PREDICTION OF ISTANBUL STOCK EXCHANGE (ISE) DIRECTION BASED ON NEWS ARTICLES

Hakan Gunduz and Zehra Cataltepe
Computer Engineering Department, Faculty of Computer and Informatics,
Istanbul Technical University, Istanbul, TURKEY
E-mail= { hakangunduz, cataltepe}@itu.edu.tr

ABSTRACT

In this paper, we examined the effects of financial news on Istanbul Stock Exchange and we tried to predict the direction of ISE National 100 Index open price after the news articles were published. In order to do this study, we got news articles from three big financial websites and we represented them as feature vectors. We obtained class labels from the ISE National 100 Index open price and assigned them to these feature vectors. After creating the datasets, we selected the informative features using Mutual Information (MI) and Term Frequency-Inverse Document Frequency (TF-IDF) Weighting methods. According to the selected features, we created feature subsets with different number of features and we trained a Naïve Bayes Classifier with them. We used precision, recall, accuracy and F-Measure metrics to evaluate the performances of our methods. We found out that feature selection can help us use a smaller number of features for classification and as a measure of how useful a feature is, TF-IDF is better than MI.

KEYWORDS

Stock market direction prediction, Mutual Information, TF-IDF Weighting, Naïve Bayes Classifier.

1 INTRODUCTION

Stock market prediction has been an attractive research topic in recent years. Because, there are many investors interested in making good profits from stock exchange. But it is hard to predict the future stock price or direction because of the complex and dynamic nature of stock markets and many factors that affect the stock market behavior.

In order to predict the future stock price or direction, there are two analysis methods used in the literature [1][2]. The first method is technical analysis which is based on historical stock prices. By examining the daily, monthly and yearly past prices of the stock, the future stock price or stock direction is predicted. In this approach, Time Series Analysis techniques are used and market timing is very important [3][4]. The second method is fundamental analysis and this analysis includes the important numbers about the structure of the economy. When the fundamental analysis is used, the features like interest rates, inflation, unemployment percentage and economic growth are considered to make a prediction [5]. In the last 10 years, alternative analysis choices have been investigated for stock market prediction. With the development of information technologies, we can access abundant information about finance especially through financial news on web sites. Several studies showed that there is a direct link between the stock markets and financial news [6][7].Thus, financial news which include general news articles, company releases, news of global economy affect the performance of stock market and we can use them to predict the future stock prices and direction of the stocks.

If we use the financial news articles to make the prediction, we need to extract important information that may have an effect on the stock market. News articles found on World Wide Web are unstructured and we need to convert them into structured form in order to analyze patterns in these articles. Natural Language Processing and Data Mining methods can be used to create feature vectors. Each element in a feature vector is named a term and it represents the words that occur in a news document [8].

In this paper, we aim to learn how news articles affect the direction of Istanbul Stock Exchange (ISE) National 100 Index. In order to achieve our goal, we collected news articles from three important finance websites in Turkey. These news articles contained general financial news, expert advice reports and company releases. We also aim to find words in news articles which have more effect than the others on the market. We used text mining techniques to analyze the news articles. After the analysis process, the articles were converted to feature vectors. Three class labels, "UP, DOWN and NEUTRAL" were assigned to these feature vectors according to the new articles publishing date and the opening price of the ISE National 100 Index. These feature vectors had thousands of features and we eliminated features which were not informative. We did feature selection process using Mutual Information (MI) and TF-IDF Weighting. We trained a Naïve Bayes classifier that uses the news articles to predict the direction of ISE National 100 Index. After the classification process, the classifier performance was measured by recall, precision, F1-Measure and accuracy metrics.

The remaining part of the paper is organized as follows. In the next section, we summarize Related Works about Stock Market Prediction. We describe data set preparation in Section 3.Section 4 gives information about Feature Selection Methods, Classification Process and Evaluation Metrics. In Section 5, Experiment Results are shown. Section 6 contains Conclusion and Future Works.

2 RELATED WORKS

There are lots of studies about stock market prediction in the literature and machine learning techniques like Artificial Neural Networks, Support Vector Machines, Genetic Algorithms and Hidden Markov Models were applied them [9][10][11][12]. In these works, numeric data (e.g. historical stock prices) and technical indicators were used. The idea of using the news articles to predict the stock market behavior was emphasized with Mitchell and Mulherin's research [7]. In this study, the direct relation between Dows Jones Announcements and the stock market was detected. Researchers also examined the intraday impacts of the news articles on the stock markets. In [13], Mittermayer examined the stock market direction 15 minutes after the publication release time. In this work, they used the Support Vector Machines for classification. In Gidofalvi's work, a Naïve Bayes Classifier was used to predict the direction of stock price of 12 tick-by-tick price data with 10 minute intervals were used to assign labels to news articles [5].

In the stock market prediction studies, in order to do feature selection from thousands of features, methods like Term Frequency-Inverse Document Frequency (TF-IDF) on Bag of Words (BOW) features were used [14].

There are a number of studies on Istanbul Stock Exchange. ISE returns were predicted using Artificial Neural Networks [11] and Adaptive Neuro Fuzzy Inference System (ANFIS) [15]. There is also a study that used technical indicators to predict the ISE Direction [16]. However, to the best of our knowledge, there aren't any studies to predict the stock market behavior using financial news articles.

3 DATASET

Our document collection contained 111587 news articles in Turkish, published between January 1st, 2010 and December 31st, 2011. These articles were in an unstructured form. In order to use them in this study, first, we preprocessed the news articles as follows:

Stop Word Removal: We found Stop Words list in Turkish prepared by Fatih University Natural Language Processing Group and we eliminated Turkish Stop Words from our document collection.

Stemming: In order to do stemming, we used **Zemberek [17],** Turkish Language Processing Framework written in JAVA, and we found the root forms of all words and reduced the dimension of word space. Turkish is an agglutinative language, thanks to stemming we can replace a set of words like "okul", "okudu","okuyacak","okutman" with their root "oku".

After these operations, we got a set of 13800 unique words from our document collection.

After finding the unique words, we represented each news article as a vector (Bag of Words (BOW) representation). Each dimension of this vector is a word. The sequence of words is not important in "Bag of Words". The vector representation of the d^{th} news article is shown in Eq. 1:

$$\mathbf{x}(\mathbf{d}) = \lfloor w_1, w_2, w_3, w_4, \ldots w_n \rfloor \quad (1)$$

$\mathbf{x}(\mathbf{d})$ is also named as a feature vector with m dimension. Each element of the feature vector is a word and if the word occurs in the news article, the value of the element will be 1, otherwise it will be 0. When we created the feature vectors from all documents in our document collection, we had a feature matrix that had 111587 rows and 13800 columns.

Only the occurrences of the words in a document do not give us idea about the effects of words on class labels. In this situation, we need to look at the number of word occurrences in a document and a document collection. For this reason, we formed a second dataset whose feature values were found by Term Frequency-Inverse Document Frequency (TF-IDF) Weighting. TF- IDF Weighting Scheme uses the following expression to calculate the feature values:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \qquad (2)$$

TF–IDF includes two statistical metrics, term frequency and inverse document frequency. The term frequency $tf(t,d)$ is the number of times that term (word) t occurs in document d. The inverse document frequency $idf(t,D)$ is a measure that shows the rarity of a term (Word) t across in the document collection D. It is found by dividing the total number of documents in document collection by the number of documents containing the term, and then taking the logarithm of that quotient. After tf and idf values for each term was found, the products of tf and idf gave TF–IDF values in the feature vectors. If a term had a high TF-IDF value in the feature vectors, it could be said that the term had a high term frequency (in the given document) and a low document frequency in the whole collection of documents.

After creating feature matrices in two representations, we assigned class labels to every feature vectors in them. We considered the release time of financial news to assign the class labels. In order to find the direction of the Istanbul Stock Exchange (ISE) National 100 Index open price at a day after the news articles became publically available, we examined the rate of ISE National 100 Index open price change day by day to create a label for each day between 2010 and 2011. Eq. 3 shows how the rate of open price change was calculated for day ***t***.

$$\text{Rate of Open Price Change} = \frac{s(t) - s(t-1)}{s(t-1)} \qquad (3)$$

In Eq. 3, $s(t)$ shows open price of day ***t***, $s(t-1)$ shows the open price on the day before ***t***. We found Mean and Standard Deviation (std.) values of rate of change during the two years and we assigned the labels to each day by considering them. If the rate of change in day ***t*** was bigger than the sum of mean and std., we assigned the label “UP” to this day. If it was smaller than the difference between mean and STD values, the label was assigned as “DOWN”. The label needed to be “NEUTRAL” if the mentioned cases above were not suitable. These formed class labels were assigned to the news articles considering the news publication time.

4 METHODS

4.1 Feature Selection

When we observed our datasets, the feature matrices had thousands of features. So, we needed to do dimensionality reduction on the feature vectors in order to throw away non-informative features. The first process that we did was to look at document frequency of the terms. We got the words which occurred more than 1000 times in document collection and the number of features in the feature matrix were reduced from 13800 to 2888. After the elimination of features according to document frequencies, we examined the relation between selected features and the class labels using Mutual Information.

Mutual Information (MI)

Mutual information is a quantity that measures mutual dependence between two random variables. In our research, we examined the dependence of the words that occurred in the financial news and the class labels that were assigned to these documents. We used MI in dataset whose feature values were binary and eliminated features (words) that were not dependent on class labels. The feature vectors and class vectors had discrete values so we calculated MI between features and class labels using Eq. 4.

$$I(W;C) = \sum_{w \in W} \sum_{c \in C} p(w,c) \log\left(\frac{p(w,c)}{p(w)p(c)}\right) \qquad (4)$$

In Eq. 4 is joint probability distribution function $p(w,c)$ of feature w and class vector c, $p(w)$ and $p(c)$ are marginal distribution functions of w and c respectively. W and C denote the set of all discrete values that w and c could take respectively.

4.2 Naïve Bayes Classifier

In this study, we used a Naïve Bayes (NB) Classifier which is based on Bayes Rule. It gives a good performance on text categorization problems [18]. NB Classifier finds the probability model using conditional probabilities of the features to compute the probability of an instance belonging to a class.

In this model, we tried to find $p(c \mid w_1,\ldots,w_n)$, where each w was the value for each feature in our feature matrix and c was the movement direction of the ISE 100 Index. $p(c \mid w_1,\ldots,w_n)$ is referred as posterior probability. And in NB classifier, we used the instances to find the posterior for each class; we assigned the instances to the class with the highest posterior probability. But it is difficult to find the posterior probabilities directly from the instances that have large feature dimensions. In this situation, we used Bayes' rule, which is shown in Eq. 5.

$$p(c \mid w_1,\ldots,w_n) = \frac{p(c)p(w_1,\ldots,w_n \mid c)}{p(w_1,\ldots,w_n)} \qquad (5)$$

In Eq. 5, $p(c)$ is the prior probability and it is computed by looking the class distributions in training set. $p(w_1,\ldots,w_n)$ is called as evidence. Evidence has no dependence on class and it can be ignored when computing the posterior. $p(w_1,\ldots,w_n \mid c)$ is likelihood and computing the likelihood is more difficult than the prior probabilities. Because, the conditional probability in likelihood is dependent on all of the features. To simplify the computation, a NB classifier considers conditional independence of the features. It means that each feature is conditionally independent of every other feature. Computing the likelihood considering this naïve assumption is shown in Eq.6.

$$p(w_1,\ldots,w_n \mid c) \,\alpha\, p(c) \prod_{i=1}^{n} p(w_i \mid c) \qquad (6)$$

The effects of this assumption were examined in [19]. In [19], Zhang explained the classification performance of NB algorithm by looking into the dependencies between features. It was stated that even if there were strong dependencies between features, the NB classifier would still have fairly good performance.

4.3 Evaluation Metrics

The evaluation metrics measure the classifier ability of identifying classes correctly in the classification process. In machine learning applications, the classifier performances are generally measured by accuracy. Accuracy is found by dividing the total number of correctly predicted instances by total number of instances. Some classification problems contain an imbalanced class distributions and one class dominates the other classes. When we assign the dominating class label to all instances in the test set and measure the prediction performance in accuracy, our classifier success will be high, but the classification ability of classifier is not good due to an unsuccessful prediction performance of minor classes in the test set. So, accuracy is not necessarily a suitable metric to evaluate performance of each class in the classification process for unbalanced data.

Instead of accuracy, we used three measures that evaluate the classifier performance on the different classes separately. These measures are precision, recall and F-measure. Before defining these measures, we need to examine a confusion matrix which shows correct and incorrect predicted instances in each class according to the actual class labels.

In Table 1, there is a confusion matrix for two-class classification task. In this matrix, tp and fp show true positive and false positive counts respectively. fn is false negative, and tn is true negative counts.

Table 1. Confusion matrix for two-class classification.

Actual/Predicted as	Positive	Negative
Positive	tp	Fn
Negative	fp	Tp

Confusion matrix shows the ability of a classifier to distinguish the different classes. Its elements are used in calculating recall, precision and F-measure as shown below:

$$\text{recall} = \frac{tp}{tp + fn} \tag{7}$$

$$\text{precision} = \frac{tp}{tp + fp} \tag{8}$$

$$\text{F} - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{9}$$

All three measures have the ability of evaluating performance on different classes. If there are more than 2 classes in classification, we can separately select each class and accept it as positive, the remaining classes will evaluated as negative. Thus, we look at prediction performance of classifier at each class level.

5 EXPERIMENTS AND RESULTS

As we mentioned in Section 3, we formed two datasets with feature values in binary representation and TF-IDF Weighting. These datasets were divided into two parts in order to train and test the classifier. News articles publication date was considered for this separation. The publication period of the articles was two years. In the two-year period, the articles published in the first 18 months were used to train the classifier and the articles published in the remaining 6 months were used to test the classifier performance.

When we looked at the class distributions in our datasets, “NEUTRAL” class had dominance over other classes, “UP” and “DOWN”. The class counts are shown in Table 2.

Table 2. Number of class counts.

Class	Training Set	Test Set
Up	7956	5262
Neutral	65726	18791
Down	8802	5050

The datasets had 2888 features (words) and when we used to The WEKA data mining software [20] to train NB classifier with them, we encountered with memory problem due to thousands of features. So, in order to start the training phase, we did the dimensionality reduction with feature selection methods, Mutual Information (MI) and TF-IDF Weighting.

In MI, we examined the relation between terms and class labels in binary valued training set. Figure 1 shows the relationship between words and class labels according to MI.

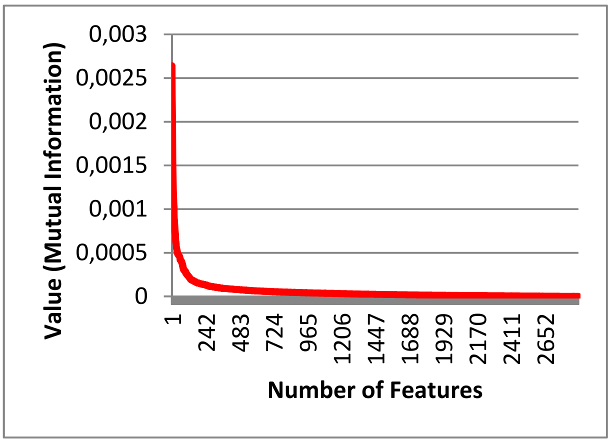

Figure 1. The relationship between features and class labels according to MI value.

Figure 1 shows that after a certain point, MI values of the features converge to zero. So, we selected the first 500 words that had higher MI values as features. After choosing the features, we created 7 different training subsets which had 10, 20, 50, 100, 200, 300 and 500 features respectively. Same features were selected for the test sets.

In order to use the feature vectors that had TF-IDF values, we summed up the TF-IDF values for each features in all documents and we sorted all terms from high to low by total TI-IDF’s. Again, we selected the first 500 words as the features and we created 7 feature subsets for the training and test sets.

Table 3. The classification results of 7 feature subsets (Binary Feature Values)

No of Features	Precision			Recall			Acc
	Down	Neut.	Up	Down	Neut.	Up	
10	0,156	**0,645**	0,125	0,003	0,989	0,003	**0,64**
20	**0,194**	0,645	0,122	0,002	**0,991**	0,004	0,64
50	0,163	0,642	0,179	0,002	0,942	0,049	0,61
100	0,194	0,645	**0,191**	0,010	0,889	0,106	0,59
200	0,191	0,645	0,178	0,024	0,736	0,236	0,52
300	0,191	0,644	0,176	0,023	0,696	0,273	0,50
500	0,195	0,645	0,176	**0,031**	0,670	**0,293**	0,49

The classification results of binary valued feature subsets are shown in Table 3. When we looked at the classifier performance at class level, for the class "NEUTRAL", we got 99,1% recall rate with 64,5% precision only using 20 features. The accuracy rate was 64% in this classification. But, with these 20 features, the classification performance is too low for "UP" and "DOWN" classes. When we increased the number of features to 500, the recall rate increased up to 3,1% for the class "DOWN" and 29,3% for the class "UP" with precision rate 19,5% and 17,6% respectively. However, the recall rate decreased to 67% for class "NEUTRAL" and with 500 features; we got the least accuracy percentage in all feature subsets.

After examining the binary valued feature subsets, we evaluated the subsets which are TF-IDF Weighted. Table 4 shows the precision, recall and accuracy rates using 7 feature subsets. In these subsets, "NEUTRAL" class got the highest recall rate with 10 features. The recall rate is 95,4% with 64,6% precision.

But, the recall rates for classes "UP" and "DOWN" were nearly zero with 10 features. In this experiment, the best recall rates for classes "DOWN" and "UP" were obtained with subset having 500 features. The recall rates are 19,6% and 6,6% for "UP" and "DOWN" respectively. Using more features to classify, the identification ability of the classifier increased despite the fact that the accuracy rates for all feature sets went down.

Table 4. The classification results of 7 feature subsets (TF-IDF Weighting Feature Values)

No of Features	Precision			Recall			Acc
	Down	Neut.	Up	Down	Neut.	Up	
10	0,000	0,646	0,170	0,000	**0,954**	0,044	**0,62**
20	0,143	0,646	0,181	0,009	0,933	0,057	0,61
50	**0,203**	0,647	0,169	0,036	0,838	0,124	0,56
100	0,192	**0,648**	0,175	0,053	0,803	0,147	0,55
200	0,183	0,646	0,174	0,058	0,776	0,163	0,54
300	0,187	0,646	**0,183**	0,062	0,756	0,189	0,53
500	0,184	0,646	0,180	**0,066**	0,741	**0,196**	0,52

We used precision and recall rates to find F-measure rates for each class. The rates are shown in Table 5. F-Measure evaluates the distinguishing performance of classifier on class based. In order to evaluate the overall performance of our classifier; we used Macro-Averaged F-Measure (M.A. F-Measure). In M.A. F-Measure, firstly F-Measure for each class is found and then the mean of F-Measure values for all classes is taken [21]. Macro-averaged F-measure gives equal weight to each class.

Table 5. F-Measure Rates for 7 feature subsets (TF-IDF Weighting and Binary Feature Values)

No of Features	TF-IDF Weighting			Binary		
	Down	Neut.	Up	Down	Neut.	Up
10	0,000	**0,770**	0,070	0,006	0,781	0,008
20	0,016	0,764	0,087	0,005	**0,782**	0,008
50	0,061	0,730	0,143	0,006	0,764	0,078
100	0,083	0,717	0,160	0,021	0,748	0,137
200	0,089	0,705	0,169	0,044	0,688	0,203
300	0,093	0,696	0,185	0,042	0,670	0,214
500	**0,097**	0,690	**0,188**	**0,053**	0,657	**0,220**

In Figure 2, the chart shows the Macro-Averaged F-Measure values for two datasets. In these datasets, the dataset which had TF-IDF representation in features had superiority against the binary-valued dataset. When we looked at all feature subsets, the highest Macro-Averaged F-Measure value was obtained from 500 features in TF-IDF Weighting and if we wanted to get acceptable performances from two datasets, we could only use 200 features. According to the highest M.A F-measure value in binary valued dataset, using TF-IDF weighting increased the performance rate nearly 4%.

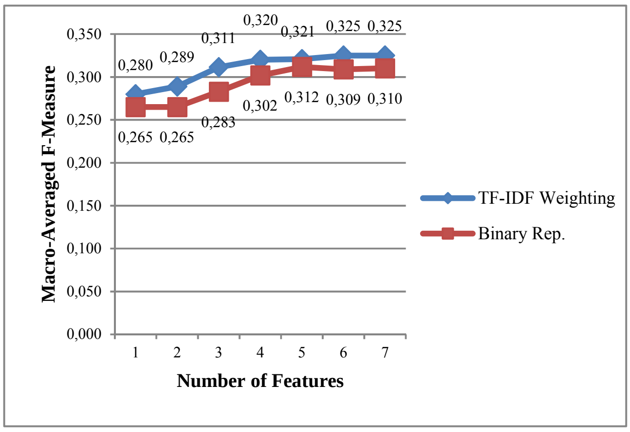

Figure 2. Macro-Averaged F-Measure value for Binary and TD-IDF Weighting Feature Values

After the classification process, we detected the most informative words in two datasets. Table 6 shows the top 10 informative words.

Table 6. The top 10 informative words in two datasets.

Words(Turkish/English)	
TF-IDF	**MI**
Kurul/**committee**	Şubat/**february**
Done/**data**	Mayıs/**may**
Yüzde/**percentage**	Rapor/**report**
Varlık/**asset**	Done/**data**
Tablo/**table**	Tekrar/**repetition**
Ol/**become**	Car/ **prevailing**
Yıl/**year**	Referans/**reference**
Genel/**general**	Azınlık/**minority**
Yatırım/**investment**	Ağustos/**august**
Pay/**share**	Eylül/**september**

6 CONCLUSIONS AND FUTURE WORK

In this paper, we investigated the use of the financial news articles in Istanbul Stock Exchange National 100 Index direction prediction. Unlike other ISE prediction studies, we used the words occurred in the news articles as features and we tried to predict the direction of opening price in the day after article publication date. We formed two dataset with different representations and used the Naïve Bayes Classifier to do the classification process. When we examined the results obtained from the two datasets, increasing the number of features to train the classifier improved the recall rates of “UP” and “DOWN” classes. But, the recall rate of class “NEUTRAL” decreased with increasing feature set sizes. Also, with the use of features that had TF-IDF values, we got higher M.A F-Measure values than the features with binary values. This situation did not change with the number of features that were used. After these experiments, we concluded that TF-IDF Weighting in feature values could improve the classifier performance at the rate of between 4% and 10%.

As a future study, we are working on different feature representations and feature selection methods to improve our classification performance. We also want to be able cope with the class imbalance problem.

7 REFERENCES

1. Elkan, C., Notes on discovering trading strategies (1999).
2. Hellström, T., Holmström, K., Predicting the Stock Market, Technical Report Series IMATOM- 1997-07, (1998).
3. Marcek, D., Stock price forecasting: Statistical, classical and fuzzy neural network approach, in MDAI, ser. Lecture Notes in Computer Science, V. Torra and Y. Narukawa, Eds., vol.3131. Springer, (2004), pp. 41–48.
4. Abdullah, M.H.L.b, Ganapathy, V.: Neural Network Ensemble for Financial Trend Prediction. Proc. TENCON 3, (2000), pp. 157-161.
5. Gidófalvi, G., Using News Articles to Predict Stock Price Movements, (2001).
6. Wuthrich, B., Permunetilleke, D., Leung, S., Cho, V., Zhang, J., Lam, W., Daily prediction of major stock indices from textual www data, in KDD, (1998), pp. 364–368.
7. Mitchell, M.L., Mulherin, J.H., The impact of public information on the stock market, The Journal of Finance Vol. 49, No. 3, Papers and Proceedings Fifty-Fourth Annual Meeting of the American Finance Association, Boston, Massachusetts, January 3-5, 1994 (Jul., 1994), pp. 923-950
8. Schumaker, R.P., Chen, H., A Quantitative Stock Prediction System based on Financial News, Information Processing and Management: an International Journal, Volume 45 Issue 5, September, (2009) pp. 571-583.
9. Kim, K., Han, I., Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index. Expert Syst. Appl. 19 (2), (2000), pp. 125–132.
10. Kim, K. J., Financial time series forecasting using support vector machines, Neurocomputing, 55(1/2), (2003), pp. 307–319.
11. Bildirici, M., Ersin, Ö.Ö., 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, (2009), pp. 7355–7362.
12. Hassan, M.R., Nath, B., Stock Market Forecasting Using Hidden Markov Model: a new approach. Proc. Of 5th Int. Conf.on intelligent systems design and applications (2005).
13. Mittermayer M., Forecasting intraday stock price trends with text mining techniques, 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, vol. 00, (2004), pp. 64-73.
14. Nikfarjam, A., Emadzadeh, E., Muthaiyah, S., Text mining approaches for stock market prediction, The 2nd International Conference on Computer and Automation Engineering (ICCAE), (2010), pp.256-260.
15. Boyacioglu, M.A., Avci, D., 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, Volume 37, Issue 12, December 2010, pp. 7908-7912.
16. Kara, Y., Boyacioglu, M.A., Baykan, Ö.K., 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, (2011), pp. 5311-5319.
17. Akın A.A., Akın M.D, Zemberek, an open source NLP framework for Turkic Languages, (2007).
18. McCallum, A., Nigam, K., A comparison of event models for Naive Bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization, (1998), pp. 41-48.

19. Zhang, H., The optimality of naive Bayes, Proceedings of the 17th International FLAIRS conference (FLAIRS2004), Menlo Park, CA, AAAI Press, (2004), pp. 562-568.
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, (2009), pp. 10-18.
21. Özgür, A., Özgür, L., Güngör, T., Text categorization with class-based and corpus-based keyword selection. In Proceedings of the 20th International Symposium on Computer and Information Sciences, volume 3733 of Lecture Notes in Computer Science, Springer-Verlag, (2005), pp. 607–616.