Sentiment Analysis and Reporting from Text Data in Film Reviews

Egehan Eralp
Computer Engineering
Istanbul University - Cerrahpasa
Istanbul, Turkey
egehaneralp@outlook.com

Abdullah Oğuz Türk
Computer Engineering
Istanbul University - Cerrahpasa
Istanbul, Turkey
oguzturk99@outlook.com

Kerem Solmaz
Computer Engineering
Istanbul University - Cerrahpasa
Istanbul, Turkey
keremsolmaz@protonmail.com

***Abstract*—Human and communication are inseparable. For this reason, it is very important to be able to express feelings towards the other side in people and their communication. Nowadays, with the digitalizing world, people perform most of their communication in a written form in computer environment. The inability of these text data to express feelings and thoughts is a big problem in terms of communication. In order to eliminate or minimize this problem, it is very important to analyze the sentiments contained in the text data. In this context, many input representation models (Word2Vec, Doc2Vec) and text classification models (Naïve Bayes, Decision Tree, Logistic Regression, LSTM) were developed and evaluated in our study. As a result of the evaluations, the sentiment contained in the text data in the film reviews and the score corresponding to the text were estimated successfully with the most successful models. A report is presented to the user/institution along with the analysis of the user-based data generated as a result of the estimations.**

Index Terms—Sentiment Analysis, Movie Reviews, Text Classification, Machine Learning, Deep Learning

I. INTRODUCTION

With the developing technology, it is seen that the number of documents produced in the computer environment is accelerated every year. Some problems, that also need to be resolved, arise with the numerous benefits of a large number of information. On this context, main problems that arise are the classification of texts in electronic environment and the analysis of those texts. The text classification problem is, in its most general definition, determining which of the predetermined classes a text belongs to or to which ones.

In today's world, text classification applications are very common, but the text analysis made specifically for movie reviews is very limited in technical and practical terms. Since the text data in film reviews are subjective, comprehensive analysis is a must. Therefore, analyzing words in the singular is insufficient, sentences should be evaluated as a whole.

With the successful results obtained from the movie evaluations, the trends and preferences of the users are analyzed. The analysis results mentioned are very important both for the users and for the institutions providing the evaluation service. Users can use this system to learn and observe the feelings/sentiment of other users on any movie. On the other hand, it is possible for service providers to analyze what kind of users are interested in what kind of movies and to create action plans accordingly.

With this article study, more advanced evaluations have been obtained with more advanced input representation and text classification models, unlike the conventional film evaluation systems today. With the application created within the scope of the article, the user will be able to review the films they want in written form, without spending time deciding on the rating. In addition, by accessing the analysis of the review data of the movie, the user will be able to learn more about the movie, and thus, the user will be able to decide whether the movie is worth watching or not, thanks to the reports presented in the application with minimum time loss. For this purpose, the visuals in the report have been created in a way that the users can easily understand and then quickly take actions.

The rest of the article is organized as follows: Part 2 includes a review and summary of some of the work done in natural language processing, text classification and sentiment analysis. Chapter 3 specifies and explains the tools, programming languages and libraries used in the creation of the system developed within the scope of the article. Chapter 4 explains the system implementation step-by-step, highlighting the key actions at each step. Chapter 5 gathers the findings under a single roof and specifies the decisions to be taken in line with the results.

II. RELATED WORK

In the digitalizing world, the number and importance of text data is increasing day by day. The increasing number of data brings with it the need to obtain information from data. Analyzing or classifying emotions in text data is carried out to extract information from raw data. For this reason, the number of studies aiming to analyze emotion in text data has increased rapidly. In this section, some literature studies related to the subject are included.

Q. Le and T. Mikolov evaluated the traditional methods used in text classification and sentiment analysis studies and mentioned alternatives in their study. In the study, it is mentioned that obtaining only the numerical equivalents of the sentences, especially the "bag-of-words" and similar methods, is insufficient in terms of analyzing their meanings. The study focuses on the importance of creating a paragraph vector for each sentence whose meaning will be analyzed [1].

B. Lakshmi Devi, V. Varaswathi Bai, S. Ramasubbareddy and K. Govinda used Naive Bayes Classifier and Decision Tree Classifier from Sci-Kit Learn library in their study and classified emotion in movie reviews. The emotion classification performed is an example of Binary classification, and it was created to predict only the 'negative' and 'positive' content of the text [2].

In the study carried out by A. Apoorv, X. Boyi, V. Ilia, R. Owen, and P. Rebecca in 2011, it was aimed to analyze the emotions in Twitter posts. The data used for the text classification models consists of 11,875 lines and 4 different target emotion classes. Included in the text data, the 5000 most frequently used abbreviations in English and the original were matched and used for analysis. Apart from text data, classification and evaluation were also carried out on the basis of 'emote's [3].

In the study carried out by R. Gupta and S. Gill Nasib in 2012, predictions about whether there is fraud were developed according to the financial statements in the data set they obtained by using Multilayer Feed Forward Neural Networks, SVM and Logistic Regression models. Similar to the other study in the literature, Binary text classification over two target classes is seen in this study [4].

In the study carried out by G. Jain, B. Manisha and B. Agarwal in 2017, the prediction of whether messages transmitted on any social network are spam is made with various machine learning and deep learning models. SVM, Naive Bayes, ANN and Random Forest were used as classification models. In addition to prediction models, bag-of-words, Term Frequency and TF-IDF methods are mentioned in text classification. The text classification performed is an example of Binary classification, and it is based only on the prediction of 'spam' and 'not spam' [5].

R. Prabowo and M. Thelwall aimed to use different classification and machine learning methods together in this study. A new unified method was created by combining rule-based classification, supervised learning and machine learning. This method has been tested with movie reviews, product reviews, and MySpace reviews. It has been suggested that each classification method will make a good contribution to other methods [6].

Lin C. and Y. He in their study; They proposed a new probabilistic modeling structure called the joint sentiment/topic model (JST), which simultaneously detects emotion and topic from the text. This structure is based on Latent Dirichlet Allocation (LDA). Preliminary trials of this JST model using an unsupervised learning approach have shown promising results [7].

In this study carried out by Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts; It has been mentioned that unsupervised vector-based approaches can make rich lexical models, but they cannot capture the emotional information that is central to the words. To solve this problem, a model using a mixture of supervised and unsupervised techniques is presented [8].

In this study, S. Kiritchenko, X. Zhu and S. M. Mohammad described a sentiment analysis system that detects emotion in short and informal text messages such as tweets and SMS, and detects the emotion of a word or phrase in these texts. The system is based on the supervised statistical text classification approach [9].

In this study carried out by X. Fang and J. Zhan, they addressed the issue of emotion polarity categorization, which is a sentiment analysis problem. The data used in the study were taken from product reviews on Amazon.com. Experiments for both sentence-level classification and evaluation-level classification yielded promising results [10].

In this study by N. Godbole, M. Srinivasaiah and S. Skiena, a text analysis system developed for newspapers and blogs on Lydia text analysis system is presented. Antonyms were mostly used to construct emotion dictionaries [11].

In this study T. Wilson, J. Wiebe and P. Hoffmann, at the expression level, an approach to sentiment analysis is presented that first determines whether an expression is neutral or polar (negative/positive), and then clarifies the polarity of polar expressions. With this approach, the contextual polarity for large emotional expression clusters could be determined more clearly [12].

K. Denecke, in this study, proposes a methodology to determine the polarity of the text in a multilingual framework. As a method, the steps were followed first to determine the language in which the text was written, then to translate the text into English and then to classify it as positive or negative. LingPipe Language Identification Classifier was used to learn the original language of the text. PROMT Excellent Translation Technology was used to translate the text into English. In the classification part, three different ways were followed as LingPipe Classifier, SentiWordNet Classifier with classification rule and SentiWordNet Classifier with machine learning [13].

The aim of this study, prepared by B. R. Patel and K. K. Rana, is to give information about Decision Tree Algorithms and to compare these algorithms and reveal their advantages and disadvantages. In the study, many decision tree algorithms such as CHID, CART, ID3 C4.5, C5.0, Hunt's Alghoritm were mentioned. The areas where Decision Tree algorithms can be used are mentioned and each algorithm is compared in Speed, Pruning, Boosting, Missing Values and Formula scales. Decision tree technologies that can be used by researchers (WEKA, GATree, Alice d'ISoft, See5, OC1) and the functions of these technologies are explained [14].

In this study carried out by S. D. Jadhav and H. P. Channe, K-NearestNeighbor, Naive Bayes Classifier and Decision Tree algorithms are explained, comparative analyzes of these algorithms are made and application results are stated. These algorithms, prepared using WEKA technology, were trained with the same data set, and accuracy results were observed, and this process was repeated for three different data sets. As a result, it has been observed that each classification algorithm has its own advantages and disadvantages [15].

III. Methods and Tools

Within the scope of the article, along with the programming languages and libraries used, many different technologies have been used.

A. Programming Languages

Background services in the study were developed with Python – Flask Web Framework. The reason why Python language is preferred when creating background services is that it contains various libraries (pandas, numpy, sklearn, matplotlib, etc.) in the fields of machine learning, deep learning and data analysis / visualization. Using Jupyter Notebook in this process, facilitated the analysis and visualization of the data, the comparison of the success of the models, and the testing of the data sets.

B. Libraries and Models

Biggest advantage of using Python as a programming language in the development processes of background services is that it contains libraries that are very useful in the relevant field. These libraries were used while obtaining the results during the processes mentioned in Chapter 4: Pandas, Numpy, Sklearn, WordCloud, Matplotlib, Math, Gensim, Nltk, Pickle, Tensorflow.

IV. RESULTS AND PREPARATION OF SYSTEM

In this section, the steps and outputs of the application developed with the tools and methods specified in Section 3 are given step by step.

A. Collection and Preparation of Data

With the collection of text and rating data from various data sources, a data set with 10 output rating values was prepared. The critical point at this stage is that there is an equal number of text data indicating each rating value in the data set. These data sets are called balanced data sets.

```
Value counts for Train codes
10      2320
9       2320
8       2320
7       2320
6       2320
5       2320
4       2320
3       2320
2       2320
1       2320
Name: rating, dtype: int64
```

```
Value counts for Test codes
7       580
3       580
10      580
6       580
2       580
9       580
5       580
1       580
8       580
4       580
Name: rating, dtype: int64
```

Fig. 1. Balanced rating values in the training and test dataset.

B. Pre-Processing of Data

By preprocessing the raw data set, in the text data;

- Stopwords removed,
- Capital letters converted to lowercase letters,
- Punctuation marks have been removed and
- Non-alphabetic symbols have been dropped..

With these processes carried out within the preprocessing, possible problems in the process of extracting meaning from words and sentences in the following steps are prevented.

C. Creating Input Representation Models

Input representation is the name given to digitization for the purpose of analyzing the meaning of a text input. Expressing words and sentences in vector space is more successful in terms of both digitization and semantic analysis compared to various old methods (Bag-of-words, TF, TF-IDF, etc.).

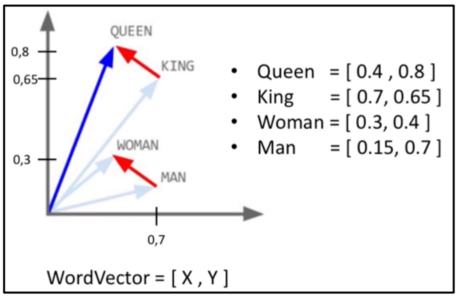


Fig. 2. Creation of word vectors and semantic relationships

In this context, two different models, Word2Vec and Doc2Vec, were tested in order to generate input representations. Since the Doc2Vec model also includes the Word2vec algorithm, digitizing the input texts using the Doc2Vec model and extracting their meanings was deemed appropriate within the scope of the application. Before the Doc2Vec model can be trained, each row of data must be expressed as a TaggedDocument. Each TaggedDocument data obtained within the scope of this project is placed in a vector space consisting of 200 numerical values. (The placement of words in vector space is simply given in Figure 2.) Thus, similar words and sentences were analyzed numerically.

D. Creation of Rating Prediction (Text Classification) Models

After each text data is expressed with 200 numerical values as a result of the Doc2Vec model, text classification is performed using the classical classification algorithms of the Sci-Kit Learn library. The text classification process performed in this step is called rating prediction. By analyzing the movie evaluation received from the user, the rating between [1-10] corresponding to this evaluation was estimated. Within the scope of this paper, the results were produced and evaluated with Decision Tree Classifier, Naive Bayes, Logistic Regression and LSTM models.

Evaluation Metric:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

TP: True Positive, TN: True Negative
FP: False Positive, FN: False Negative

Decision Tree Classifier:

Decision trees are a classification method that creates a model in the tree structure form, consisting of decision nodes and leaf nodes according to feature and target. The decision tree algorithm is developed by dividing the data set into smaller or even smaller pieces. A decision node may contain one or more branches. The first node is called the root node. A decision tree can consist of both categorical and numerical data. Classification is done using the Gini Impurity or Entropy equations. The main aim is to classify the data by making the Total Gini Impurity zero [16].

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

$$H = -\sum p(x) \log p(x)$$

(Entropi Denklemi, Claude Shannon, 1948.)

Fig. 3. Gini Impurity and Entropy Equation

Naïve Bayes Classifier:

Naive Bayes classifier is based on Bayes Theorem. It is a lazy learning algorithm and can also work on unstable datasets. Bayes' theorem shows the relationship between conditional probabilities and marginal probabilities within the probability distribution for a random variable [17].

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

Fig. 4. Bayes Theorem

Logistic Regression:

Logistic regression is a regression problem where the dependent variable is a categorical variable. It is widely used in linear classification problems. Although it is called regression, there is a classification [18]. Logistic regression is used to find the most appropriate model to describe the relationship between a two-way characteristic (dependent variable = response -or outcome variable-) and a set of independent (predictive or explanatory) variables. Logistic regression generates the coefficients -and standard errors and significance levels- of a formula to estimate the probability of the existence of characteristics of interest by logit transform:

$$\boxed{logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k}$$

Above, the p-value is the probability that the characteristic exists.

$$\boxed{odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}}$$

$$\boxed{logit(p) = \ln\left(\frac{p}{1-p}\right)}$$

Rather than choosing parameters that minimize the sum of square root errors (like standard regression), estimation in logistic regression chooses parameters that maximize the probability of observation of sample values [18].

LSTM (Long Short-Term Memory):

It is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can handle not only single data points (like images) but also entire data sequences (like speech or video). For example, LSTM is applicable for tasks such as unpartitioned, linked handwriting recognition, speech recognition [19][20], and detection of anomaly in network traffic or IDSs (intrusion detection systems). An ordinary LSTM unit consists of a cell, an entrance gate, an exit gate and a forget gate. The cell remembers values at arbitrary time intervals and these three gates regulate the flow of information entering and leaving the cell [21].

The hyperparameters and values of the text classification models used in the article are given in the table below.

Text Classification Model	Hyperparameters and others
Decision Tree	criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort='deprecated', ccp_alpha=0.0
Naïve Bayes (Gaussian)	priors=None, var_smoothing=1e-09
Logistic Regression	penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None
LSTM	units = 100, dropout=0.2, recurrent_dropout=0.2, activation="tanh", recurrent_activation="sigmoid", use_bias=True, kernel_initializer="glorot_uniform", recurrent_initializer="orthogonal", bias_initializer="zeros", unit_forget_bias=True

The accuracy results obtained as a result of the estimation of the text classification models trained with the hyperparameter values specified in the table above are given in the table below.

Text Classification Model	Prediction Accuracy (Approximate Preds.)	Prediction Accuracy (Exact Preds.)
Decision Tree	47.1%	20.8%
Naïve Bayes (Gaussian)	59.7%	35.1%
Logistic Regression	64.9%	34.9%
LSTM	61.3%	31.4%

As a result of the evaluation, the most successful text classification model was determined as Logistic Regression. In the text classification results performed using the Logistic Regression model, 64.9% accuracy was obtained in convergent predictions. Since the data collected on the films within the scope of the project are subjective, minor deviations in the results obtained can be ignored. For example, if the expected rating value to correspond to a text data is 5, it is very likely to be estimated at 4 or 6 and can be considered successful. For this reason, the 64.9% success achieved in convergent estimates is extremely important.

On the other hand, the text classification success value obtained with the LSTM model is very close to the text classification success value obtained with the Logistic Regression. For this reason, in the stage of choosing which model to use, it is necessary to examine other factors besides the forecast performance. In this direction, it has been revealed that the main difference between the RNN-LSTM and Logistic Regression models examined is the deployment processes. While the Logistic Regression model can be directly saved with the Pickle library and then used for estimation easily, it has been observed that it is necessary to save the architecture and weights separately for Neural Network models. So, Logistic Regression method was preferred as the text classification model due to its maximum estimation success and ease of deploy.

E. *Creation of Sentiment Analysis (Text Classification) Models*

In addition to predicting the user rating/score, it is aimed to analyze the emotion of the movie evaluation made by the user. The main purpose of creating and using the sentiment analysis model is that the user comments on the movie are subjective. Subjective text data taken as input within the scope of the study was estimated in terms of positivity and negativity using the sentiment analysis model.

In this process, the texts that received 1/2/3 Ratings from the available data were marked as Negative (0) content. The texts of the 8/9/10 ratings are marked as Positive (1). Then, steps B, C, D were repeated for this prepared data set. As a result of the evaluation of various models, one more Doc2Vec input representation model and Logistic Regression text classification model were obtained.

The most successful emotion analysis model obtained was Logistic Regression, with a success rate of 92.4%.

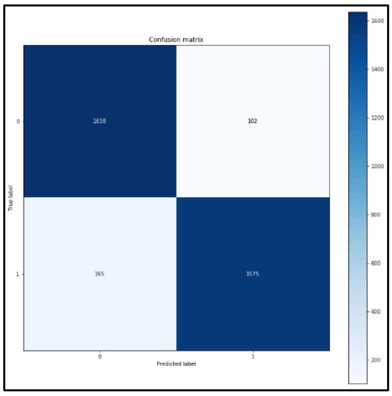


Fig. 5. The most successful emotion analysis model obtained was Logistic Regression, with a success rate of 92.4%.

As a result, with the models produced in the steps D and E, it was analyzed how many stars the user gave to the relevant movie out of 10 and whether the relevant comment had negative or positive content.

F. *Creation of Web Services*

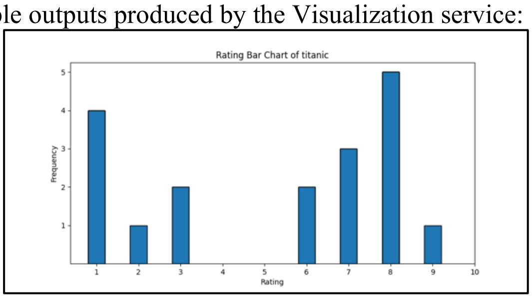
After the most successful models are deployed, there is the stage of obtaining the outputs by giving inputs to these models recorded in the queue. Therefore, Web Services are needed in this process where various inputs can be taken and various outputs can be produced. The APIs created within the scope of the study are coded with Python-Flask Web Framework and the background of the project basically consists of two web services.

Implemented web services and included functionalities:

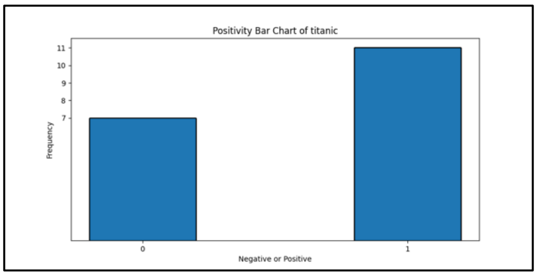
a. <u>Sentiment Analysis Service:</u>
- It preprocesses the text data it receives as input.
- Identifies important 'keywords' in text data.
- It provides digitization of text data for rating prediction and determination of its meaning in vector space. (in a 200 dimensional vector space, with the Doc2Vec model)
- Rating prediction of the digitized text data is performed using Logistic Regression. (Text Classification)
- Meaningful text data obtained at the end of Step 3 is digitized for emotion estimation and its meaning is determined in vector space. (in a 100-dimensional vector space, with the Doc2Vec model)
- Sentiment prediction is achieved using Logistic Regression on digitized text data. (Text Classification)
- Results obtained in the classifications for Rating Estimation and Sentiment Analysis are written and stored in the table of the movie the user commented on.
- Returns the Rating result and Emotion of the evaluation made to the user as output.

b. <u>Visualization Service:</u>
- Gets the name of the movie for which the user wants to view the Evaluation Report.
- Reads the table in the database where the data is stored.
- Checks whether the table exists or not.
- Rads the original evaluation texts from the table containing the meanings of the evaluations and the classification results and identifies the most frequently used words. (Words with maximum frequency). These detected words and their frequencies are used in Word Cloud visualization.
- Calculations are made on the text classification results stored in the table, and the results and visualizations are displayed to the user. (Average rating value and the positivity rate of the comments according to the reviews of the relevant film)
- As the output, from the previous reviews of the movie whose report is requested to be displayed; rating average, ratings bar chart, rating of positive/negative reviews, positive/negative reviews bar chart and WordCloud created as a result of word analysis are reported to the user.

Sample outputs produced by the Visualization service:



➔ Average of Review Ratings: 5.22



➔ Positive Review Rate: 61.11%

(WordCloud image created with the most frequently used words in evaluations)

V. Conclusion

Doc2Vec models were created to digitize text data and analyze their meaning. On these numerically expressed text data, classification was performed for rating ([1-10]) and sentiment (negative, positive) prediction. In this process, Decision Tree, Naive Bayes, Logistic Regression and LSTM algorithms were used. As a result of the tests, it was observed that the Logistic Regression Algorithm produced the best results for text classification.

The movie evaluations made by the users were recorded after being classified with the models created within the scope of the study. Then, Exploratory Data Analysis (EDA) was performed on the text data based on the movie, various graphs (rating barchart, positivity barchart, word cloud) were created and reported to the user. With these reports, users will be able to learn whether the relevant movie is worth watching or not, with minimum time loss.

As a result of this study, successful models created in the sub-branches of Input Representation and Text Classification have contributed to the Natural Language Processing (NLP) literature by analyzing and visualizing text data.

Acknowledgment

We would like to thank to the RA Ebu Y. Güven for their help in the development process of the study and the application and to the information he provided on the article, and express our special thanks to Assoc. Prof. Muhammed A. Aydın, with whom we worked together from the beginning to the end of the project.

References

[1] Le Q., Mikolov T., 2014. “Distributed Represenations of Sentences and Documents”. In: International Conference on Machine Learning. (pp. 1188-1196) (https://cs.stanford.edu/~quocle/paragraph_vector.pdf)

[2] Lakshmi Devi B., Varaswathi Bai V., Ramasubbareddy S., Govinda K. (2020) “Sentiment Analysis on Movie Reviews”. In: Venkata Krishna P., Obaidat M. (eds) Emerging Research in Data Engineering Systems and Computer Communications. Advances in Intelligent Systems and Computing, vol 1054. Springer, Singapore (https://link.springer.com/chapter/10.1007%2F978-981-15-0135-7_31)

[3] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, 01.2011. “Sentiment Analysis of Twitter Data” (https://www.researchgate.net/publication/247935218_Sentiment_Analysis_of_Twitter_Data)

[4] Gupta R., Gill Nasib S. “Financial Statement Fraud Detection using Text Mining”, 2012. (https://www.researchgate.net/publication/270554223_Financial_Statement_Fraud_Detection_using_Text_Mining)

[5] Jain G., Manisha, Agarwal B. “Spam Detection on Social Media Text”, 05.2017 (https://www.researchgate.net/publication/322791510_Spam_Detection_on_Social_Media_Text)

[6] Prabowo R., Thelwall M., 04.2009, “Sentiment analysis: A combined approach” (https://www.sciencedirect.com/science/article/abs/pii/S1751157709000108)

[7] Lin C., He Y., 11.2009, “Joint sentiment/topic model for sentiment analysis” (https://dl.acm.org/doi/abs/10.1145/1645953.1646003)

[8] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, Christopher Potts, 06.2011, “Learning Word Vectors for Sentiment Analysis”(https://www.aclweb.org/anthology/P11-1015.pdf)

[9] Kiritchenko S., Zhu X., Mohammad S. M., 20.08.2014, “Sentiment Analysis of Short Informal Texts”(https://www.jair.org/index.php/jair/article/view/10896)

[10] Fang X., Zhan J., 16.06.2015, “Sentiment analysis using product review data”(https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2)

[11] Godbole N., Srinivasaiah M., Skiena S., 2007, “Large-Scale Sentiment Analysis for News and Blogs”(http://www.uvm.edu/pdodds/files/papers/others/2007/godbole2007a.pdf)

[12] Wilson T., Wiebe J., Hoffmann P., 09.2009, “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis ”(https://www.aclweb.org/anthology/H05-1044.pdf)

[13] Denecke K., “Using SentiWordNet for Multilingual Sentiment Analysis” (https://www.researchgate.net/profile/Kerstin-Denecke/publication/4330825_Using_SentiWordNet_for_multilingual_sentiment_analysis/links/00463526769e2e3c11000000/Using-SentiWordNet-for-multilingual-sentiment-analysis.pdf)

[14] Brijain R Patel, Kushik K Rana, 2014, “A Survey on Decision Tree Algorithm For Classification”(http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=85E70CA16041457C881117E29015EDF8?doi=10.1.1.673.2797&rep=rep1&type=pdf)

[15] Sayali D. Jadhav, H. P. Channe, 2013, “Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques “(https://www.ijsr.net/archive/v5i1/NOV153131.pdf)

[16] Uzun E., “Decision Tree (Karar Ağacı) Algoritması - Sınıflandırma”, 07.13.2018, (https://erdincuzun.com/makine_ogrenmesi/decision-tree-karar-agaci-id3-algoritmasi-classification-siniflama/)

[17] Hatipoglu E., “Machine Learning – Classification – Naïve Bayes”

[18] Veri Bilimcisi, “Lojistik Regresyon (Logistic Regression)” 18.07.2017, (https://veribilimcisi.com/2017/07/18/lojistik-regresyon/)

[19] Sak Hasim, Senior Andrew, Beaufays Francoise, "Long Short-Term Memory recurrent neural network architectures for large scale acoustic modeling", 2014

[20] Li Xiangang, Wu Xihong, "Constructing Long Short-Term Memory based Deep Recurrent Neural Networks for Large Vocabulary Speech Recognition",10.15.2014

[21] Wikipedia, Long Shot-Term Memory, 12.5.2020 (https://en.wikipedia.org/wiki/Long_short-term_memory)