UNIVERSITÀ DI PISA

Sentiment Analysis for Financial News

Text Analytics a.y. 2022/2023

Edited by:
Salvatore Giovino (640413)
Ismaele Gorgoglione (544853)
Mariam Narchemashvili (634789)

University of Pisa

April 17, 2023

Contents

1 Introduction

The dataset that is being analyzed in this paper is a collection of 4840 sentences. The provided collection has been annotated by 16 people, all with relevant background knowledge of finances. So, in the end, we have the dataset with two columns, in the first, we have the provided annotation, a "Sentiment" (in our case the target class) which can be either *positive*, *negative*, or *neutral*, and in the second, we have the sentence itself. Figure 1 below is the representation of the dataset with the first 5 instances.

Sentiment	News
neutral	According to Gran , the company has no plans to move all production to Russia , although that is where the company is growing .
neutral	Technopolis plans to develop in stages an area of no less than 100,000 square meters in order to host companies working in computer technologies and telecommunications , the statement said .
negative	The international electronic industry company Elcoteq has laid off tens of employees from its Tallinn facility ; contrary to earlier layoffs the company contracted the ranks of its office workers , the daily Postimees reported .
positive	With the new production plant the company would increase its capacity to meet the expected increase in demand and would improve the use of raw materials and therefore increase the production profitability .
positive	According to the company 's updated strategy for the years 2009-2012 , Basware targets a long-term net sales growth in the range of 20 % -40 % with an operating profit margin of 10 % -20 % of net sales .

Figure 1: Original Dataframe

Three of the annotators were researchers and the remaining 13 annotators were master's students at Aalto University School of Business with majors primarily in finance, accounting, and economics.

Since the study is focused only on financial and economic domains, the annotators were asked to consider the sentences from the viewpoint of an investor only; i.e. whether the news may have a positive, negative, or neutral influence on the stock price. As a result, sentences that have a sentiment that is not relevant from an economic or financial perspective are considered neutral.

The motivation of this project is to train the best classification model in such a way that in the face of a new set of data relating to the financial theme, it can label the sentiment related to individual news and then exploit it as a possible indicator to act in the market.

The objective of the work is to classify each example sentence into a positive, negative, or neutral category by considering only the information explicitly available in the given sentence.

We summarize the project in four tasks:

- Data Understanding and Preparation
- Unsupervised Learning: Clustering
- Supervised Learning: Classification

2 Data Understanding & Preparation

2.1 Data Semantics

In the following section, the distribution of variables is represented with the help of different kinds of visualization tools. In figure 2, we can see the distribution of class labels throughout the whole dataset. While looking at it we notice a strong presence of neutral annotations, which takes 59.4% of the whole data, while we have quite a low portion of positive and negative ones (respectively 28.1% and 12.5%). This was the reason that lead us to perform balancing for subsequent analysis.

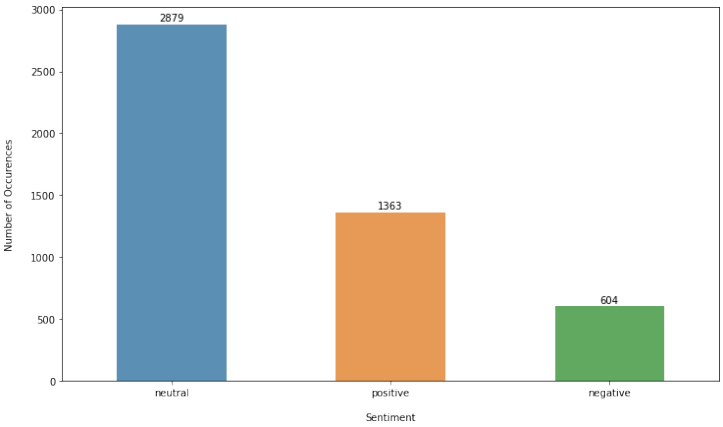
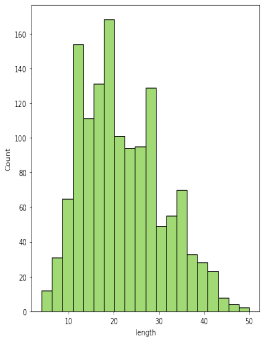


Figure 2: Class Distribution

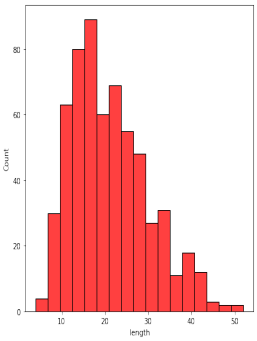
Studying the individual classes in more detail, we learned some statistical measures for each of them, such as mean, minimum and maximum sentence length, standard deviation and etc., that can be seen in the figures below.

	length
count	1363.0
mean	21.9
std	8.9
min	4.0
25%	15.0
50%	21.0
75%	28.0
max	50.0



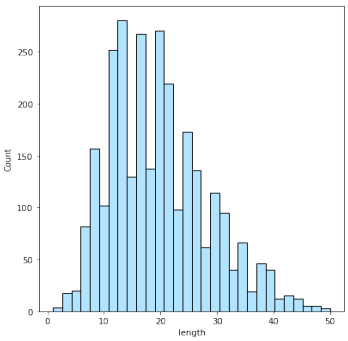
(a) Positive

	length
count	604.0
mean	21.24
std	8.88
min	4.0
25%	15.0
50%	20.0
75%	27.0
max	52.0



(b) Negative

	length
count	2879.0
mean	19.67
std	8.63
min	1.0
25%	13.0
50%	18.0
75%	25.0
max	50.0

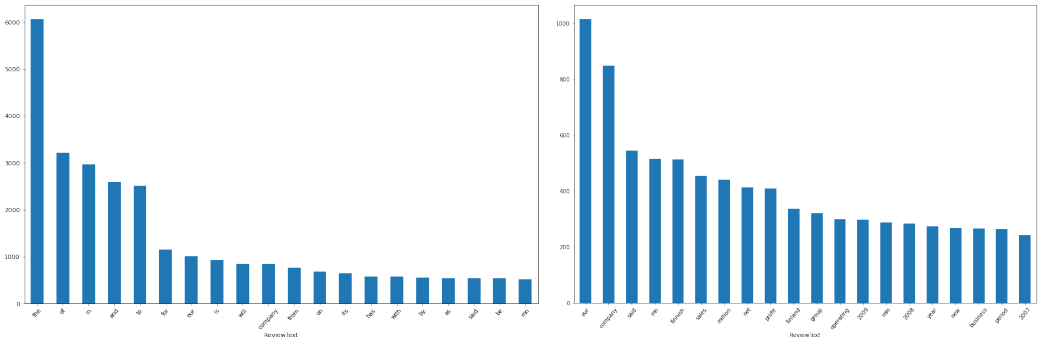


(c) Neutral

Figure 3: News Distribution among Classes

From figure 3 it is obvious that the largest number of sentences of each class is in the range of 10 to 20, and the average length is around 20. Considering all the other statistical measures, except for the *count* measure, they are more or less similar for each class. A sharper distinction, however, is highlighted by their distribution, in which we see that sentences classified as neutral have peaked between 11 and 22, while negative sentences are more concentrated between 10 and 18, and positive sentences have a pattern more similar to neutral sentences. Although a good presence is also found up to 28. In general, however, they follow a similar trend, increasing until around 20 and then decreasing until around 50.

Figure 4 provides evidence of dramatic change in the most frequent words before and after removing *Stopwords* from the dataset.



(a) Before Removing Stop Words (b) After Removing Stop Words

Figure 4: Top 20 Frequent Words

In fact, instead of having conjunctions, prepositions, or articles, there are words like "eur", "company" or even notable mentions of specific years 2007, 2008, and 2009. It seems to be because of the financial crisis of those years.

Additionally, the top 5 Parts of Speech (POS) were analyzed, which can be seen in figure 5. Most of the news sentences include singular nouns (NN), person nouns (NNP) prepositions (IN), followed by definite articles (JJ), and cardinal digits (DT).

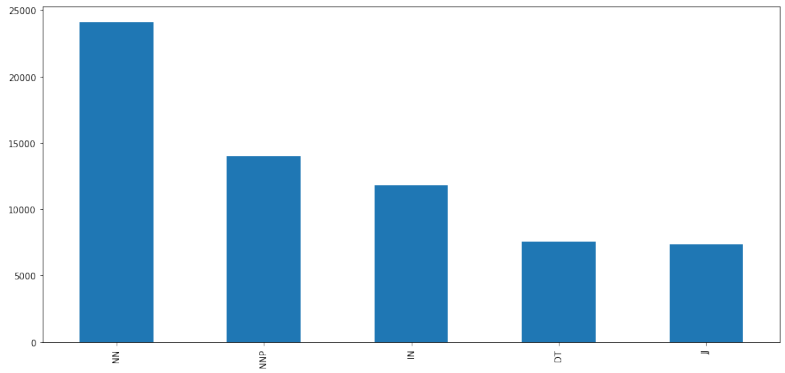


Figure 5: Top 5 POS Tagging

2.2 Preprocessing

In this section, initial data cleaning steps have been carried out, followed by a phase of balancing data for subsequent analysis.

Missing Values and Duplicates. In order to clean the data, first we checked for missing values and discovered that in our dataset we do not face this problem. After we looked for duplicates, which resulted in finding 6 news sentences absolutely identical to each other, so we removed them from the original dataset.

Special Characters. Then, we continued the process by removing special characters from the text with the help of REGEX. In particular, it has been used to identify and remove any *Hashtags*, *URLs*, and *Mentions*. Also, to keep only alphanumeric characters, another expression to remove is the decimal digits of numbers with commas. Finally, other expressions to transform were the words like *"EUR79"*, which have been changed by *"EUR 79"* since the number of such instances was extremely high.

Words of Same Meaning. After an initial analysis of the word clouds, we realized that different words came up which had a high frequency (occurrence rate) but were only different acronyms to indicate the same thing. An example would be the case of *"Oyj"* and *"Oy"* indicating Public Limited Company or even the different ways to indicate *"millions"* such as *"m"*, *"mln"*, or *"million"*. Another example is the case of *"EUR"* and all the different ways to indicate it. After discovering all these occurrences we came up with one unique expression for each case (such as *"eur", "m" and etc.*) to replace all the others of the same meaning in order to avoid any misinterpretations.

Lemmatization. As the last step of the data cleaning task, we applied the lemmatization algorithm from the TextBlob library in such a way that two or more similar words can be traced back to the same lemma, relying on morphological analysis of the word through predefined dictionaries in the library itself. This task was performed in order to avoid the occurrence of the same original words in different forms, such as *"decreases", "decreasing", or "decrease"* and etc.

In general, from the figure 6 we can say that all three classes have many words in common, such as *"EUR", "Company", "million" or "Finnish"*, (we probably have *"Finnish"* as one of the most relevant words because of the experts). However, we notice that for the positive class there are words such as *"profit", "contract", "Agreement", or "increase"*, while in the negative class we have *"decreased", "fell", "lower", or "loss"*.

(a) Positive

(b) Negative

(c) Neutral

Figure 6: Word Clouds for Each Class

After cleaning the dataset we created a new one to balance the classes with the technique of Undersampling. We did not perform the oversampling technique as after splitting between train and test sets (70%-30%) the minority class has only 423 records compared to 1995 records of the neutral class and 970 of the positive one, therefore we did not want to create a lot of artificial records. Two undersampling techniques have been tried random undersampling, which however reduced the dataset too much by flattening the number of records of each class equal to the one of the minorities (423 records), and undersampling by specifying a suitable percentage of random removal of the records, 40% for the neutral and positive classes thus obtaining the following distribution of the classes after the split:

Before Undersampling			
	Neutral	Positive	Negative
Train	2011	958	419
Test	862	405	185
After Undersampling			
	Neutral	Positive	Negative
Train	1192	592	419
Test	532	226	185

3 Clustering

In this part of the report, we provide an analysis of the unsupervised learning task, Clustering. It is considered as unsupervised as we do not have any prior knowledge of an actual class of given data, meaning that we are blindfolded in the process. The main idea is that clustering exploits similarities between the data to be analyzed, similarities that can be of various natures, but which are essentially a distance between the dataset points. Different algorithms can be used for clustering analysis, but the following section will examine K-Means. The reason would be the type of data that we have (text) and also the fact that it is said to be one of the most famous algorithms being used for this task. Before applying the technique, it is required to go through some preprocessing steps. Especially when working with text, there are several steps that have to be taken prior to clustering.

As a first step of preprocessing, we drop the "sentiment" column from the dataset, as it is a class label and should not be taken into consideration in unsupervised learning. After, we apply natural language processing tools (from the Natural Language Toolkit package), specifically the ones for tokenization and vectorization, to transform collection of human-readable text into a matrix of token counts. All these results in the type of data that could be fed to K-means.

3.1 K-means

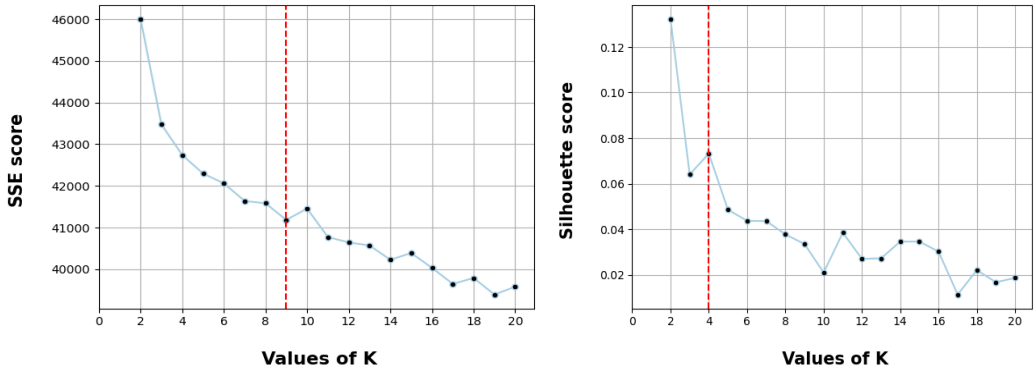
In order to run the algorithm, first we need to define the number of clusters. One of the most common ways to define this number is using an Elbow Methos. It helps to plot the WCSS (Within Clusters Summed Squares) values and selects the point where the parameter value falls more than the previous value. For each point, the Sum of Squared Error (SSE) is the distance to the nearest cluster. To get SSE, we square these errors and sum them. Finally, we choose the optimal number of clusters considering the lowest error with respect to the lower number of clusters. The lower the SSE the better.

Another method that can help to choose the number of clusters is the Silhouette method. In this procedure, the silhouette coefficient is plotted, and the maximum value is selected. The higher the Silhouette Coefficient the better.

Figure 7a represents the SSE score with respect to the number of K (clusters). It is quite obvious from the graph that it was not easy to identify the optimal number for K, as we did not have a significant drop in terms of error score. The reason why we chose K equal to 9 is the noticeable drop in the range of (7, 10).

On the other hand, figure 7b represents the Silhouette score with respect to the number of clusters. As we can see the more, we increase the number of clusters the score decreases, the highest score we could achieve is in the case of K = 4.

After considering both metrics, we decided to test the results on both cases (K = 9 and K = 4). Unfortunately, after running the K-means algorithm the obtained results were not promising in either case as we achieved quite unbalanced clusters.

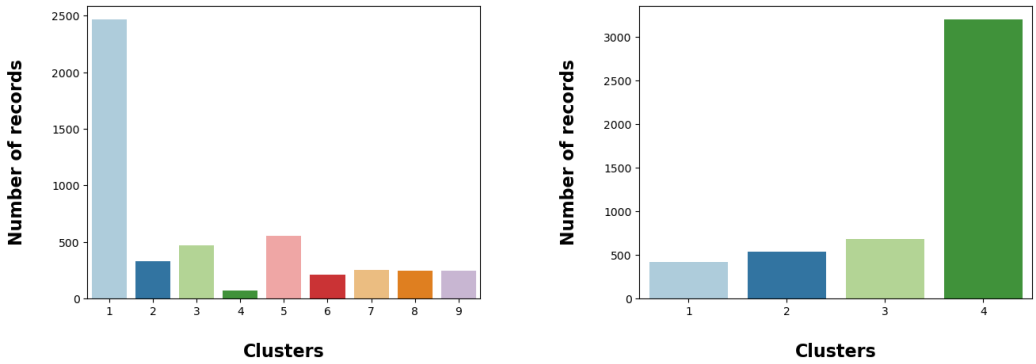


(a) SSE score (b) Silhouette score

Figure 7: SSE and Silhouette

	SSE	Silhouette
9 Clusters	41066.9176	0.0273
4 Clusters	42302.7753	0.0426

Figure 8a demonstrates the distribution of records into clusters, and as we can see it is done poorly as most of the records end up in the first cluster. The same conclusion can be derived from the case of 4 clusters (see figure 8b).



(a) Case 1: 9 clusters (b) Case 2: 4 clusters

Figure 8: No. of Records per Clusters

In order to capture significant information (if any) obtained from this task, we decided to represent some statistics. For instance, figure 9 shows the class distribution inside cluster 1 obtained in Case 1 (K = 9). All the other clusters share almost the same distribution of the label, meaning that none of them is highly representative of any class.

We also tried to extract the most common features shared among each cluster. Figure 10 is the representation of the most frequent words with their corresponding scores in cluster 1 of Case 1.

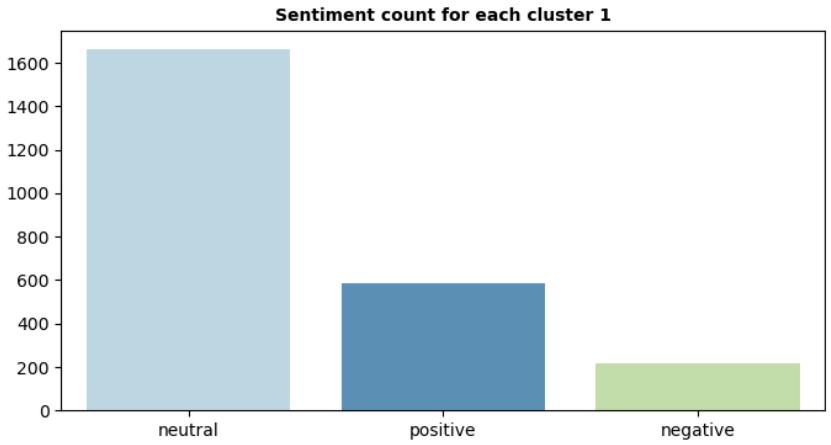


Figure 9: Sentiment count

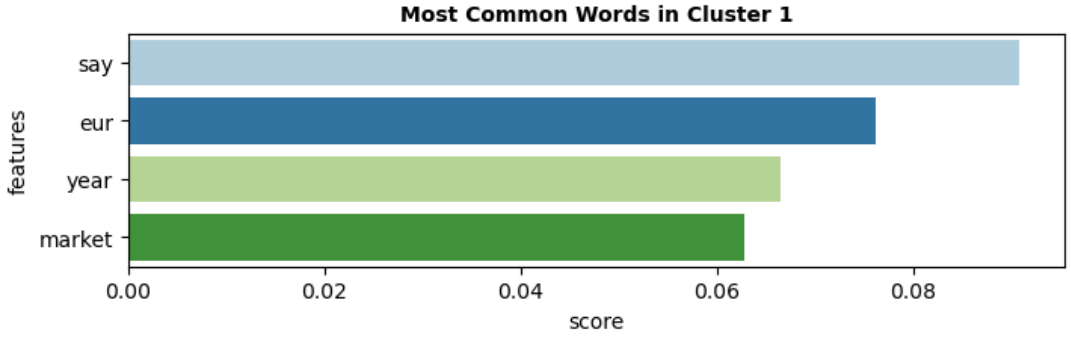


Figure 10: Most frequent word

To conclude the reasoning, we think that carrying out unsupervised learning tasks on this type of data is not powerful, meaning that the obtained results do not provide any valuable insights from the data.

4 Classification

Data for the classification task consists of a collection of instances/records and each of them is characterized by the tuple (x, y). X – represents attribute/predictor and Y- represents class/response. In the case of the dataset that we are analyzing, the predictor is the news (sentence itself) while the response is the sentiment. In this part of the analysis, we examined and evaluated different classification algorithms, compared the obtained results, and identified the "winner" algorithm.

As the dataset is not balanced with respect to the target attribute ('sentiment'), all the steps have been repeated not only on the original sets but also on the balanced ones. Before diving into algorithms, we need to prepare the data. As a first step, the dataset has been separated into training and test sets. After that, the categorical attribute

('sentiment') has been dropped from each set. To the 'news' attribute the same step of vectorization has been carried out, to transform the text into vectors.

	No. of Records
Train	3388
Test	1452

Below are the steps that have been followed after preparing the sets:

- Learning algorithm: a systematic approach to learning classification model on the training set.
- Induction: by using a learning algorithm, building a classification model.
- Deduction: applying classification model on the test set (unseen test instances).

For each of the learning algorithms, all the significant parameters have been tuned in order to achieve the best result possible. Therefore, in the following sections, all the results are ones obtained after Hyper-parameter Tuning techniques from the model selection class of 'sklearn' library. More specifically, the 'Grid Search' method has been used with cross-validation *(cv = 5)* which helps to search for the best settings over specified parameter values for an estimator.

4.1 Random Forest Classifier

The first model that has been tested was the Random Forest Classifier, which is one of the ensemble learning algorithms, that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

The parameters used for this model are *"min_samples_leaf = 1", "min_samples_split = 7", "n_estimators = 400", "max_features = 'auto'", and "criterion = 'gini'".*

Figure 11 reports the results that have been achieved after examining this specific classifier. We can see that the 'neutral' class has the highest scores in precision, recall, and F1-score, which is not surprising as most of the data points are of this class. We can say that the results are quite satisfactory.

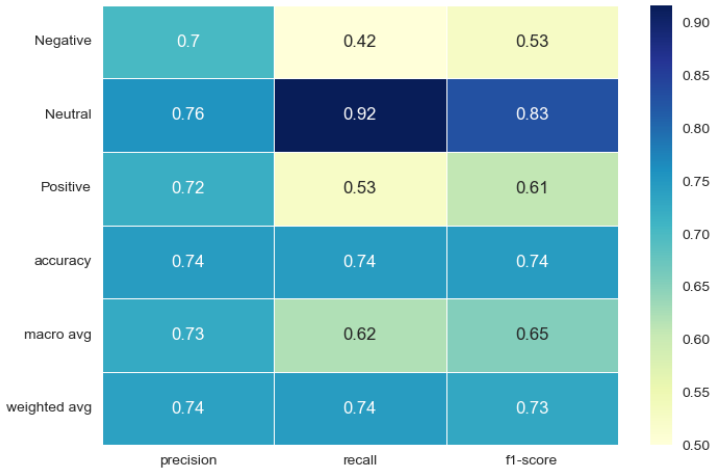


Figure 11: Random Forest Classification Report

4.2 SVM

As a next step, we tested SVM (Support Vector Machines), more specifically SVC (Support Vector Classification). We decided to apply nonlinear SVM in order to use the approach "one vs one" for classification. After performing tuning, the parameters used for this model are *'kernel' = Radial Basis Function (rbf) and C = 10.* Also, in this case, the results of classification are good, even better in terms of the 'negative' class, which can be seen in the following figure 12.

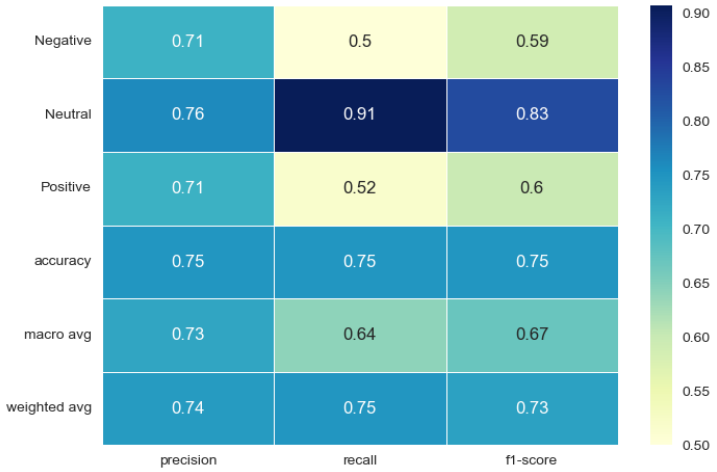


Figure 12: SVM Classification Report

In this case, even if the model performs well, the convergent trend of the two curves represented on the learning curve in the figure 13 shows that if we had more training data it would result in much higher scores.

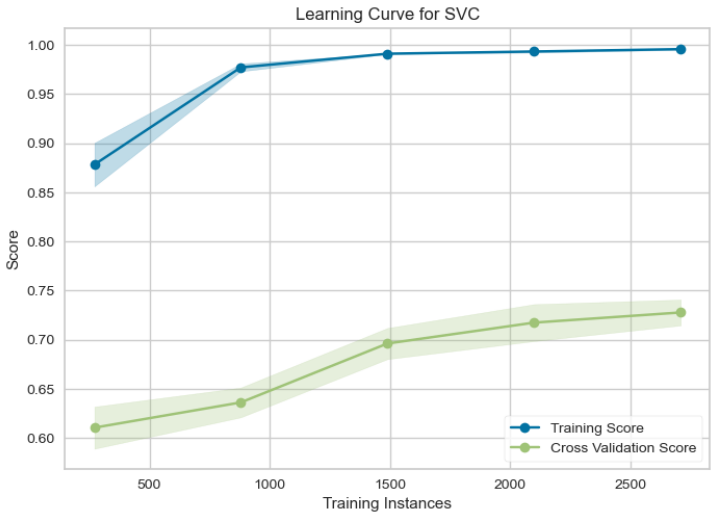


Figure 13: SVC Learning Curve

4.3 Bert

The next classifier that we applied on our dataset is from the BERT family. The architecture of BERT has been wildly successful in a variety of tasks in NLP. They compute vector-space representations of natural language that are suitable for use in deep-learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after. BERT models are usually pre-trained on a large corpus of text, then fine-tuned for specific tasks. It has to be emphasized that the computational complexity of this classifier in terms of time is really high. Therefore, it has been hard to run the algorithm several times. However, even during initial implementation the outcome has been very satisfactory, also considering other classifier performances. Figure 14 shows the classification report of the BERT model, and we can see a significant improvement compared to the previous results.

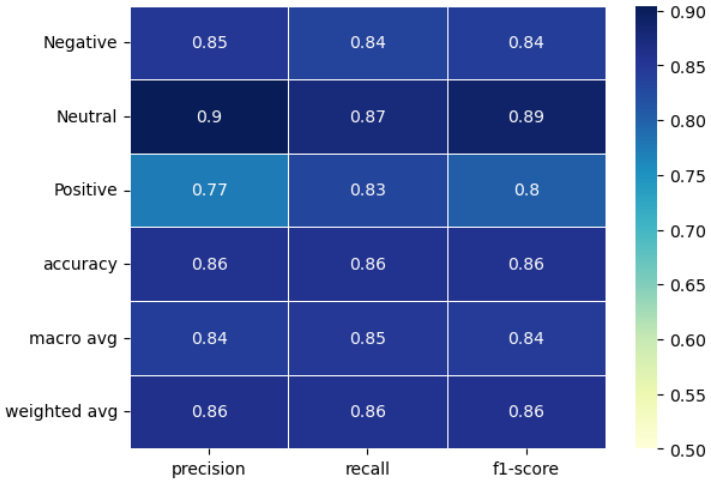


Figure 14: BERT Classification Report

We can say that the 'winner' classifier is BERT, as it gives very positive responses, especially for the minority classes, proving to be the best-performing algorithm in this task.

Even though under-sampling techniques have been applied to balance the data as discussed above, and all the models were tested on sampled data as well, the results did not show any improvement in terms of higher accuracy, therefore we decided not to report them in this paper. This is probably caused by the fact that all the models do not have enough training data. Therefore, all the outcomes reported above are the ones achieved from the original data.

Then we saw that BERT obtained results that are better than the other algorithms, this is probably caused by the fact that this one is made ad-hoc for text classification. There are at least two reasons why BERT is a powerful language model. First of all, it is pre-trained on unlabeled data extracted from Books Corpus, which has 800M words, and from Wikipedia, which has 2,500M words. Second, it is pre-trained by utilizing the bidirectional nature of the encoder stacks. This means that BERT learns information from a sequence of words not only from left to right but also from right to left.

4.4 Neural Network - LSTM

We choose LSTM because it is effective in memorizing important information. In LSTM we can use a multiple-word string to find out the class to which it belongs. This is very helpful while working with NLP. If we use appropriate layers of embedding and encoding in LSTM, the model will be able to find out the actual meaning in the input string and will give the most accurate output class.

We decide split the dataset into *training*, *validation*, and *test* set because we wanted to check if the network was overfitting.

	X shape	y shape
Train Set	(2541, 1131)	(2541, 3)
Validation Set	(847, 1131)	(847, 3)
Test Set	(1452, 1131)	(1452, 3)

Taking into consideration a pre-build network and modifying its various shapes, we choose this as our model:

Layer (type)	**Output Shape**	**Param #**
Embedding	(None, None, 32)	160000
Conv1D	(None, None, 32)	3104
MaxPooling1D	(None, None, 32)	0
Bidirectional	(None, 64)	16640
Dropout	(None, 64)	0
Dense	(None, 3)	195

Keras offers an *Embedding layer* that can be used for neural networks on text data. It requires the input data to be integer encoded so that each word is represented by a unique integer. This data preparation step was performed using the *CountVectorizer*. The Embedding layer is initialized with random weights and will learn an embedding for all of the words in the training dataset. As the last layer, we choose *Softmax* because the result could be interpreted as a probability distribution.

Testing the model with our unbalanced dataset we can see in figure 15 that the network was not overfitting so we didn't apply early stopping.
We also checked whether the network was overfitting after balancing, but again it was not.

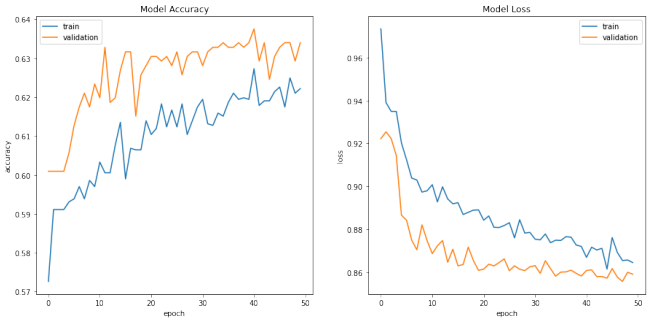


Figure 15: LSTM on unbalanced dataset

The following table shows the results obtained with both the balanced and unbalanced datasets. As we can see, they are nearly equal.

	Unbalanced Dataset	*Balanced* Dataset
Accuracy	0.6185	0.6185
Precision	0.6385	0.6485
Recall	0.5413	0.5413
F1-Score	0.5859	0.5901

In figure 16 we can see the errors of the model and what the model classifies without errors.
Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class.

Most members of the neutral class are correctly classified. However, the other two classes, have a higher level of misclassification probably due to the discrepancy in the number of sentences. Again, balancing the dataset did not lead to better results.

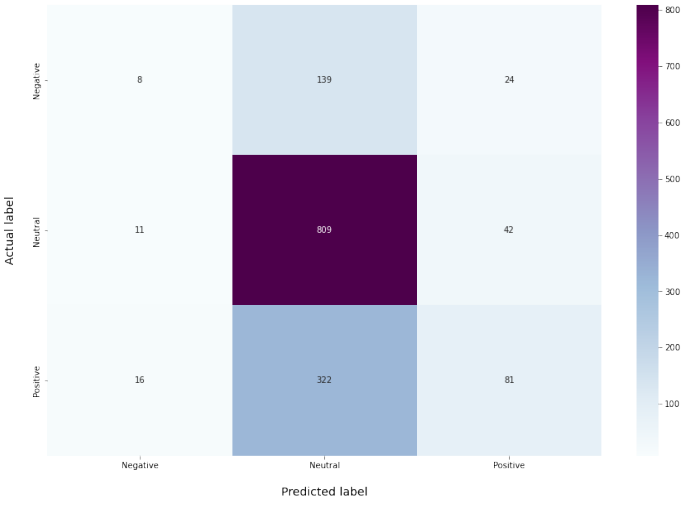


Figure 16: LSTM Confusion Matrix

5 Conclusion

Several aspects can be highlighted after this analysis of the Financial News dataset.

First, the results obtained after balancing the data were never better than the results obtained from the original one. This is probably because of the fact that all the models do not have enough training data.

Additionally, the fact that clustering technique was not really successful in learning the textual characteristics of our data. This is also understandable because of the type of dataset that we have

Finally, to conclude the work, we can say that the objective of finding the best classification model has been reached, as we announce our "winner" model to be BERT. It allowed us to label the unknown test data (previously unseen by the model) with an accuracy of 86%.

The following figure is produced on a negative class of test data already classified with the BERT model. We can easily notice the words such as *"decrease", "loss", "fell"*, that seem to appear frequently in these sentences.

Figure 17: Word Cloud of 'Negative' Class by BERT Model