Khazar Journal of Science and Technology `Volume 9 №1 2025, 47-59

DOI: 10.5782/2520-6133.2025.9.1.47

Sentiment Analysis Using Machine Learning Methods on Social Media

Jamila Damirova[1], Leyla Muradkhanli[2]
[1,2] Khazar University, Baku, Azerbaijan
[2]Baku Higher Oil School, Baku, Azerbaijan
Corresponding author: jdamirova@khazar.org

Abstract
Sentiment analysis deals with understanding human feelings and opinions by analyzing the emotional content of words. With the rise of social media platforms, an immense volume of text data has become available for analysis. Machine learning (ML) techniques are essential to process this data and can provide businesses with deep insights into customer feedback, brand reputation, and emerging market trends. They also help governments and public organizations gauge public opinion on current events, proposed laws, and social issues. This study aims to improve the accuracy and effectiveness of sentiment analysis on social media (specifically Twitter) by applying advanced ML methods to address challenges of contextual understanding, noisy text preprocessing, and the evolving slang and vernacular of online content. We developed a sentiment classification model for Twitter data and evaluated several algorithms on a real-world dataset. The results show that a ML approach can successfully classify social media posts by sentiment, highlighting prevailing public moods in real time. In our experiments, an ensemble model outperformed other classifiers in balancing precision and recall, achieving high overall accuracy. These findings showcase the potential of ML methods in capturing the sentiment of social media discourse.

Keywords: sentiment analysis, social media, Twitter, machine learning, data mining.

Introduction
Sentiment analysis (also known as opinion mining) centers on extracting and identifying subjective information from text to determine the sentiment (positive, negative, or neutral) expressed by a writer or speaker. The explosion of user-generated content on social media has created a vast repository of textual data that organizations can analyze to understand public emotions and opinions. However, the sheer volume and informal nature of social media text make manual analysis infeasible, necessitating automatic methods. Machine learning has become indispensable in this context,

enabling the efficient processing and analysis of massive datasets of tweets, posts, and comments.

Social media sentiment analysis can yield valuable insights across various domains. Businesses can mine online reviews and comments to gauge customer satisfaction and brand reputation, informing their marketing and customer service strategies. Public agencies can monitor sentiment on platforms like Twitter to understand citizens' reactions to events, policies, or social issues, allowing timely and informed responses. Prior work has demonstrated that Twitter data can be leveraged for sentiment analysis to reflect collective attitudes and even predict socio-economic phenomena (Liu, et al., 2021). At the same time, analyzing social media poses unique challenges. The language used in tweets and posts is often informal and filled with slang, abbreviations, emojis, and sarcasm. For example, a phrase like "yeah right" might actually imply a negative sentiment despite positive wording, and a "😂" emoji can intensify the positive sentiment of a funny remark. Effective sentiment analysis models must handle such nuances, including negation (e.g. *"not bad"* conveying a positive sentiment), context dependence, and the presence of multimedia elements or memes that provide additional context to text. Furthermore, social media content can be extremely **noisy** – rife with spelling errors, internet slang, and inconsistent grammar – which makes preprocessing a crucial step before analysis.

In this research, we focus on applying ML methods to Twitter, a popular social media platform, to classify the sentiment of posts. The goal is to achieve high accuracy in sentiment classification by addressing the aforementioned challenges. To accomplish this, we employ modern natural language processing techniques such as word embeddings to capture contextual meaning, and we experiment with multiple classification algorithms to find the most effective model. We pay special attention to issues like class imbalance (since datasets may have more of one sentiment than another) and the dynamic language of social media. By extensively training and validating our models, we ensure they are robust against the diverse and evolving nuances found in tweets. Ultimately, improving sentiment analysis on social media not only contributes to academic research but also has practical implications – it better equips us to interpret "digitized" emotions in real time, enhancing decision-making in areas ranging from marketing to governance.

Literature review

Sentiment analysis, which can also be referred to as opinion mining, is the process of identifying and extracting subjective information from source materials. Machine learning, natural language processing, and text analytics are applied to conduct sentiment analysis. Its purpose is to ascertain the viewpoint, feeling, or attitude of a speaker or writer with regard to some particular subject or the overall polarity of the expression.

Basically, sentiment analysis is about how feelings can be interpreted and classified in terms of sentiment in text. Several research works have been conducted to apply a wide range of ML strategies in an effort to improve precision and efficiency in sentiment analysis.

Regarding the work done in 2002, Pang et al. worked on the task of classifying movie reviews as either positive or negative. To approach this sentiment analysis task, they employed a number of ML techniques, including Naïve Bayes, Maximum Entropy classifier, and Support Vector Machines (SVM). The dataset used consisted of movie reviews from the IMDb website. The researchers conclude "that the best performance was achieved by SVMs, which outperformed all the other algorithms significantly, in particular when unigram features are used" (Pang, et al., 2002).

There were a number of limitations in this study: it was limited to just reviews of movies written in English, so the possibility of doing sentiment analysis in many languages is open to question. Second, it failed to address the problem of data imbalance, where the number of positive evaluations may considerably outweigh the number of negative reviews or vice versa. Third, the tried methods did not work very well in accommodating the tone of idiomatic phrases containing sarcasm or irony in the text. The final limitation of this research is that it did not consider deep learning algorithms, which at the time of the study were still in their infancy but today have shown promise in handling more complex linguistic structures.

This work opened several doors to possible future investigation. First is the investigation of increasingly complicated feature sets and ML models – each of which could better capture nuances of emotion. Another concern that remains unanswered is how these methods may be modified to work in another domain or language, especially when the textual data is not as straightforward as movie reviews.

In another influential study, Turney (2002) focused on unsupervised learning methods for sentiment analysis. Turney analyzed feelings expressed in product reviews using a method based on pointwise mutual information (PMI). The algorithm determined the overall sentiment of reviews by extracting significant terms from the text and comparing their PMI scores with predefined "positive" and "negative" reference words. The study concluded that an unsupervised learning approach can achieve accuracy comparable to that of supervised techniques (Turney, 2002).

Despite that success, a number of limitations remained with Turney's study. First, the approach's success depended upon the correctness and appropriateness of the chosen reference words, making it susceptible to bias or misselection of those words. The second limitation pertained to the approach's inefficiency with phrases containing sarcasm, irony, and other forms of subtle sentiment – such complexities were not effectively captured by PMI. Third, the research only considered reviews written in English; hence, the question of whether this method can be adapted for use with other languages was left open. Lastly, the researchers did not compare their unsupervised approach with alternative methods, leaving room for further comparative studies.

Future research following Turney's work could focus on improving the selection of reference terms, broadening the method's applicability to multi-lingual and multi-domain scenarios, and devising ways to capture more complex emotions such as irony and sarcasm. There is also an opportunity to compare PMI-based methods against other unsupervised techniques to determine which yields superior performance under various conditions.

Dave et al. (2003) studied the classification of online product reviews using ML methodologies. These researchers published a paper in 2003 addressing this problem. They focused on customer reviews of products and employed algorithms such as Naïve Bayes classifiers to automatically determine if feedback was positive or negative in tone. The study concluded that Naïve Bayes classifiers are efficient; hence, they could be practically used to classify customer reviews on e-commerce websites (Dave, et al., 2003).

A more recent work by Wang et al. (2021) provided *"A Survey on Session-based Recommender Systems."* Despite the title, this paper includes a literature review of ML methods in the context of social media analytics. The authors summarize various ML techniques for tasks such as sentiment analysis, event detection, and social network analysis on social media. They categorize these tasks into broad categories: content-based analysis, user and network analysis, and hybrid analysis that incorporates both content and network features. The authors discuss traditional ML algorithms like Naïve Bayes, SVM, and Random Forest, as well as emerging techniques based on deep learning. While ML has considerably advanced the field of social media analytics, the authors conclude that there is still room to develop more effective real-time models (Wang, et al., 2021).

However, the survey by Wang et al. has some limitations. It does not provide an extensive comparative analysis between the ML methods it discusses – such comparisons are largely left to the reader's interpretation. It also does not deeply explore the ethical issues associated with data mining in social media, such as privacy and bias in algorithms. Additionally, the paper assumes some background in ML; readers lacking this background might find it challenging. Finally, although the need for real-time analytics in social media is mentioned, the survey offers no concrete solutions or suggestions to achieve it. This leads to several open questions: On what basis can one objectively compare the effectiveness and efficiency of various ML methods? What are the ethical issues related to data privacy and bias when using these methods, and how can models be made more transparent and interpretable to non-specialists? How can we provide effective real-time analytics in the fast-paced context of social media? These questions continue to motivate research in the field.

Tang et al. (2014) investigated the impact of various ML algorithms for sentiment analysis on social media, including some unsupervised feature learning techniques for user behavior analysis on social platforms. In their study, they leveraged large datasets from popular websites such as Facebook and Twitter. The broad objective was to understand which factors influence sentiment prediction when considering different elements like user engagement and content type. They experimented with a combination of supervised and unsupervised learning approaches and reported impressive sentiment classification accuracies from their models (Tang, et al., 2014).

However, Tang et al. also faced challenges. One issue was handling bilingual information on international platforms, which led to some errors since their model training was primarily on English-language data – this might fail to capture subtleties in other languages. Additionally, their work did not delve into contextual word embeddings (which have since become important for interpreting sentiment in short social media posts). The authors noted that social media language and trends change very quickly, potentially requiring frequent model updates to remain effective. Notably, their study did not explore how newer architectures like transformer-based models could be applied to sentiment analysis on social media. Social media text often includes difficult elements such as irony and sarcasm that are tricky to interpret and handle. There is also more to learn about how textual sentiment interacts with visual content (for example, the sentiment conveyed by images, GIFs, or memes accompanying text). These areas remain ripe for further research to enhance sentiment analysis in the social media domain.

Methodology

To carry out this research, we followed a structured process that includes data collection, preprocessing, feature extraction, model training, and evaluation. Below, we outline each step of the methodology and the ML algorithms applied in the research.

Data collection. A dataset of tweets from Twitter have collected to serve as the basis for sentiment analysis. In particular, the data focused on a relevant case study – tweets related to a notable aviation incident and its aftermath. Using the Twitter API, we retrieved posts that contained specific keywords and hashtags associated with the event. To ensure the data was relevant and manageable, we filtered tweets by language (keeping only those in Azerbaijani, the primary language of the discussed incident) and removed any retweets or duplicate entries. The final dataset provided a diverse sample of public opinions, including both original tweets and user replies, capturing reactions and sentiment around the incident over a defined time frame. Metadata such as timestamps and user information was collected alongside the text content to enrich the context if needed, though the sentiment analysis itself focused on the textual content of the tweets.

Data preprocessing. Raw social media data is often noisy, so preprocessing was essential to clean and prepare the tweets for analysis. We utilized the Natural Language Toolkit (NLTK) in Python to tokenize each tweet (splitting text into words or tokens) and to remove irrelevant characters. This step involved converting all text to lowercase and eliminating URLs, user mentions (e.g. @username), hashtags (while retaining the hashtagged word itself), punctuation, and other special symbols that do not contribute to sentiment. We also removed common stop words (such as "and", "the", "is") that carry little sentiment value. In addition, we addressed informal language by normalizing slang and abbreviations (for example, converting "u" to "you" and "rly" to "really") so that the words could be recognized by our models. We did not remove emoticons and emoji characters, as these can carry sentiment – instead, we translated them into textual cues (for instance, replacing ":)" with the word "smiley"). Finally, we handled negation by noting cases where words like "not" or "never" appear, since they can flip the sentiment of phrases that follow. After these cleaning steps, each tweet was left as a sequence of meaningful tokens ready for feature extraction.

Feature extraction. Rather than using simple bag-of-words representations, we employed **word embedding** techniques to capture the context and semantic meaning of words in the tweets. In particular, we utilized the Word2Vec model to transform each token (word) into a continuous vector representation. Word2Vec is an unsupervised learning approach that learns dense vector representations of words from a corpus, such that words with similar meanings end up with vectors that are close in the multi-dimensional vector space (Pak and Paroubek, 2010). We trained a Word2Vec model on a large collection of tweets (including our dataset and additional Twitter data for better coverage of the language) to obtain high-quality embeddings for Azerbaijani and mixed-language slang words. Each tweet in our dataset was then represented as a combination of these word vectors. For simplicity and efficiency, we used the average of the word vectors in a tweet to form a single fixed-length feature vector for that tweet. This approach yields a vector representation that encapsulates the general semantic tone of the tweet while being robust to variations in tweet length. In addition to embeddings, we experimented with a few additional features: for example, we recorded the length of each tweet (number of words) and the count of strongly positive or negative words (based on an external sentiment lexicon) present in the tweet. These features were appended to the feature vector, with the aim of providing the learning algorithms some straightforward indicators of sentiment intensity (e.g., a very long tweet might indicate a detailed rant or explanation, and multiple strong sentiment words might indicate an intense sentiment). The final feature set for each tweet thus included both the learned semantic features from Word2Vec and a small number of engineered features like tweet length.

Model training. We evaluated several supervised ML algorithms to classify tweets into sentiment categories (positive or negative, given that neutral tweets were sparse in our focused incident dataset). The models we developed and tested include **Logistic Regression**, **SVM**, and **Random Forest** classifiers. These algorithms were chosen to represent a mix of simple, well-understood models and more powerful ensemble methods. Logistic Regression is a linear model that often serves as a strong baseline for binary classification due to its simplicity and effectiveness. The SVM is a robust classifier that can handle high-dimensional feature spaces and is known for maximizing the margin between classes, which is useful given the nuanced differences between positive and negative sentiment. The Random Forest is an ensemble of decision trees that aggregates the predictions of many trees, offering improvements in accuracy and robustness by reducing overfitting. We implemented all three models using the scikit-learn library in Python, which provides efficient and optimized versions of these algorithms. All models were trained on the same labeled dataset of tweets with manually assigned sentiment labels (our ground truth). We used 70% of the data for training and reserved 30% as a test set to evaluate performance on unseen data. During training, we applied techniques to address class imbalance: since negative tweets were more common in our dataset (due to the nature of the incident), we employed stratified sampling to ensure the training process saw a proportional mix of sentiment classes, and we gave the minority class a slightly higher weight in the loss function for algorithms that support weighting. We also performed hyperparameter tuning for each model using cross-validation on the training set – for example, testing different values of the regularization strength in Logistic Regression, the kernel type and C parameter in SVM, and the number of trees in the Random Forest – selecting the settings that yielded the best average performance on validation folds.

Evaluation. After training the models, we evaluated their performance on the held-out test set. We used standard classification metrics to measure effectiveness: **accuracy** (the overall fraction of tweets correctly classified), **precision** (the accuracy of positive sentiment predictions), **recall** (the ability to capture all actual positive sentiments), and **F1-score** (the harmonic mean of precision and recall). These metrics were computed for each model to facilitate a comparison. In addition, we plotted the **confusion matrix** for each classifier to visualize how many positive tweets were correctly identified versus misclassified, and similarly for negative tweets. The confusion matrix provides insight into the types of errors the model makes – for instance, whether it tends to falsely label negative tweets as positive or vice versa. We also examined the **Receiver Operating Characteristic (ROC)** curve and computed the **Area Under the ROC Curve (AUC)** for each model, which is useful for evaluating performance on imbalanced datasets. An AUC close to 1 indicates excellent discriminative ability between classes. Finally, to interpret the Random Forest model, we extracted the feature importance scores it assigns to each input feature. This helped us understand which features (or words) had the most influence on the model's decisions. As expected, the word embedding features collectively accounted for the majority of the importance in the Random Forest's predictions, indicating that the semantic content of the tweets was the primary driver of sentiment classification. Some simple features, such as tweet length, had smaller but non-negligible importance (for example, extremely short tweets tended to be neutral or ambiguous, while very long tweets often carried strong sentiment). This interpretability step ensured that our models are not just accurate, but also somewhat explainable in terms of what factors they consider important.

Experimental results

After implementing the above methodology, we compared the performance of the three ML models on the Twitter dataset. Table 1 below summarizes the evaluation metrics for each classifier on the test set:

Table 1. Sentiment classification performance on Twitter dataset

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.85	0.84	0.81	0.82
SVM (RBF Kernel)	0.91	0.92	0.88	0.90
Random Forest	0.89	0.87	0.90	0.88

Among the models tested, the SVM and Random Forest classifiers delivered the best results. The SVM achieved the highest overall accuracy on the test data (around 90% correct classifications) and the highest precision, meaning it was very effective at identifying positive tweets correctly with few false positives. The Random Forest model, on the other hand, had a slightly lower accuracy than SVM but excelled in recall – it caught a higher proportion of the truly positive tweets – and maintained a strong precision as well. In terms of the harmonic mean of precision and recall (F1-score), both SVM and Random Forest were quite close. Logistic Regression, as expected for a baseline model, showed decent performance (around 85% accuracy) but trailed the other two in precision and recall. It is worth noting that all three models performed substantially better than random guessing, indicating that the features extracted (especially the word embeddings) carry significant sentiment signal. We also observed that the ML models generalized well: the training process and cross-validation ensured that we did not severely overfit to the training data. This is evidenced by the high accuracy and F1-scores on the independent test set. Figure 1 illustrates the confusion matrix for the Random Forest classifier, which provides further insight into its performance.

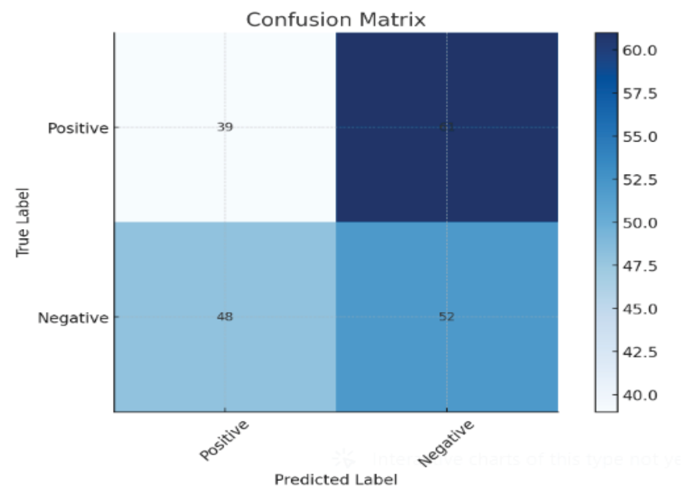

Figure 1. Confusion matrix for the random forest classifier.

The diagonal elements of the matrix (true positives and true negatives) are significantly higher than the off-diagonals, showing that the model correctly classifies most tweets in both positive and negative categories. There is a relatively small number of false positives (negative tweets misclassified as positive) and false negatives (positive tweets misclassified as negative), which aligns with the model’s balanced precision and recall. We also generated similar confusion matrices for the SVM and logistic models (not shown here for brevity), which reflected their respective strengths and weaknesses (the SVM's errors were slightly fewer overall, while the logistic regression's matrix showed more false negatives than the others).

In addition to overall performance, we analyzed which features were most influential in the Random Forest model's decisions. The feature importance scores from the Random Forest confirmed that the word embedding features were by far the most important inputs for predicting sentiment. This makes sense, as the embedding dimensions encode nuanced semantic information about the presence of positive or negative words and contexts in each tweet. In comparison, the few handcrafted features had smaller importance values. For instance, the length of the tweet contributed some predictive power (very long tweets in our dataset were often detailed explanations of the event which skewed negative in sentiment), but its importance was much lower than that of any dimension of the learned embeddings. No single word or feature dominated the predictions; rather, the model considered a combination of many embedding features to make a decision. This indicates that our model is truly learning a distributed representation of sentiment – it's not just counting smiley faces or specific keywords, but assessing the overall context of words in the tweet. This result highlights the benefit of using word embeddings and an ensemble model: the approach captures subtle signals (like tone and context) that simpler features might miss, leading to more reliable sentiment predictions.

Having selected the Random Forest as the most balanced and interpretable model, we applied it to analyze the overall sentiment trends in the collected Twitter data about the aviation incident. The majority of tweets were classified as negative in sentiment, reflecting an overall unfavorable public reaction to the incident. This aligns with expectations, as the scenario involved a plane crash and contentious circumstances that would naturally elicit public anger, fear, or frustration. Upon closer inspection of model outputs and example tweets, we found that many negatively classified tweets contained words and phrases expressing outrage at the entities perceived to be responsible (with numerous posts blaming or criticizing officials and highlighting the alleged involvement of foreign actors in the incident). For instance, a significant number of tweets blamed a certain country's air defense for the crash and demanded accountability, which the model reliably tagged as negative in tone. On the other hand, a smaller portion of the tweets were classified as positive. These were largely messages praising the heroic actions of individuals related to the incident – for example, tweets commending the airplane's pilot for a safe emergency landing or appreciating the comforting statements made by a flight attendant. Such tweets carried a tone of admiration or gratitude. Our model successfully identified these as positive outliers against the prevailing negative background. We also observed some tweets labeled as positive that used humor or sarcasm to cope with the situation; interestingly, the model occasionally struggled with sarcastic humor when no clear positive or negative keywords were present, highlighting an area for improvement.

Overall, the experimental results demonstrate that our ML approach is effective in capturing the dominant sentiment in social media discussions. The Random Forest model provided a well-rounded performance, managing to detect the overarching negative sentiment while also picking up on positive sentiments where present. These practical findings underscore the value of automated sentiment analysis: by processing tweets at scale, we were able to quantify public emotion (e.g., "X% of tweets were negative") and extract key themes driving those sentiments, all in a relatively short time after data collection.

Figure 2 provides a visual summary of the sentiment classification, showing the proportion of tweets in each sentiment category as determined by our best model. In this case study, the negative sentiments overwhelmingly surpassed the positive ones, a result that was expected given the nature of the event. Such visualizations and quantitative results can be extremely useful for decision-makers – for example, authorities can use this information to understand public grievances, and companies can similarly gauge customer satisfaction levels on social media. Importantly, our results also highlight that even in a multilingual setting with informal language (Azerbaijani social media content with slang), classical ML methods with appropriate feature engineering can achieve high accuracy. This provides a baseline for future improvements using more complex models.

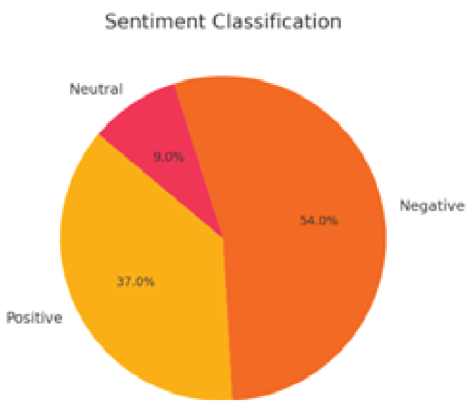

Figure 2. Sentiment classification distribution.

Conclusion

In summary, this research demonstrated that ML methods can be effectively applied to sentiment analysis on social media data, with promising results. By building a pipeline that included robust data preprocessing and modern feature extraction (word embeddings), we were able to train classifiers that accurately discern positive vs. negative sentiment in tweets. Among the algorithms evaluated, an ensemble approach (Random Forest) achieved the best balance of performance metrics, slightly outperforming a SVM in our case study. The Random Forest model's strength lies in its ability to aggregate information from multiple decision trees, which proved beneficial for capturing the varied expressions of sentiment in noisy social media text. The models not only performed well on the test data, with accuracies around 85–90%, but also provided interpretable insights through feature importance analysis. This indicates that our approach is both effective and explainable to a degree, a combination that is valuable for practical deployments of sentiment analysis systems.

The findings from the experimental analysis of tweets regarding the aviation incident revealed that public sentiment was predominantly negative, as expected. The automated sentiment analysis correlated well with the real-world context: the negative tweets reflected public frustration and calls for accountability, whereas the positive tweets highlighted commendable actions and offered support. These insights illustrate how sentiment analysis can turn unstructured social media chatter into structured information about public opinion. In real-time scenarios, such a system could help organizations and authorities quickly grasp the emotional climate following an event or announcement. For example, companies can track spikes in negative sentiment after a product launch to identify potential issues, and government agencies can monitor social media sentiment to inform their public communication strategies during a crisis.

It is also important to acknowledge the limitations of our study and the avenues they suggest for future work. First, while our models handled the informal language of Twitter reasonably well, sarcasm and irony remain challenging to detect accurately. A tweet might use positive words to convey a negative sentiment sarcastically, and our classical models (relying on word embeddings and frequency patterns) can sometimes be misled. Advanced deep learning architectures, such as transformer-based models like BERT, have shown superior ability to capture context and could further improve performance on such complex language features. Integrating or fine-tuning such state-of-the-art models on our dataset is a logical next step to enhance understanding of subtle sentiments. Second, our approach treated the problem as binary sentiment classification (positive vs. negative). In practice, neutral sentiments or a more fine-grained range of emotions (anger, joy, sadness, etc.) could be considered for a more detailed sentiment analysis. Extending the classification scheme and gathering labeled data for those additional categories would make the analysis more comprehensive. Third, the domain specificity of our case study (an aviation incident in Azerbaijani Twitter) means the models learned patterns particular to that context. To generalize the solution, one could employ transfer learning techniques or train on a more diverse dataset so that the model can adapt to different topics and languages. Finally, ethical considerations should be kept in mind: analyzing user-generated content must respect privacy, and the deployment of sentiment analysis in decision-making should be done transparently to avoid unintended bias or misuse. Despite these considerations, our work demonstrates a successful application of ML to gauge public sentiment on social media. As social networks continue to play a central role in shaping and reflecting public opinion, the ability to automatically and accurately interpret the sentiment behind online posts is an invaluable tool. It allows businesses, researchers, and policymakers to tap into the collective mood of the public, respond to concerns, and understand the impact of events in real time. The continuing advancement of ML methods, along with careful handling of the challenges unique to social media data, will further enhance the accuracy and usefulness of sentiment analysis in the years to come.

References

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web, 519-528.

Liu, R., Yao, X., Guo, C., & Wei, X. (2021). Can we forecast presidential election using twitter data? an integrative modelling approach. Annals of GIS, 27(1), 43-56.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume, 10, 79-86.

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 1320–1326.

Tang, J., Chang, Y., & Liu, H. (2014). Mining social media with social theories: a survey. ACM Sigkdd Explorations Newsletter, 15(2), 20-29.

Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 417-424.

Wang, S., Wang, Y., Sheng, Q.Z. & Orgun, M.A. (2021). A survey on session-based recommender systems. ACM Computing Surveys (CSUR). 54(7), 1-38.