Comparison of Machine Learning Algorithms on Sentiment Analysis of Tweets

Student paper

Vedad Fejzagić
Master Studies
International Burch University, Faculty of Engineering, Natural and Medical Sciences
Sarajevo, Bosnia and Herzegovina

Abstract—**In today's era, using internet platforms to convey information to others, whether family, friends or strangers has become the norm. One of the leading social platforms in that regard is "Twitter" (now "X"). Sentiment analysis is considered a classification problem of determining whether an input is positive or negative. The aim of this research was to show to what extent, for the given subset of data, certain ML models, outperform others depending on the choice of preprocessing steps within the sentiment analysis domain. This paper presented results on analyzing sentiment of tweets using different Machine Learning (ML) methods.**

Keywords-sentiment analysis; machine learning; preprocessing; natural language processing; twitter

I. INTRODUCTION

Twitter is a social media platform that has gained enormous popularity over the last decade. It has tens of millions active users worldwide. Text that "Twitter" users write is often referred to as a "tweet" and has a certain limit of maximum number of characters. This platform has become an enormous source of user-generated data, that are condensed within the limitations of the tweet.

Sentiment Analysis is a Natural Language Processing (NLP) and Information Extraction task that aims to obtain writer's feelings expressed in positive or negative comments, questions and requests, by analyzing a large number of documents [1]. It aims to determine the tonality of a written document within a certain topic of interest. Sentiment analysis has become an important factor in the online world. It helps brands measure customer satisfaction for a given product, or their brand overall. In politics, it is used to understand public opinion on certain events and decisions. Sentiment analysis can also be used as one of the elements in the pipeline for studying trends and the change in behavior of society online. Table I shows example tweets that have positive and negative sentiment assigned to them.

Machine learning is the sub-domain of artificial intelligence, which gives ability to a computer system to perform a certain task without being programmed exhaustively [2]. Machine learning algorithms (models) can detect patterns of data and use that to infer conclusions on newly seen information. Nowadays, there are numerous viable models that can be applied to a particular issue, and it has become increasingly important to measure and compare the accuracy of each viable model.

TABLE I. TWEETS WITH ASSIGNED SENTIMENT

Sentiment	Tweet
Positive	I'm so excited to watch this #movie
Negative	it makes me sad when @jackson is sick

One way to automate sentiment measuring for online platforms is using Machine Learning models trained specifically on user data from the chosen platform. Research that was presented in this paper models the sentiment analysis problem as a binary classification problem, meaning that sentiment from user input can be either classified as positive or negative. Additionally, this paper presented the accuracy of several machine learning models solving the classification problem based on the choice of different data processing steps.

II. LITERATURE REVIEW

The landscape of sentiment analysis within the machine learning field has witnessed a surge of research endeavors aiming to detect and categorize sentiments expressed in diverse textual data.

Pang et al. [3] used machine learning methods (Naive Bayes, Maximum Entropy and Support Vector Machines) to perform sentiment analysis on movie reviews. They selected reviews where the rating was provided either with stars or a numerical value from the Internet Movie Database (IMDb). The authors mentioned that they had extracted and converted the reviews into one of the three categories: positive, negative or neutral. In terms of preprocessing, the authors used features based on unigrams and bigrams and used three-fold cross-validation and reported that Support Vector Machines tend to have the best performance in relation to Naive Bayes and Maximum Entropy, but the differences are not very large. Moreover, authors report that the unigram presence had been the most effective in terms of model performance.

Go et al. [4] used several different machine learning classifiers and feature extraction techniques. They used the

following classifiers: Naive Bayes, Maximum Entropy and Support Vector Machines and feature extraction methods such as unigrams and bigrams. The authors mentioned that they stripped emoticons from training data since they negatively impacted Support Vector Machines and Maximum Entropy classifiers (but had little effect on Naive Bayes). They also pointed out that emoticons can be misleading at defining the correct sentiment. In their research, they changed usernames from tweets (which is straightforward to do as Twitter usernames always start with '@' symbol) into 'USERNAME' class token. Similar is done with URLs which were replaced by the 'URL' token. Interestingly, the authors also mention words that have repeated letters (such as 'huuungry') which they always normalized to words that have two repeated letters (ex. 'huungry'). Using Unigram and Bigram feature extraction led to an increase in accuracy for both Naive Bayes and Max Entropy, however it led to a decline in accuracy for Support Vector Machines. The authors also state that using only Bigrams is not useful and that it is better to combine Unigrams and Bigrams.

Medhat et al. [5] pointed out that the sentiment analysis task is a classification problem and that firstly features should be extracted. They note that some of the features are terms presence and frequency (i.e. n-grams with their frequency counts), part of speech (i.e. finding adjectives), opinion words and phrases (i.e. words often used to express certain opinions, such as "like"), and negations (i.e. terms such as "not good" is same as "bad"). Additionally, the authors note that one approach to tackle this classification problem is using machine learning. They mention several supervised learning methods, such as Naive Bayes classifier, Support Vector Machines Classifiers (SVM), Neural Networks, Decision Tree classifiers and others, as potential candidates. Moreover, in their research they point out ML algorithms are usually used for this type of problem as they can use the train data that can be domain-specific, hence the models themselves become attuned to the domain.

Kouloumpis et al. [6] researched the utility of linguistic features for sentiment analysis of Twitter messages. They used three different sets of data. For training they used the hash-tagged dataset (HASH) from the Edinburgh Twitter corpus and extracted data that contained emoticons from the dataset created by [4], which they referred to as EMOT. For evaluation they used a dataset produced by the iSieve Corporation (ISIEVE). The authors mention several ideas they did regarding data preprocessing. Namely, they replaced abbreviations with their full meaning (e.g., BRB was converted to "be right back"). Next, they had applied lowercasing on words and had replaced repeating characters within words by a single character. Finally, the authors mention replacing special Twitter characters such as hashtags, usernames and URLs with placeholders that indicate their type (what authors of previously analyzed research did). In terms of features, the authors mention using n-gram features, lexicon features, part-of-speech features and micro-blogging features. The authors had used 10% of the HASH dataset for validation, and the remainder had been used for training the AdaBoost.MH model with 500 rounds of boosting. The authors repeated this process ten times and had taken the average performance of the models. The authors did not include the [4] data in the initial experiment. Those data were used as an expansion to HASH data to improve the sentiment classification. They mention that 19,000 messages from [4] that had been divided equally between positive and negative sentiments are randomly selected and appended to the HASH data. The authors report their findings based on the results from the validation set first. Namely, authors found that adding data from the dataset [4] did lead to improvements in accuracy when all features are used. Next, the authors evaluated the models using the data produced by the iSieve Corporation (ISIEVE). They reported that the best performance comes from using the n-grams together with lexicon and microblogging features, while including part-of-speech features contributes to a drop in model performance.

This collective body of research underscores the versatile nature of sentiment analysis and the diverse methodologies employed to unravel the complicated network of sentiments embedded in textual data.

III. MATERIALS AND METHODS

A. The data

Source data that are used for the purpose of this research contains a collection of tweets stored in a csv format created by [4]. The csv data contains the following information for each tweet: polarity, id, date, query, user and text.

The polarity field represents whether the tweet is negative, positive or neutral in its sentiment, represented by integers 0, 4 and 2 respectively. The id is a unique id of the tweet within the dataset. The date represents the date of the creation of the tweet. The query field represents whether the tweet has been collected with some specific keyword (otherwise NO_QUERY value is assigned). And finally, the text field represents the content of the tweet itself. For this research, polarity, text and id fields are of most interest, while others are not needed. Table II shows an example of three tweets found within the data when the unneeded fields are ignored.

B. Data preprocessing

Since the data contains raw tweets, several preprocessing steps were applied onto the text of each tweet. This was done to reduce noise and dimensionality of the data during training of machine learning models. The choice of some preprocessing steps was also driven by findings from previously mentioned researchers. Preprocessing of the data consists of several steps (Table III):

1) Placeholder transformation: converts URLs, hashtags and usernames into placeholder tokens: "URL", "HASHTAG" and "USERNAME" respectively (e.g. "https://www.google.com/" was transformed into "URL").

2) Tokenization: converts the tweet text into a list of tokens. Punctuations and other symbols were kept as their own separate tokens (e.g. "just landed!" was transformed into the list: "just", "landed", "!").

3) Token normalization: applies transformation functions onto each token. Firstly, English contractions and internet abbreviations were expanded (e.g. "BRB" was converted into "be right back", and "it's" was converted into two tokens "it"

and “is”). Secondly, all but one repeating neighboring characters were removed (e.g. “hooouse” has been converted into “house”). Finally, English punctuations and English stop words were removed from the token list, and all records containing empty tokens were ignored altogether.

All tokens, except the placeholders from 1), were transformed to their lowercase representation. According to findings by [4], emoticons can be misleading, hence there was no special handling for emoticons and, as per the nature of preprocessing steps, they are removed. Additionally, the Term Frequency-Inverse Document Frequency (TF-IDF) was applied onto the preprocessing results to be used as a feature for machine learning models.

The outputs of data preprocessing are two sets of data. One output dataset (A) has all preprocessing logic applied onto it, while the other output dataset (B) does not have its usernames, hashtags and URLs transformed into placeholders. Each output dataset was then split into five subsets (i.e. 10 subsets in total) of six thousand records (i.e. thirty thousand records in total representing A, and same for set B) ignoring records that have zero tokens. Also, each subset contains records with identical ids, and contains exactly 50% of tweets labeled as having polarity 0, and 50% tweets labeled as having polarity of 4. Additionally, records that have a polarity of 2 are ignored since their total representation in the source dataset was only 139 records.

C. Training the Machine Learning models

In this research, several machine learning models were used and had their accuracies compared based on the preprocessed data. Namely: Naive Bayes, Support Vector Machines (SVM), Random Forest, XGBoost, Decision Trees and Feed Forward Neural Network (NN) were trained.

1) Naive Bayes

Naive Bayes model uses the Bayes Theorem to predict the probability that a given set of features belong to a particular label [5]. It is the simplest and most used classifier.

2) Support Vector Machines (SVM)

The goal of an SVM classifier is to calculate a hyperplane that separates one set of data from another. This leads to a creation of a nonlinear decision boundary [7]. It’s a “nonparametric” model meaning that the parameters for SVM are not predefined, and their number depends on the training data, hence we say they are data-driven [8].

3) Random Forest

Random forest is a method of classification that utilizes decision trees such that output of each tree is aggregated into the one final result [9]. It is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [10]. It is one of the ensembles of decision trees methods.

TABLE II. REPRESENTATIVES FROM SOURCE DATASET

id	polarity	text
1467833799	0	I think my arms are sore from tennis
137	2	Just landed at San Francisco
389	4	I loved night at the museum!!!

TABLE III. DATA PREPROCESSING STEPS

preprocessing step	dataset	example text
raw input data	A & B	more of them? #earthquake BRB
placeholder transformation	A	more of them? HASHTAG BRB
	B	more of them? #earthquake BRB
tokenization	A	more, of, them, ?, HASHTAG, BRB
	B	‘more’, ‘of’, ‘them’, ‘?’, ‘#’, ‘earthquake’, ‘BRB’
Normalization	A	‘HASHTAG’, ‘right’, ‘back’
	B	‘earthquake’, ‘right’, ‘back’

4) Decision Trees

Other than to classify correctly as much of the training data as possible, and to generalize beyond the training data, the goal of the decision tree is to be incremental and to have as simple structure as possible [11]. The decision-tree classifiers SLIQ and SPRINT have been shown to achieve good accuracy [12].

5) XGBoost

XGBoost (Extreme Gradient Boosting) utilizes gradient boosting on decision trees and has recently become a method of choice for applied machine learning and Kaggle competitions. Chen et al. [13] note that the XGBoost model was used by 17 solutions that won challenges on the Kaggle platform.

6) Feed Forward Neural Network

The neural network consists of a set of neurons that are connected to each other. The multi-layered feed-forward neural networks are the most popular type of neural network. The neurons within said network are ordered into layers where the first layer is called the input layer, and the last layer is called the output layer [14].

Each model was trained on 90% of records from each subset of data, and then tested on the remaining 10%. Additionally, the training was reinforced by utilizing 5-fold cross validation with hyperparameter tuning. The training was done on a machine with 16GB RAM and a 6-core CPU with 3.7GHz frequency.

IV. RESULTS

For subsets of both A and B, the accuracy of each model for each of the subsets was recorded, and then average value was calculated. For each model, the average accuracies of five subsets of (A) and (B) were plotted in a bar chart (Fig. 1). Table IV contains the exact numbers.

D. Use case (A)

For subsets of dataset (A), the Naive Bayes model has an average accuracy of 74% followed by Support Vector Machines

(73.6%), Random Forest (73%) and XGBoost (72.6%). The Decision Trees model seems to be underperforming with an average accuracy of 62.2%, followed by Feed Forward Neural Network with an average accuracy of 69.8%. For the use case of (A), the Naive Bayes model seems to perform the best.

E. Use case (B)

For subsets of dataset (B), the findings seem to be similar. The Naive Bayes model performed the best in this use case as well, with 73.8% average accuracy, followed by Support Vector Machines at 72.6%, and Random Forest and XGBoost both at 72.2% average accuracy. Feed Forward Neural Network, being at 70.4% average accuracy seems to perform better in this use case, while the accuracy of Decision Trees model in this case is the worst overall at 60.4%.

Overall, the consistent performance of the Naive Bayes model across both datasets suggests its robustness and reliability in capturing patterns within the given data. Additionally, the varying performance of other models between the two datasets highlights the dataset-specific nuances influencing model accuracy. These findings provide valuable insights for model selection and optimization tailored to specific use cases and datasets.

V. CONCLUSION

This study showed that for the problem of sentiment analysis of tweets, the Naive Bayes seemed to perform the best and seemed to be the clear model of choice for this type of analysis. It was followed by SVM, Random Forest, XGBoost and Forward Neural Network, while the Decision Trees model was proving not to be the best choice for this type of analysis.

TABLE IV. AVERAGE ACCURACY PER MODEL BASED ON THE DATASET

	(A)	**(B)**
SVM	0.736	0.726
Naive Bayes	0.74	0.738
XGBoost	0.726	0.722
Random Forest	0.73	0.722
Decision Trees	0.622	0.604
Feed Forward NN	0.698	0.704
Total	**4.252**	**4.216**

To conclude, preprocessing steps that were applied onto (A) and missing from (B) seemed to be almost negligible when it comes to the performance of Naive Bayes and XGBoost models, while the importance increases when using other models (especially SVM and Decision Trees). Additionally, when using a Feed Forward Neural Network model, adding the placeholders (as done in dataset A) was detrimental to the overall accuracy of the model.

F. Limitations and potential improvements

There are several ways in which this research could be improved upon. One of the ways is using more data for model training. As mentioned before, this study used six thousand labeled records to train and test the previously mentioned machine learning models. As the input to the models were

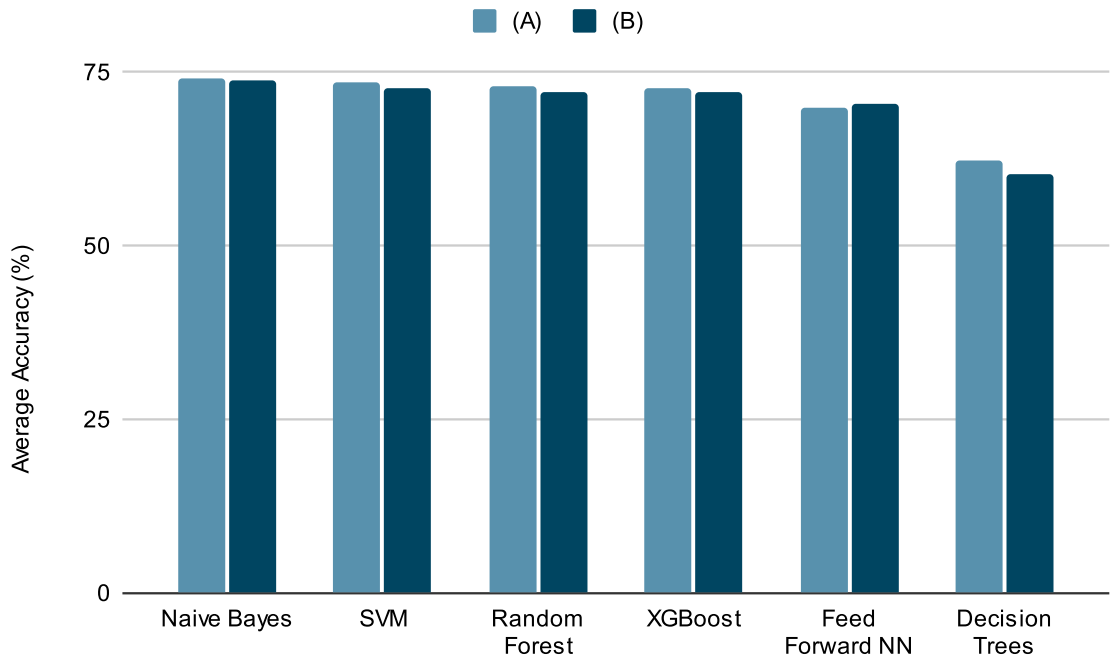

Figure 1. Average Accuracy Comparison of Machine Learning Models per Dataset

features based on user text from the internet, better accuracy can be achieved with more data that are correctly labeled.

Upon analyzing the data used in this study, it can be shown that certain records have their polarity fields annotated differently from actual sentiment they represent. Some tweets had been labeled as having positive sentiment, even though in reality they were negative, and vice versa. Furthermore, the number of records labeled as having neutral sentiment could be improved such that future studies can focus more on tweets with that dimensionality, as opposed to only focusing on positive and negative sentiments.

Finally, further research can be done with utilization of more features and potentially more preprocessing steps. The argument can be made as to which feature or preprocessing step played a significant part in the performance of which machine learning model in this field of study.

REFERENCES

[1] S. Mukherjee, "Sentiment analysis.," *ML. NET Revealed: Simple Tools for Applying Machine Learning to Your Applications*, pp. 113--127, 2021.

[2] A. Mishra, "Machine Learning Classification Models for Detection of the Fracture Location in Dissimilar Friction Stir Welded Joint," *Applied Engineering Letters : Journal of Engineering and Applied Sciences*, vol. 5, pp. 87--93, 2020.

[3] B. Pang, L. Lillian and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing ({EMNLP} 2002)*, Association for Computational Linguistics, 2002, pp. 79--86.

[4] W. Medhat, A. Hassan and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, pp. 1093--1113, 2014.

[5] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, 2009

[6] E. Kouloumpis, T. Wilson and J. Moore, "Twitter sentiment analysis: The good the bad and the omg," in *Proceedings of the international AAAI conference on web and social media*, 2011, pp. 538--541.

[7] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, pp. 18--28, 1998.

[8] V. Kecman, "Support vector machines--an introduction," in *Support vector machines: theory and applications*, Springer, 2005, pp. 1--47.

[9] L. Breiman, "Random Forests," *Machine learning*, pp. 5--32, 2001

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss and V. Dubourg, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825--2830, 2011

[11] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, pp. 660--674, 1991.

[12] K. Alsabti, S. Ranka and V. Singh, *CLOUDS: A decision tree classifier for large datasets*, 1998.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785--794.

[14] D. Svozil, V. Kvasnicka and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and intelligent laboratory systems*, vol. 39, pp. 43--62, 1997.