PREDICTING STOCK MOVEMENTS WITH MACHINE LEARNING USING TEXTUAL DATA

MERYEM ÖZDEMİR

BOĞAZİÇİ UNIVERSITY

2020

PREDICTING STOCK MOVEMENTS WITH MACHINE LEARNING USING TEXTUAL DATA

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Meryem Özdemir

Boğaziçi University

2020

DECLARATION OF ORIGINALITY

I, Meryem Özdemir, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature .....

Date 21.09.2020

ABSTRACT

Predicting Stock Movements With Machine Learning

Using Textual Data

Economic events perceive great attention from information retrieval community. As one of the popular practices, language models on economy related textual data are proven to be advantageous for anticipating economic events. However, studies on Turkish stock market with textual sources are still limited as language models focus on popular languages. Fortunately, a significant step is taken on language models via the Transformer architecture, and its novel methodology widened the horizons of Natural Language Processing (NLP) studies for over 100 languages with the help of transfer learning. Ergo, in this study, it is aimed to incorporate both the latest advances and the traditional methods of NLP with machine learning classifiers to foresee the stock movements of the companies publicly traded in BIST market, using their official disclosures. To this end, 69,806 material events disclosures of BIST companies are fetched from Public Disclosure Platform (KAP) and labeled with stock movement directions. During the experiments, announcements are represented with Term Frequency Inverse Document Frequency (TFIDF) vectors and Bi-directional Encoder Representations for Transformers (BERT) embeddings so as to be classified with six different learners, namely Multinomial Naïve Bayes, Logistic Regression, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Categorical Boosting (CatBoost), and pre-trained classification layer of the Turkish case of BERT, namely BERTurk. While all setups yielded promising results, best performance is delivered by LightGBM on TFIDF with 39.7% F1-macro score.

ÖZET

Hisse Senedi Hareketlerinin Makine Öğrenmesi ile Metinsel Veri Kullanılarak Tahmin Edilmesi

Finansal konular veri bilimi dünyasından da büyük ilgi görmektedir. Bu alanda sıkça uygulanan ve ekonomik olayları tahmin etmedeki başarısı kanıtlanmış bilgi çıkarımı pratiklerinden biri de doğal dil işlemedir. Öte yandan, Türkiye ekonomisi üzerinde metinsel verilerin dahil edildiği çalışmaların sayısı hala kısıtlı bir seviyededir. Bu durumun sebeplerinden birinin dil işleme yöntemlerindeki gelişmelerin birkaç popüler dil üzerinde yoğunlaşması olduğu söylenebilir. Fakat son zamanlarda atılan önemli adımlar ile doğal diş işleme teknikleri görece daha az kullanılan diller için de büyük imkanlar sunmaktadır. Bu önemli adımlardan biri olan Transformatör tekniği gelişmiş mimarisi ve öğrenim aktarımı sayesinde son yıllarda farklı dillerdeki metinler üzerinde başarılı sonuçlara ulaşmıştır. Bundan hareketle, bu çalışmada, doğal dil işleme tekniklerinin hem geleneksel yöntemleri hem de en son gelişmeleri makine öğrenmesi yöntemleri ile bir arada kullanılarak BIST şirketlerinin hisse senedi hareketlerini tahmin etmeye yönelik deneyler yapılmıştır. Metinsel veri olarak, bu şirketlerin özel durum bildirimleri KAP'ın internet sitesinden toplanmış ve hisse senedi hareket yönleri ile sınıflandırılmıştır. Hem terim sıklığı/ters belge sıklığı, hem de Türkçe üzerinde ön eğitimli BERTurk modelinin kelime gömme vektörleri ile temsil edilen metinler üzerinde altı farklı sınıflandırıcı eğitilmiştir. Tüm modeller ümit vadeden sonuçları getirirken, Çok Terimli Naive Bayes, Lojistik Regresyon, XGBoost, LightGBM, CatBoost ve ön eğitimli BERTurk sınıflandırıcısı ile yapılan deneylerde en iyi sonuç %39.7 F1-makro skoru ile LightGBM algoritmasından elde edilmiştir.

ACKNOWLEDGEMENTS

I would like to express my gratitude to the people who helped and guided me through the process of this study.

First and foremost, I would like to thank my thesis advisor, Assist. Prof. Ahmet Onur Durahim for his precious guidance which was beyond essential to the study. I am deeply grateful that when I asked for his advisory to build up a remarkable work, he accepted and encouraged me for more. I hope this thesis to be an achievement worthy of the time he devoted to it.

Secondly, I would like to thank to Özcan Gündeş for sharing his knowledge with me and helping to achieve this study. Thanks to his kind and helpful attitude, I felt safe as I had a friend to counsel when I needed.

Also, I would like to express my gratitude to Prof. Ceylan Onay Şahin and Prof. Ali Rana Atılgan for being in my thesis defense committee. They devoted their time to assess this study and conveyed their valuable feedbacks. I am beyond honored that this study is endorsed by them.

Lastly, I would like to thank to my family for their endless support. Through this work, their support was beyond anything to me, as always. I once again understand, achievements are meaningful only when you have your supporters by your side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

API:	Application Programming Interface
BERT:	Bi-directional Encoder Representations for Transformers
BIST:	Borsa Istanbul
CatBoost:	Categorical Boosting
CNN:	Convolutional Neural Network
GRU:	Gated Recurrent Unit
KAP:	Public Disclosure Platform
LDA:	Latent Dirichlet Allocation
LightGBM:	Light Gradient Boosting Machine
LSTM:	Long Short-Term Memory
MI:	Mutual Information
MLP:	Multi-Layer Perceptron
NLP:	Natural Language Processing
OpenIE:	Open Information Extraction
RNN:	Recurrent Neural Network
SMOTE:	Synthetic Minority Over-sampling Technique
S&P:	Standard & Poor's
SVM:	Support Vector Machines
SVR:	Support Vector Regressor
TFIDF:	Term Frequency Inverse Document Frequency
XGBoost:	Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

Economic growth is the hottest topic of all times no matter what other emerging threats the world faces. This priority rules not only the governmental politics, but also every individual's life, with a never-ending force. Being such a critical issue for all levels of the society, economics and related issues persistently attracts not only corporate actors, but also academics from different fields of science. Psychology studies about consumer behaviors and choices on economic activities (Hursh & Roma, 2016), sociology papers investigating the relationship between disproportionate economic power and employee salaries (Wilmers, 2018), medical imaging experiments trying to reveal the brain activity during economic decision making (San Martín, Appelbaum, Huettel, & Woldorff, 2016), machine learning practices on historical data to anticipate economic downturns (Heiberger, 2018), and stock prediction methodologies using web text corpora (Yang, Xu, Ng, & Dong, 2019) are only a few examples from this extensive ground of study. Likewise, Sorto, Aasheim, and Wimmer (2017) mentions stock level prediction as an eye-catching research area, especially with the advance of the engineering practices on data. Even with a glance at the literature, their opinion can be assumed fully grounded, as a substantial effort is seen spent on stock prediction starting from decades ago (Kimoto, Asakawa, Yoda, & Takeoka, 1990; Mizuno, Kosaka, Yajima, & Komoda, 1998; Guresen, Kayakutlu, & Daim, 2011; Hiew et al., 2019).

Amidst all the exertion invested on stock forecasting task, predictability is still a matter of debate. While the stock price is a reflection of traders' belief of the future value of the company, rationality of this belief is severely contradictory. As one of the main

ideas on this debate, proposed by Fama (1965), Efficient Market Hypothesis is based upon the fair market prices which are the result of all available information related to the market. Li et al. (2014) stated that the Efficient Market Hypothesis, with its weak form of efficiency, has motivated many academic researchers for trying to predict securities with the help of companies' fundamentals. On the other hand, as mentioned by Li et al. (2014), some of them were not able to predict the prices based on quantitative information and the assumption of traders' unemotional decision making, since traders cannot be thought as fully rational. This emotional side of traders brings another point of view to the effort on stock market prediction. For over two decades, researchers have been utilizing textual information that is spread through web by different resources and proven to have influence on investors' emotions, to predict the market movements (Wüthrich et al., 1998; Gidofalvi & Elkan, 2001; Shynkevich, McGinnity, Coleman, & Belatreche, 2015). Among these researchers, Li, Chen, Jiang, Li, and Chen (2016) endorsed textual information as an essential part of predictive studies, especially with the advance in internet and its diffusing power over society.

As Hirschberg and Manning (2015) stated in their review, after years of progress, studies on NLP has evolved from statistical and structure-independent techniques to learning frameworks with semantic and syntactic dimensions. This evolution can also be seen in stock prediction studies which have a wide portfolio of methods differing from each other with their language representations, data pipelines, predictive algorithms, and outputs. However, the most common side among all is being able to obtain better predictions with the help of their NLP approaches.

In this study, a text classification approach is built upon NLP to anticipate BIST stock market using financial announcements of related companies. As the stock market

of a prospering economy, BIST is studied and discussed in many academic papers by means of performance prediction at both index and stock level (Gunduz, Yaslan, & Cataltepe, 2017). However, among these valuable studies, methods that incorporate textual data and employ NLP techniques are still limited, compared to foreign markets with English textual data sources. To address this relatively untouched domain of study with the help of the latest advances in NLP, this thesis presents a study in which a series of classification experiments carried out using both the deep learning approach of BERT and TFIDF vectors for text representation. Upon representations of the text pieces, several machine learning classifiers are employed with the aim of anticipating BIST securities' daily price movements.

Proposed by Devlin, Chang, Lee, and Toutanova (2018) from Google Research Team, BERT is a language model which uses transformer-encoder structure with language masking methodology. Being able to adapt to different tasks and languages, BERT is a sight for NLP tasks on relatively unpopular languages, such as Turkish. That being the case, this study aims first and foremost to exploit the BERTurk model, on public disclosures in Turkish language so as to predict the stock directions of BIST companies, with the help of transfer learning. To this end, 69,806 announcements declared by BIST companies during 2015-2019 are gathered from KAP's website. These announcements are not only classified using BERTurk model's deep architecture with pre-trained embedding and sequence classification layers, but also processed using bag of words approach with TFIDF scores so as to be classified through mainstream machine learning models for comparison.

The study consists of six main sections in the succeeding pages. Firstly, in Literature Review, previous works exerted on NLP and stock prediction are surveyed

with their key findings. Then, the second part defines the methodology and clarifies the predictive pipeline of the experiments, comprehensively. Afterwards, the Results section displays the outcomes of all prediction pipelines stated in the previous part, with the key performance criteria. While detailed comparisons and key findings are discussed in the Conclusion part, constraints on proposed study and ideas for future work are presented in the last section.

CHAPTER 2

LITERATURE REVIEW

2.1 Text classification

The literature on knowledge discovery widens perpetually, owing to the ever-growing data which continuously streams from all kind of sources in all kind of forms. Amid these forms, history of textual data began at late 1940s with researches on machine translation, then enriched by many valuable studies in the following years (Jones, 1994). In one of early examples of text classification studies, Cohen (1996) compared two learning methodologies to classify e-mails into several categories using their textual parts with TFIDF weights. The study concluded in an efficient learning process and a promising performance on e-mail categorization with an error rate around 6%. TFIDF is a way of feature weighting which serves to textual knowledge discovery studies since the early stages. Along with on bag of words approach which considers text pieces as group of tokens, TFIDF vectors highlight the words that are not only frequent in the document being processed, but also rare in terms of occurrence in different documents. However, bag of words technique treats text bodies as groups of n-grams, that is, structural or syntactic features of the language are not considered while representing the text pieces. Despite, being grounded on occurrence statistics, TFIDF is proven to be an efficient and effective style of weighting features by many studies exerted on diverse machine learning tasks. Joachims (1996) used TFIDF weights in a probabilistic manner and compared it to the classic TFIDF with a Naive Bayes classifier. On the task of classifying Reuters news into pre-defined categories, the study achieved its best result with the proposed method yielding 90.3% of accuracy. Years after, in the study of

Abualigah and Khader (2017), TFIDF was again on stage, this time for clustering news articles into meaningful groups along with K-means algorithm. Together with proposed feature selection methodology, the paper presented a successful performance in conclusion of the experiments held on eight different news datasets.

Yilmaz and Abul (2018) utilized TFIDF weights in their study with the aim of deducing public opinion through messages posted on Twitter around the time of a political election in Turkey. They trained three different classifiers (Support Vector Machines (SVM), Random Forest, and Decision Tree) on top of several feature sets. Among all settings, SVM with linear kernel on top of Latent Dirichlet Allocation (LDA) came out as the winner with an accuracy level of 89.9%.

On one hand, this simple and efficient method of TFIDF weighting of n-grams is still popular among NLP communities and yields significant performance on variety of tasks. On the other hand, as being negligent about syntactic and semantic dimensions of the text pieces, statistical methods are not alone on the stage as the advances in NLP introduced new methods to understand the text without omitting the essence of the language. One of these methods is word embedding concept, whose foundations was firstly spoken at 1954 by Harris (1954) based on the idea of revealing the contextual relationship through the co-occurrence patterns (as cited in Levy and Goldberg, 2014, p. 1). As an example, Savigny and Purwarianti (2017) studied on a classification problem on YouTube comments in Indonesian language and compared word embeddings to a baseline model with TFIDF vectors. Proposed methodology which incorporates word embeddings with a Convolutional Neural Network (CNN) model achieved 76.2% classification accuracy and beat the SVM model on TFIDF which yielded 74.1% accuracy.

Aydoğan and Karci (2020) presented a Turkish text classification study with word embeddings which were trained on a huge corpus crawled from Turkish Wikipedia website. Their main purpose was incorporating Turkish word embeddings with powerful deep learning frameworks so as to perform an automated labeling on a secondary dataset consisting of consumer feedbacks with 10 different categories. To this end, the study experimented with six different deep learning frameworks (CNN, Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Bi-directional LSTM) along with three different word embedding methods (Word2Vec with continuous bag of words, Word2Vec with Skip-gram, and GloVe). While pre-trained word embeddings proved themselves with additional performance they brought in all combinations, the winner of the three word embedding methodologies was Word2Vec with continuous bag of words almost with all deep learning approaches. Among all settings, GRU on Word2Vec with continuous bag of words presented as the most successful classification pipeline with a validation accuracy of 85.82%.

Aside from the task specific text categorizations in literature, such as labeling consumer comments' categories, splitting news into main topics, sentiment analysis is another popular text classification concept which tries to understand the emotional polarity of text pieces. Literature on sentiment analysis has several methodologies that exploit the text in different manners, such as dictionary-based classification, statistical methods, and classification with word embeddings. Martineau and Finin (2009) utilized TFIDF vectors in a different course which they claimed to be better at differentiating word distributions between the two opposite sentiment labels. They created weight vectors for negative and positive comments separately, then subtracted one from the other to get the delta. After the experiments held on two different datasets, results

proved that they were right at their claims, as the Delta-TFIDF method achieved 88.1% accuracy which was higher than the two baseline models depending on word count and classic TFIDF with 84.65% and 82.85% accuracies respectively.

In their study, Wang, Liu, Luo, and Wang (2018) proposed a deep learning framework to label short text pieces gathered from online channels with their sentiment polarity. After being evaluated on three different review datasets, their LSTM model applied on top of Word2Vec Skip-gram text representation outperformed its rivals (Naive Bayes, Extreme Learning Machine) in most of the experiments in terms of F-measure. Moreover, Wang et al. (2018) observed that the performance increases as the training size increases, with all deep learning methods.

Catal and Nangir (2017) presented a sentiment classification approach which utilized three different learners with majority voting. They evaluated their method on three different Turkish datasets including book, movie, and product reviews. Experiments on all datasets showed that the proposed multi-model learner resulted in a better accuracy than the individual performances of the three voters, namely Naive Bayes, SVM, and Bagging.

2.2 Stock prediction with NLP

Amid countless valuable information retrieval studies on textual data, finance is one of the most popular domains with many researches exploiting all levels of NLP advances to deduce meaning from related text sources. In the following lines, indicative papers that benefit from NLP for finance related tasks will be discussed with their key points and main contributions to the literature, along with their experimental results.

Researchers have considered the news articles as the main sources of economic events and sought for the best employment of many powerful estimators on them. Although impressive progress made in NLP techniques in time was their best booster, early attempts with simple configurations were also yielded successful results. Even in recent studies, simplicity of early methods is still preferred because of the computational efficiency and the ease of use. Bag of words representation is one of these old-but-gold methods which shapes textual data into a model friendly form within prediction pipelines which can be highly diverse in terms of complexity, such as plain statistical models and multi-layer deep learning frameworks.

Lavrenko et al. (2000) designed a system to discover the relation between historical data of economic indicators and news stories. The system used price history to retrieve the stock trends which were thought as important to traders' decisions. Historical data included stock prices with intervals of 10 minutes, and trends were retrieved via piecewise linear fitting over the time series. These trends were then associated with the news articles which were released before the trend within the specified amount of time. The goal of this association was to build a model that learns the words related to specific trends such as increasing, decreasing, and straight. This brought out the ability to anticipate future trends with news articles in hand. Classification of the documents according to their relationship with trend types were done with Naive Bayes classifier, with bag of words representation and the assumption of word independence. The model used 127 different stocks, and 38,000 news articles which were related at least one of those stocks. While the evaluation of the classification was done using detection error tradeoff, whole system was subjected to a 40-day trading simulation to see if the model gave better earnings than a random strategy. After the experiments, Lavrenko et al.

(2000) observed that over 1000 times of random trading trials, only eight of them were able to beat the designed model in terms of the return on investment. Meaning a significance level of 1%, the model was proven to be useful to trading strategy. The study also put effort on time delay of news influence on stock market by experimenting over news time window ranging from simultaneous to 10 hours. Results showed that while simultaneous matching of the news and price trends gave the highest return, its limiting effect on training set was undesirable. After simultaneous timing, the second-best result was obtained on 10 hours window. Another discussion made on the paper was about the coverage of the stock prediction model. Also, Lavrenko et al. (2000) mentioned about global and local model types which worked on the dataset as a whole or uses stock specific relations to learn language models. As both approaches gave promising results on return on investment, they suggested that a mixture of them would be the best solution.

Schumaker, Zhang, Huang, and Chen (2012) created a trading machine named Arizona Financial Text with the motivation of providing additional prediction ability from subjective news articles. Their machine forecasted stock prices using both numerical features and news text bodies 20 minutes after the release of a news article. Using bag of words representation with binary vectors, Support Vector Regressor (SVR) model predicted price levels of the stocks. While the examinations proved that the polarity of news brings out 9.5 points of additional accuracy, news with negative tones found to be more effective on prediction process.

Li, Huang, Deng, and Zhu (2014) are another group of researchers who worked on stock prediction problem using news data with bag of words representation. Unlike many other papers written on the same subject, their model aimed to predict the stock

price level in short term, instead of predicting price movement direction. In addition to the historical prices of the related stocks, proposed system used top 1000 words from the news text corpus having the highest Chi-square value with their TFIDF scores. These two types of data were processed through two different sub-kernels so as to be combined based on the weights of their importance for predicting the stock price. Experiments were done with SVR on four different set up: news data only, financial indicators only, naive combination of news and financial data, and multi kernel processing of news and financial data. Experiments showed that using Multiple Kernel Learning gave the highest performance among all.

After baseline models have been developed on stock market prediction with textual data, further enhancements concentrated on both algorithmic structure and data usage design. An obvious example of a bettered data design is the study conducted by Shynkevich, McGinnity, Coleman, and Belatreche (2015), which aimed to predict the price levels of the stock market by combining the sector specific news with the firm specific ones using Multiple Kernel Learning. The feature set of this study was represented with bag of words technique and refined according to the Chi-square values, then shaped as a vector composed of TFIDF weights. Proposed method was tested with different combinations of kernels (polynomial, linear, gaussian), and compared to SVM and K-Nearest Neighbor methods. In terms of accuracy, Shynkevich et al. (2015) achieved to beat other two algorithms with their Multiple Kernel Learning for all stocks.

Li et al. (2016) brought another issue into discussion for stock prediction with textual data: speed. Their model considered not only the accuracy but also the pace of the prediction model which has a significant role on intra-day stock orders. In addition to the historical prices and technical indicators, news data was used with TFIDF vectors

which included top 1,000 words with the highest Chi-square value. Pre-processed data was committed to an Extreme Learning Model which is basically a single layer neural network providing extra speed than fully connected multi-layer networks. To measure the effect of the proposed methodology, comparison experiments were also included SVM and Back Propagation Neural Network models. Results showed that while SVM had the highest accuracy for most of the cases, Extreme Learning Model yielded the second-best results in terms of accuracy, and the best result in terms of speed.

In most of the studies carried out on stock market prediction using news articles, news corpora are composed of firm and sector related statements. That approach, deliberately or not, pretends that individual stocks or stock markets are closed systems and are not influenced by other macro events, or treats them like they are exposed to the same affect equally by subtracting a composite indicator's level from the individual prices. Verma, Dey, and Meisheri (2017) proposed their solution to this, as they included six different news categories in their system. Their system was distinguished from others also with its causal relation measurements between the news categories and the stock movements and the time windows used for predictions. They experimented with 1-day, 2-day, 1-week (5-day) time windows to measure the delays of different news categories' effects. Using LSTM network, their predictions on price movement of Indian market outperformed SVM in terms of accuracy and Matthews correlation coefficient. Other than that, they revealed that different categories of events had varying levels of influence on different sectors, and, while legal news affected immediately, political events had a lag on influencing the stock market.

Nam and Seong (2019) wrote a paper on stock market prediction with a complex setup which took sectoral relations into account. Their bet was on predictive power of

combining news belonging to other companies that were related with the target stock's news at different levels. As another distinctive side of their study, they did not assume a basic bi-directional interaction between the companies operating in the same sector. Instead, they learned the complicated relationship via Transfer Entropy, then learned related firms' contribution to the target firm's prediction via Multiple Kernel Learning. Studies on market prediction focuses on weakly efficient markets as they are influenced by information resources. Nam & Seong (2019) used Hurst Exponent to decide if the market was in the weak form of efficiency or it was mainly driven by the historical price levels. While greater Hurst Exponent value means stronger market, they worked on Korean market which had nearly 0.5 Hurst Exponent indicating weak form of efficiency. Financial news of Korean market was represented as bag of words, and feature selection done by Chi-square technique. Then resulting set of features weighted with their TFIDF scores, so as to be fed into Multiple Kernel Learning model. Their predictive design achieved successful results in terms of accuracy and F-measure at all sectors, namely pharmacy, material, and food expenses.

In addition to news articles, financial reports announced by the companies are also examined for their contribution to the stock prediction task. Lee and Suh (2018) studied to predict the stock returns of Standard & Poor's (S&P) 500 companies and utilized 10Q reports which are necessarily filed by publicly traded United States companies for each of the first three quarters of the year. Their configuration included bag of words representation of the documents with TFIDF weighting, and a multilayer deep neural network to predict the return after the announcement, within different time frames such as 30-, 60-, 90-days. Their use of data differed from other studies as they included only the words that did not appear on the previous filings of the related

companies. After the experiments, their two-class classification problem gave its best area under curve value as 54.94 with the 60-day framework. Class precisions of the same time frame reported as 27.0% for up class which constituted 25% of the dataset, and 81.0% for down class which constituted 75% of the dataset.

Sakarwala and Tanaydin (2019) incorporated numeric financial performance indicators and textual data from earnings releases of the S&P500 companies to predict the stock price movements within the following days. They represented the textual parts of the releases with GloVe vectors so as to utilize them as input for four different deep learning architectures, namely Multi-Layer Perceptron (MLP), CNN, RNN, CNN-RNN. Their experiments resulted in 68% accuracy at its best, with CNN-RNN model which was also reported as the best in terms of generalization.

Kraus and Feuerriegel (2017) compared deep learning frameworks with traditional machine learning algorithms for anticipating stock performances using announcements declared by German companies. Their study contained both the magnitude and the direction of the stock as the target of the prediction task. Text representation was carried out in three different ways: bag of words with TFIDF weights, bag of words with binary representation of occurrences, and word embeddings which were created with transfer learning. While their experiments showed that deep learning methods performed better than the traditional machine learning methods, word embeddings increased the model performance on both regression and classification tasks. Also, their best configuration for both tasks was the LSTM model with pre-trained word embeddings which achieved 57.8 accuracy on classification and 3.104 mean absolute error on regression.

Sentiment analysis, which detects the mood of a piece of text, is a typical discussion when it comes to using textual data in machine learning experiments. Even so, studies which treats sentiment as a domain-specific dimension are still limited. While a word may express completely opposite sentiments even in the same domain, assuming some dictionary of plain words as a one-fits-all sentiment bible may disappoint the audience. Stock prediction studies have already adopted both styles and some of them are mentioned in the following lines.

News data has been subjected to lots of scientific papers for its power on predicting stock market's future, likewise Li, Xie, Chen, Wang, and Deng (2014)'s study. Their approach differentiated from others by their sentiment analysis methodology with a special sentiment category set to represent each word in a text as a vector composing of sentiment values. This special sentiment category set, named as Harvard IV-4 psychological dictionary, included fifteen categories for general purpose classification of words into sentiments. They utilized word vectors as inputs of an SVM model, together with the historical prices to predict stock direction of the next day. Instead of analyzing the sentiment factor with bag of words technique as positive or negative, preferring a more granular dictionary brought out them an additional accuracy on their prediction task. However, researchers admit that the Loughran-McDonald categories was better at results as being not general purpose but specific to the financial glossary.

Sorto et al. (2017) studied on a stock prediction system with the belief of the influence of media's mood on investor decisions and trading strategies. News articles were summarized and analyzed for their sentiments with SenticNet framework in Python. These sentiments were then used for scoring each stock on a daily basis for their

sentiment polarity and used as input to a Logistic Regression model. Their experimental design was based on comparison of titles and summarized texts of news articles and showed that title usage resulted in slightly better performance in terms of accurately classified days and ROC curve values.

Khedr, Salama, and Yaseen (2017) studied on a prediction model which used sentiment of financial news to predict the movement of the companies' securities. Their system was built on two main classification steps: detecting article sentiment based on TFIDF scores via Naive Bayes classification algorithm, then, tagging the stock for upward/downward direction via K-Nearest Neighbor classification algorithm using textual and numeric data. With experiments for sentiment classification, Naive Bayes was found to be the best sentiment predictor compared to SVM and K-Nearest Neighbor.

Liu, Lu, and Du (2019) distinguished from others by considering relations between companies, which was detected through news articles, when predicting stock market. Relationship types were listed as shareholding, collaboration, administration, and supply-customer, and explored via an enterprise knowledge graph. Then, for each stock, daily sentiment vectors were calculated with the emotion scores of words. With fundamental ratios of companies, relationship matrices and news vectors were fed into a GRU to produce final predictions. Compared to the configuration without company correlations, proposed method showed a significant improvement with additional 16.1 points of accuracy.

Although economic events and any other field that has an effect on it widely and immediately covered on news, public opinion on these events cannot be captured by the articles. Since it levels the playing field for public to voice their attitude towards matters,

social media data is being widely utilized by all means in scientific researches. As such, stock market researchers have its own way to mine it.

Si et al. (2013) saw social networks as a great pool of messages that were posted by individuals to depict their attitude over a wide range of issues. They used Twitter posts with the symbols of S&P100 stocks to predict index's future direction. To this end, tweets were analyzed with Dirichlet Process Mixture model, and then sentiment scores of resulting topic labels were calculated to get the aspect-based sentiments. Sentiment scores were calculated with a vocabulary of opinion words with their binary labels. Resulting model was compared to two different approaches; first one was a predictive model which used time series of prices, and the second one was an approach considering both the time series information and the overall sentiment score of the tweets. After experiments with different time windows for inclusion of the tweets, Si et al. (2013) proved that topic-based sentiment analysis with 3-day lag beat the two other baselines with a significant improvement.

Makrehchi, Shah, and Liao (2013) proposed a predictive model for stock market, based on tweet sentiments, and compared two different sentiment detection approaches in their study: lexicon-based sentiment analysis, supervised sentiment classification. It was stated that the first one was much easier to use as there was no need for a labeled dataset to train a model. Whereas, supervised models were mentioned as outperforming to lexicon-based methods. To overcome the lacking labeled data issue, researchers designed an automated labeling process which relates tweets to decreasing and increasing stock movements, then, labels their sentiments accordingly. Also, another indicator of the tweet sentiment was considered to be the statements of companies about their financial performance that exceeded the forecasted level. With this automatically

labeled data, Rocchio classifier was used for sentiment classification. After combining tweet sentiment scores to get the overall sentiment of the day, next day's price movement was predicted at both index and individual stock level. Makrehchi et al. (2013) compared their system to lexicon-based sentiment and historical price-based predictions and found their model as outperforming to both of them.

Si et al. (2014) were one of those who analyzed and utilized social network messages to unveil the relationship between the investors' emotional expressions and the security market. Their way distinguished from others because of the network architecture they build between stock symbols mentioned in Twitter messages. Named as Semantic Stock Network, their network graph was built through cooccurrences of the stock symbols and used for unveiling the hidden link between them. Their design detected topics of edges between the nodes with labeled LDA, then calculated a score of sentiment using a dictionary to be used in a Vector Auto Regression model. Their work was reported with promising results in terms of accuracy of predicted directions.

Nguyen, Shirai, and Velcin (2015) aimed to discover the relationship between stock market deviations and people's mood stated through finance discussion board messages, with a distinguishing method by its simultaneous style for topic and sentiment detection. They examined the stock movement prediction problem with six different approaches which had particular configurations in terms of data pipeline but using the same model of SVM. Essentially, those configurations were for testifying not only the contribution of sentiment extraction, but also the way it was extracted for the accuracy of prediction. While the baseline model was using only historical prices, others were extracting topics and sentiments of the messages with distinct ways, such as sentiment lexicon or supervised classification techniques. On the other hand, proposed method's

claim was about simultaneous extraction of sentences' topics and sentiments with a multilabel output, through Joint Topic/Sentiment model. Joint Topic/Sentiment model, which is an extension of LDA, uses a weakly supervised operation based on a sentiment glossary and known for its domain independent prediction power. However, while the highest accuracy resulted from the aspect-based sentiment method which depended on Stanford CoreNLP for topic and SentiWordNet for sentiment, Joint Topic/Sentiment based model was beaten by all other configurations, even by the price only design. After that, Nguyen and Shirai (2015) published another paper to enhance their former method and reported that they achieved better results than both the former setup and price only setup. Named as Topic Sentiment LDA, unlike the Joint Topic/Sentiment model, their new method was assuming that one sentence might have only one topic, and sentiment word distribution of each topic differ from each other. While this time proposed method overperformed other designs with a serious improvement on accuracy, having nearly same results with the price only, LDA based, and Joint Topic/Sentiment based configurations may cause doubt on the health of the experimental design.

Chen, Yeo, Lau, and Lee. (2018) conducted a study which predicts stock movement based on selective data sources, by utilizing social media posts only if it was shared by verified accounts. Their intent was focusing on only the influential texts and ignoring other noises which were not only useless for prediction power, but also may trouble it. Their approach included social media messages with topic-based sentiments calculated through LDA and a specific sentiment glossary, and numerical features of companies. Using GRU and boosting method together, they raised the concentration on wrongly predicted samples on each prediction. Their experiments showed that boosting method enhanced the predictive power of RNN, and whole configuration was in a

competitive position compared to other methods, such as Artificial Neural Network, MLP, SVR.

Li et al. (2014) studied on a prediction framework, namely Media-Aware Quantitative Trader, which drew its strength from web textual data including news and social discussion messages. Their framework tried to examine each news article by extracting significant words via part of speech tagger, then combine it with public sentiment that was deduced from discussion forum messages, based on a finance specific sentiment dictionary. While textual data was being represented by bag of words technique and weighted with TFIDF scores, its effect on prediction was tuned as to fade out as time passed. On the other hand, sentiment dictionary was built with a conditional probability calculation, with the assumption of that negative news were followed by a decline in securities, while positive ones were followed by rising. Their experiments proved the contribution of using textual information in addition to numeric features and found that negative sentiment had more predictive power than the positive. At the end, their SVR model predicted stock level prices with a 26 min lag and found to be useful for trading strategy with 166.1% return in three months.

Li et al. (2016) concentrated on identification of particular contributions of different information sources to stock prediction problem. They used both public mood and news articles, in addition to the fundamental features of the securities. To deal with these three sources, they set up a tensor-based regression model which revealed each one's effect on stock price individually. Textual data was represented as bag of words and TFIDF values, and sentiment was detected through financial sentiment dictionary and statistical method. After deciding empirically on a 26-minutes time window, they compared their model to a list of powerful methods: Classification and Regression Tree,

Back Propagation Neural Network, SVR with and without Principal Component Analysis. Experiments concluded that while tensor-based framework gave the best results on directional accuracy, SVR model was the winner of the regression task.

Early attempts of natural language models were mostly based on bag of words, which processes text bodies or sentences as groups of word stems. That was a clear and computationally effective way of text representation, still, using this approach brought its own drawback by preferring simplicity to language structure. Then, the next target for the researchers was to find a way of representing text data without missing its structure which is severely important to not to lose the meaning. Event extraction approach was one of those ways which detects word groups composed of actors, actions, objects and mentioning about special matters. These events were believed to be the essence of the sentences, so the main message of the text could be understood only by extracting them. However, labeling events in a text body requires tons of labeled training data, or advanced techniques must be applied to overcome this requirement of supervision. As one of those techniques which deals with labeled data issue, Banko, Cafarella, Soderland, Broadhead, and Etzioni (2007) introduced a ground-breaking study on event extraction with their Open Information Extraction (OpenIE) system trained on 9,000,000 text pieces from web to learn entities and their relations. As OpenIE offers a domain independent frame, it is exploited by many researchers from different areas and resulted in successful papers.

Chen, Xu, Liu, Zeng, and Zhao (2015) addressed two different issues about event extraction from text data: lexical properties, and sentence structure properties. Their effort was specially exerted on automatic extraction of the meaning of a sentence without missing any part of it. Although literature has a number of studies using

convolutional layers in language processing, Chen et al. (2015) emphasized that the common practice had a handicap on capturing the whole picture because of the max pooling practice. With this problem at hand, they proposed a dynamic multi-pooling CNN, which classified a sentence in two consecutive steps: classification of each word for trigger/non-trigger labels, classification of the trigger words for event arguments. The second step involved an unsupervised subprocess which learned the semantic representations of the words and produces embedding vectors with skip-gram model, then used it for argument classification. Working on embedding vectors, their CNN model did max pooling over different regions of the sentences to reveal the whole meaning. After experimenting on a common news text corpus, their approach was reported as outperformer to the former baseline models in terms of F-measure. After that, Chen, Liu, Zhang, Liu, and Zhao (2017) enhanced their model with automatically labeled training data constructed with distant supervision, and their results was competitive.

Ding, Zhang, Liu, and Duan (2014) suggested a model which handled news text data to predict stock price direction, by extracting structured events using OpenIE methodology to resolve the costly problem of labeled data of event types. Resulting tuples of event extraction step were composed of subject, action, object, and the time of action so as to relate them with the stock price data. Extracted events were then converted into their general versions to minimize the disadvantage coming from the sparse nature of event types. Ding et al. (2014) experimented two different modelling methods on two different data representation techniques: TFIDF scores and events for text representations, linear model of SVM and a nonlinear deep neural network for predictions. Those experiments had also a time dimension for the interval between the

news and related stock prices as one day, one week, and one month. Evaluation was done using two different measures: accuracy as a general indicator of predictive performance, and Matthews Coefficient Correlation as a common practice for stock price prediction researches in literature. After all experiments, it was observed that regardless of the modelling approach, event extraction results in better performance than the bag of words representation. Additionally, researchers noted that the reaction of the market to the news were captured best within one day interval.

After their paper in 2014, Ding, Zhang, Liu, and Duan (2015) proposed another recipe to fix the drawback at their first study coming from the sparse nature of the event tuples. This time, their model depended on a deep neural tensor network which was trained to learn event embeddings that located events with similar meaning closer to each other. Resulting vectors then used as three different groups according to their time spans: short-term (1 day), mid-term (1 week), long-term (1 month). Before all were fed into a feed forward neural network, while the short-term event embeddings were directly used, others were processed through convolutional filters. Their performance criteria were again the same, and their results proven the contribution of event embeddings with CNN filters with a significant gain on accuracy.

Huang, Ji, Cho, and Voss (2017) tried to use transfer learning method to deal with the labeled data problem of event extraction tasks and developed a new framework to classify unobserved events using their semantic distances to observed ones. They benefited from available event dictionaries to get the observed event types, such as FrameNet, then they adapted Zero Shot Learning to event tagging task. Possible events detected by Abstract Meaning Representation, then processed through CNN filters to

form the event tuples. Compared to LSTM network, their configuration did not bring a winning F-measure, but still has promising scores.

Zhang, Qu, Huang, Fang, and Yu (2018) utilized both financial news and social network messages in addition to historical prices, to predict stock market's ups and downs. Their Multiple Instance Learning framework stated as a useful tool to harmonize different types of data together, as it measured relative importance of each source particularly. News data was used with events' dense representations generated via Sentence2Vec, and social media messages were used with their topic-based sentiment labels extracted via LDA and a sentiment dictionary. With an extensive set of experiments, their method reported as the winner by means of F-measure.

Oncharoen and Vateekul (2018) proposed a method to predict stock market behavior using text and numeric data, and applied word and event embeddings with deep neural network in their configuration. Text context was subtracted from news titles and processed via OpenIE framework to extract event tuples, which were then transformed into embedded words using pre-trained GloVe model. After obtaining embedded representations of words, next step was training a neural tensor network to get event embeddings. Resulting event vectors were summarized as three different groups: short-term (1 day), mid-term (1 week), long-term (1 month). While long-term and mid-term vectors were subjected to a CNN layer with max pooling, short-term vectors directly participated in a fully connected layer with other two groups. On the other hand, numerical features were processed with LSTM as time series data, then both parts were activated via softmax layers. Outputs of the softmax layers of text and numeric data were then merged into a vector so as to be fed into the final layer of prediction pipeline. Proposed method was tested on three different data set, and compared to different

combinations of input data, such as numeric-only, text-only, event-only, text and numeric. Both directional accuracy and investment return declared the proposed method as the winner at overall performance.

Minh, Sadeghi-Niaraki, Huy, Min, and Moon (2018) studied on a deep learning model that used numerical features and financial news to predict the directional movement of the stock market and examined it on different time windows. Their key contributions to the literature were summarized in three bullets: a two-stream GRU model, stock2vec embedding approach, time dimension of the experiments. Stock2vec is a special embedding model trained on word embeddings and Harvard IV-4 sentiment lexicon, to represent each stock with a vector. Using those vectors with financial indicators as input to the two-stream GRU model, textual effect was learned bi-directionally and achieved nine points of additional accuracy compared to GRU model. Comparisons with former studies in literature shows that two-stream GRU was an outperformer, even to the other RNN based methodologies. Other key findings were that financial indicators have a notable contribution on predictions, and the model had its best performance at one-day interval prediction.

Deep Learning community has been introduced to the new gear of machine learning by Bahdanau, Cho, and Bengio (2014), which brought a substantial improvement to the discipline and named as attention mechanism. Attention mechanism took a solid step forward in the way of imitating human intellectual by concentrating on the essential content while deducing the meaning. This way of learning made its reputation through language and image processing as they are the most famous ones when it comes to being rich in input data. Stock prediction papers have studies of attention-based language processing models which benefits from the attention approach.

Hu, Liu, Bian, Liu, and Liu (2018) pointed out to not only the contribution of textual data to market prediction, but also the quality issues that can be faced when web sourced data is used. Their model was designed for mimicking human decision-making process exactly: distinguishing the quality, paying attention to different sources with different amounts, and being able to decide correctly even when one of the information resources was missing. While quality distinction and attention weighting carried out through Hybrid Attention Networks, Self-Paced Learning was used for focusing on informative training instances instead of missing parts. Their input set consisted of two main parts: financial news represented as word embedding vectors, and stock prices. To decide news' importance level, an attention layer was applied on days' news vectors, then summarized accordingly for each day. Afterwards, summarized versions were fed into bi-directional GRU. Bi-directional GRU modeled the sequence of events considering both the past and the future, then given its output to another attention layer to discriminate importance level between days. At the end, processed data fed into an MLP model with weighted instances to classify stocks' directions as up, down, preserve. They constructed an experiment to prove their method, and compared it to Random Forest, MLP (without attention layers and Self-Paced Learning), and RNN based models. Both for accuracy and investment return, proposed method demonstrated an outstanding performance among other methods.

Liu, Cheng, Su, and Zhu (2018) implemented a successful method of attention network which processed news body texts to predict whether the market will rise or fall. After creating embeddings via bi-directional GRU, their design included two hierarchical steps: word level attention to capture the essential words of the sentence, sentence level attention to capture the essential sentences of the news article. Then,

summarized version of the textual data was turned into a prediction score, by the help of a dense layer and a softmax output layer. Their method successfully passed their examination with 61.38% accuracy and beaten its competitors.

NLP is being refined on its existing approaches, while at the same time moving forward with brand new techniques perpetually. However, as Hirschberg and Manning (2015) pointed out in their review, most of these advances are capable on a limited number of popular languages such as English, Chinese, German. Hence, academic researchers are concentrated on these dominant languages, while others still need to be improved even for the earliest methodologies of NLP. With this deadlock in hand, models with multi-lingual capabilities deserves much more attention than the others. As one of these multi-lingual models, BERT is developed and trained by Google research team, at 2018, to solve broad range of NLP tasks such as sentiment labeling, dialogue generation, sentence classification. Devlin et al. (2018) introduced their bi-directional transformer-based model as a remedy to the common problem of existing studies coming from one-way processing. In the simplest terms, BERT's random masking allows to learn the context without directional limits, by trying to predict masked parts depending on the rest. As being trained on a huge corpus of web text with multi-lingual scope, BERT seems to be the dominant method of further researches on NLP tasks through upcoming years. With the same motivation, stock prediction tasks have already started to explore and exploit BERT with the studies cited in the following paragraphs.

Hiew et al. (2019) conducted one of the earliest examples of financial prediction studies which benefited from BERT to analyze social sentiment. At first, they compared sentiment classification of BERT to other state-of-the-art models, listed as attention-based transformer, Pointwise MI with CNN and bi-directional LSTM, and Facebook's

FastText. On their finance forum dataset, BERT achieved the best performance in terms precision, recall, and f-measure with 7.2 points of F-measure distance to the closest competitor. After measuring sentiment polarity of user messages, price prediction task performed with LSTM, and compared to Vector Auto Regression model. Researchers concluded that while LSTM had higher MSE for one-year period, for longer time spans, machine learning algorithm selection lost its authority as all methods overfitted to the same trend.

Yang et al. (2019) claimed that treating all negative words as the same in terms of influential power on stock prices was one big mistake of former studies. To master at this problem, they studied on a self-attention-based prediction framework which was inspired by Vaswani et al. (2017)'s method. Their input dataset consisted of web search terms which were thought as indicators of financial sentiment of investors, along with historical returns. After creating embeddings of top search terms with BERT, combined vectors of numeric and text data were fed into feed-forward neural network, so as to anticipate short term price returns. In conclusion of experiments, their method achieved the best result compared to the simple aggregation method of web search sentiment and proved the contribution of BERT model in sentiment analysis.

2.3 Predictive studies on BIST market

Turkey has a prospering economy with a stock market which attracts local and foreign investors with its volatile nature (Balcilar & Demirer, 2015). Since volatility means opportunity rather than risk to the adventurists, BIST is an eventful field with lots of occasions for traders. Ergo, BIST is subjected to many important researches that inspect the market in a variety of ways (Gunduz et al., 2017). While textual data seems to be not

exploited enough for this emerging market, some of the predictive studies exerted on BIST are mentioned in the following lines with their approaches and findings about the market.

Bildirici and Ersin (2008) studied on a forecasting framework and utilized nearly 20 years of historical data on BIST market. Using closing prices, Bildirici and Ersin (2008) incorporated neural networks to the classical probabilistic methods of econometry to predict the index price. Their approach proven to be useful in most of the cases compared to the statistical approaches' individual results by means of RMSE.

Boyacioglu and Avci (2010) developed a fuzzy methodology to anticipate return of the BIST index relying on numeric input. On a monthly basis, Boyacioglu and Avci (2010) not only collected economic indicators of BIST market, but also included historical data from other countries' index prices which were claimed to be related to BIST market through the selected time period. Their predictions on monthly returns were found to be successful with an RMSE of 0.0068.

Kara, Boyacioglu, and Baykan (2011) compared two different predictive frameworks in terms of their ability to anticipate BIST100 index direction using technical signs. Utilizing 10 years of history, they found that their Artificial Neural Network performed significantly better than the polynomial SVM algorithm, with an accuracy level of 75.74%. Also, authors pointed out that model performance fluctuated between years, as the time they included in the study comprised economic crisis which severely influenced stock market.

Among restricted stock prediction studies that incorporates NLP on the prediction task using Turkish language sources, Gunduz and Cataltepe (2013) examined news in their study in terms of the effect on the stock market. Their efforts were to

predict BIST100 index's direction which was claimed to be sensitive to the financial news. In their proposed configuration, textual data was represented with TFIDF vectors of stemmed words. Then, they set a threshold for the document frequency to eliminate rare words which existed less than 1,000 documents in their corpora. Afterwards, they measured Mutual Information (MI) value of the remaining words to distill the input space to get the most useful feature set. The experiments made on the remaining set of features resulted in 32.5% F1-macro measure.

Gunduz and Cataltepe (2015) presented another study which incorporates textual data to numeric indicators to predict index level price direction of BIST100 index. They studied on a feature selection method to deal with the class imbalance problem while selecting the best set of features. Their proposed method for feature selection, namely Balanced MI, yielded 68% F1-macro score along with a Naïve Bayes classifier. Their input space consisted of both online news articles and official declarations of the companies. However, their experiments showed that announcements declared by companies did not helped the prediction process, as involving them decreased the performance.

Gunduz, Yaslan, and Cataltepe (2018) studied on a specific group of companies to predict price movements in a short time window. Their experiments included nine different companies operating in finance sector. Using online news articles with word embeddings and dictionary-based sentiment classification techniques, they compared three different prediction method, namely LSTM, Naïve Bayes, and random labelling. Although the best performance was yielded by LSTM with 54% F1-macro measure, they concluded that attention mechanism used along with LSTM did not bring additional performance to the model.

In a literature full of papers exercised on a limited number of celebrity languages such as English, Chinese, German, BERT is a milestone for the NLP domain with its multi-language and multi-task extent of ability. Moreover, BERT has a special version that is trained on a huge corpus in Turkish language. This special configuration, named as BERTurk, as stated by Aras, Makaroglu, Demir, and Cakir (2020), gains popularity in Turkish NLP community with its remarkable performance on different NLP tasks. Ergo, in this study, it is aimed to make the best of this state-of-the-art framework while predicting the stock directions of BIST market. Moreover, traditional NLP methods such as bag of words representation and TFIDF vectors are also employed not only for comparison, but also for inquiring the best fit for the specific task of the study. To the best of my knowledge, this is the first academic study applying BERTurk framework to KAP announcements of the companies traded on BIST market to predict their stock directions.

CHAPTER 3

METHODOLOGY

This chapter presents the approach of the thesis in detail. First, the data used in the study is explained comprehensively with its qualitative and quantitative features. Then, the following sub-section shares the pre-processing activity carried on the data, with its effects on the input space. Also, the target variable is explained with its calculation in the related sub-section. Subsequently, processed data's representation method is explained with its constraints. Afterwards, dimensionality reduction and over-sampling methods are mentioned which employed in the study to get a better input space for the experiments. Then, state of the art machine learning classifiers and BERTurk for the sequence classification task are described elaborately with their concepts and the usage in this study. An illustration of the study's predictive pipeline is given in Figure 1.

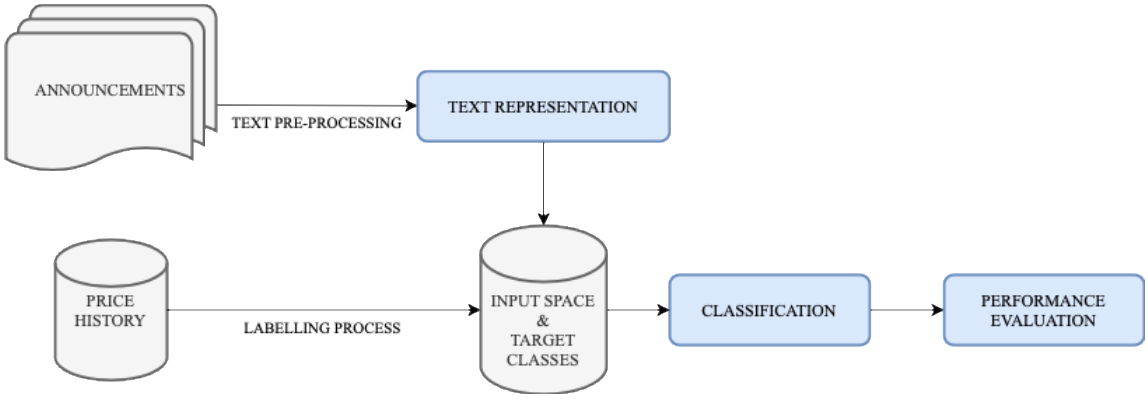

Figure 1. Prediction pipeline of the study

3.1 Data collection

Approach presented in this study, requires numeric and textual data to be collected and related to each other as the main goal is to predict numeric variables' move based on

related text pieces. So, numeric variables that constitute the target of the experiments, and textual data that form the input space are collected and processed for proper usage. Details of the data and related procedures are presented in the following lines.

3.1.1 Price data

In order to relate public disclosures with the stock movements, stock prices are collected in two different sources, namely BIST data store and Yahoo Finance. BIST data store has a product on their website (BIST Historic and Reference Data Platform) named Daily Closing Prices and Trade Volumes with monthly files which hold comprehensive information about prices and trading volumes of the BIST companies. Although these files contain useful data, it is not safe to use them directly since their price columns are not adjusted. Hence, adjusted prices are collected via Python Application Programming Interface (API) of Yahoo Finance.

Whilst prices used in the study is obtained mainly from Yahoo Finance, the API could not get the prices for some of the ticker-date pairs. For these pairs, BIST files are utilized if the related date is not coinciding with a corporate action of the related company, as corporate actions cause marginal difference between the regular and adjusted versions of the stock prices. For the ticker-date pairs which could not be fetched from both sources, related announcements are excluded from the study.

In addition to the individual stock prices, BIST index price data is also utilized in the study to assign the labels. Daily data of the index is gathered from tr.investing.com with a historical quest.

3.1.2 Textual data

KAP is the official platform in which publicly traded companies in Turkey are obliged to place their announcements with their electronic signs so as to expose related information to their investors. This related information can include financials and/or specific cases of the company, also it can be disclosed by regulatory establishments, such as Central Securities Depository. Announcements used in this study consist of material event disclosures of the BIST companies, which are declared during the 2015-2019 period. To collect the data from KAP's website (www.kap.org.tr), a Python library (BeautifulSoup 4.8.2: Python Package Index, 2019) written for web crawling activities is utilized, then the quested announcements stored in text files. Following fields of the announcements are fetched and stored in the scope of this study:

- ID: unique identifier of the announcement
- URL: web address of the announcement
- Title: official title of the company
- Ticker: company's stock symbol in BIST market
- Publish date: timestamp when the announcement is placed
- Disclosure type: category of the announcement
- Summary: summary of the announcement
- Related companies: tickers of the companies related to the announcement
- Related funds: symbols of the funds related to the announcement
- Items of the agenda: this area exists if the announcement is related to a general committee meeting
- Statement explanation: company's explanation for the disclosure

KAP's website assigns unique identifiers to the announcements, which are also placed in their web addresses. In order to be able to visit the original pages of the crawled announcements, ID and web address information are stored in the text files. Official titles and stock symbols are the identifiers of the companies which are also utilized to match the announcements with related data, such as price history, sector information.

Publish date area reveals the disclosure time up to seconds, which helps to distinguish if the announcement is made during the market sessions. Effect of the announcements are examined with one-day time window which assumes that an announcement made on a particular date will show its influence on the same trade date's closing price when compared to the preceding trade date's closure. Announcements which are declared on weekend or on national holidays are related to the next stock date. It is also assumed that the announcements made after 04:00 pm would affect the next stock date's closure as the amount of time until the market closing time of 06:05 pm (Borsa Istanbul, 2020) is not enough to digest the revealed information. With these assumptions in hand, a new column is created to associate each announcement with a stock date which they influence. Fetched announcements are then related with these stock dates' price data gathered via afore mentioned sources.

Announcements that are placed on KAP are grouped under four main categories: financial reports, material event disclosures, regulatory authority announcements, and others. Financial reports mainly include periodic disclosures of the companies about their financial actions and related performance indicators. Since these reports consist of numeric indicators mostly, and placed in attached files instead of web pages, they are not included in this study. On the other hand, regulatory authority announcements are

brief reports of the related establishments' executions on the market, such as trading suspensions, and transaction cancellations. As they are not rich in terms of textual data, they are not utilized in the experiments. The category of other contains several types of disclosures, such as regular forms of the companies which displays general information about the company, valuation, transaction, and analysis reports related to the companies and their corporate actions. Yet, as none of these disclosure types are text-intense and their contents are mostly delivered via attachments, they are not utilized for the study. However, material events category covers the disclosures about the special cases of the companies as well as the reports of the usual activities such as agendas of the general committee meetings. Being mainly consisted of textual parts which narrate the disclosed case, material events disclosures are eligible for being processed with NLP techniques. Ergo, in this study, BIST companies' material events disclosures are queried and fetched from the official platform of KAP so as to be utilized in the prediction pipelines. As the result of the query, 69,806 announcements are collected in total which belong to the companies publicly traded on BIST market.

Textual information is taken from the items of the agenda and the statement explanation parts of the announcements, as they carry the essence of the disclosed case. Some of the announcements have very short or no explanation part, instead they convey the information through attached documents. These attachments could not be fetched via automated codes; thus, they are not included in the study. Afterwards, if the announcement has no explanation or items of the agenda parts, it is filtered out from the dataset.

After matching announcements with price changes, the ones with no price information and has zero text length in terms of word count are filtered out as they

would serve a proper row neither as input nor as output. This filtering left 9,429 announcements out, and 60,377 of them remained. Descriptive statistics for the remaining disclosures are given in the Table 1.

Table 1. Statistics of Crawled Announcements After Pre-processing

Announcement count	60,377
Company count	402
Announcement date count	1,499
Company x stock date count	42,074
Peak date with the highest announcement count	22.09.2013
Highest announcement count in a day	213
Maximum word count	6,762
Minimum word count	1
Average word count	107
Unique word count	56,898

3.2 Data pre-processing

Prior to the experiments, both numeric and textual data are pre-processed so as to get them in proper format to be used in the prediction pipelines of the study. Details of the pre-processing steps are clarified in the following lines.

3.2.1 Target variable

Direction labels of the stock dates are calculated in relative manner, by considering the market's composite indicator. This way of relative calculation of the stock change is believed to be a more accurate way to measure the extra effect of the revealed information as it isolates the stock from the macro environment. Moreover, direction of the relative change is assigned with a threshold on the magnitude of the change so as not to polarize slight changes. Calculation of nominal and relative changes are given in Equation 1 and 2.

$$C_i = \left(\left(\frac{p_i}{p_{i-1}} \right) - 1 \right) * 100 \quad (1)$$

$$R_i = C_i - B_i \quad (2)$$

Equation 1 calculates the change (C_i) between the closure prices of the stock date (p_i) and the previous stock date (p_{i-1}) in percentages. For relative change (R_i), BIST index's change (B_i) for the same time window is subtracted from C_i, then its sign used in the target column calculation to clarify the effect of the announcement by isolating it from the macro environment. In order to omit slight price changes and accept them as neutral, relative change is used along with a threshold on the minimum absolute change. So, the target column is created using R_i and labeling done using standard deviation of the relative stock changes as the threshold. The rule set applied on the relative change is given below.

- Down (-1) ➔ if $R_i < -\sigma$
- Stationary (0) ➔ if $-\sigma <= R_i <= \sigma$
- Up (+1) ➔ if $R_i > \sigma$

Class distribution of the target column is given in Table 2, together with the price direction distributions according to nominal and relative changes.

Table 2. Distribution of the Stock Change Direction

Direction	Nominal Change	Relative Change	Target Column
-1.0	39.5%	52.2%	5.8%
0.0	21.0%	0.1%	86.5%
1.0	39.5%	47.7%	7.8%

3.2.2 Text pre-processing

Text pre-processing steps that are carried out in this study mainly aim to remove unnecessary characters as well as the non-alphanumeric parts of the announcements.

Subsequent to the cleaning steps, text pieces are subjected to a language detection procedure as some of them contain English version of the declaration. These English parts of the announcements are detected and removed with the help of the Langdetect (langdetect 1.0.8: Python Package Index, 2020) library, with a sentence-based approach. Then, in order to merge announcements that are declared by the same company and expected to affect the same stock date, rows are summed by company and stock date columns. After all, pre-processing resulted in 42,074 rows of data with proper input and output columns.

3.3 Document representation

In this study, it is aimed not only to explore and exploit the latest advances of the NLP field, but also to utilize the traditional NLP approaches which brought about successful performances for many years. Ergo, two different text representation methods are used and compared, namely bag of words and BERT embeddings.

3.3.1 Bag of words

Machine learning classifiers are firstly carried out on bag of words representation with unigrams. These unigram features (t) of the documents (d) are then weighted with TFIDF scores which depend on the following formula (3) of the Scikit-learn's TFIDF vectorizer (Pedregosa et al., 2011).

$$TFIDF(t,d) = TF(t,d) * IDF(t) \tag{3}$$

$$IDF(t) = \log\left(\frac{n+1}{DF(t)+1}\right) + 1 \tag{4}$$

TFIDF formula penalizes being common among different documents, while emphasizing frequent occurrence in the present document. In this way, generic words of the public disclosures are hoped to fade out with low TFIDF weights, while discriminative words are paired with higher values. Whilst being special to a limited number of documents is valuable, too infrequent words are not useful for the learning process. Therefore, weight vectors of the experiments are constructed with a document frequency filter which obliged during TFIDF calculations and set to a minimum of 10. For the overall set of announcements, 15,572 unigrams are found to be mentioned in 10 or more documents.

In addition to the original TFIDF weights, vectors are utilized with discretization by slicing each feature into 10 groups of equal intervals. For each group, mean value of the interval is used instead of the original weights, except for zero TFIDF values.

3.3.2 BERT embeddings

Devlin et al. (2018) presented BERT architecture as dependent on three types of embeddings to represent textual data, namely token, segment, and position embeddings. These pre-trained embedding schemes create three different vectors over the text, that are then summed for the final representation of the text piece to be used with the attention masks. Devlin et al. (2018) illustrated their embedding structure with the image given in Figure 2.

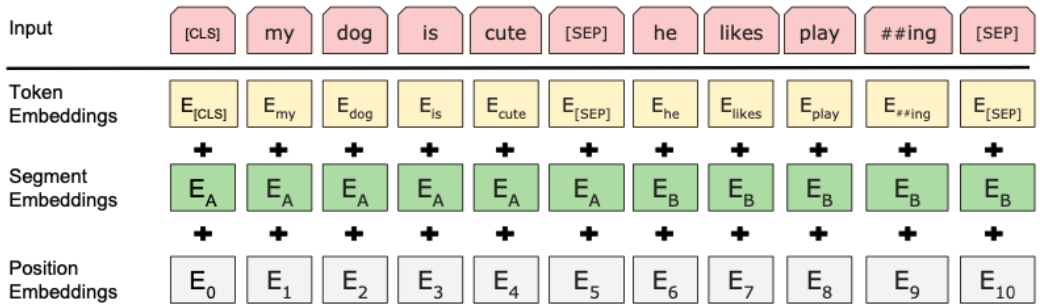

Figure 2. BERT embedding layers
Source: Devlin et al. (2018)

Parallelly created vectors of BERT represent different aspects of the text which are all important to reveal the relations. Token embeddings are created for contextual side of the words, while position embeddings help the model to know which word located in which position within the sentence. Other than that, segment embeddings are useful when the input is a sentence pair instead of a single text piece, as it is in a next sentence prediction task. Whilst all three embeddings exist in all BERT models, segment embeddings are not discriminative for text classification tasks, like it is in this study. Further information about BERT's method to represent textual data will be discussed elaborately under the section dedicated to BERT architecture.

Embeddings utilized in this study are derived from BERTurk model, and not only used with BERTurk sequence classifier, but also experimented with machine learning classifiers. For machine learning classifiers, embedding output of the BERTurk is utilized by extracting CLS embedding of each announcement since it represents the whole text.

3.4 Dimensionality reduction

Textual sources are prone to create sparse input spaces especially when statistical approaches are in use. This sparse nature of the data may cause difficulties for the learners. However, dimensionality reduction techniques are highly useful for not only overcoming the sparsity but also enhancing the input space. They not only increase the learning performance but also ease the training process by reducing the column size. In the study, two different ways of dimensionality reduction, namely feature selection and matrix factorization, are utilized to get a denser feature set.

3.4.1 Feature selection

Feature selection helps the prediction pipelines by discarding ineffective features and shrinking the input space to a more useful subset of the original. In this study, two different feature selection methods are employed to see if they help the learners to yield better performances. Following subsections explain utilized feature selection techniques.

3.4.1.1 Feature selection with Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a feature selection method which seeks for the best feature subset according to an importance indicator. The algorithm uses an estimator to assign each feature an importance level. Then, it visits the feature set repeatedly to eliminate the least important ones until the set shrinks to the intended size.

In this study, RFE function of Scikit-learn library is utilized with Support Vector Classifier as the estimator (Pedregosa et al. 2011). A reduced version of the feature set with the most important 1,000 columns of the TFIDF matrix is subjected to the experiments with machine learning classifiers.

3.4.1.2 Feature selection with MI

MI is a measurement which quantifies the contribution of a feature to the prediction of the true label. Using entropy values, it gives credit to the features that ease the classification with either their existence or their absence. Equation 5, 6, and 7 formulizes the process.

$$H(X) = -\sum_{i=1}^{N} P(x_i)\log P(x_i) \tag{5}$$

$$H(X|Y) = -\sum_{x\in X, y\in Y} P(x,y) log \frac{P(x,y)}{P(x)} \tag{6}$$

$$I(X,Y) = H(X) - H(X|Y) \tag{7}$$

Equation 5 calculates the entropy for X variable to measures the variance of it. For a constant column, entropy value is zero since the variable has no variance at all. On the other hand, Equation 6 measures the conditional entropy for X with Y in hand. For example, in a case where X takes the same value for each known value of Y, conditional entropy is zero as knowing Y gives a definite impression on X's value.

In Equation 7, by subtracting the conditional entropy, MI ($I(X,Y)$) tells how much of the variance of the X is coming from Y and vice versa. So, two independent variables' MI is zero as their variances are not conditioned on each other, while for dependent variables it equals to a non-negative value.

In this study, MI scores are used to create a subset of 1,000 features via SelectKBest function of Scikit-learn library (Pedregosa et al. 2011). Then, TFIDF vectors are incorporated to the experiments with their reduced versions.

3.4.2 Matrix factorization

Matrix factorization aims to create a representation of a high dimensional matrix by constructing a lower version which stores the core of the original space. Thus, learning process is expected to be more efficient and accurate as it deals with a smaller and more relevant space of data.

Singular Value Decomposition (SVD) is a matrix factorization method which is good at projecting sparse matrices into dense spaces. This linear way of dimensionality reduction uses eigenvectors to decide on the important cells of the matrix. On the other hand, Truncated SVD uses a subset of the singular values by setting the remaining ones zero and brings out extra speed.

In this study, Scikit-learn's TruncatedSVD function is employed to get a reduced input space with 1,000 columns (Pedregosa et al. 2011). Afterwards, reduced version of TFIDF matrices are experimented via machine learning classifiers.

3.5 Over-sampling

Classification problems with imbalanced set of labels come with a disadvantage as labels have not the same opportunity in terms of being represented. This may easily cause a low performance learning with a bayes towards the majority class. To overcome this issue, input space can be manipulated in different ways so as to level the class distribution. Over-sampling is one of these methods in which minority class(es) are augmented in particular ways. In this study, two methods are applied before the models' training phases: Random Over-sampling and Synthetic Minority Over-sampling Technique (SMOTE).

3.5.1 Random Over-sampling

In this method, minority class(es)'s instances are augmented by randomly selecting between the existing rows of the data. For the CLS embedding incorporated to the machine learning classifiers, existing rows balance the distribution by being replicated without any process on them. And, for the pre-trained model of BERTurk, announcement texts are replicated with Random Overs-sampling technique before their incorporated into the model. At each fold, all classes but the majority are augmented by repeating randomly selected input rows with the help of Imbalanced-learn library of Python (Lemaître, Nogueira, & Aridas, 2017).

3.5.2 SMOTE

Chawla, Bowyer, Hall, and Kegelmeyer (2002) proposed an over-sampling strategy, named SMOTE, which creates new lines of input data for the minority class(es) instead of re-visiting the existing ones. Basically, SMOTE forms new inputs using existing tuples and the distances to their nearest neighbors. By doing so, it also forms a more general value set for the minority class(es).

In this study, SMOTE method is applied on the training sets of TFIDF vectors so as to deal with the imbalance problem. Using Imbalanced-learn library of the Python, all classes but the majority are augmented with five nearest neighbors to create synthetic input rows (Lemaître, Nogueira, & Aridas, 2017).

3.6 Machine learning classifiers

Concept of learning from data comes out to be incessant at getting superior to its yesterday since it continually deepens. Despite the progress made with deeper

frameworks, traditional machine learning methods still bring about competitive results compared to more sophisticated approaches (Kraus and Feuerriegel, 2017; Wang et al., 2018; Khedr et al., 2017). To benefit from this predictive power, several machine learning models are leveraged in this study, on top of afore mentioned TFIDF vectors and BERTurk embeddings. These models are introduced in the following lines in detail.

3.6.1 Multinomial Naïve Bayes

Naïve Bayes is a simple yet powerful framework which depends on the assumption of independent features. It's efficient and effective method is still being employed as either benchmark or main model in many NLP studies. In their study Catal and Nangir (2017) experimented on Turkish text to classify them into sentiment categories with a voting-based machine learning approach. Apart from their proposed method being the winner, among the individual classifiers Naïve Bayes was the best performer on all three datasets used in the study. Another sentiment classification study by Wang et al. (2018) showed that Naïve Bayes could get a performance as good as an LSTM model, when the feature set was smaller.

Assuming that all instances in the input vector are independent from each other, Naïve Bayes classifier relies on the following formula of Bayes theorem.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{8}$$

$P(C|X)$: posterior probability of the class C given the input vector X

$P(X|C)$: likelihood of the input vector X when the class is equal to C

$P(C)$: posterior probability of the class C

$P(X)$: prior probability of the input vector X

Likelihood of the input vector given the class C is calculated by multiplying individual likelihoods of each member of the vector, since they assumed to be independent. That yields the Equation 9 for the $P(C|X)$, then the class with the highest posterior probability takes the crown for that particular X vector. However, as $P(X)$ is constant for each row of the input matrix, it is not used in the classifiers.

$$P(C|X) = \frac{P(C)\prod_{i=1}^{n} P(x_i|C)}{P(X)} \tag{9}$$

Multinomial Naïve Bayes is a version of Naïve Bayes classifier, which is also a popular technique among text classification tasks. This version uses the below distribution (Equation 10) for each class c, with n being the number of words in the corpus and θ_{ci} being the probability of word i existing in a document belonging to class c.

$$\theta_c = (\theta_{c1}, \theta_{c2}, \dots, \theta_{cn}) \tag{10}$$

θ_{ci} is calculated with the Equation 11:

$$\theta_{ci} = \frac{N_{ci} + \alpha}{N_c + \alpha n} \tag{11}$$

In this formula, N_{ci} stands for the count of the documents that belong to the class c and include word i, while N_c is the sum of all N_{ci} values derived for each word in the corpus for class c. And, α is a non-negative regularization factor that avoid the formula being zero for the words that are not present in the training set. To implement Multinomial Naïve Bayes classifier, Scikit-learn library's MultinomialNB function is used in the study with a custom three-fold time series cross validation and its parameters are tuned with the help of a grid search for optimization (Pedregosa et al. 2011).

3.6.2 Logistic Regression

Logistic Regression is a linear classifier which in simplest words is a linear regression with a sigmoid on top of it. Owing to the sigmoid it applies on the regression's result, outcome of a logistic regression model lays between zero and one. That is, if the result of the regression part goes to infinity, outcome takes one, whilst for a value closer to negative infinity outcome takes zero, and for a value around zero outcome takes 0.5. Related formulas are given with Equation 12 and 13:

$$z = ax + b \tag{12}$$

$$y = \frac{1}{1+ e^{z}} \tag{13}$$

As y takes a value between zero and one, a threshold must be set to convert it to a class label. However, this approach accounts for only binary classification tasks which a threshold would be enough to separate two classes from each other. To convert this concept into a multiclass classification technique, softmax function must be used instead of sigmoid function. Thus, y becomes a vector of classes with one for the true class and zero for the rest, then the value for class y_i is calculated with the Equation 14:

$$y_i = \frac{e^{zi}}{\sum_{j=1}^{n} e^{zj}} \tag{14}$$

Equation of the softmax function, n being the number of classes, ensures that $\sum_i^n y_i = 1$. Then, as Alpaydın (2014) states in his book, Logistic Regression basically takes the maximum of y_i values, in a differentiable manner. Afterwards, learning process is carried out with cross-entropy function, by minimizing it via gradient descent.

In this study, Logistic Regression is applied with the help of Scikit-learn library, using LogisticRegression function (Pedregosa et al. 2011). Using a three-fold custom time series cross validation, a grid search is done so as to get optimum hyperparameter

set for the classifier. In this way, it is possible to see and choose the best regularization method and regularization strength for the problem in hand. Also, multinomial loss is used for the multiclass classification problem, instead of one-vs-all approach.

3.6.3 XGBoost

Machine learning community has a leverage in hand, named boosting, to overcome low performance learners. Boosting method basically combines multiple low performance learners to get a strong one. Alpaydın (2014) described this process as learning from the predecessors' mistakes. With gradients involved in this boosting process, learners are trained on loss gradients of their formers, so as to minimize the error.

Being presented by Chen and Guestrin (2016), XGBoost is one of the popular boosting methods which depends on parallelly created decision trees. Chen and Guestrin (2016) summarizes their contribution with following bullets:

- High scalability
- Efficient framework with the proposed weighted quantile sketch algorithm
- Automated handling of sparsity
- Optimized hardware usage

Although XGBoost's source usage method enables it to search for tree splitting points efficiently, its way of considering sparsity as missing values makes it hard to work on TFIDF vectors, like other tree-based learners. Yet, popularity of the latest boosting machines is not neglectable. Thus, they are employed on both original data and reduced versions of it. In the study, XGBoost implementation is done with the help of the XGBoost package of Python with optimized hyperparameters.

3.6.4 LightGBM

Proposed by Ke et al. (2017), LightGBM is one of the tree-based gradient boosting algorithms which speeds up the training process by its structural differences, such as relying on a subset of the data while computing the gradients which brings out a significant reduction in time without much sacrifice from the accuracy by using tuples with larger gradients. Moreover, Ke et al. (2017) introduced another novel feature grouping and binning strategy that they defined as their remedy for the sparsity issue. Basically, LightGBM groups feature columns together in which non-zero values do not coincide, such as one-hot-encoding columns. By adding margins to the individual features, the model ensures that their values lay in different ranges in the composite column. That is to say, each individual can be represented by a different value bin.

LightGBM differentiates from many other machine learning algorithms by its ability to use categorical data without requiring former processing such as one-hot-encoding. In this study, LightGBM is employed for not only its high performance in terms of accuracy and speed coming from its efficient structure, but also the way it handles categorical variables. To leverage its advantages, in this study, LightGBM algorithm is used with the help of LightGBM package of Python.

3.6.5 CatBoost

CatBoost is another gradient boosting based ensembled tree algorithm with a novel approach for both boosting weak learners and handling non-numeric features, as Prokhorenkova, Gusev, Vorobev, Dorogush, and Gulin (2018) stated. CatBoost derives target statistics for categorical variables in an ordered fashion. In their study,

Prokhorenkova et al. (2018) pointed out that the algorithm does this ordering again and again for each step, so as to obtain a different permutation and prevent high variance in target statistics between former and latter rows of the input. This ordering approach is claimed to be the solution for the prediction shift in the existing gradient boosting machines. Other than target statistics of the categorical features, the study mentioned that this shift is also caused by gradients which are trained on the same set of inputs, so are biased. To overcome this, instead of using the same ordering, CatBoost algorithm derives different versions of the input with random orderings. Using these versions to measure performance of the trees is claimed to be the way to reduce the variance of predictions.

CatBoost is a novel structure of gradient boosting trees with substantial differences from its rivals. Hence, it is involved in the study using CatBoostClassifier method of CatBoost package in Python.

3.7 BERT

Proposed by Devlin et al. (2018), BERT is a framework in which text pieces are examined from both directions (left-to-right and right-to-left) to create a more accurate representation of the context. This bi-directional architecture is pre-trained on a huge library and claimed to be able to function well for different kind of NLP tasks with only an additional output layer. While it is undeniably a huge step being able to fine-tune this pre-trained model for different NLP tasks such as next sentence prediction, question answering, and text classification, real excitement comes from its multi-lingual side, especially for the ones who work on non-English text.

Devlin et a. (2018) explained the architecture behind as the transformer blocks that utilizes attention mechanism. To capture the math and logic of the BERT model more comprehensively, these two concepts will be mentioned first, then the other details of BERT such as text representation, masking language model, pre-training, and fine-tuning parts will be presented. Afterwards, BERTurk, which is a variant of BERT language model trained on a huge Turkish corpus, will be explained in detail.

3.7.1 Attention mechanism

Attention mechanism is proposed for language translation tasks at first, by Bahdanau, Cho, and Bengio (2014). Bahdanau et al. (2014) argued that former encoder-decoder architectures could suffer from their one-sized vector approach to represent the sentences, especially when it comes to longer ones, as compressing a sentence would sacrifice more performance as the sentence gets longer. Instead, their aim was to focus on the more important pieces of the text which would say more about the meaning of it. Then, a sentence was encrypted into vectors coming from its words being summarized in varying sizes. In their study, Bahdanau et al. (2014) utilized bi-directional construction of RNNs and they visualized the structure with the following image and the annotation belonging to it.

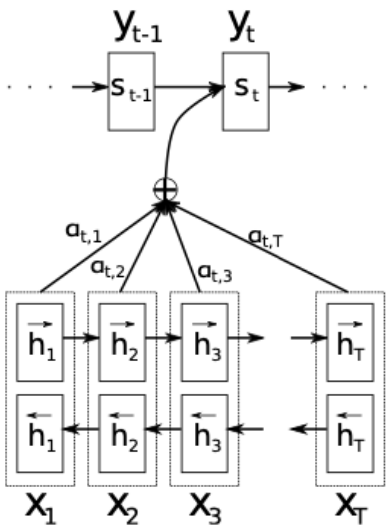

Figure 3. Attention mechanism
Source: Bahdanau et al. (2014)

yi: i-th target word

si: i-th hidden state

aij: weight of j-th annotation for the i-th target word

Xj: j-th word of the sentence

hj: j-th annotation belonging to Xj

The process pictured in Figure 3 tries to predict y_t, given the former state's output and the current hidden state. However, unlike RNN, Bahdanau et al. (2014) created a unique set of representations with respect to the vicinity of each word. Then, for each target word, these vectors are summarized with different set of weights, in other words: attention levels. Bahdanau et al. (2014) formulizes this process with the following equations.

$$c_i = \sum_{j=1}^{T_x} a_{ij} h_j \tag{15}$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{16}$$

Context of a sentence (c_i) is a selective aggregator of the neighbors (h_j) which gives credit (a_{ij}) according to the influence that each neighbor has on the target word. These credits are being learnt along with the whole procedure.

3.7.2 Transformers

Transformers are set of neural layers which come from encoders and decoders with parallel attention layers. Vaswani et al. (2017) created this block structure to get a thorough representation of the text without recurrently visiting sub-parts of it. In their study, construction of a transformer block was picturized as Figure 4.

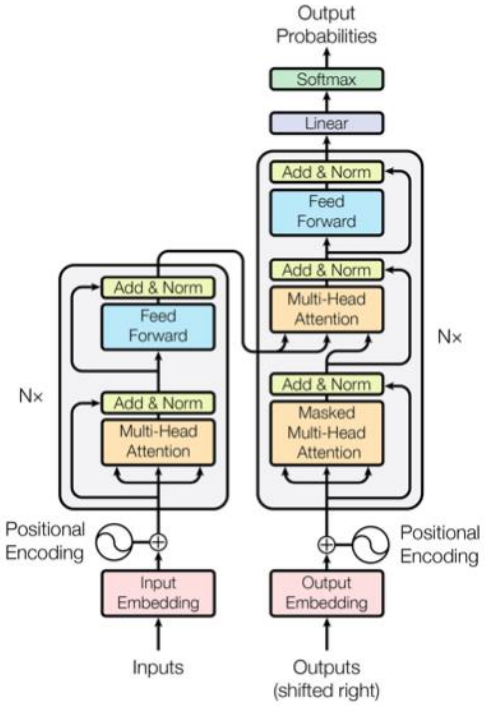

Figure 4. Structure of a transformer block
Source: Vaswani et al. (2017)

On one hand, as the figure indicates, each layer of an encoder consists of two subordinates: an attention layer for contextual representation and a neural network for positions. On the other hand, decoding part has an additional subordinate attention layer

that also processes the encoder's output. In the study, both an encoder and a decoder are stated to have six of these layers with the same structure.

Multi-head approach for attention layers enable the model to learn the contextual representation in a parallel concept which work on different locations of the sentence at the same time. Then, these parallelly created attention vectors are aggregated into a final version so as be fed into the last layer of the encoder, which is a feed forward neural network. Vaswani et al. (2017) stated that this approach brings out computational efficiency and would bring out even more when the attention span is restricted with a particular number of words.

3.7.3 Text representation

BERT's model design names an input instance as a sequence which may involve one or two text pieces in it. This pair-wise concept is invented for the NLP tasks which relates two text pieces to each other, such as next sentence prediction. Devlin et al. (2018) enclosed each sequence with two special receptors, namely CLS and SEP, and set a boundary between the two instances with another SEP token. CLS token, standing for classification, learns to represent whole sequence, while SEP distinguishes the two members from each other. Along with these guardians at the two edge and the separation point of the sequence, words within the sentence are tokenized into pieces, and then subjected to three different embedding layers: token, segment, and position embeddings.

Token embeddings are constructed via WordPiece embedding, introduced by Wu et al. (2016) from Google Research team. WordPiece embedding scheme leverages common partitions to represent the rare whole. In this way, Wu et al. (2016) created a

method that stands between char-level and word-level processing, which is claimed to increase the performance by having the advantages of both sides.

Segment embeddings are learnt for labeling each token with the sentence they belong to within the sequence. While this layer is an essential part for the sentence pair tasks, for NLP studies which deal with only one sentence at a sequence this layer is only symbolic.

Position embeddings aim to store and use the information of each token's location within the sequence. Vaswani et al. (2017) pointed out that as the transformer blocks have not a sequential course of action, thus not aware of the location by nature, this layer helps to relate each token with its place. Authors stated that they choose sinusoid function to learn positions for its ability to adapt sentences with unprecedented lengths. After all, afore mentioned three layers of embeddings are summed into a single vector so as to be fed into the self-attention layer with masks on it.

3.7.4 Masked language modelling

Language masking is an old concept, spoken by Taylor (1953) firstly, which depends on the theory of the ability of deducing the whole from a diminished version of it, depending on the existing pieces. That is to say, one can read a piece of text with missing words and complete it intuitively as available words and structure of the language imply the absents. This old but gold theory, re-emerged in the recent ground-breaking architecture of BERT, enabling it to work in both directions.

In essence, bi-directional practice of BERT relies on the Masked Language Modelling (MLM), as stated by Devlin et al. (2018). During the pre-training process, Devlin et al. (2018) masked a random part of each sequence before creating their

embeddings. While masked word pieces construct the 15% of the sequence, remaining tokens are exploited for predicting the masked part. However, as this approach is exclusive to pre-training process, to narrow the gap between fine-tuning these randomly selected parts are masked only for 80% of the time. For the rest of the time, tokens are replaced with another random word piece or remained same.

3.7.5 Pre-training of BERT

BERT's novel architecture is pre-trained on an English corpus, consisting of 3,3 million words, for predicting the next sentence. This task requires the model to learn the relationship between consecutive sentences. Devlin et al. (2018) constructed the dataset with half randomly paired sentences and half actual successive ones. Two models, namely BERT-base and BERT-large, are pretrained on this dataset with 12 and 24 transformers blocks, 768 and 1024 hidden layers, and 12 and 16 attention heads, respectively. Base model was built with the same size as OpenAI so as to compare their performances. At the end, two models with millions of parameters are pre-trained for four days not only to predict if the sentence is next to its mate or not, but also to learn embedding layers so as to be the starting point of many other NLP tasks with an additional layer on top.

3.7.6 Fine-tuning of BERT

Transfer learning is one of the blessings that happened to the information retrieval community which basically gives the opportunity for exploiting other's experiences. That is to say, researchers benefit from a model with set of parameters that are already leaned on a similar dataset. Apart from time saving, transfer learning is also a redeem for

the cases where enough data is not available or hard to collect, or additional sources would help anyway. Pre-trained BERT, along with its history with 3,3 million words, is a wealthy legator that hands down millions of parameters learned on a massive textual source that is nearly impossible for an individual to fetch.

Novel architecture of BERT enables the model to be adaptive for different NLP tasks by only re-visiting the learnt parameters along with relative datasets. Devlin et al. (2018) performed fine-tuning for a set of different NLP tasks with additional datasets. Those setups involved not only tasks with paired sentences such as question answering, but also single sentence sequences for classification models.

Adapting BERT to a classification problem required an additional layer to learn mapping input representations to the classes. Devlin et al. (2018) did it with a log loss function, upon the CLS token's embedding, as it is the big picture representing the whole sequence. After extensive trials on a benchmark dataset, BERT with both its base and large setups beat its rivals, including the OpenAI.

3.7.6 BERTurk

BERTurk is a variant of the original BERT model, which is pre-trained on a large Turkish corpus. With the help of Turkish NLP enthusiasts, BERT's state-of-the-art design is trained on 4,404,976,662 Turkish tokens (Hugging Face). Budur, Özçelik, Güngör, and Potts (2020) compared this Turkish version to the multi-lingual BERT and the original BERT models on a translation task from English to Turkish. After training each model for 3 epochs, with their cased and uncased versions separately, their findings were that being trained on a huge Turkish corpus is important for the performance, since

BERTurk came as the winner with its cased setup. In addition to this, authors pointed out that all model variants performed better with cased versions.

Aras, Makaroglu, Demir, and Cakir (2020) compared transformer models to recurrent architectures with their performances on Turkish named entity recognition task. Their experiments showed that transformer blocks perform better than the recurrent structure of bi-directional LSTM, since all transformer-based architectures out-performed any of the bi-directional LSTM setups. However, in the study, cased version of BERTurk with a conditional random field layer on top of it, was the best performer among the transformer models including the original BERT.

As many other pre-trained NLP models, BERTurk models can also be found in the model hub called Hugging Face (huggingface.co). So, in this study, BERTurk's base version, namely bert-base-turkish-cased, is fetched from the repository of Hugging Face through Pyton. This pre-trained model of BERTurk is not only fine-tuned for its pre-trained sequence classifier, but also employed for its embeddings to be used with machine learning classifiers. As the deep architecture of the model requires intense parallel computations for fine-tuning, experiments are carried out on Google Colab notebook with Tesla P100-PCIE-16GB GPU.

Model configuration of bert-base-turkish-cased is placed in Table 3 as stated in Hugging Face. The model allows a sequence to have up to 512 tokens. That is to say, each document attended to the model with their first 512 tokens if they have, else, their existing tokens are padded with zeros to 512. Constraints and other details that are present in the study are also given with Table 4.

Table 3. Parameters of the Berturk-Base-Turkish-Cased Model

Parameter	Value
Drop-out probability of attention	0.1
Activation function (encoder and pooler)	gelu
Drop-out probability of fully connected layers	0.1
Number of hidden units	768
Maximum number of tokens	512
Number of attention heads	12
Number of hidden layers	12
Token id for padding	0

Source: https://huggingface.co/dbmdz/bert-base-turkish-cased, 2020

Table 4. BERTurk Model Configuration Used in the Study

Parameter	Value
Maximum number of tokens	512
Truncating and padding	post
Batch size	8
Learning rate (Adam optimizer)	2E-05
Epsilon (Adam optimizer)	1E-08
Number of epochs	4
Number of classes	3

CHAPTER 4

RESULTS

This chapter presents the experiments that are carried out in the scope of this study, with their performances. First, performance measures are defined, then all model configurations are evaluated using given measures.

4.1 Performance measure

Classification tasks are often evaluated with a confusion matrix which compares true labels to model outcomes. Illustrated in Figure 5, the orthogonal emphasized with a darker fill cover the instances that is labelled correctly, namely true positive (TP) and true negative (TN), while remaining cells carry the ones mislabeled by the model, namely false positive (FP) and false negative (FN).

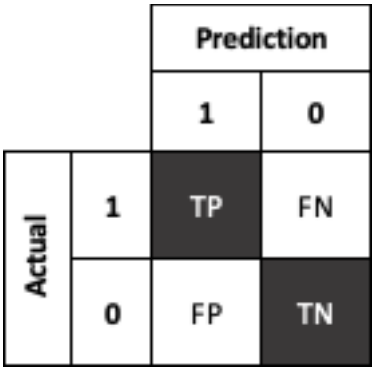

Figure 5. Confusion matrix for a binary class classification problem

Performance of a classification model can be quantified using following formulas that are derived from the confusion matrix the model resulted in.

$$A = \frac{TP+TN}{TP+TN+FP+FN} \tag{17}$$

$$P = \frac{TP}{TP+FP} \tag{18}$$

$$R = \frac{TP}{TP+FN} \tag{19}$$

$$F_1 = \frac{2*P*R}{P+R} \tag{20}$$

Accuracy (A) is the ratio of truly labeled instances over the total set. However, this measure can be misleading especially when the classes have an imbalanced distribution. Instead, F_1 measure is highly preferred since it is a harmonic indicator which considers both Precision (P) and Recall (R) metrics. Precision tells about how many of positive labels the model gives are truly positive, whilst Recall shows to what extent the model can capture real positives labels. However, when it comes to multi-label classification problems, F_1 measures are calculated for each label separately. To get the whole picture, different aggregation methods are employed on individual scores of the classes. As one of them, macro averaging takes a plain average of all individual measures, without considering the class distribution. Another option comes with a weighted average formula, which is true to its name; F_1-weighted. And the last option handles the issue more globally by calculating global Precision and Recall, then calculating the F_1-micro over these ratios, so basically invents another path to get the Accuracy. Among these metrics, F_1-macro is chosen as the most suitable indicator for this study and used in model selection. However, label based F_1 scores and F_1-weighted for overall set is also given in the results, together with the Accuracy.

4.2 Experiments with machine learning classifiers

Machine learning classifiers are utilized with both TFIDF vectors and BERTurk embeddings, as stated in the Methodology section. During the experiments, all machine

learning classifiers are subjected to grid search with a three-fold sliding window time series cross-validation scheme, in order to find their optimal hyperparameter settings.

Cross-validation scheme is customized to work with months instead of row indices and designed to slide forward six months at each fold. So, at each fold, six half-years are utilized as training data, while subsequent two half-years was the test set. Although this method caused twelve months of data to be used twice and their contribution to the over-all performance to increase, this overlap is accepted as negligible as it extends the usage of the data. With a larger dataset with more years of announcements, monthly sliding cross-validation scheme can be used without overlapping test months.

Table 5 exhibits average validation F1 scores of the classifiers with their optimum hyperparameter sets on TFIDF vectors and BERTurk embeddings. Scores are given for individual labels and with their averaged versions. Best result of each indicator is emphasized with a light shading.

Multinomial Naïve Bayes gives its best performance along with the feature space reduced with MI. However, feature selection with RFE brings out the poorest performance with its linear approach on dimensionality reduction, and second-worst performance with BERTurk embeddings. Another point that common with all feature spaces but MI, Multinomial Naïve Bayes grasps the negative sign better than the upward stock movement.

Logistic Regression delivers its best result on the original TFIDF vectors with F1-macro of 38.5%. While dimensionality reduction methods do not bring additional performance to the learner, poorest results are observed with the MI based feature selection. This non-linear way of feature selection worsens the performance by 4.2

points of F1-macro score with respect to the original feature set that brings out the best performance. Other than that, Logistic Regression performed poor with CLS embeddings and learns negative direction better than the positive in most of the feature spaces, similar to Multinomial Naïve Bayes. However, among the learners, Logistic Regression shows the best performance in terms of distinguishing up class.

As opposed to the common belief, XGBoost's performance on sparse TFIDF vectors is not far away from the dense spaces as it delivers the second best F1-macro score. While the highest F1-macro comes with RFE, in all configurations negative direction is learnt better than the positive one. Apart from the prediction performance, XGBoost was the slowest learner during the experiments among discussed methodologies.

Among all setups, LightGBM with RFE dimensionality reduction yields the best F1-macro score with 39.7%. However, F1-macro winner's classification accuracy is 7.8 points lower than the highest accuracy which is yielded by LightGBM on BERTurk embeddings. Similar to the other classifiers, LightGBM distinguishes down class better than the up, on all feature spaces. Top 50 features of each fold of the best classifier are presented in Appendix A, with a decreasing order in importance.

CatBoost, the second-best learner according to the F1-macro score, presents its best performance along with the matrix factorization on the input space. Similar to other gradient boosting trees, it performs well enough on the original TFIDF vectors. Better discrimination of the negative direction is existing in CatBoost's results, also. In addition, CatBoost gives the highest F1-weighted score overall with CLS embeddings which is slightly better than the LightGBM's performance on the same input space.

Table 5. F1 Scores of Machine Learning Classifiers on Different Feature Spaces

Experiment	Down (-1)	Stationary (0)	Up (+1)	Macro	Weighted	Accuracy
Multinomial Naive Bayes						
TFIDF (original)	20.2%	79.4%	14.8%	38.2%	69.4%	65.0%
TFIDF (MI)	20.1%	84.7%	11.1%	38.6%	73.5%	72.4%
TFIDF (RFE)	13.8%	64.9%	14.6%	31.1%	56.8%	48.3%
TFIDF (SVD)	19.4%	80.7%	15.8%	38.6%	70.5%	66.6%
BERTurk (CLS)	16.1%	61.3%	16.7%	31.4%	54.1%	45.3%
Logistic Regression						
TFIDF (original)	18.6%	80.3%	16.7%	38.5%	70.2%	66.2%
TFIDF (MI)	17.0%	69.6%	16.4%	34.3%	61.1%	53.4%
TFIDF (RFE)	18.8%	77.2%	15.7%	37.2%	67.6%	62.1%
TFIDF (SVD)	18.0%	74.3%	16.8%	36.4%	65.2%	58.7%
BERTurk (CLS)	17.0%	69.0%	17.3%	34.4%	60.7%	52.8%
XGBoost						
TFIDF (original)	17.8%	87.8%	11.8%	39.1%	76.0%	77.3%
TFIDF (MI)	17.5%	85.7%	12.0%	38.4%	74.2%	74.0%
TFIDF (RFE)	19.2%	84.7%	13.7%	39.2%	73.7%	72.3%
TFIDF (SVD)	18.1%	87.2%	11.2%	38.9%	75.5%	76.5%
BERTurk (CLS)	15.7%	87.0%	11.8%	38.1%	75.2%	76.0%
LightGBM						
TFIDF (original)	17.0%	88.8%	10.6%	38.8%	76.6%	78.8%
TFIDF (MI)	15.7%	86.6%	12.6%	38.3%	75.0%	75.5%
TFIDF (RFE)	19.0%	85.2%	14.9%	39.7%	74.2%	73.1%
TFIDF (SVD)	17.7%	87.0%	12.1%	38.9%	75.3%	76.2%
BERTurk (CLS)	13.1%	90.0%	7.4%	36.8%	77.1%	81.1%
CatBoost						
TFIDF (original)	17.1%	87.2%	11.8%	38.7%	75.4%	76.2%
TFIDF (MI)	16.4%	85.6%	12.3%	38.1%	74.1%	73.7%
TFIDF (RFE)	18.9%	82.8%	15.4%	39.0%	72.2%	69.5%
TFIDF (SVD)	18.1%	87.0%	13.2%	39.4%	75.5%	76.3%
BERTurk (CLS)	14.3%	89.8%	7.9%	37.4%	77.1%	80.9%

Even though sentence embeddings created with BERTurk yielded promising results especially with the tree boosters, they could not beat TFIDF vectors with any of the learners by means of F1-macro. BERTurk embeddings has a maximum token length

limitation which could be a cause of decrease in performance. As 12.4% of the input texts have more than 512 tokens, this portion of the announcements could not be utilized truly because of the model limitations. Yet, the highest F1-weighted and accuracy scores are delivered with CLS embeddings in combination with the gradient boosting tree learners.

Other than the original TFIDF weights, classification experiments are conducted on discretized vectors, also. Similar to the original vectors, dimensionality reduction and over-sampling methods are employed during these experiments. While there is not a consistent superiority between the original weights and the discretized values in terms of performance, results on discretized vectors are placed in Appendix B.

Moreover, LightGBM and CatBoost algorithms are experimented with categorical features in addition to the text representation vectors. To this end, sector information of the companies is included as a new feature both with main sector and sub-sector columns on separate trainings to see if they result in a performance increase. Table 6 displays the results of these experiments along with the performances of LightGBM and CatBoost without categorical columns for comparison. Best validation performances of each algorithm are emphasized with light shading. While the highest F1-macro scores are not changed, sector information enhanced the prediction performance on up class for both classifiers.

Table 6. F1 Scores of LightGBM and CatBoost Classifiers With Categorical Features

Experiment	Down (-1)	Stationary (0)	Up (+1)	Macro	Weighted	Accuracy
LightGBM						
TFIDF (original)	17.0%	88.8%	10.6%	38.8%	76.6%	78.8%
Main Sector	17.1%	88.5%	10.0%	38.5%	76.4%	78.4%
Sub-Sector	16.7%	88.5%	12.1%	39.1%	76.6%	78.7%
TFIDF (MI)	15.7%	86.6%	12.6%	38.3%	75.0%	75.5%
Main Sector	15.6%	85.8%	11.8%	37.7%	74.2%	74.2%
Sub-Sector	15.2%	86.5%	12.6%	38.1%	74.8%	75.4%
TFIDF (RFE)	19.0%	85.2%	14.9%	39.7%	74.2%	73.1%
Main Sector	17.6%	85.2%	14.0%	38.9%	74.0%	73.0%
Sub-Sector	15.3%	84.4%	14.9%	38.2%	73.3%	71.9%
TFIDF (SVD)	17.7%	87.0%	12.1%	38.9%	75.3%	76.2%
Main Sector	15.9%	86.4%	14.1%	38.8%	74.9%	75.3%
Sub-Sector	14.2%	86.3%	13.7%	38.1%	74.7%	75.1%
BERTurk (CLS)	13.1%	90.0%	7.4%	36.8%	77.1%	81.1%
Main Sector	12.9%	89.7%	7.5%	36.7%	76.9%	80.7%
Sub-Sector	11.8%	89.7%	8.5%	36.7%	76.9%	80.8%
CatBoost						
TFIDF (original)	17.1%	87.2%	11.8%	38.7%	75.4%	76.2%
Main Sector	17.7%	87.1%	12.3%	39.0%	75.5%	76.1%
Sub-Sector	17.6%	87.1%	12.5%	39.0%	75.4%	76.1%
TFIDF (MI)	16.4%	85.6%	12.3%	38.1%	74.1%	73.7%
Main Sector	15.9%	85.2%	13.3%	38.1%	73.8%	73.1%
Sub-Sector	15.6%	85.3%	12.8%	37.9%	73.9%	73.3%
TFIDF (RFE)	18.9%	82.8%	15.4%	39.0%	72.2%	69.5%
Main Sector	17.8%	83.2%	15.6%	38.8%	72.5%	70.0%
Sub-Sector	17.0%	81.3%	16.2%	38.2%	70.9%	67.4%
TFIDF (SVD)	18.1%	87.0%	13.2%	39.4%	75.5%	76.3%
Main Sector	17.0%	85.0%	15.3%	39.1%	74.0%	73.1%
Sub-Sector	16.2%	85.2%	14.3%	38.6%	74.0%	73.3%
BERTurk (CLS)	14.3%	89.8%	7.9%	37.4%	77.1%	80.9%
Main Sector	14.3%	89.6%	8.8%	37.6%	77.0%	80.5%
Sub-Sector	13.9%	89.4%	8.0%	37.1%	76.8%	80.2%

4.3 Experiments with BERTurk pre-trained classifier

BERTurk embeddings are processed with the pre-trained sequence classification model which has a linear layer on top for classification. During the fine-tuning process, cross validation scheme that is applied on machine learning classifiers is employed in exact fashion, and training data is over-sampled randomly. Average training losses during the epochs are given in Figure 7, and average validation performance of the folds is yielded as given in Table 7.

Table 7. Performance of Pre-Trained Classifier on BERTurk Embeddings

Experiment	Down (-1)	Stationary (0)	Up (+1)	Macro	Weighted	Accuracy
BERTurk	16.3%	88.1%	12.1%	38.8%	76.1%	77.8%

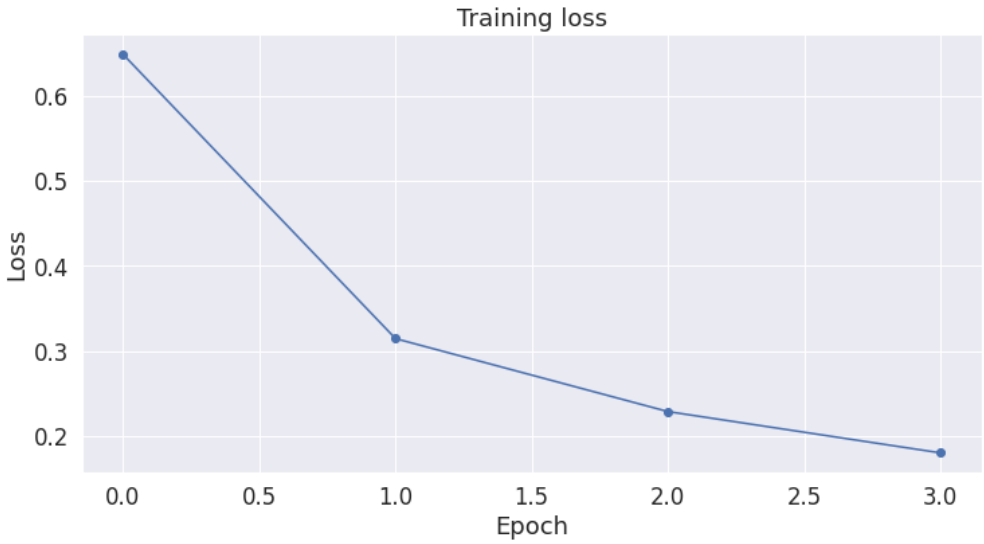

Figure 6. Average training loss during the epochs

Pre-trained classifier delivered the highest F1-macro score on BERTurk embeddings among the learners. And as it is observed with the TFIDF vectors and machine learning classifiers, pre-trained classifier is also able to learn negative direction better than the positive one.

CHAPTER 5

CONCLUSION

Finance is a greedy phenomenon which seeks for the best employment of any advances in related fields. NLP is one of these fields with lots of enthusiasts who are trying their best to make the most of any tool in hand, like it is with the stock prediction task. In Literature Review section of the thesis, many valuable studies are mentioned, and benefited from so as to build up this work. Despite of the dominance of a few languages in NLP researches about stock prediction, an emerging market is examined in this study along with related textual sources in Turkish language. In this frame, diverse range of machine learning algorithms are incorporated with different ways of textual representation.

Public disclosures are in close attention of the investors, undoubtedly. Ergo, they are involved in prediction pipelines to anticipate markets' future (Lee & Suh, 2018; Gunduz & Cataltepe, 2015; Kraus & Feuerriegel, 2017). In this study, material events announcements of BIST companies are utilized to predict the price direction at stock level. Crawled announcements are related to market days to be matched with the price movements of them. While each ticker is only related to its own announcement, macro facts are excluded by subtracting the BIST index change.

To reveal the relation between announcements and stock prices, TFIDF weighting and BERTurk embeddings are utilized with a diverse set of learners. Through a three-fold monthly sliding time series cross validation scheme, five different machine learning algorithms trained on both the statistical view of TFIDF and deep structure of BERTurk embeddings. Employed machine learning classifiers, namely Multinomial

Naïve Bayes, Logistic Regression, XGBoost, LightGBM, and Catboost, have different ways of learning from the data. While all algorithms performed well on experiments, gradient boosting machines are found to be the best performers with a more robust structure under the changing input space characteristics.

Feature engineering methods are the helpers of the learning algorithms with their remedial transforming strategies on the input space. In this study, three different ways of dimensionality reduction are incorporated to the prediction pipelines to see if they bring any additional performance to the learners. While the winner of the study is a configuration with RFE feature selection method, Multinomial Naïve Bayes and Logistic Regression models are observed to be very sensitive to the reduction technique as it may cause up to 7.5% decrease in the F1-macro. On the other hand, gradient boosting tree methods performed well under all techniques of feature engineering and yielded a more stable performance with different feature spaces.

Among machine learning classifiers, in terms of F1-macro measurement, LightGBM algorithm found to be the best performer with 39.7% F1-macro and 74.2% F1-weighted on TFIDF vectors. Also, efficient structure of LightGBM is found to be a good learner with not only dense input spaces but also sparse matrices.

Although statistical approaches are still on stage with competent performances, NLP community has much more sophisticated tools in hand. As one of them, BERTurk is utilized as both a representation methodology and a pre-trained classification framework. Unlike other studies with critically outperforming performances of BERT relative to its rivals, embeddings' performance in this study falls closely behind the TFIDF, with 38.1% F1-macro with machine learning classifiers on CLS and 38.8% F1-macro with the pre-trained classification layer of BERTurk. One thing that may affect

the embeddings' performance negatively is the token length constraint of the model. Especially with the common formal tone of the announcements with generic sentence patterns, truncating tokens may discard meaningful parts of the text pieces. Truncated text pieces consisted of 12.4% of the input set with a nearly same class distribution with the overall data.

Consistent with the former studies on stock prediction using textual sources, in all experimental setups, public disclosures are found to be more influential on the investors when the tone is negative (Schumaker et al., 2012; Li et al., 2014). That may arise from the fact that existing investors of the stock are more concerned about the words from the company that may make them consider changing their positions on the shares they hold.

Even though public disclosures seem more relevant to the investors by nature, being committed by the company makes it harder to distinguish the tone of the language. On one hand, as the announcements are official documents placed by corporations, they mostly consist of generic words and their language is more formal than the other resources, such as news articles and social media messages. On the other hand, as the announcements are shaped by the subject of it, wording would be more careful and neutral than a news article which is written by an external person without any concern about the effect of it. Thus, it would not be a baseless claim to suggest that catching the real story of a public disclosure is harder than the other sources. However, results show that even without numeric data, KAP announcements give notable hints about the future moves of the market. Proven by the experiments, one could anticipate the right strategy to play by paying attention only to the disclosures. And moreover, an automated system for this task would indeed constitute a useful decision support unit.

In this study, an untouched problem is discussed with the help of NLP techniques and valuable former studies which presented great domain guidance. Predicting BIST companies' daily stock directions by examining companies' public disclosures is an exciting task which may bring out strong benefit to the investors by foreseeing what should be done under the circumstances of the related stocks. With the promising results observed in this study, prediction approaches with both the statistical and the deep configurations are believed to be highly useful for trading strategies at BIST market even when there is no preliminary price information about the traded companies. Over and above all, by processing the public announcements with both traditional methods and the latest advances of the NLP domain, this study presents a significant practice on Turkish textual data as it is intended to. Especially in the finance domain, NLP studies in Turkish languages are still very limited due to the populist attitude of NLP advances until so far. Though, the latest advances in the field give great opportunity to whom are concerned with relatively less popular languages. Ergo, in this study, it is mainly aimed to exploit the latest progress in the NLP by employing BERTurk model to predict the stock movement directions of BIST companies using their official declarations. It is hoped and believed that the experiments carried out in the scope of this study would present a notable work, so inspires and encourages others for more, who are eager to widen the Turkish NLP literature with their valuable efforts.

CHAPTER 6

LIMITATIONS AND FUTURE WORK

In this study, although the aim was to make the most of the textual resources, there were some constraints on data collection and processing steps.

Special case announcements are collected via an HTML crawler from the website of KAP. Although being able to fetch five years of announcements with a code block was an important advantage, attachments of the announcements could not be utilized in the study. Since some of the announcements give only brief explanation on the web page and disclose the important part via an attached file, essence of the case could not be gathered for this kind of cases.

Historical prices are mainly gathered from the Python API of Yahoo Finance. Because of unstable performance of the API, some of the ticker-stock date queries did not ended successfully, so the absent rows had to be filled with unadjusted prices of BIST data store file. Although the missing rows did not count much, unstable work of the API was a limitation to the study.

Public disclosures have a more formal and neutral structure compared to other textual sources proven to be useful for predicting stock changes, such as online news articles and social media messages. Being written by the companies, announcement explanations have less words with polarity than these sources, even when they disclose a negative incident. With this fact in hand, distinguishing an announcement's tone becomes harder than the sources written by third companies with no concern about the possible negative effects of it.

Although BERTurk framework is a novel methodology to employ to the task being discussed, limitations of the model caused some sacrifice from the input. Due to the maximum token length constraint, text pieces are truncated for 14.2% of the announcements. While this limitation would not cost much when working on news articles as they give the main message in early lines of the text, public disclosures do not share the same characteristic.

In this study, main goal is to make the most of the textual data with NLP techniques. To do so, experiments are focused on the announcements rather than numeric data, such as price histories of the stocks, government bonds, commodity prices. However, it is believed that the utilized data and learners would yield higher performance with additional numeric data, as their predictive power is notable even in the absence of it.

In this study, announcements are utilized for predicting daily directions of the stock movements. However, events that are declared by companies through an official platform may show their influence on the stock price in longer time spans. Time delay of the events' influences on stock markets are examined in many studies in literature (Lavrenko et al., 2000; Verma et al., 2017; Lee & Suh, 2018; Si et al., 2013; Li et al., 2016; Ding et al., 2014). From simultaneous to 90-day matchup, researchers investigated the best time window for observing the financial event effect on the markets they studied. Similar to these studies, a time frame dimension can be added to the experiments so as to discover the delay between the announcements and the stock movements.

Relative change calculation and standard deviation threshold are the corrections that made to distill the information effect from the stock movements. Though, prices can

also be used after being purified from seasonal effects and usual trends. In this way, event effect can be parsed clearly.

Although the pre-trained embeddings of BERTurk are resulted in promising performance measures, fine-tuning them on the study's own corpus would enhance the performance. As most of the announcements contain financial terms or sector specific concepts, context would be represented better by a fine-tuned model.

APPENDIX A

TOP 50 FEATURES OF THE BEST CLASSIFIER

Fold 1	mevcut, gereğince, birliği, üyelerinin, yetki, ederiz, açıklanmamış, açıklanan, buna, geçmiş, miktar, tarihine, proje, verilmiş, haklarının, borsada, geçen, karşılığı, bağış, dağıtılabilir, bedel, gören, biri, paylarımız, yolu, taraf, bankamızca, ortaklık, etmekte, kervansaray, değer, yine, eski, hak, farklı, karşılanmak, nitelikteki, ortaklarımıza, planlanmaktadır, toplantısının, ihale, yönünde, menfaat, sürelerinin, tamamlanması, değerinin, senetleri, hesaplarının, kayıtlara, yatırımcılarımıza
Fold 2	tebliği, gereğince, olağanüstü, yetki, yeniden, ederiz, üyelerine, kullanım, imzalanmıştır, temsil, açıklanan, borsada, işlemin, açıklanmamış, maddeleri, bilanço, bilgiler, kişiler, miktar, müzakeresi, bağışlar, sebebiyle, yine, geçen, başvurusu, dağıtılması, projesi, hukuki, gösteren, başlanmıştır, tamamlanması, oldukları, olmasına, etmiş, bilgilerinize, bankamız, yatırımcılarımıza, biçiminde, nitelikteki, karşılanmak, kaynakların, hizmet, denetimi, metro, açıklamaları, menfaat, eski, kefaletler, hakkındaki, bankamızca
Fold 3	ayrıca, yetki, üyelerine, geçmiş, açıklanmamış, kullanım, yeniden, temsil, imzalanmıştır, tarihine, buna, ali, haklarının, gibi, sebebiyle, başvuruda, elinde, nitelikteki, etmiş, hukuki, icra, yine, beyan, tahsisli, eski, takiben, başvurusu, üyeliklerine, kayıtlara, biçiminde, oldukları,

stratejik, kamunun, menfaatler, bedel, kervansaray, görüşmelere, başına, bulunulmasına, yıllar, yatırımcılarımıza, değerinin, bankamızın, tasarruf, olmasına, beher, bazı, kaynakların, almak, belirlendiği

APPENDIX B

RESULTS ON DISCRETIZED TFIDF VECTORS

Experiment	Down (-1)	Stationary (0)	Up (+1)	Macro	Weighted	Accuracy
Multinomial Naive Bayes						
TFIDF (original)	20.4%	79.8%	14.9%	38.4%	69.7%	65.4%
TFIDF (MI)	19.3%	84.1%	12.8%	38.7%	73.1%	71.4%
TFIDF (RFE)	14.0%	64.7%	14.7%	31.2%	56.6%	48.1%
TFIDF (SVD)	17.9%	85.7%	2.3%	35.3%	73.4%	74.0%
Logistic Regression						
TFIDF (original)	18.5%	80.4%	17.0%	38.6%	70.3%	66.4%
TFIDF (MI)	17.5%	69.9%	16.4%	34.6%	61.4%	53.7%
TFIDF (RFE)	18.5%	77.4%	15.7%	37.2%	67.7%	62.3%
TFIDF (SVD)	18.0%	74.6%	17.1%	36.6%	65.5%	59.0%
XGBoost						
TFIDF (original)	15.6%	90.5%	6.4%	37.5%	77.7%	82.1%
TFIDF (MI)	17.0%	89.9%	9.0%	38.6%	77.5%	81.0%
TFIDF (RFE)	19.8%	86.9%	12.0%	39.6%	75.4%	75.7%
TFIDF (SVD)	17.8%	85.4%	14.1%	39.1%	74.2%	73.7%
LightGBM						
TFIDF (original)	14.8%	90.4%	7.6%	37.6%	77.6%	81.8%
TFIDF (MI)	14.3%	89.9%	8.9%	37.7%	77.3%	81.0%
TFIDF (RFE)	18.9%	86.0%	13.3%	39.4%	74.7%	74.2%
TFIDF (SVD)	16.5%	85.9%	14.1%	38.8%	74.5%	74.4%
CatBoost						
TFIDF (original)	16.0%	90.1%	8.8%	38.3%	77.6%	81.3%
TFIDF (MI)	15.0%	90.0%	9.2%	38.1%	77.4%	81.1%
TFIDF (RFE)	19.1%	85.1%	14.7%	39.6%	74.1%	72.9%
TFIDF (SVD)	17.7%	84.2%	15.3%	39.1%	73.3%	71.9%

REFERENCES

Abualigah, L. M., & Khader, A. T. (2017). Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering. *The Journal of Supercomputing, 73*(11), 4773-4795. doi:10.1007/s11227-017-2046-2

Alpaydin, E. (2014). Introduction to machine learning. *3rd Ed.* London, England: MIT Press.

Aras, G., Makaroğlu, D., Demir, S., & Cakir, A. (2020). An evaluation of recent neural sequence tagging models in Turkish named entity recognition. *arXiv preprint arXiv:2005.07692*.

Aydoğan, M., & Karci, A. (2020). Improving the accuracy using pre-trained word embeddings on deep neural networks for Turkish text classification. *Physica A: Statistical Mechanics and its Applications, 541, 123288*. doi:10.1016/j.physa.2019.123288

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Balcilar, M., & Demirer, R. (2015). Effect of global shocks and volatility on herd behavior in an emerging market: Evidence from Borsa Istanbul. *Emerging Markets Finance and Trade, 51*(1), 140-159. doi:10.1080/1540496X.2015.1011520

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. *Ijcai*, *7*, 2670-2676.

Bildirici, M., & Ersin, Ö. Ö. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul stock exchange. *Expert Systems with Applications, 36*(4), 7355-7362. doi:10.1016/j.eswa.2008.09.051

Borsa Istanbul Historic and Reference Data Platform. (n.d.). Retrieved from https://datastore.borsaistanbul.com.

Borsa Istanbul. Trading Hours (n.d.). Retrieved March 1, 2020, from https://www.borsaistanbul.com/en/sayfa/2948/trading-hours

Boyacioglu, M. A., & Avci, D. (2010). An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: The case of the Istanbul stock exchange. *Expert Systems with Applications, 37*(12), 7908-7912. doi:10.1016/j.eswa.2010.04.045

Budur, E., Özçelik, R., Güngör, T., & Potts, C. (2020). Use of machine translation to obtain labeled datasets for resource-constrained languages. *arXiv preprint arXiv:2004.14963*.

Catal, C., & Nangir, M. (2017). A sentiment classification model based on multiple classifiers. *Applied Soft Computing, 50,* 135-141. doi:10.1016/j.asoc.2016.11.022

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16,* 321-357. doi:10.1613/jair.953

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Krishnapuram, B. (Ed.), *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). San Francisco, CA, USA.

Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2018). Leveraging social media news to predict stock index movement using RNN-boost. *Data & Knowledge Engineering, 118,* 14-24. doi:10.1016/j.datak.2018.08.003

Chen, Y., Liu, S., Zhang, X., Liu, K., & Zhao, J. (2017). Automatically labeled data generation for large scale event extraction. In Barzilay, R., & Kan, M. Y. (Eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 1* (pp. 409-419). Vancouver, Canada.

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In Zong, C., & Strube, M. (Eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1* (pp. 167-176.). Beijing, China.

Cohen, W. W. (1996). Learning rules that classify e-mail. In *AAAI Spring Symposium on Machine Learning in Information Access* (pp. 18-25).

Danilak, M. M. (2020). *langdetect (Version 1.0.8) [Computer software].* Retrieved May 1, 2020, from https://github.com/Mimino666/langdetect

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2014). Using structured events to predict stock price movement: An empirical investigation. In Pennington, J., Socher, R., & Manning, C. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1415-1425). Doha, Qatar.

Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). Deep learning for event-driven stock prediction . In Yang, Q., & Wooldridge, M. (Eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence* (pp. 2327-2333). Buenos Aires, Argentina.

Fama, E. F. (1965). The behavior of stock-market prices. *Journal of Business 38, 1*(1965), 34–105.

Gidofalvi, G., & Elkan, C. (2001). Using news articles to predict stock price movements. Department of Computer Science and Engineering, University of California, San Diego.

Gunduz, H., & Cataltepe, Z. (2013). Prediction of Istanbul stock exchange (ISE) direction based on news articles. *The Society of Digital Information and Wireless Communication*.

Gunduz, H., & Cataltepe, Z. (2015). Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection. *Expert Systems with Applications, 42*(22), 9001-9011. doi:10.1016/j.eswa.2015.07.058

Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2017). Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems, 137*, 138-148. doi:10.1016/j.knosys.2017.09.023

Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2018). Stock market prediction with deep learning using financial news. In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). Izmir, Turkey.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications, 38*(8), 10389-10397. doi:10.1016/j.eswa.2011.02.068

Heiberger, R. H. (2018). Predicting economic growth with stock networks. *Physica A: Statistical Mechanics and its Applications, 489*, 102-111. doi:10.1016/j.physa.2017.07.022

Hiew, J. Z., Huang, X., Mou, H., Li, D., Wu, Q., & Xu, Y. (2019). BERT-based financial sentiment index and LSTM-based stock return predictability. *arXiv preprint arXiv:1906.09024*.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science, 349*(6245), 261-266. doi:10.1126/science.aaa8685

Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T. Y. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In Wu, L., & Liu, H. (Eds.), *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 261-269). Marina Del Rey, CA, USA.

Huang, L., Ji, H., Cho, K., & Voss, C. R. (2017). Zero-shot transfer learning for event extraction. *arXiv preprint arXiv:1707.01066*.

Hugging Face. (2020). Retrieved from https://huggingface.co/dbmdz/bert-base-turkish-cased

Hursh, S. R., & Roma, P. G. (2016). Behavioral economics and the analysis of consumption and choice. *Managerial and Decision Economics, 37*(4-5), 224-238. doi:10.1002/mde.2724

Joachims, T. (1996). *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. School of Computer Science, Carnegie Mellon University, Pittsburgh.

Jones, K. S. (1994). Natural language processing: A historical review. In A. Zampolli, N. Calzolari, & M. Palmer (Eds.), *Current issues in computational linguistics: In honour of Don Walker* (pp. 3-16). Springer, Dordrecht.

Kara, Y., Boyacioglu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. *Expert systems with Applications, 38*(5), 5311-5319. doi:10.1016/j.eswa.2010.10.027

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference* (pp. 3146-3154). MIT Press.

Khedr, A. E., Salama, S. E., & Yaseen, N. (2017). Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications, 9*(7), 22. doi:10.5815/ijisa.2017.07.03

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *1990 IJCNN International Joint Conference on Neural Networks* (pp. 1-6). IEEE.

Kraus, M., & Feuerriegel, S. (2017). Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems, 104*, 38-48. doi:10.1016/j.dss.2017.10.001

Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000). Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining, 2000*, 37-44.

Lee, W., & Suh, B. (2018). Modeling Stock Prices with Text Contents in 10-Q Reports. In Kim, H. K., Miao, H., Ito, T., Yeo, H., Hong, C. S., Yeom, G. H., ... & Hwang, H. J. (Eds.), *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 224-229). Busan, South Korea.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research, 18*(1), 559-563.

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. In Toutanova, K., & Wu, H. (Eds.), *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 302-308). Baltimore, Maryland, USA.

Li, Q., Chen, Y., Jiang, L. L., Li, P., & Chen, H. (2016). A tensor-based information framework for predicting the stock market. *ACM Transactions on Information Systems (TOIS), 34*(2), 11. doi:10.1145/2838731

Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. *Information Sciences, 278*, 826-840. doi:10.1016/j.ins.2014.03.096

Li, X., Huang, X., Deng, X., & Zhu, S. (2014). Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information. *Neurocomputing, 142*, 228-238. doi:10.1016/j.neucom.2014.04.043

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. *Knowledge-Based Systems, 69*, 14-23. doi:10.1016/j.knosys.2014.04.022

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F. .., & Deng, X. (2016). Empirical analysis: stock market prediction via extreme learning machine. *Neural Computing and Applications, 27*(1), 67-78. doi:10.1007/s00521-014-1550-z

Liu, J., Lu, Z., & Du, W. (2019). Combining enterprise knowledge graph and news sentiment analysis for stock price prediction. In *Proceedings of the 52nd Hawaii International Conference on System Sciences* (pp. 1247-1255). Hawaii, USA.

Liu, Q., Cheng, X., Su, S., & Zhu, S. (2018). Hierarchical complementary attention network for predicting stock price movements with news. In Cuzzocrea, A. M., James, A., Paton, N. W., Divesh, S., Rakesh, A., Broder, A. Z., ... & Haixun, W. (Eds.), *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 1603-1606). Torino, Italy.

Makrehchi, M., Shah, S., & Liao, W. (2013). Stock prediction using event-based sentiment analysis. In Raghavan, V., Hu, X., Liau, C. J., & Jan, T. (Eds.), *The 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (pp. 337-342). Georgia, USA.

Martineau, J., & Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis. In Bastian, M., Heymann, S., & Jacomy, M. (Eds.), *Proceedings of the 3rd AAAI International Conference on Weblogs and Social Media* (pp. 258–261). San Jose, CA, USA.

Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access, 6*, 55392-55404. doi:10.1109/ACCESS.2018.2868970

Mizuno, H., Kosaka, M., Yajima, H., & Komoda, N. (1998). Application of neural network to technical analysis of stock market prediction. *Studies in Informatic and Control, 7*(3), 111-120.

Nam, K., & Seong, N. (2019). Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market. *Decision Support Systems, 117*, 100-112. doi:10.1016/j.dss.2018.11.004

Nguyen, T. H., & Shirai, K. (2015). Topic modeling based sentiment analysis on social media for stock market prediction. In Zong, C., & Strube, M. (Eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 1354-1364). Beijing, China.

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications, 42*(24), 9603-9611. doi:10.1016/j.eswa.2015.07.052

Oncharoen, P., & Vateekul, P. (2018). Deep learning for stock market prediction using event embedding and technical indicators. In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)* (pp. 19-24). Krabi, Thailand.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. .., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research, 12*, 2825-2830.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, 6638-6648.

Richardson, L. (2019). Beautiful soup documentation. Retrieved June 1, 2020, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Sakarwala, M. A., & Tanaydin, A. (2019). Use advances in data science and computing power to invest in stock market. *SMU Data Science Review, 2*(1), 17.

San Martín, R., Appelbaum, L. G., Huettel, S. A., & Woldorff, M. G. (2016). Cortical brain activity reflecting attentional biasing toward reward-predicting cues covaries with economic decision-making performance. *Cerebral Cortex, 26*(1), 1-11. doi:10.1093/cercor/bhu160

Savigny, J., & Purwarianti, A. (2017, August). Emotion classification on youtube comments using word embedding. In Farhan, A. N., Khodra, M. L., Tomioka, S., Sougawa, H., Ishimura, H., Okamoto, A., ... & Blume, Y. (Eds.), *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)* (pp. 1-5). Denpasar, Indonesia.

Schumaker, R. P., Zhang, Y., Huang, C. N., & Chen, H. (2012). Evaluating sentiment in financial news articles. *Decision Support Systems, 53*(3), 458-464. doi:10.1016/j.dss.2012.03.001

Shynkevich, Y., McGinnity, T. M., Coleman, S., & Belatreche, A. (2015, July). Stock price prediction based on stock-specific and sub-industry-specific news articles. In Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., & Pfeiffer, M. (Eds.), *2015 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). Killarney, Ireland.

Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013). Exploiting topic based twitter sentiment for stock prediction. In Schütze, H., Fung, P., & Poesio, M. (Eds.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 24-29). Sofia, Bulgaria.

Si, J., Mukherjee, A., Liu, B., Pan, S. J., Li, Q., & Li, H. (2014). Exploiting social relations and sentiment for stock prediction. In Pennington, J., Socher, R., & Manning, C. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1139-1145). Doha, Qatar.

Sorto, M., Aasheim, C., & Wimmer, H. (2017). Feeling the stock market: a study in the prediction of financial markets based on news sentiment. In *Proceedings of the Southern Association for Information Systems Conference* (pp. 1-8). St. Simons Island, Georgia, USA.

Taylor, W. L. (1953). "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly, 30*(4), 415-433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems* (pp. 5998-6008). Long Beach, CA, USA.

Verma, I., Dey, L., & Meisheri, H. (2017). Detecting, quantifying and accessing impact of news events on Indian stock indices. In Alt, R., Tao, X., & Unland, R. (Eds.), *Proceedings of the International Conference on Web Intelligence* (pp. 550-557). Leipzig, Germany.

Wang, J. H., Liu, T. W., Luo, X., & Wang, L. (2018). An LSTM approach to short text sentiment classification with word embeddings. In Lee, C. C., Yang, C. Z., & Chien, J. T. (Eds.), *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing* (pp. 214-223). Hsinchu, Taiwan.

Wilmers, N. (2018). Wage stagnation and buyer power: How buyer-supplier relations affect US workers' wages, 1978 to 2014. *American Sociological Review, 83*(2), 213-242. doi:10.1177/0003122418762441

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W. .., & Klingner, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Wüthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., & Zhang, J. (1998). Daily prediction of major stock indices from textual www data. *Hkie Transactions, 5*(3), 151-156.

Yang, L., Xu, Y., Ng, J., & Dong, R. (2019). Leveraging BERT to improve the FEARS index for stock forecasting. In Chen, C. C., Huang, H. H., Takamura, H., & Chen, H. H. (Eds.), *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 54-60). Macao, China.

Yilmaz, K. E., & Abul, O. (2018). Inferring Political Alignments of Twitter Users. In *2018 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). Rome, Italy.

Zhang, X., Qu, S., Huang, J., Fang, B., & Yu, P. (2018). Stock market prediction via multi-source multiple instance learning. *IEEE Access, 6*, 50720-50728. doi:10.1109/ACCESS.2018.2869735