Machine Learning - 2023 - Mini Project II

Aicha Moussaid
Aicha.moussaid@abo.fi
EDISS Master's Programme - Åbo Akademi University, Turku, Finland

Introduction

The growing popularity of social media platforms like Twitter has led to increased interest from businesses, individuals, and governments in comprehending how users feel about their products, services, policies, or events. Sentiment analysis, a branch of natural language processing, is dedicated to uncovering subjective information from text, including emotions, attitudes, and opinions. Through analyzing substantial quantities of Twitter data, we can acquire significant insights into people's viewpoints and sentiments on a diverse range of topics.

In this report, we will explore different approaches to sentiment analysis of Twitter data using machine learning algorithms. We will start by collecting and preprocessing the data, including text cleaning, vectorizing and data transformation. We will then compare the performance of several machine learning models, including Bernoulli Naïve Bayes, Linear SVC, Logistic Regression, and Random Forest, in terms of performance and speed.

We will evaluate the models using a popular standard dataset for sentiment analysis, namely the Sentiment140 dataset. We will also discuss some of the challenges of sentiment analysis, such as sarcasm, irony, ambiguity, and context dependence.

In essence, the primary objective of this report is to offer an outline of the latest advancements in utilizing machine learning techniques for analyzing sentiment in Twitter data.

Data Processing

The Sentiment140 dataset is a prominent dataset employed in sentiment analysis research. It is comprised of 1.6 million tweets that are categorized with either positive, negative, or neutral sentiments. This dataset was produced by researchers at Stanford University and is widely utilized for training and assessing various machine learning models used in sentiment analysis. Its significant advantage is its vast size, which enables more precise model training and testing. Additionally, its straightforwardness is beneficial since every tweet is tagged with a single sentiment category, making it user-friendly for both novices and experts in the field.

The version of this dataset provided for this mini project has two columns, sentiment_label, holding values of 0 for negative and 4 for positive, and tweet_text, holding the tweet text as it is posted. The number of rows provided is also tweaked and amount to 160,000 rows. The data processing step is started by first checking for any null values, then decoding the labels given (0 and 4) to sentiments (positive and negative).

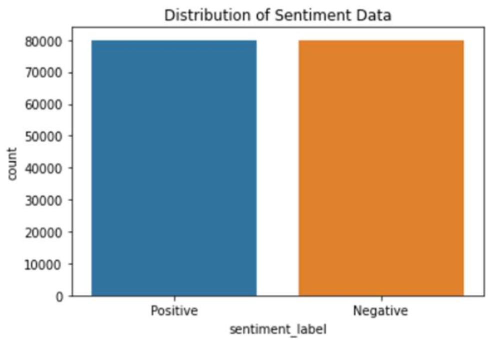
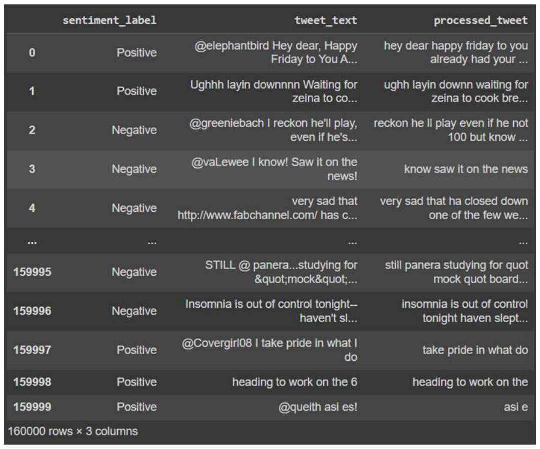


Figure 1. Distribution of Sentiment Data

A simple data visualization of the balance of this dataset is performed, and after ensuring that our dataset is balanced, as shown in Figure 1., we move on to tweet text preprocessing.

In Natural Language Processing (NLP), Text Preprocessing has long been regarded as a crucial step. Its purpose is to modify text into a more comprehensible format, making it easier for machine learning algorithms to operate efficiently. The preprocessing steps taken are: lower casing, where each text is converted to lowercase, replacing URLs, Emojis, Usernames, removing non-alphabets, consecutive letters, short words less than 2 characters, stop words, and finally, lemmatizing where a word is converted to its base (e.g. “Great”, “Good”). After 34 seconds, we obtain the new dataset shown in Figure 2.



	sentiment_label	tweet_text	processed_tweet
0	Positive	@elephantbird Hey dear, Happy Friday to You A...	hey dear happy friday to you already had your ...
1	Positive	Ughhh layin downnnn Waiting for zeina to co...	ughh layin downn waiting for zeina to cook bre...
2	Negative	@greeniebach I reckon he'll play, even if he's...	reckon he ll play even if he not 100 but know ...
3	Negative	@vaLewee I know! Saw it on the news!	know saw it on the news
4	Negative	very sad that http://www.fabchannel.com/ has c...	very sad that ha closed down one of the few we...
...	...	...	...
159995	Negative	STILL @ panera...studying for &quot;mock&quot;...	still panera studying for quot mock quot board...
159996	Negative	Insomnia is out of control tonight--haven't sl...	insomnia is out of control tonight haven slept...
159997	Positive	@Covergirl08 I take pride in what I do	take pride in what do
159998	Positive	heading to work on the 6	heading to work on the
159999	Positive	@queith asi es!	asi e

160000 rows × 3 columns

Figure 2. Dataset after Text Preprocessing

To get further understanding of the data, we generate Word Clouds, shown in Figure 3, for positive and negative tweets from our dataset and see word occurrence. It can be seen that words evoking positive sentiment are, for example, *love*, *thank*, *good*, which generally are used in positive contexts, meanwhile words like *work*, *miss*, *sorry*, and *hate*, are usually used in negative contexts. This does not mean we cannot find words in both sentiments like *love* which can be used in both contexts.

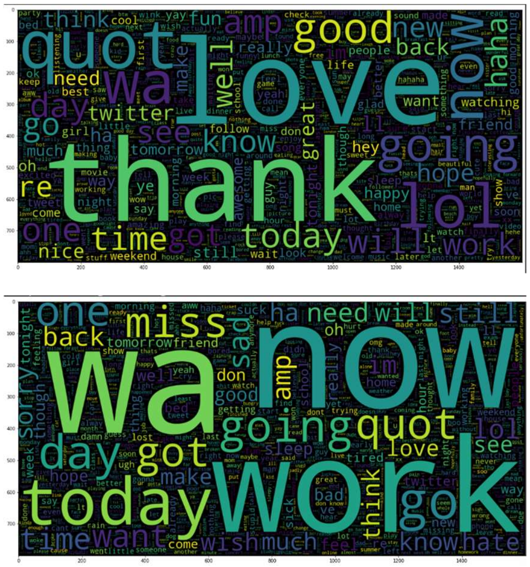


Figure 3. Word Clouds of Tweet Sentiments

After this step, the preprocessed data is split in 95% training data and 5% testing data, since we need our models to learn the best they can on as much data as possible. We perform vectorization using TF-IDF vectorization.

TF-IDF is a technique that assists in determining the significance of words within a document or dataset. To elaborate, let's consider an example: suppose a dataset comprises essays on the topic "My House." Among the words that appear frequently, "a" is one such word. However, other words like "home," "house," and "rooms" occur less frequently and hence carry more weightage in terms of conveying information compared to the word "a." This is the basic idea behind TF-IDF. The TF-IDF Vectorizer converts raw documents into a TF-IDF feature matrix. Typically, the Vectorizer is trained solely on the X_train dataset. We then transform the X_train and X_test sets into matrix of TF-IDF features by using our trained TF-IDF Vectorizer.

This sums all the steps needed to be performed on the dataset before feeding it to our models.

Modelling

In this section, four machine learning models are used and evaluated to observe their performance on the processed data for our sentiment analysis problem. Bernoulli Naïve Bayes, Linear Support Vector Classification, Logistic Regression, and Random Forest are the final models we opted for. Given that our dataset is balanced, meaning it has an equal number of positive and negative predictions, we have opted to use Accuracy as the evaluation metric. Additionally, we are constructing a Confusion Matrix to gain insights into how our model is performing for both types of classification. We also added a speed evaluation, or in other words, how much time each model takes, and we use it as an additional evaluation metric to compare the performance of the chosen models. Figure 4 sums the resulting confusion matrix for all 4 models, but a closer inspection is needed still to further understand how they performed.

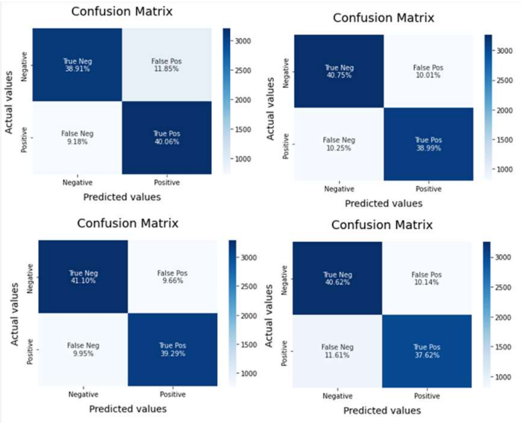


Figure 4. Confusion Matrix of ML models used.

The models performed as the following: 79% accuracy for BernoulliNB, 80% for LinearSVC, 80% for Logistic Regression, and 78% for Random Forest. As for the time metric, they respectively took 2 seconds, 4, seconds, 37 seconds, and 6170 seconds (1h 43min).

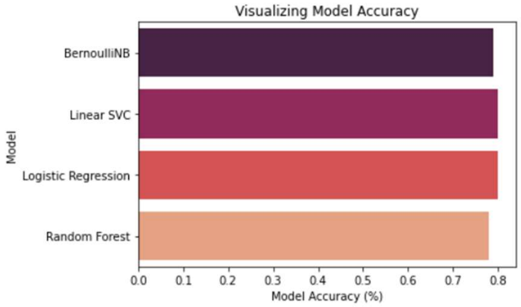


Figure 5. Model Accuracy Visualization

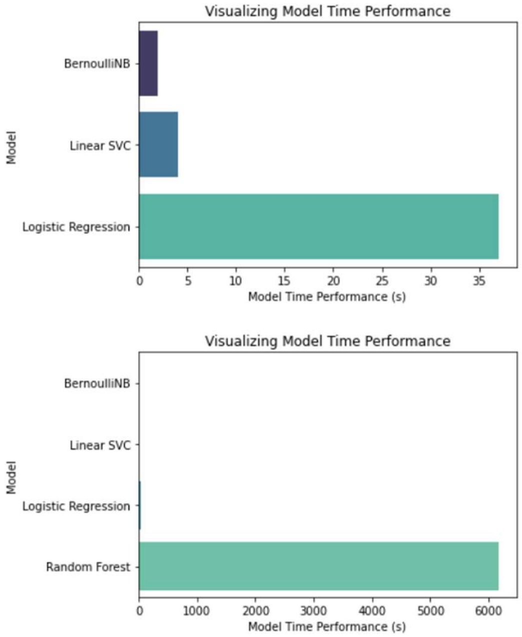


Figure 6. Model Time Performance

From Figure 5 and 6, we can see how the accuracy was close for all of them ranging from 78% to 80%, which makes it difficult to decide which model would be best for such a problem. This is why the Time performance metric was added in this study, since it clearly shows the "right" from the "wrong" model to opt for. The ranking system could go as the following, where LinearSVC performed the best in only 4 seconds, followed closely by BernoulliNB, which of course has 1% accuracy less, but is considered the fastest. Random Forest performed poorly in terms of both, Time and Accuracy, which could be due to the number of trees taken (100).

This led us to tweaking the number of trees to see how the performance would improve. We went with 20 trees, and it gave a performance of 77%, decreasing by only 1%, for a duration of 21 minutes, as shown in Figure 6.1, which could be considered as a fair tradeoff. Random Forests tend to perform well in high-dimensional datasets, but in sentiment analysis, the number of features (words) is typically limited, making it difficult for Random Forests to capture the nuanced meanings and associations between words.

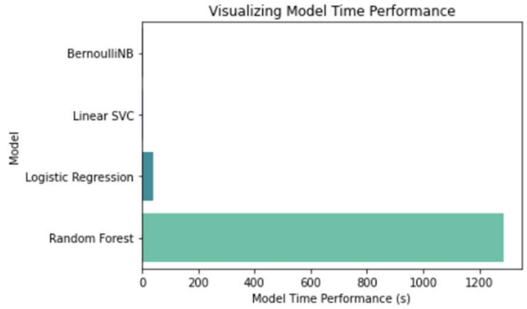
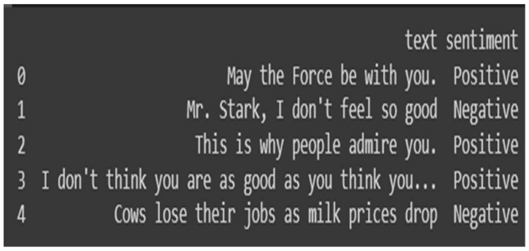


Figure 6.1. Updated Model Time Performance.

The last step of this project was to save the models and assess the performance on new sentences. We picked a few sentences and tried to gauge the classification of these sentences using one of the models, as shown in Figure 7.



```
                                      text sentiment
0                     May the Force be with you.  Positive
1                   Mr. Stark, I don't feel so good  Negative
2                     This is why people admire you.  Positive
3  I don't think you are as good as you think you...  Positive
4             Cows lose their jobs as milk prices drop  Negative
```

Figure 7. Text-Sentiment Table using best performing model.

The sentiment analysis model is more accurate when given straightforward sentences that harbor more common, clearer sentiment words like “hate”, “love”, “like”, “don’t like”. Although, when using sentences with sarcasm or context-dependent words in another sentence, like the 4th and the 5th sentences in the table, we can see that the model misclassifies the sentiment.

This forms a limitation to our models when trying to interpret the meaning and further highlights how machine learning models can only learn the most common combos of the words, given different contexts in order for them to be reliable enough.

Conclusion

To summarize, sentiment analysis of Twitter data through machine learning is a quickly developing area with tremendous potential for diverse applications. This report provides an overview of the state-of-the-art in sentiment analysis of Twitter data, outlines the required preprocessing steps for text data, and delivers a comprehensive analysis of various machine learning models' performance using the widely used Sentiment140 dataset.

Our findings emphasize the importance of selecting the appropriate model and preprocessing techniques, which can significantly impact sentiment analysis accuracy. Furthermore, we've explored the time taken as a performance metric to differentiate between the models, as well as highlighted the difficulties and restrictions of sentiment analysis, including the challenges of handling sarcasm, and context-dependent sentiment.

This report provides a beneficial resource for anyone seeking insights into people's attitudes and opinions towards various topics through Twitter data analysis, including researchers, businesses, and individuals. Although, more research is required to produce more robust and accurate models that can handle the nuances of sentiment in natural language.