Computers, Materials & Continua
DOI:10.32604/cmc.2021.017827
Article

Tech Science Press

Multi-Class Sentiment Analysis of Social Media Data with Machine Learning Algorithms

Galimkair Mutanov, Vladislav Karyukin* and Zhanl Mamykova

Al-Farabi Kazakh National University, Almaty, 050040, Kazakhstan
*Corresponding Author: Vladislav Karyukin. Email: vladislav.karyukin@kaznu.kz
Received: 13 February 2021; Accepted: 02 April 2021

Abstract: The volume of social media data on the Internet is constantly growing. This has created a substantial research field for data analysts. The diversity of articles, posts, and comments on news websites and social networks astonishes imagination. Nevertheless, most researchers focus on posts on Twitter that have a specific format and length restriction. The majority of them are written in the English language. As relatively few works have paid attention to sentiment analysis in the Russian and Kazakh languages, this article thoroughly analyzes news posts in the Kazakhstan media space. The amassed datasets include texts labeled according to three sentiment classes: positive, negative, and neutral. The datasets are highly imbalanced, with a significant predominance of the positive class. Three resampling techniques (undersampling, oversampling, and synthetic minority oversampling (SMOTE)) are used to resample the datasets to deal with this issue. Subsequently, the texts are vectorized with the TF-IDF metric and classified with seven machine learning (ML) algorithms: naïve Bayes, support vector machine, logistic regression, k-nearest neighbors, decision tree, random forest, and XGBoost. Experimental results reveal that oversampling and SMOTE with logistic regression, decision tree, and random forest achieve the best classification scores. These models are effectively employed in the developed social analytics platform.

Keywords: Social media; sentiment analysis; imbalanced classes; machine learning; oversampling; undersampling; SMOTE; russian; Kazakh

1 Introduction

It has become a common practice for people to actively share their thoughts and opinions about local and global events through social media. As new occasions happen almost every day, and their actuality varies remarkably, it is imperative to monitor the most critical topics in different spheres of life (i.e., politics, economics, civil society, education, healthcare, ecology, culture, and sports). The volume of facts and opinions about them shared on social media renders such a tracking impracticable without automated methods, and this has made analytical platforms indispensable. Generally, the core element of these platforms is the sentiment analysis tool. Sentiment analysis [1] has been extensively explored since the early works by Mantyla et al. [2]. The analytical

platforms [3] have been developed to automate and increase the social media processing speed. They are customarily targeted at monitoring actual social and political situations [3,4], using social networks under governmental control [5], quantitatively analyzing unstructured data [6], forming analytical material [7], and extracting pertinent information from texts [8].

In this paper, we present the OMSystem, the first automatic tool developed to analyze Kazakh users' opinions expressed through social media and over-the-top (OTT) platforms. This system enables monitoring web resources and social networks with subsystems for modeling "social well-being," estimating the sentiment of user's messages and comments, supporting the sentiment dictionaries of the Russian and Kazakh languages, and machine learning (ML) algorithms. This OMSystem supports Kazakhstan's leading news portals, the most popular social networks, such as Facebook, VKontakte, Instagram, Twitter, and YouTube, and accounts of famous bloggers. The system's chief objectives are prompt monitoring of the information space and social networks on the most relevant themes. They unambiguously define the purview of the problem, determine public opinions and their quick explanation, analyze the dynamics of a commercial brand, events, and activity mentions, and, in turn, the evaluation of the extent of "social well-being."

The architecture of the OMSystem, schematically illustrated in Fig. 1, includes the following components:

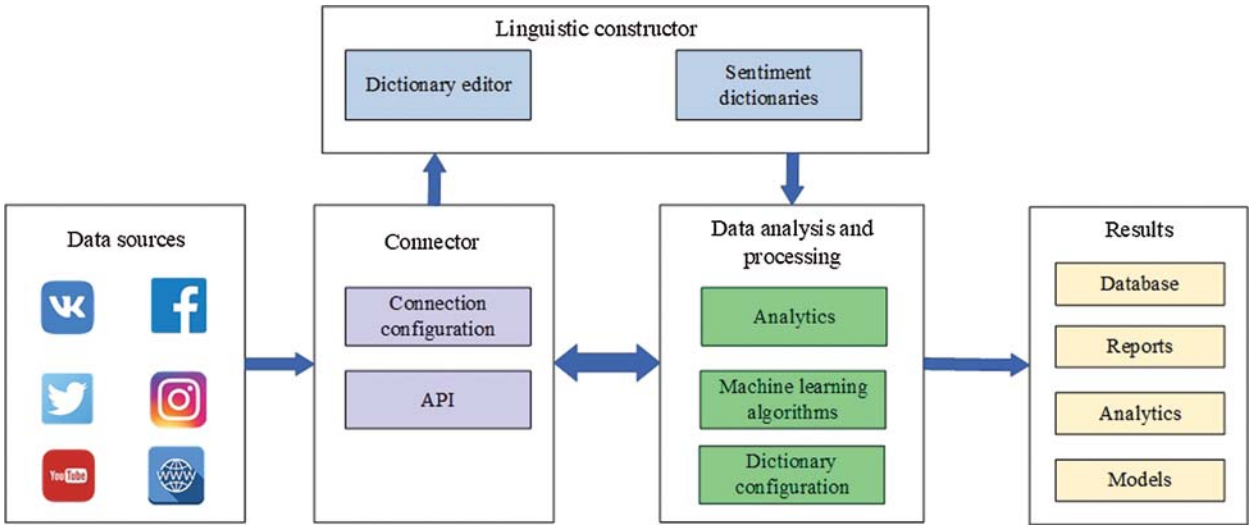

Figure 1: The architecture of the OMSystem

- Data sources: They include news portals, blogs, and social networks.
- Connector module: It is used to configure the connection to sources and the API of the target data sources.
- Linguistic constructor module: It is used to create sentiment dictionaries comprising words belong to either of the three classes: positive, negative, and neutral.
- Data analysis and processing module: It is based on sentiment dictionaries and deploys ML algorithms for sentiment analysis. Furthermore, it builds social analytics that reveals the sentiment concerning momentous events and people's attitude toward and interest in them.

- Results module: It encompasses a newly formed relational database of texts and comments, models of sentiment analysis, social analytics, and visualized reports of "social well-being."

The core element of the OMSystem is the sentiment analysis tool capable of identifying three sentiment categories (positive, neutral, and negative) of parsed texts.

There are several approaches to the sentiment definition:

- Lexicon-based [9]
- ML-based [10]
- Deep learning (DL)-based [11]

The **lexicon-based approach** [12,13] relies on assigning sentiment categories to words. Words are typically labeled in two categories (positive and negative), three categories (positive, neutral, and negative), or five categories (very positive, positive, neutral, negative, and very negative). The effectiveness of the lexicon-based approach [13] depends on the high quality of sentiment dictionaries containing the large corpus of words labeled in the categories mentioned earlier. A notable drawback [14] of this approach is the need to include a large number of linguistic resources to find the essential words for sentiment analysis.

The **ML-based approach** [15] includes supervised and unsupervised learning methods [16]. In the former, instead of words, whole texts are labeled with sentiment categories. It is an intricate, time-consuming, and error-prone method, which requires meticulously-designed guidelines. Therefore, the elaboration of semi-automatic methods, using sentiment dictionaries, is a reasonable solution in accelerating text labeling and enhancing its quality. After labeling, the dataset is segregated into training and testing portions. In the next step, the TF-IDF measure is used to extract features from texts. Subsequently, texts are classified with ML algorithms (naïve Bayes (NB), logistic regression (LR), support vector machine (SVM), k-nearest neighbors (k-NN), decision tree (DT), random forest (RF), XGBoost, CatBoost, etc.). Unsupervised learning [16] does not include any labeled training data and therefore does not require human participation. The most commonly employed unsupervised method is the k-means clustering [17]. This method groups similar data points together around centroids, representing the clusters' centers, and discovers their mutual features. Although clustering-based approaches do not require a preliminary stage of dataset preparation by human experts, they are susceptible to the position of centroids. Moreover, the clustering method groups instances together based on criteria that are not explicitly evident.

A number of recent studies have been devoted to a **DL-based approach** [18–20] that focuses on enhancing text classification performance by dint of its superiority in terms of accuracy when trained with a considerable amount of data. To this end, the use of deep neural networks (DNNs) [20], recurrent neural networks (RNNs) [21,22], and convolutional neural networks (CNNs) [23] is well documented in the literature. A DNN is a type of neural network (NN) that includes several layers: an input layer processing a representation of the input data, hidden layers abstracting from this representation, and an output layer that predicts a class based on the inner abstraction. A CNN is a DNN composed of convolutional [23] and pooling [24] layers. While convolutional layers filter inputs to extract features, pooling layers reduce the dimension of features. A final layer reduces the vector dimension to the length of the categorical representation of the class. An RNN is an NN where connections between neurons create a directed cycle that forms feedback loops. This type of an NN can remember previous computation steps and reuse the information in the following input sequence.

This paper focuses on the supervised ML-based approach, which is computationally fast and exhibits promising classification results. The rest of the paper is organized in the following way:

Section 2 provides an overview of the related works pertinent to the theme of this paper. In Section 3, we introduce the benchmark datasets, preprocessing steps, and ML algorithms used for the sentiment classification. In Section 4, we discuss our experimental setting, providing an extensive analysis of our results. Finally, in Section 5, we briefly delineate all the steps taken, suggest the best ML models for use in the OMSystem, and outline directions for future research.

2 Related Works

This section reviews the literature devoted to sentiment classification approaches. Research in sentiment analysis has been reflected in a large number of works in the last couple of years. As the emotional aspect of texts is generally exacting to determine unambiguously, lexicon-, ML-, and DL-based approaches have been explored in diversified ways.

A number of recent works [12–14] have presented extensive studies on the usage of lexicons and have introduced various labeling schemes for lexicon generation and news classification. Reference [25] experimented with several categories: politics, business, sports, entertainment, and technology. The lexicon dictionary was used to find the positive and negative words in a document. The whole document's sentiment score was computed by considering the sentiment value of all its words. Although the assignment of the document's sentiment with a lexicon dictionary is defined well, a few studies [14,25] did not discuss the manual check of the quality of the lexicon-based labeling by human annotators. This step is vital for sentiment analysis and is elucidated in Section 3 of this paper.

In the framework of ML-based approaches, a number of works focused on comments from the Twitter platform [10,15,18], releasing or exploiting existing large-scale datasets available for building their classifiers. The classification [26,27] of tweets with NB, k-NN, and SVM classifiers have been explored in [28–31], revealing fairly satisfactory and expeditious results despite the simplicity of their implementation. Preprocessing techniques and Bernoulli NB, SVM, and LR algorithms were used to improve the efficacy of sentiment classification [29]. Stemming and removal of redundant symbols and stop words helped to increase the accuracy of their classification results.

DNNs have also been used, among other works, in [32,33]. CNNs have been implemented for sentiment classification from Chinese text in [34]. Results computed on the Chinese datasets indicated that the accuracy was comparable with traditional ML methods. Focusing on Arabic sentiment classification, Reference [35] explored both CNNs and long short-term memory networks (LSTMs) for binary sentiment classification. Experimental results manifested an outstanding performance with an accuracy of 88% and 85% for CNN and LSTM, respectively. The combinations of CNN with LSTM and gated recurrent unit models were implemented in [36]. The binary classification was applied to five reviews and three Twitter datasets. In the experiments, an average accuracy of 90% was attained.

Most of the mentioned works focused on processing the English language that has numerous available and accessible resources. This paper observes the sentiment analysis of texts in the Russian and Kazakh languages, which has heretofore received minimal attention. Reference [37] explored sentiments of Russian tweets using LR, XGBoost, and CNNs. Reference [38] focused on ML algorithms on classifying Russian texts, but it does not provide a detailed comparison of the previously employed algorithms. Reference [39] implemented a dictionary for sentiment analysis from Kazakh texts. In [40], the sentiment analysis was performed by formalizing rules for defining the sentiment of phrases in texts. These works neither conducted a thorough study

of the sentiment classification with various lexicon-and ML-based approaches nor presented a comparison with the results attained by the previous similar works. Thus, this paper delivers a more comprehensive sentiment analysis of Russian and Kazakh texts with seven extensively deployed ML algorithms.

3 Methodology

This section describes the principal steps of text preprocessing [41,42], class resampling [43–45], feature selection, and text classification with the use of ML algorithms [46]. These steps and the underlying logic are graphically represented in Fig. 2.

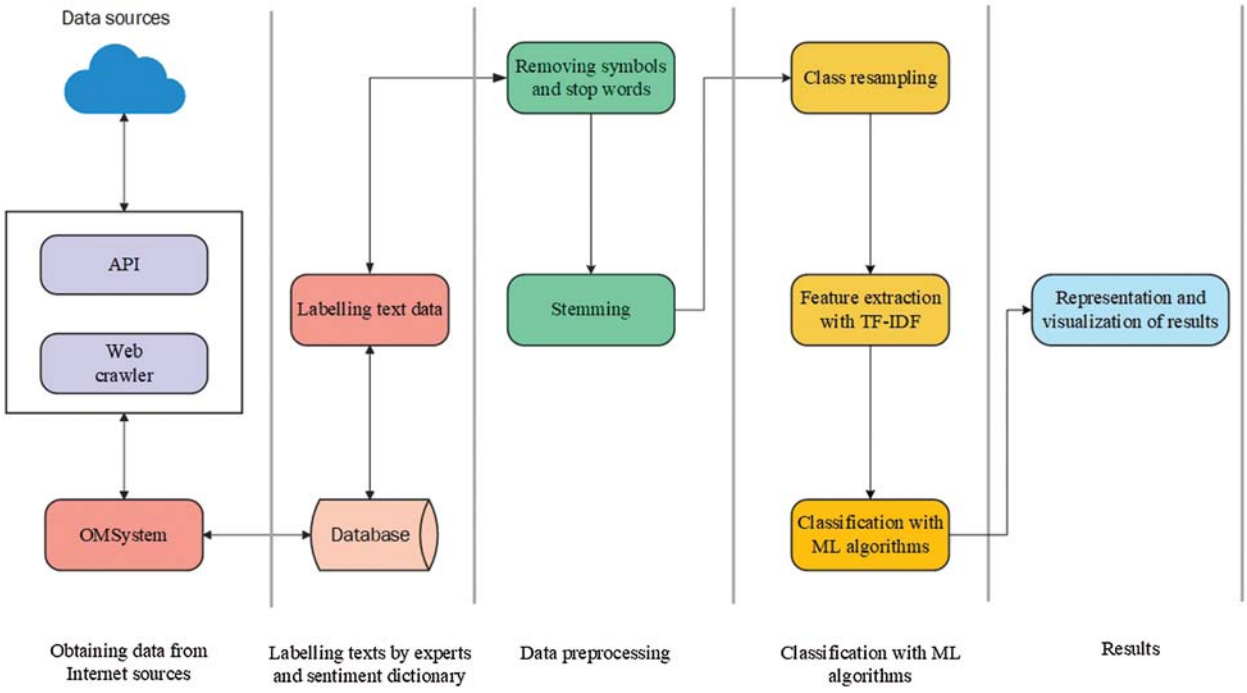

Figure 2: Stages of classification with ML algorithms

3.1 Datasets

The texts used to build our training and testing datasets were collected with the web-crawler provided by the OMSystem. The primary sources were the leading news portals of Kazakhstan, namely: "Nur" (https://www.nur.kz/), "Informburo" (https://www.informburo.kz/), "Today" (http://www.today.kz/), "Kazinform" (https://www.kazinform.kz/), "KazTag" (https://www.kaztag.kz/ru/), "Holanews" (https://www.holanews.kz/), "Forbes" (https://www.forbes.kz/), "Zakon" (https://www.zakon.kz/), "Time" (https://www.time.kz/), "Vlast" (https://www.vlast.kz/), "Tengrivews" (https://www.tengrinews.kz/), "Kapital" (https://www.kapital.kz/), and "The village" (https://www.the-village-kz.com/).

The downloaded texts were labeled according to three sentiment classes: positive, negative, and neutral. The initial labeling was realized through a sentiment dictionary. Subsequently, the labeled texts were manually examined and corrected by Masters and Ph.D. students in political science. Each text was commonly reviewed by three annotators separately, and the final label was assigned

on the basis of the majority of votes. The total number of manually-revised sentiment-labeled texts is 80,873 in Russian and 15,933 in Kazakh. Tab. 1 provides a distribution of the downloaded texts over three classes.

Table 1: Distribution of texts over classes

Language	Positive	Negative	Neutral
Russian	59,425	17,494	3,954
Kazakh	14,071	1,366	496

3.2 Data Processing

The retrieved texts are required to be preprocessed prior to the subsequent steps. First, all words were transformed to the lowercase register. Afterward, the punctuation marks, digits, special symbols, and links were dropped as they did not carry any pertinence in most instances [37]. Additionally, it was necessary to remove the extremely frequent words (i.e., stop words such as "и" (and), "В" (in), "еще" (yet), "бұл" (this), "әрбір" (each), "үшін" (for)).

Furthermore, stemming or lemmatization has to be performed to reduce the number of words with similar emotional meanings [37,38]. The difference between these approaches is that the latter obtains an infinitive form of the words, whereas the former eliminates affixes and endings of words to gain a root. In this paper, stemming was used because there is no well-designed lemmatizer for the Kazakh language. Its complete development is overly taxing. "SnowballStemmer" from Python NLTK library was applied for words in the Russian language, and our own "KazakhStemmer," based on a full set of affixes and endings, was designed to process words in the Kazakh language.

3.3 Class Resampling

Imbalanced classes act as a notable challenge in training a good classifier, both for binary and multi-class classification tasks [43–45]. As the classes are highly imbalanced, a majority classifier would yield fairly accurate results labeling all instances with the most represented class. However, failing on all the items belonging to the other two classes would perform poorly in terms of precision, recall, and F1-score, representing our primary evaluation metrics. Class resampling techniques provide us with different alternative solutions to avoid this problem. Among them, for our experiments, we chose three widely used techniques (random undersampling, random oversampling [43–45], and synthetic minority oversampling (SMOTE)), leaving the exploration of alternative approaches for future research (Fig. 3).

The undersampling method eliminates the segment of the training dataset belonging to the majority class to make it close to or equal in size to the minority class. The drawback of such a solution is that the minority class is too small to reduce the other two classes to its size, and therefore, a large part of pertinent and valuable information is lost. In the oversampling method, a contrasting operation is realized. The minority class is increased in its size to match the majority class by coping multiple times its instances to reach the desired size. This solution has the advantage of preserving all the valuable information in the dataset.

SMOTE is another prevalent oversampling technique wherein new points are synthesized between the existing ones. The procedure is typically contemplated as a hypercube between each

point of the minority class and its k nearest neighbor points. New artificial points are created inside the hypercube. This solution has a conspicuous advantage of preserving useful information and even increasing its size.

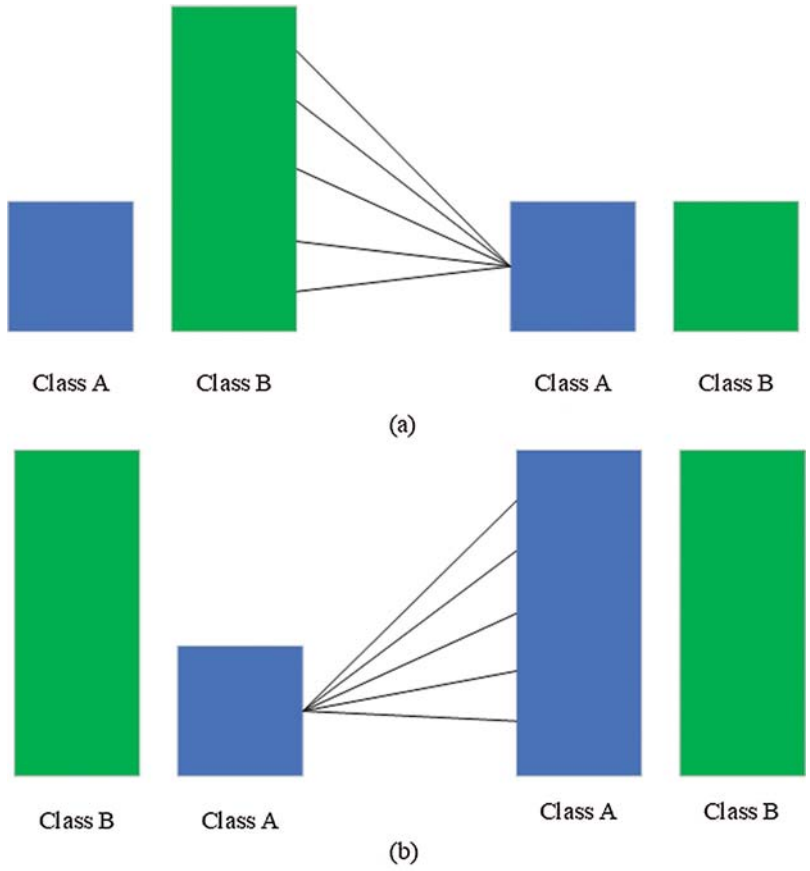

Figure 3: Class resampling—(a) random undersampling and (b) random oversampling

3.4 Text Vectorization

The text vectorization step aims at transforming texts into a numeric vector representation on which ML algorithms can be readily applied. "Bag of words" is a simple vectorization approach wherein every text from the dataset is represented as a vector with a length equal to the vocabulary of the dataset. In this encoding model, a vector is filled with the frequency of each word that appears in the text. Despite the simplicity of this approach, vectors are generally observed to be very long with lots of zeros. Besides, it does not consider the importance of the words. Along this direction, a valid alternative is represented by TF-IDF [47]. It is based on "Bag of words" but targets the pertinence of words in a given document. TF-IDF is calculated by:

$$TF\_IDF = TF \times IDF. \tag{1}$$

TF is the ratio of a word occurrence in a document to the number of words in the document:

$$TF_{d,w} = \frac{count\,(w,\, d)}{count\,(N,\, d)}, \tag{2}$$

where, $count\,(w, d)$ is the frequency of a word w in a document d and $count\,(N, d)$ is the number of words N in a document d.

IDF provides the weight of each word based on its frequency in the *corpus D*:

$$IDF_{w,d} = \log \frac{count\,(D)}{count\,(w, d)}, \tag{3}$$

where, $count\,(D)$ is the number of documents.

3.5 Classification with ML Algorithms

ML text classification has been performed with various algorithms, including NB, LR, SVM, k-NN, DT, RF, and XGBoost.

An NB classifier [28], based on the Bayes' theorem, is a probabilistic ML model used for the task of classification. It is fast to implement and yields promising results. The classifier is based on the conditional probability that a document d belongs to a class c. Bayes' formula lies in the foundation of the algorithm. For sentiment analysis, the formula of the classifier is given by:

$$P\,(c \mid d) = \frac{P\,(c) \times P\,(d \mid c)}{P\,(d)}, \tag{4}$$

where, $d = \{x_1, x_2, \ldots, x_n\}$, x_i is a weight of the ith word in a document d and c is the document class.

A multi-class SVM [30,31] has the objective to allocate the hyperplane that can effectively divide the input data into multiple separate classes. The type of hyperplane depends on multiple features. If the number of features is two, trivially, the hyperplane is a line. If the number of features is three, the hyperplane is in the form of a two-dimensional plane. If the number of features exceeds three, the hyperplane takes a complex form. An equation of the hyperplane can be written as:

$$y_i\left(\vec{w} \times \vec{x} + b\right) \geq 0, \tag{5}$$

where, $\vec{x} = (x_1, x_2, \ldots, x_n)$ is an input vector; $\vec{w} = (w_1, w_2, \ldots, w_n)$ is a weight vector; y_i is an output value; b is the bias. If the value is more than or equal to zero, it belongs to a positive class. Otherwise, it belongs to a negative class.

An LR [29] predicts the probability of an outcome by fitting data to a logistic function. The classifier uses a linear function $f\,(x) = w_0 + w_1 x_1 + \ldots + w_r x_r$, where $w_0, w_1, \ldots, w_r$ are the predicted weights or coefficients. The LR function $p\,(x)$ is a sigmoid function

$$p\,(x) = \frac{1}{1 + e^{-f\,(x)}}. \tag{6}$$

The resultant p(x) is a probability value between 0 and 1. The document d belongs to the first class if the value of p(x) is close to zero. Otherwise, it is placed in the second class. For multi-class classification, we adopted the one-*vs.*-one (OvO) approach to identify a particular class. In this approach, a multi-class dataset is split into several binary classification issues where each binary classifier is trained on instances belonging to one class and instances belonging to one other class. In the one-*vs.*-all (OvA), numerous binary classifiers are trained to distinguish instances from one

class from all other instances. An advantage of the OvO over the OvA is that the datasets of all individual classifiers are balanced when the whole multi-class dataset is balanced.

A k-nn [30] is a traditional non-parametric algorithm for classifying data instances. It calculates the distances between the vectors and assigns the points to the class of its k nearest neighbor points. This algorithm generally classifies documents with the most widely used distance measure called the Euclidean distance, defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^{N} \left(a_{ix} - a_{iy}\right)^2}, \tag{7}$$

where, $d(x, y)$ is the distance between two documents; a_{ix} and a_{iy} are weights of the ith terms in the documents x and y, respectively; N is the number of unique words in a list of documents.

A DT [45] is a structure with N nodes. In the first step, a word is chosen, and all documents holding the same are placed on one side, and documents not containing it are put on the other side. This way, two separate sets are created. Subsequent to that, a new word is selected in these sets, and all the previous steps are repeated. The entire procedure continues until a set in which all documents are assigned to the same class is attained. In the RF classifier [48], a bunch of independent trees is built. Every document is classified by the trees independently. The class of the document is defined by the largest number of votes of all trees.

XGboost [45] is one of the most extensively used ML algorithms. It has a good performance and solves most regression and classification problems. Boosting represents an ensemble technique where previous errors are resolved in a new model. The diversions of the trained ensemble's predictions are computed on a training set at each iteration. Thereby, the optimization is done by adding the new tree predictions to the ensemble, decreasing the model's mean deviation. This procedure continues until the required level of the error is reached, or the "early stopping" criterion is achieved.

4 Experiments and Discussion

In this study, Python 3.8 with efficient NLTK, Scikit-learn, Imbalanced-learn, Matplotlib, and Seaborn libraries were used for the experiments. Tokenization, removal of stop words, and stemming were performed by NLTK. Python's Imbalanced-learn package was utilized for class resampling. Vectorization and classification were accomplished by Scikit-learn. The plots of Matplotlib and Seaborn were used to visualize the experimental results. The required steps were taken in the following order. Texts were preprocessed, resampled with three techniques (undersampling, oversampling, and SMOTE), vectorized with TF-IDF, and classified with the ML algorithms described in Section 3.5. Different metrics were used depending on the classification task to measure the performance of classifiers.

Binary classification (into positive/negative)

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{8}$$

$$precision = \frac{TP}{TP + FP}, \tag{9}$$

$$recall = \frac{TP}{TP + FN}, \tag{10}$$

$$F1\_score = 2\frac{precision \times recall}{precision + recall}, \tag{11}$$

where, *TP* (true positive) indicates a test instance correctly classified with the *positive* sentiment class; *TN* (true negative) indicates a test instance correctly classified with the *negative* sentiment class; *FP* (false positive) indicates a test instance wrongly classified with the *positive* sentiment class; *FN* (false negative) indicates a test instance wrongly classified with the *negative* sentiment class.

Multi-class classification into positive/negative/neutral. The following metrics were implemented: precision-macro, precision-micro, precision-weighted, recall-macro, recall-micro, recall-weighted, F1-score macro, F1-score micro, and F1-score-weighted. Precision-macro is the arithmetic mean of all precision scores for all classes. Precision-micro is the sum of all true positives for all classes, divided by all the positive predictions.

$$precision\_macro = \frac{precision_1 + precision_2 + precision_3}{3}, \tag{12}$$

$$precision\_micro = \frac{TP_1 + TP_2 + TP_3}{TP_1 + TP_2 + TP_3 + FP_1 + FP_2 + FP_3}. \tag{13}$$

Recall-macro and recall-micro are defined in a similar manner

$$recall\_macro = \frac{recall_1 + recall_2 + recall_3}{3}, \tag{14}$$

$$recall\_micro = \frac{TP_1 + TP_2 + TP_3}{TP_1 + TP_2 + TP_3 + FN_1 + FN_2 + FN_3}. \tag{15}$$

The weighted average is computed like the macro average; however, each class has a weight according to the number of entries that belong to it. Weighted precision and recall are calculated in the following way.

$$precision\_weighted = \frac{w_1 \times precision_1 + w_2 \times precision_2 + w_3 \times precision_3}{3}, \tag{16}$$

$$recall\_weighted = \frac{w_1 \times recall_1 + w_2 \times recall_2 + w_3 \times recall_3}{3}, \tag{17}$$

where, w_1, w_2, *and* w_3 are the weights of the corresponding classes.

In experimental results, datasets were randomly divided into training 70% and testing 30% sets. The seven ML algorithms were then applied to texts, and the corresponding results were obtained. The classification results computed on the imbalanced Russian and Kazakh language datasets are shown in Tab. 2. Multi-class classification metrics and a confusion matrix of texts in the Russian language with LR are shown in Fig. 4.

The results of the classification of oversampled datasets are encapsulated in Tab. 3. Multi-class classification metrics and a confusion matrix of Russian texts with LR are shown in Fig. 5.

The results of the classification of SMOTE datasets are shown in Tab. 4. Multi-class classification metrics and a confusion matrix of Russian texts with LR are shown in Fig. 6.

The results of the classification of undersampled datasets are shown in Tab. 5. Multi-class classification metrics and a confusion matrix of Russian texts with LR are shown in Fig. 7.

Table 2: Classification of imbalanced datasets

Classifier	NB	SVM	LR	k-NN	DT	RF	XGBoost	Average
Russian texts								
Accuracy	0.74	0.73	0.81	0.77	0.76	0.80	0.75	0.77
Precision-macro	0.49	0.24	0.75	0.62	0.62	0.75	0.64	0.59
Precision-micro	0.74	0.73	0.81	0.77	0.76	0.80	0.75	0.77
Precision-weighted	0.70	0.54	0.80	0.75	0.76	0.80	0.72	0.72
Recall-macro	0.35	0.33	0.57	0.53	0.60	0.57	0.42	0.48
Recall-micro	0.74	0.73	0.81	0.77	0.76	0.80	0.75	0.77
Recall-weighted	0.74	0.73	0.81	0.77	0.76	0.80	0.75	0.77
F1-score macro	0.33	0.28	0.61	0.56	0.60	0.63	0.44	0.49
F1-score micro	0.74	0.73	0.81	0.77	0.76	0.80	0.75	0.77
F1-score-weighted	0.65	0.62	0.79	0.76	0.76	0.79	0.70	0.72
Kazakh texts								
Accuracy	0.89	0.89	0.90	0.89	0.88	0.92	0.89	0.89
Precision-macro	0.30	0.30	0.77	0.59	0.58	0.82	0.69	0.58
Precision-micro	0.89	0.89	0.90	0.90	0.88	0.92	0.89	0.90
Precision-weighted	0.80	0.80	0.90	0.87	0.88	0.91	0.87	0.86
Recall-macro	0.33	0.33	0.43	0.48	0.57	0.52	0.40	0.44
Recall-micro	0.89	0.89	0.90	0.89	0.88	0.92	0.89	0.89
Recall-weighted	0.89	0.89	0.90	0.89	0.88	0.92	0.89	0.89
F1-score macro	0.31	0.31	0.48	0.51	0.57	0.60	0.44	0.46
F1-score micro	0.89	0.89	0.90	0.89	0.88	0.92	0.89	0.89
F1-score-weighted	0.84	0.83	0.87	0.88	0.88	0.90	0.86	0.87

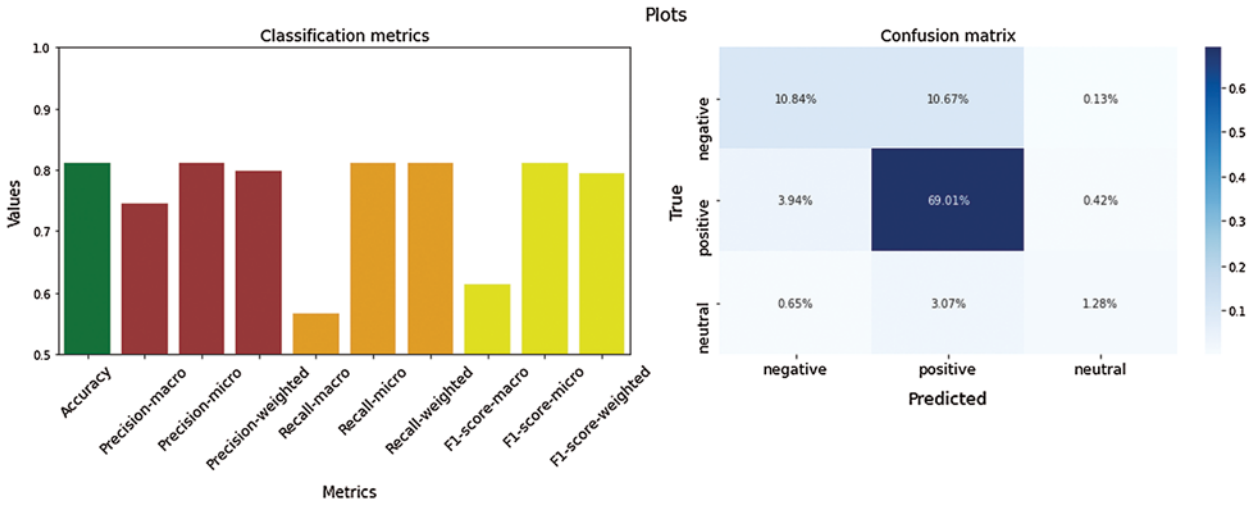

Figure 4: Classification metrics of imbalanced Russian texts

Table 3: Classification of oversampled datasets

Classifier	NB	SVM	LR	k-NN	DT	RF	XGBoost	Average
Russian texts								
Accuracy	0.73	0.63	0.86	0.66	0.92	0.93	0.67	0.77
Precision-macro	0.75	0.63	0.86	0.77	0.92	0.93	0.67	0.79
Precision-micro	0.73	0.63	0.86	0.66	0.92	0.93	0.67	0.77
Precision-weighted	0.74	0.63	0.86	0.77	0.92	0.93	0.67	0.79
Recall-macro	0.73	0.63	0.86	0.65	0.92	0.93	0.66	0.77
Recall-micro	0.73	0.63	0.86	0.66	0.92	0.93	0.67	0.77
Recall-weighted	0.73	0.63	0.86	0.66	0.92	0.93	0.67	0.77
F1-score macro	0.73	0.62	0.86	0.63	0.91	0.93	0.66	0.76
F1-score micro	0.73	0.63	0.86	0.66	0.92	0.93	0.67	0.77
F1-score-weighted	0.73	0.62	0.86	0.63	0.91	0.93	0.66	0.76
Kazakh texts								
Accuracy	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
Precision-macro	0.85	0.61	0.95	0.94	0.96	0.99	0.76	0.87
Precision-micro	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
Precision-weighted	0.85	0.61	0.95	0.94	0.96	0.99	0.76	0.87
Recall-macro	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
Recall-micro	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
Recall-weighted	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
F1-score macro	0.85	0.61	0.95	0.93	0.96	0.99	0.76	0.86
F1-score micro	0.85	0.62	0.95	0.93	0.96	0.99	0.76	0.87
F1-score-weighted	0.85	0.61	0.95	0.93	0.96	0.99	0.76	0.86

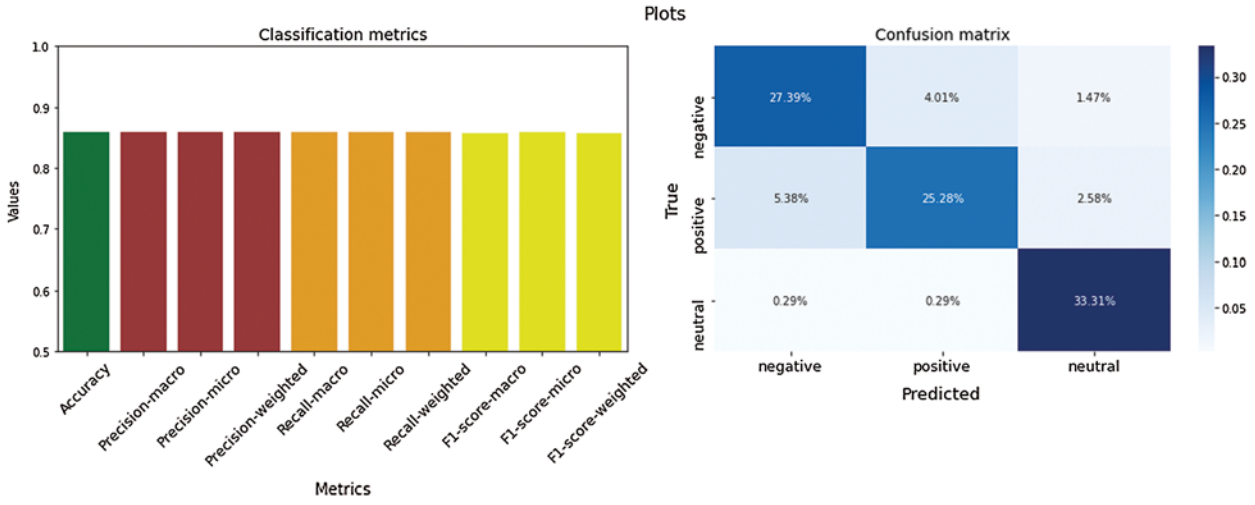

Figure 5: Classification metrics of oversampled Russian texts

Table 4: Classification of SMOTE datasets

Classifier	NB	SVM	LR	k-NN	DT	RF	XGBoost	Average
Russian texts								
Accuracy	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
Precision-macro	0.75	0.65	0.87	0.80	0.85	0.90	0.67	0.78
Precision-micro	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
Precision-weighted	0.75	0.65	0.87	0.80	0.85	0.90	0.67	0.78
Recall-macro	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
Recall-micro	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
Recall-weighted	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
F1-score macro	0.74	0.64	0.87	0.66	0.85	0.90	0.67	0.76
F1-score micro	0.73	0.65	0.87	0.70	0.85	0.90	0.67	0.77
F1-score-weighted	0.74	0.64	0.87	0.66	0.85	0.90	0.67	0.76
Kazakh texts								
Accuracy	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
Precision-macro	0.87	0.61	0.95	0.84	0.94	0.97	0.73	0.84
Precision-micro	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
Precision-weighted	0.87	0.61	0.95	0.84	0.94	0.97	0.73	0.84
Recall-macro	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
Recall-micro	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
Recall-weighted	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
F1-score macro	0.87	0.61	0.95	0.75	0.94	0.97	0.72	0.83
F1-score micro	0.87	0.62	0.95	0.79	0.94	0.97	0.73	0.84
F1-score-weighted	0.87	0.61	0.95	0.75	0.94	0.97	0.72	0.84

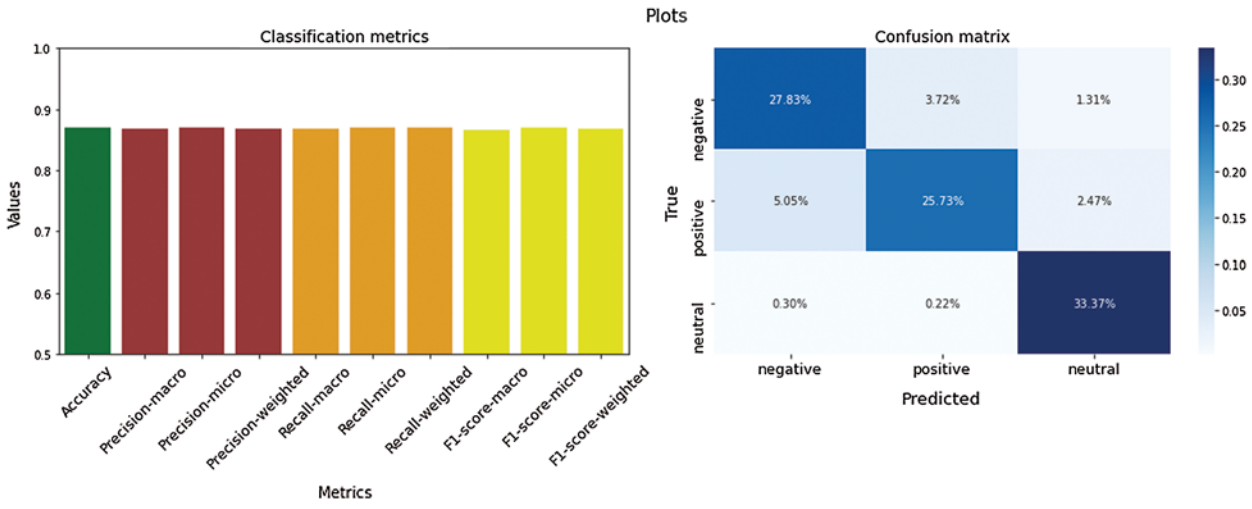

Figure 6: Classification metrics of SMOTE Russian texts

Table 5: Classification of undersampled datasets

Classifier	NB	SVM	LR	k-NN	DT	RF	XGBoost	Average
Russian texts								
Accuracy	0.55	0.63	0.74	0.62	0.67	0.70	0.65	0.65
Precision-macro	0.66	0.64	0.74	0.63	0.66	0.70	0.65	0.67
Precision-micro	0.55	0.63	0.74	0.62	0.67	0.70	0.65	0.65
Precision-weighted	0.66	0.64	0.74	0.63	0.66	0.70	0.65	0.67
Recall-macro	0.55	0.64	0.74	0.62	0.67	0.70	0.65	0.65
Recall-micro	0.55	0.63	0.74	0.62	0.67	0.70	0.65	0.65
Recall-weighted	0.55	0.63	0.74	0.62	0.67	0.70	0.65	0.65
F1-score macro	0.52	0.63	0.73	0.61	0.66	0.69	0.64	0.64
F1-score micro	0.55	0.63	0.74	0.62	0.67	0.70	0.65	0.65
F1-score-weighted	0.52	0.63	0.73	0.61	0.66	0.69	0.64	0.64
Kazakh texts								
Accuracy	0.57	0.60	0.75	0.65	0.68	0.68	0.71	0.66
Precision-macro	0.66	0.61	0.76	0.65	0.68	0.68	0.71	0.68
Precision-micro	0.57	0.60	0.75	0.65	0.68	0.68	0.71	0.66
Precision-weighted	0.66	0.61	0.77	0.65	0.68	0.68	0.71	0.68
Recall-macro	0.57	0.61	0.76	0.65	0.69	0.68	0.71	0.67
Recall-micro	0.57	0.60	0.75	0.65	0.68	0.68	0.71	0.66
Recall-weighted	0.57	0.60	0.75	0.65	0.68	0.68	0.71	0.66
F1-score macro	0.55	0.59	0.75	0.65	0.68	0.68	0.71	0.66
F1-score micro	0.57	0.60	0.75	0.65	0.68	0.68	0.71	0.66
F1-score-weighted	0.55	0.59	0.75	0.65	0.68	0.68	0.70	0.66

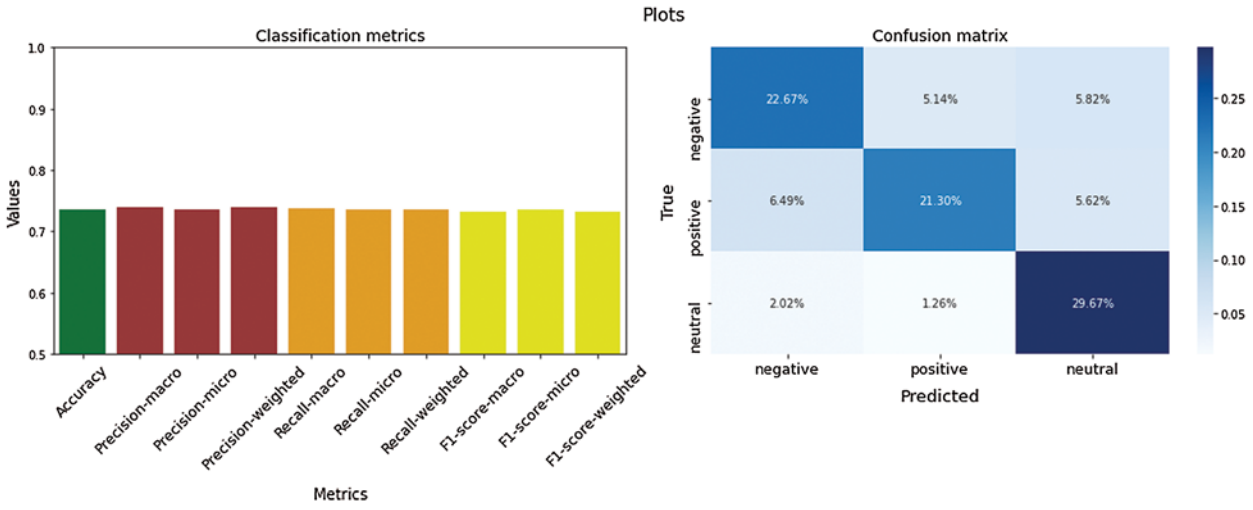

Figure 7: Classification metrics of undersampled Russian texts

The results of different classification models reveal that the models trained on imbalanced data achieve the lowest performance. Data undersampling obtains medium results, possibly owing to the fact that the resulting models cannot take full advantage of the whole training material available. As expected, the oversampled and SMOTE models, which make better use of the available data, achieve the best results. Among the various ML models tested, LR, DT, and RF yielded the best performances. Although the NB classifier performs well, it is worth remarking that the algorithm suffers from the known limitations associated with the assumption that all its features are mutually independent. Despite its simplicity, k-NN attains satisfactory results on datasets with a small size. Nevertheless, it tends to be slower and less accurate with larger corpora. As the RF classifier uses a number of independent DTs, and it is apparent that its performance is superior to a single DT. In a previous study, singular value decomposition [49] was applied to texts where they were classified with SVM and XGBoost. It was done to speed up the algorithms' training, so it is one reason explaining that these classifiers are under-performing compared to others. The classification results across the Russian and Kazakh languages are comparatively equal with slightly better performance for the latter in the oversampled and SMOTE datasets, having a smaller testing size. In summary, it could be seen that large balanced datasets, obtained with oversampling and SMOTE approaches, are the best ones and preferable to be used in the social analytics platforms.

5 Conclusion

We described the OMSystem, the advanced analytical system for monitoring Kazakhstan's most popular news portals and social networks. We focused on the sentiment analysis component for automatic text labeling. We described its core functionalities, processing steps, and algorithms (NB, SVM, LR, k-NN, DT, RF, and XGBoost), discussing their strengths and weaknesses given our text classification task. Before applying these ML algorithms, texts were preprocessed to remove punctuation, extra symbols, and stop words, stemmed, and resampled to account for the highly imbalanced data the system has to be trained on. Specifically, resampling was performed with random undersampling, random oversampling, and SMOTE. As far as the features are concerned, in our work, we concentrated on feeding our models with word frequency information supplied in the form of TF-IDF values. Classification performance was measured with different metrics (accuracy, precision, recall, and F1-score), taking into account the various data conditions (imbalanced and balanced through resampling). Besides, the corresponding histograms were built to visualize the classification metrics. The analysis of our results reveals that LR, DT, and RF with random oversampling and SMOTE are the most suitable ones to address the said task.

Based on this research, the best ML classification models for estimating social mood are included in the OMSystem for evaluating people's attitude toward significant events in society and their level of interest and involvement in different topics. The social mood on specific topics is determined by finding the largest number of texts belonging to one of three sentiment categories. As the corpora of labeled texts and the base word thesaurus used to understand their content are constantly growing, our ML models are periodically retrained to improve their sentiment classification performance. Moreover, additional future works will include strengthening these ML models by applying CNN, RNN, and bi-directional encoder representation for transformers.

Acknowledgement: We would like to thank the Center for data analysis and processing of Al-Farabi Kazakh National University for providing the datasets obtained with the OMSystem.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Dave, S. Lawrence and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. of the 12th Int. Conf. on World Wide Web*, Budapest, Hungary, pp. 519–528, 2003.

[2] M. V. Mantyla, D. Graziotin and M. Kuuntila, "The evolution of sentiment analysis–a review of research topics, venue, and top cited papers," *Computer Science Review*, vol. 27, no. 1, pp. 16–32, 2018.

[3] I. Moutidis and H. T. P. Williams, "Good and bad events: Combining network-based event detection with sentiment analysis," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–12, 2020.

[4] M. Haselmayer and M. Jenny, "Sentiment analysis of political communication: Combining a dictionary approach with crowdcoding," *Quality & Quantity*, vol. 51, no. 6, pp. 2623–2646, 2017.

[5] O. Simek, D. Shah and A. Heier, "Prototype and analytics for discovery and exploitation of threat networks on social media," in *European Intelligence and Security Informatics Conf.*, Oulu, Finland, pp. 9–16, 2019.

[6] N. Seman and N. A. Razmi, "Machine learning-based technique for big data sentiments extraction," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 3, pp. 473–479, 2020.

[7] W. N. Chan and T. Thein, "Sentiment analysis system in big data environment," *Computer Systems Science and Engineering*, vol. 33, no. 3, pp. 187–202, 2018.

[8] P. Jadon, D. Bhatia and D. K. Mishra, "A new methodology on sentiment analysis," in *1st FICR Int. Conf. on Rising Threats in Expert Applications and Solutions*, Jaipur, India, pp. 617–625, 2020.

[9] K. Kour, J. Kour and P. Singh, "Lexicon-based sentiment analysis," in *Advances in Communication and Computational Technology Select Proc. of ICACCT*, Singapore, pp. 1421–1430, 2019.

[10] K. Zarisfi, F. Sadeghi and E. Eslami, "Solving the Twitter sentiment analysis problem based on a machine learning-based approach," *Evolutionary Intelligence*, vol. 13, no. 3, pp. 381–398, 2020.

[11] C. Dang, M. García and F. De La Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3: 483, pp. 1–29, 2020.

[12] D. Abd, A. Abbas and A. Sadiq, "Analyzing sentiment system to specify polarity by lexicon-based," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 283–289, 2021.

[13] Y. M. Aye and S. S. Aung, "Contextual lexicon-based sentiment analysis in Myanmar text reviews," in *23rd Conf. of the Oriental COCOSDA Int. Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques*, Yangon, Myanmar, pp. 160–165, 2020.

[14] S. Yadav and N. Saleena, "Sentiment analysis of reviews using an augmented dictionary approach," in *5th Int. Conf. on Computing, Communication and Security*, Patna, India, pp. 1–5, 2020.

[15] G. R. Kumar, K. V. Sheshanna and G. A. Babu, "Sentiment analysis for airline tweets utilizing machine learning techniques," in *Int. Conf. on Mobile Computing and Sustainable Informatics*, Lalitpur, Nepal, pp. 791–799, 2020.

[16] P. H. Jigneshkumar, V. J. Prakash and A. Patel, "Unsupervised learning-based sentiment analysis with reviewer's emotion," in *Int. Conf. on Evolving Technologies for Computing, Communication and Smart World*, Noida, India, pp. 69–81, 2020.

[17] S. Narynov, D. Mukhtarkhanuly, B. Omarov, K. Kozhakhmet and B. Omarov, "Machine learning approach to identifying depression related posts on social media," in *20th Int. Conf. on Control, Automation and Systems*, Busan, Korea (South), pp. 6–11, 2020.

[18] A. Alharbi and E. Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," *Cognitive Systems Research*, vol. 54, pp. 50–61, 2019.

[19] R. Ghosh, K. Ravi and V. Ravi, "A novel deep learning architecture for sentiment classification," in *Proc. of the 2016 3rd Int. Conf. on Recent Advances in Information Technology*, Dhanbad, India, pp. 511–516, 2016.

[20] Z. Z. Wint, Y. Manabe and M. Aritsugi, "Deep learning-based sentiment classification in social network services datasets," in *IEEE Int. Conf. on Big Data, Cloud Computing, Data Science & Engineering*, Yonago, Japan, pp. 91–96, 2018.

[21] D. Röchert, G. Neubaum and S. Stieglitz, "Identifying political sentiments on YouTube: A systematic comparison regarding the accuracy of recurrent neural network and machine learning models," in *Disinformation in Open Online Media*, Leiden, The Netherlands, pp. 107–121, 2020.

[22] R. Socher, C. C. Lin, C. Manning and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. of the 28th Int. Conf. on Machine Learning*, Bellevue, WA, USA, pp. 129–136, 2011.

[23] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *IEEE Access*, vol. 6, pp. 13949–13957, 2018.

[24] S. Smetanin and M. Komarov, "Sentiment analysis of product reviews in Russian using convolutional neural networks," in *IEEE 21st Conf. on Business Informatics*, Moscow, Russia, pp. 482–486, 2019.

[25] S. Taj, A. Meghji and B. Shaikh, "Sentiment Analysis of News Articles: A lexicon based approach," in *Int. Conf. on Computing, Mathematics and Engineering Technologies–iCoMET*, Sukkur, Sindh, Pakistan, pp. 1–5, 2019.

[26] A. P. Jain and D. Padma, "Application of machine learning techniques to sentiment analysis," in *2nd Int. Conf. on Applied and Theoretical Computing and Communication Technology*, Karnataka, India, pp. 628–632, 2016.

[27] X. Zhang and X. Zheng, "Comparison of text sentiment analysis based on machine learning," in *Proc. of the 15th Int. Symp. on Parallel and Distributed Computing*, Fuzhou, China, pp. 230–233, 2016.

[28] M. Vikas and A. Kumar, "Sentiment analysis of twitter data using naive Bayes algorithm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 6, no. 4, pp. 120–125, 2018.

[29] P. Sharma and A. K. Sharma, "Score prediction model for sentiment classification using machine learning algorithms," in *4th Int. Conf. on Information and Communication Technology for Intelligent Systems*, Ahmedabad, India, pp. 745–753, 2020.

[30] M. R. Huq, A. Ali and A. Rahman, "Sentiment analysis on twitter data using KNN and SVM," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 6, pp. 19–25, 2017.

[31] K. Lavanya and C. Deisy, "Twitter sentiment analysis using multiclass SVM," in *Int. Conf. on Intelligent Computing and Control*, Coimbatore, India, pp. 1–6, 2017.

[32] A. M. Ramadhani and H. Goo, "Twitter sentiment analysis using deep learning methods," in *Proc. of the 7th Int. Annual Engineering Seminar*, Yogyakarta, Indonesia, pp. 1–4, 2017.

[33] P. Cen, K. Zhang and D. Zheng, "Sentiment analysis using deep learning approach," *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 17–27, 2020.

[34] F. Xu, X. Zhang, Z. Xin and A. Yang, "Investigation on the Chinese text sentiment analysis based on convolutional neural networks in deep learning," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 697–709, 2019.

[35] A. Ombabi, W. Ouarda and A. Alimi, "Deep learning CNN-LSTM framework for Arabic sentiment analysis using textual information shared in social networks," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–13, 2020.

[36] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, no. 3, pp. 279–294, 2021.

[37] A. Znovarev and A. Bilyi, "A comparison of machine learning methods of sentiment analysis based on Russian language twitter data," in *11th Majorov Int. Conf. on Software Engineering and Computer Systems*, Saint Petersburg: Russian Federation, pp. 1–7, 2020.

[38] M. A. Hamada, K. Sultanbek, B. Alzhanov and B. Tokbanov, "Sentimental text processing tool for Russian language based on machine learning algorithms," in *Proc. of the 5th Int. Conf. on Engineering and MIS*, Astana, Kazakhstan, pp. 1–6, 2019.

[39] G. Bekmanova, G. Yelibayeva, S. Aubakirova, N. Dyussupova, A. Sharipbay *et al.,* "Methods for analyzing polarity of the Kazakh texts related to the terrorist threats," in *19th Int. Conf. on Computational Science and Its Applications*, Saint Petersburg: Russian Federation, pp. 717–730, 2019.

[40] B. Yergesh, G. Bekmanova and A. Sharipbay, "Sentiment analysis of Kazakh text and their polarity," *Web Intelligence*, vol. 17, no. 1, pp. 9–15, 2019.

[41] Y. Bao, C. Quan, L. Wang and F. Ren, "The role of preprocessing in twitter sentiment analysis," *Lecture Notes in Computer Science*, vol. 8589, pp. 615–624, 2014.

[42] E. Haddi, X. Liu and Y. Shi, "The role of text preprocessing in sentiment analysis," *Procedia Computer Science*, vol. 17, pp. 26–32, 2013.

[43] A. Mukherjee, S. Mukhopadhyay, P. K. Panigrahi and S. Goswami, "Utilization of oversampling for multiclass sentiment analysis on Amazon review dataset," in *IEEE 10th Int. Conf. on Awareness Science and Technology*, Morioka, Japan, pp. 1–6, 2019.

[44] W. D. Alnatara and M. L. Khodra, "Imbalanced data handling in multi-label aspect categorization using oversampling and ensemble learning," in *Int. Conf. on Advanced Computer Science and Information Systems*, Depok, Indonesia, pp. 165–170, 2020.

[45] D. P. Chatterjee, S. Mukhopadhyay, S. Goswami and P. K. Panigrahi, "Efficacy of oversampling over machine learning algorithms in case of sentiment analysis," *Data Management, Analytics and Innovation. Advances in Intelligent Systems and Computing*, vol. 1175, pp. 247–260, 2020.

[46] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 5, pp. 1–14, 2015.

[47] H. T. Sueno, B. D. Gerardo and R. P. Medina, "Converting text to numerical representation using modified Bayesian vectorization technique for multiclass classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5618–5623, 2020.

[48] M. Ghosh and G. Sanyal, "An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning," *Journal of Big Data*, vol. 5, no. 1, pp. 1–25, 2018.

[49] P. Symeonidis, I. Kehayov and Y. Manolopoulos, "Text classification by aggregation of SVD eigenvectors," in *16th East European Conf. ADBIS*, Poznan, Poland, pp. 385–398, 2012.