Sentiment Analysis and Opinion Mining Using Deep Learning for the Reviews on Google Play

Sercan Sari(✉) and Murat Kalender

Department of Computer Engineering, Yeditepe University, Istanbul, Turkey
{ssari,mkalender}@cse.yeditepe.edu.tr

Abstract. Sentiment analysis and opinion mining have an important role to trace consumer behavior. With the recent advances in machine learning techniques, this issue has been addressed and come a long way in English. However, in agglutinative languages such as Turkish, it has been still one of the hot topics. In this study, we compare several classification methods and deep learning methods to make a sentiment analysis for Turkish reviews that we have collected from Google Play. We have 87.30% prediction accuracy for multinomial Naive Bayes and 95.87% prediction accuracy for deep learning model. We have significant results both from the machine learning classifiers and the deep learning model. While there is no difference between machine learning classifiers when we use different vectorizers, there is a difference when we build a deep learning model to predict target value. This model can also be applied to any data from Twitter, Facebook, or any other microblogging.

Keywords: Sentiment analysis · Opinion mining · Machine learning · Text classification · Deep learning

1 Introduction

Behaviors of online users have transformed as the growing rate of data on the internet. According to the statistics [11], 70% of online users who like to purchase electronics, stated that they have read online reviews before purchasing the product. This situation accelerates the researches about sentiment analysis and opinion mining for online resources such as Twitter and Facebook on the internet. There are several studies [4,14,18] which analyze microblogging platforms such as Twitter and Facebook to devise people's opinions and classify them according to the sentiment. Various kinds of information can be extracted from these resources. For example, manufacturer companies can collect information about their products, and political parties and social organizations may determine their events according to the results of this researches. With the recent advances in machine learning techniques, this issue has been addressed and come a long way in English. However, in agglutinative languages such as Turkish, it

M. Ben Ahmed et al. (Eds.): SCA 2020, LNNS 183, pp. 126–137, 2021.
https://doi.org/10.1007/978-3-030-66840-2_10

has been still one of the hot topics [13,16,25]. While there are some advantages to use microblogging especially Twitter such as a variety of data and the velocity of data [18], some of the tweets can have sarcasm in it [5] and it can be problematic to label such a huge data by just looking at some features in it [7]. These can lead to inefficient results and to degrade the performance of machine learning models. To solve these issues and to make models that predict more accurately we come up with a different approach. In the literature, there have been studies which they used movie reviews [8–10,19,24]. The approach that we have followed is similar to these studies. We use reviews on Google Play to get rid of the ambiguity of texts. Because in such statements, people directly mention their opinions and rate them before submitting them. As a result of rating before submitting, the aforementioned problems can be elucidated easily.

Although, our study has similarities with the current state of literature, our main difference over current state of studies is that we build a deep neural network to do sentiment analysis and opinion mining for the Turkish text corpus. We also contribute the literature by sharing our corpus that we have prepared for this study [1]. In our study, we apply different machine learning algorithms and build a deep neural network to analyze how the reviews on Google Play in Turkey can be utilized for sentiment analysis and opinion mining purposes. We select the reviews for the following reasons:

- Application reviews section is directly used to express the opinions of users.
- Application reviews contain enormous texts and up to date.
- There are a lot of different people using an enormous number of different applications which means different contexts.

We collected a corpus of 11000 reviews from Google Play Turkey by adjusting their ratings evenly between two sets of reviews:

1. reviews belonging four and five stars as positive emotions
2. reviews belonging one and two stars as negative emotions (Table 1).

Table 1. Examples of reviews on Google Play

Rating	Review
5	Eskiden beri oynuyorum. Oyun muhteşem. Kesinlikle herkese tavsiye ediyorum
4	Oyunu çok beğendim. Yüklediğim oyunlarün en güzellerinden biri
2	Telefonuma yaptığım son güncelleme ile fotoğraftaki renkler bozulmaya başladı
1	Program sürekli olarak kendini İngilizce yapıyor. Sorunu çözün lütfen

The rest of the paper will be analyzed as follows: In Sect. 2 we will mention the relevant work. In the third section, we will clarify how we collect the data

for our study. The fourth will give an analysis of the corpus that we use the training models. Section 5, we present model building and the results of our study. Section 6 will conclude the paper.

2 Related Works and Background

2.1 Related Works

There have been studies that focus on sentiment analysis and opinion mining in the literature. The main motivation behind these studies comes from the track-ability of consumer behavior and the wide-range of data.

In 2012, *Kaya et al.* [13] integrate sentiment classification techniques into the domain of political news for Turkish news sites. They compare supervised machine learning algorithms which are Naïve Bayes, Maximum Entropy, SVM and the character based N-Gram Language Model for sentiment analysis of Turkish political news. *Kucuk et al.* [16] work on named entity recognition (NER) and report experiments about NER on Turkish tweets. In 2015, *Yildirim et al.* [25] reports the effects of preprocessing layers on the sentiment classification of Turkish social media texts. While there are some benefits to utilize microblogging especially Twitter and Facebook such as a variety of data and the velocity of data [18], some of the microblogging texts can have sarcasm in it [5] and it can create ambiguity to work with such a huge data by just looking at some features in it [7]. These kind of features of the data may cause inefficient results. We use reviews on Google Play to get rid of the ambiguity of the texts. Because in such statements, people directly mention their opinions and rate them before submitting them. As a result of this, the aforementioned problems can be elucidated easily. In the literature, there have been other studies which they used movie reviews [8–10,19,24]. The approach that we have followed is similar to these studies.

The background of our study consists of three main parts which are feature engineering, machine learning classifiers and sequential deep learning model. In the feature engineering part, we are going to explain why we need different representations of the text data to extract information from them. In the second part, we are going to mention the machine learning classifiers that we have used in this study and finally, we going to give details about deep neural network that we have used in our research.

2.2 Feature Engineering for NLP

In the natural language process (NLP), we cannot directly use the text to extract information and to build machine learning models. In order to utilize the text information, we need to convert it into numerical values. A simple and adequate model to make us able to use text documents in machine learning is called the Bag-of-Words Model, or BoW [12]. This simple model actually focuses on the occurrence of each word in a text. By using this method, we can easily encode

every text as an encoded fixed-length vector with the length of the vocabulary which we have known. In the scope of this study, we will explain the vectorization for text-based features and analyze two of the three different ways to use this model in the scikit-learn library [20] which are CountVectorizer, TfidfVectorizer, and HashVectorizer. Since we will use CountVectorizer and TfidfVectorizer, the following subsections will explain the details about them.

CountVectorizer. The *CountVectorizer* provides and implements both occurrence counting and tokenization. It builds a vocabulary of known words. It uses this vocabulary to encode new text data. Although it is a good solution, there are some drawbacks to use *CountVectorizer* such as irrelevant words that occur so many times.

TfidfVectorizer. There is an alternative method to calculate word frequencies. It is called *Term Frequency - Inverse Document* frequency (TFIDF) which is the part of the scores belonged to each word.

- **Term Frequency:** How often a given word appears with in a text
- **Inverse Document Frequency:** This actually adjusts words which are appear so many times in texts.

2.3 Machine Learning Classifiers

In this subsection, we are going to give some details about the machine learning classifiers that we have used in our study.

Multinomial Naive Bayes. Multinomial Naive Bayes classifier is based on Naive Bayes theorem [17]. It is a simple and baseline approach to build classifiers since it is fast and easy to implement. In [21], they discuss that while it has really good efficiency, it affects the quality of results because of its assumptions. In order to eliminate such drawbacks, they introduce the multinominal Naive Bayes (MNB) method. MNB models the distribution of words in a corpus as a multinomial. They pretend the text as a sequence of words and assume that the location of words is produced independently of each other.

K-Nearest Neighbors. KNN (K-Nearest Neighbors) is one of the simplest and widely used classification algorithms. KNN is a non-parametric and lazy learning algorithm and it is used for classification and regression [3]. The main idea behind the nearest neighbor method is to find a label for the new point according to the closest distance metric to it and predict the label from these. The number of neighbors can be defined by the user.

Decision Tree Learning. Decision tree learning is one of the commonly used methods in predictive analysis [22]. The purpose is to build a model that predicts the right label of target variables from the input variables. A tree is created by splitting the input variables. Classification features have an important role while splitting the tree [23]. There are some advantages to use decision trees. It is simple to interpret decision trees and it uses white-box model.

In our study, we pick the above classifiers and analyze the effects of different vectorizers on different models. We have also built a sequential deep learning model to predict the right target value. The deep learning network that we have used in our study can be seen in Fig. 1. We are going to explain it in detail in the Model Building and Results section.

2.4 Sequential Deep Learning Model

Deep learning is a variety of a family of machine learning methods. It is based on artificial neural networks. The use of multiple layers makes the learning process deep and it is where the adjective "deep" comes from. The sequential model is one of the simplest ways to build a model in Keras which is a deep learning framework [6]. It is built on top of TensorFlow 2.0 [2]. It provides us to build the model layer by layer.

In our study, we have also built a sequential deep learning model to predict the target value according to textual information.

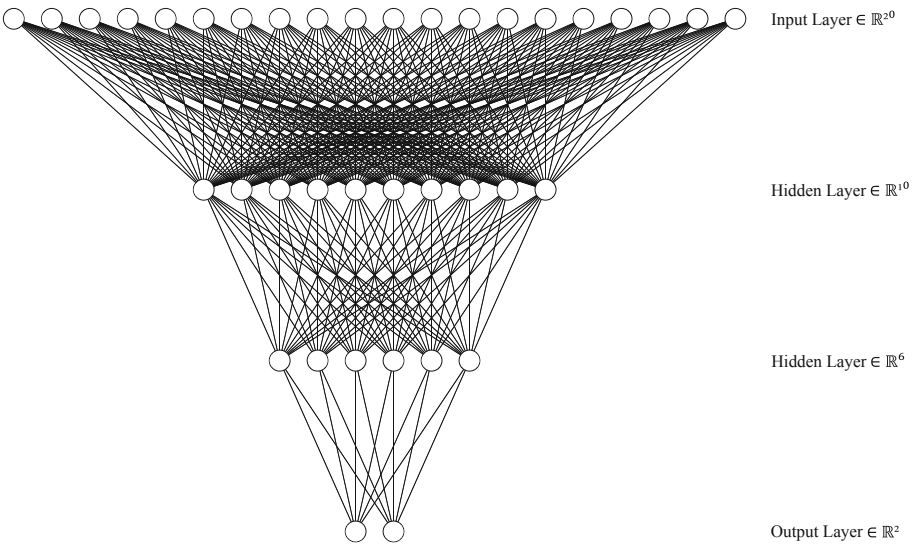

Fig. 1. Deep neural network with 2-hidden layers

3 Corpus Collection

There are several data collection methods such as APIs, We have used web scraping methods to extract the data from Google Play Turkey. We select the most popular 112 applications and we also extract the most useful 100 reviews from each application. By doing that, we achieve to collect the most relevant reviews. We simply collected the text part of the reviews and their ratings.

We eliminate the ratings that take 3 stars. We use the below procedure to label reviews according to their ratings:

- Positive label for the reviews that take 4 stars or 5 stars
- Negative label for the reviews that take 1 star or 2 stars

These two types of labeled data are used to train a classifier to predict whether the given review has a positive or negative sentiment. In our research, we specifically use the Turkish language. Categories of 112 applications that we have selected to extract reviews can be seen in the below Table 2.

Table 2. Categories of applications in Google Play

Categories	
Art & Design	Dating
Augmented reality	Daydream
Auto & Vehicles	Education
Beauty	Entertainment
Books & Reference	Events
Business	Finance
Comics	Food & Drink
Communication	Health & Fitness

4 Corpus Analysis

In order to build a machine learning model, we examine our data and explore new features such as length of the review, capital letter percentage in the review that we can use for classification. Before modeling, we check the correlation between extra features and the target value. The below Fig. 2 depicts the correlation between the percentage of digits and the percentage of capital letters in the reviews.

Other features that we extract from our corpus are digit percentage, exclamation mark usage, length of the review, and capital letter percentage. We see that none of these features are related to the target value. Correlation values

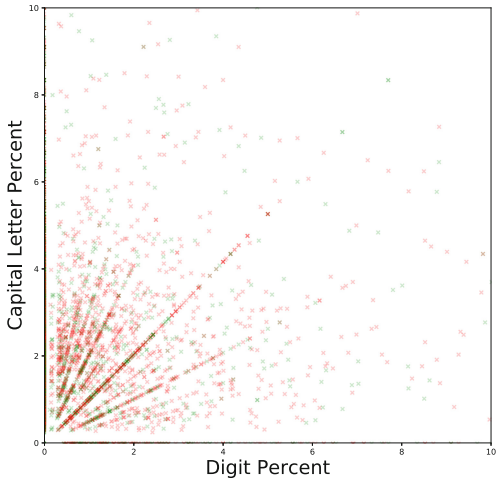

Fig. 2. Distribution between the percentage of digits and the percentage of capital letters in the reviews

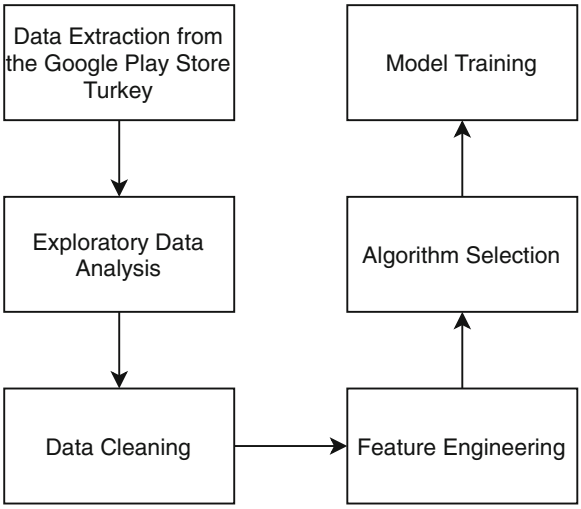

Fig. 3. Overall design

such as in Fig. 2 show us there is no relation among these features and the target value which is used for sentiment analysis. Our overall design can be seen in below Fig. 3. As we have mentioned above, we have collected over 10.000 reviews from the Google Play Turkey.

After we have collected the data, we have explored the data and we have shared some of our findings like in the Fig. 3. After these steps, we have started to clean our data by applying all procedures as follows:

- Replacing similar emoticons with a determined keyword
- Removing punctuations
- Replacing some Turkish letters with their corresponding English letters
- Lowercasing the text
- Removing digits
- Removing extra white spaces and punctuation

After these procedures, there are also special requirements to extract features from the text data because of its structure. The text should be parsed to remove tokens and after that, the words must be encoded as numerical values such as integer or float to be used in machine learning models. In order to that, we use the scikit-learn library which is a free software machine learning library [20]. By using scikit-learn we both perform tokenization and feature extraction of our corpus.

We have used two types of feature extraction methods which are *CountVectorizer* and *TfidfVectorizer* and compared their results in terms of effects to the prediction accuracy.

In our study, we both used machine learning classifiers and deep learning models for algorithm selection and model training. We are going to give the details about algorithm selection and model training parts in the following sections.

5 Model Building and Results

We build a classifier using multinomial Naive Bayes which is based on Bayes' theorem [15]. We also build models with decision trees and K-Nearest neighbors. As we have mentioned earlier, we both use CountVectorizer and TfidfVectorizer for feature engineering and analyze the results.

5.1 Results for Machine Learning Classifiers

The bar chart in Fig. 4 shows the prediction accuracy of the machine learning classifiers which we have used in our study. As can be seen from the figure, there is no significant difference in prediction accuracy when we use CountVectorizer or TfidfVectorizer.

The prediction accuracy results can be seen in Table 3. As can be seen, while there is a difference when we use a multinomial Naive Bayes classifier, there is no difference when we use different vectorizer for decision tree and KNN classifier.

5.2 Results for Deep Learning Model

As we have mentioned, we have also built a sequential deep learning model. In our model, we have 2 hidden layers and we apply dropout for the 20% of the nodes in order to avoid overfitting in our model. Figure 1 depicts the deep learning network that we have used in our study.

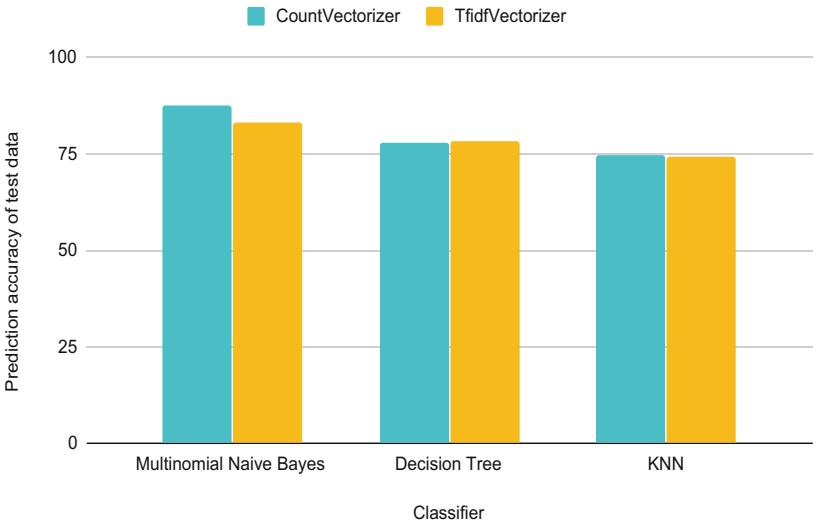

Fig. 4. Prediction accuracy of the classifiers

Table 3. Prediction accuracy table

	CountVectorizer	TfidfVectorizer
MNB	87.30%	83.13%
DT	77.93%	78.09%
KNN	74.48%	74.15%

As can be seen in Table 4, our results indicate that we have 95.87% prediction accuracy on test data for *TfidfVectorizer* and 95.71% prediction accuracy on test data for *CountVectorizer*. However, as can be seen in Fig. 5 and Fig. 6, even though we exactly use the same model for both, there may be overfitting when we use *CountVectorizer*. Also, when we look at the results in Table 4, it can be seen that there is overfitting in the model. While the prediction accuracy is 99.73% for training data, the test data accuracy is less than training data almost 5%. This may give us a clue that there can be overfitting when we use the same model with the *CountVectorizer*. In order to get rid of this overfitting, we may need to optimize the parameters of the model.

Table 4. Prediction accuracy results for the deep learning model

	Training data	Test data
CountVectorizer	99.73%	95.71%
TfidfVectorizer	95.97%	95.87%

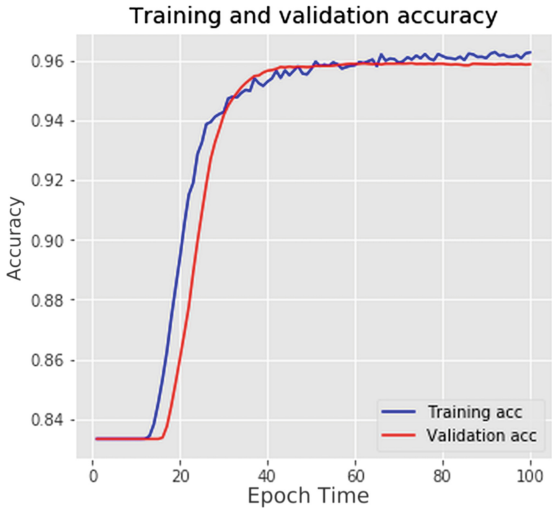

Fig. 5. Training and validation accuracy for *TfidfVectorizer*

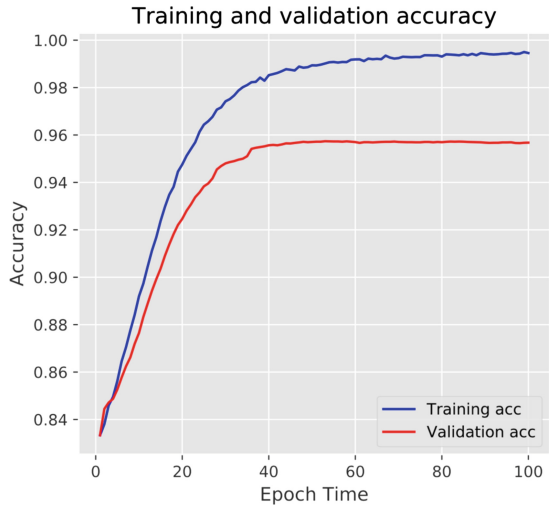

Fig. 6. Training and validation accuracy for *CountVectorizer*

6 Conclusions

It is conspicuous that sentiment analysis and opinion mining have an important role to trace consumer behavior. With the recent advances in machine learning techniques, this issue has been addressed and come a long way in English. However, in agglutinative languages such as Turkish, it has been still one of the hot topics. In this study, we compare several classification methods and deep learning

methods to make a sentiment analysis for Turkish reviews that we have collected from Google Play. We have explained the machine learning classifiers that we have used and we have depicted our overall design. We have significant results both from the machine learning classifiers and the deep learning model. Our prediction accuracy results are satisfactory. We had 87.30% prediction accuracy for multinomial Naive Bayes and 95.87% prediction accuracy for deep learning model. Our experiments showed that we could have overfitting when we use different vectorizer techniques. While there is no difference between machine learning classifiers when we use different vectorizers, there is a difference when we build a deep learning model to predict target value. Although we build our model from the reviews that we have extracted from Google Play, this model can also be applied to any data from Twitter, Facebook, or any other microblogging.

References

1. https://github.com/ssari-memory/corpus-turkish-reviews
2. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org
3. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
4. Anjaria, M., Guddeti, R.M.R.: Influence factor based opinion mining of Twitter data using supervised learning. In: 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS), pp. 1–8. IEEE (2014)
5. Bouazizi, M., Ohtsuki, T.: Opinion mining in Twitter how to make use of sarcasm to enhance sentiment analysis. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pp. 1594–1597 (2015)
6. Chollet, F., et al.: Keras (2015). https://keras.io
7. Çoban, Ö., Özyer, B., Özyer, G.: Türkçe twitter mesajlarının duygu analizi. In: Signal Processing and Communications Applications Conference (SIU) (2015)
8. Demirtas, E., Pechenizkiy, M.: Cross-lingual polarity detection with machine translation. In: Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, pp. 1–8 (2013)
9. Gezici, G., Yanıkoğlu, B.: Sentiment analysis in Turkish. In: Turkish Natural Language Processing, pp. 255–271. Springer (2018)
10. Ghorbel, H., Jacot, D.: Sentiment analysis of French movie reviews. In: Advances in Distributed Agent-Based Retrieval Tools, pp. 97–108. Springer (2011)
11. Gordon, K.: Topic: Online reviews. https://www.statista.com/topics/4381/online-reviews/
12. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
13. Kaya, M., Fidan, G., Toroslu, I.H.: Sentiment analysis of Turkish political news. In: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, vol. 1, pp. 174–180. IEEE (2012)

14. Khan, F.H., Bashir, S., Qamar, U.: TOM: Twitter opinion mining framework using hybrid classification scheme. Decis. Support Syst. **57**, 245–257 (2014)
15. Klon, A.E., Glick, M., Davies, J.W.: Combination of a Naive Bayes classifier with consensus scoring improves enrichment of high-throughput docking results. J. Med. Chem. **47**(18), 4356–4359 (2004)
16. Küçük, D., Steinberger, R.: Experiments to improve named entity recognition on Turkish tweets. arXiv preprint arXiv:1410.8668 (2014)
17. Maron, M.E.: Automatic indexing: an experimental inquiry. J. ACM (JACM) **8**(3), 404–417 (1961)
18. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. LREc **10**, 1320–1326 (2010)
19. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, pp. 79–86. Association for Computational Linguistics (2002)
20. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
21. Rennie, J.D., Shih, L., Teevan, J., Karger, D.R.: Tackling the poor assumptions of Naive Bayes text classifiers. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 616–623 (2003)
22. Rokach, L., Maimon, O.Z.: Data Mining with Decision Trees: Theory and Applications, vol. 69. World Scientific, Singapore (2008)
23. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge (2014)
24. Vural, A.G., Cambazoglu, B.B., Senkul, P., Tokgoz, Z.O.: A framework for sentiment analysis in Turkish: application to polarity detection of movie reviews in Turkish. In: Computer and Information Sciences III, pp. 437–445. Springer (2013)
25. Yıldırım, E., Çetin, F.S., Eryiğit, G., Temel, T.: The impact of NLP on Turkish sentiment analysis. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi **7**(1), 43–51 (2015)