similarity matrix was created from modified initial **text** variable. Picture below showing the most interesting relations between documents, where threshold of cosine similarity is 0.9. Blue parts displays pairs of documents that have Cosine similarity within this interval: [0.9; 1), hence green parts displays pairs with Cosine similarity equals to 1.

This algorithm reached 58 unique document pairs.

Pic. 15: Similar document pairs

9. Collocation analysis:

Initial preprocessed **text** variable was modified once more (see. Basic preprocessing). Here you can see top 20 bigrams of modified corpus:

net sale	operating profit	corresponding period	mln euro	oyj hel
omx helsinki	stora enso	said today	third quarter	per share
net profit	stock exchange	first quarter	real estate	second quarter
year earlier	alma medium	nine month	board director	adp news

Table 1: Top 20 bigrams in the corpus

Table below showing us top 20 words with highest TF-IDF score:

('forecast', 1.0)	('alone', 1.0)	('could', 1.0)	('welcome', 1.0)	('think', 1.0)
('thousand', 0.9007865546564954)	('gearing', 0.8906513436532357)	('capman', 0.8824613712544839)	('sekm', 0.8822750380611936)	('layoff', 0.8551528546185981)
('xa', 0.8540616406543488)	('catalyst', 0.8492098122051356)	('aspo', 0.8426974649984073)	('reserved', 0.8362268497484426)	('kemira', 0.8283433549616038)
('nwc', 0.8208188924095403)	('mln', 0.8160874489708007)	('nd', 0.8143222844294745)	('know', 0.8101744580566537)	('billion', 0.8061246849117512)

Table 2: 20 words with highest TF-IDF score in the corpus