Project report

Toktaganov Turlykhan

BDA-2106

Twitter account sentiment text analysis

Youtube video: https://youtu.be/jC8Qo7MdxYE

Github page: https://github.com/MightyDarkPs/sentiment_analysis_project

2023 Astana IT University

Outline of the report

1. Introduction

1.1 Problem

Today, many influential people through their twitter account massively promote their ideas and views. Many readers, without realizing it, begin to follow these trends and spread it further. For example, one tweet by Elon Musk was able to bring fluctuations in the cryptocurrency exchange. Moreover, Donald Trump's message even influenced the ruble exchange rate. If we can analyze these messages, it will help to know their mood and tone and how it will be reflected in the future.

1.2 Literature review

One of the examples I used is the code on the github page on twitter sentiment analysis (mehranshakarami, 2022). The author of this code decided to write his own code using libraries. One such library is TextBlob, which is a Python library that provides a simple API for common natural language processing tasks, including sentiment analysis. However, the author used a ready-made machine learning model. The only thing he did was write the code to split the offer into tokens using a special library. Despite that, all possible stop words and emojis also included in ready-made model.

1.3 Current work

In my Twitter sentiment analysis code, I utilized the NLTK library for various text preprocessing tasks such as tokenization, identifying emojis and stemming. The NLTK library provides a range of functions for these tasks, which can help improve the accuracy of sentiment analysis. I also used the scikit-learn library to build and train a machine learning model for sentiment classification.

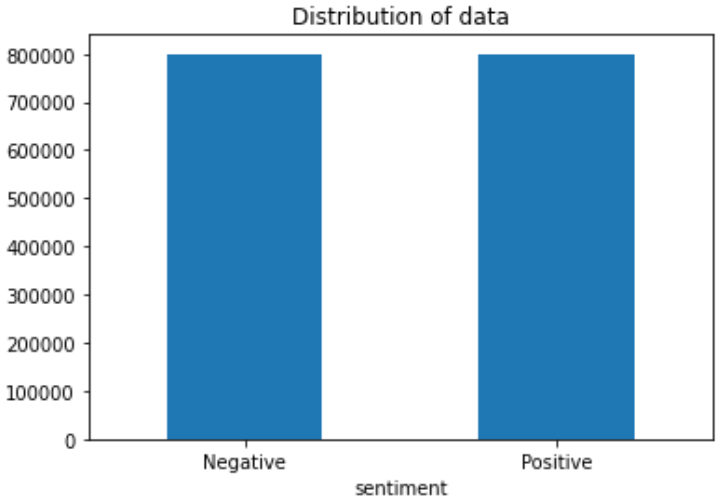
By leveraging these libraries, my Twitter sentiment analysis code was able to accurately classify the sentiment of tweets as positive or negative. These results were obtained by pre-processing the tweets with NLTK, extracting features using scikit-learn, and then training and evaluating a machine learning model.

Then I made a dump with pickle and used in on another code, where I just input my text and program try to identify whether it positive or negative.

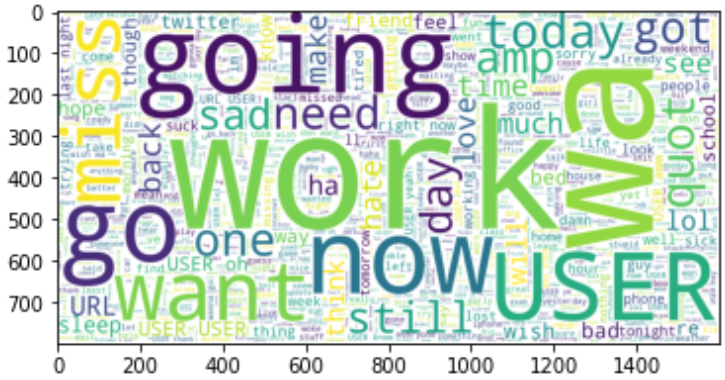
2. Data and Methods

2.1 Information about data

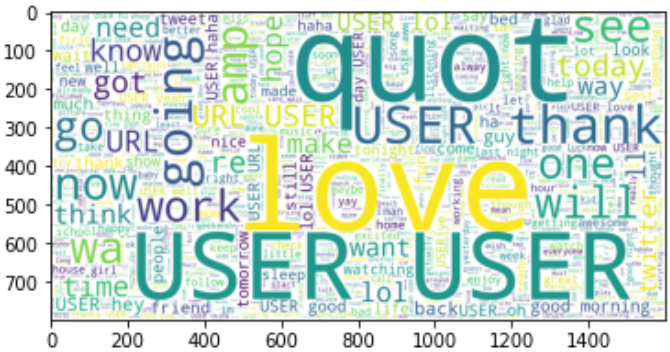
For my project I used the Sentiment140 dataset from Kaggle that contains 1,6 million tweets to train and test my machine learning model. The dataset was pre-labeled with positive and negative sentiment from 0 to 4 where 0 is negative and 4 is positive, which allowed me to train my model to accurately classify tweets as either positive or negative. However, I only used two columns out of six. For example, “target” and “text”, which I renamed to “sentiment” and “text”. Then, I also changed my grading system for reflecting sentiment to 0 and 1.



According to that graph, we can see, that in this dataset we have equally distributed positive and negative tweets where each has 800,000 rows of data.



I also made a wordcloud graph for better understanding my dataset. For example, on picture above we can see wordcloud of negative words. Here we can see that words like “word”, “going”, “now”, “go” mostly used on negative tweets.



Here we can see another example of wordcloud of positive words. The most common words here are “love”, “thank”, “quot”, “user”, “hope”. With that data we can train our model and predict mood of the message.

2.2 Description of the ML models

First, I divided my data into training and test datasets. I have put 10 percent of the original data in the test dataset and rest into training dataset.

In natural language processing, feature extraction is a critical step in preparing text data for analysis. For that I have used TF-IDF vectorizer. I used the TF-IDF vectorizer to extract features from the pre-processed tweets. The vectorizer allowed me to assign weights to each word in the tweets based on their frequency, which helped to improve the accuracy of my sentiment analysis. I first pre-processed the tweets using the NLTK library to tokenize and stem the text. Next, I used the TF-IDF vectorizer from scikit-learn to transform the pre-processed tweets into a matrix of feature vectors, where each row represented a tweet and each column represented a word with its corresponding TF-IDF weight.

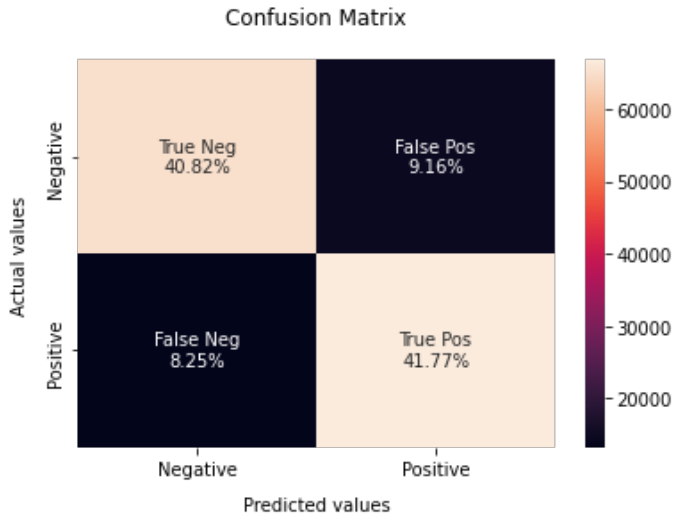
Then I have trained vectorizer with my training dataset. For further training model I have transformed my training and test datasets into matrix of TF-IDF features by using TF-IDF vectorizer. After that, these datasets are used to build a model and test it. In this project, I have decided to use logistic regression model. The logistic regression model is a popular binary classification algorithm that uses a logistic function to estimate the probability of a binary response variable.

After fitting the model, I used it to predict the sentiment of the test set, and evaluated its performance using metrics such as accuracy, precision, recall, and F1 score.

3. Results

	precision	recall	f1-score	support
0	0.83	0.82	0.82	79975
1	0.82	0.84	0.83	80025
accuracy			0.83	160000
macro avg	0.83	0.83	0.83	160000
weighted avg	0.83	0.83	0.83	160000

According to classification report, accuracy of the model is 83%. This is a pretty high number that allows you to predict positive and negative tweets.



I also made a confusion matrix, which shows that our model mostly makes predictions well.

	text	sentiment
0	I am proud of you	Positive
1	I love computer games	Positive
2	I hate to do homeworks	Negative
3	i don't feel so good	Negative
4	Good luck Kazakhstan!	Positive
5	I will get A score in advanced programming	Positive

After creating a logistic regression and vectorizer models, I exported them in pickle format and then imported it into a separate code to run. As we can see, I manually typed text in field and program started to make sentiment analysis.

4. Discussion

4.1 Critical review of results

According to all graphs, metrics and tests, it can be noted that the program is effectively fulfilling its task. During the test, there were incorrect predictions. This is because some words are used in both negative and positive tweets.

4.2 Next steps

The next step in this project is to improve it. There are several possible options for this. First, get an official developer twitter account. With it, we can access all user data. Second, increase the amount of data to train the model. This is related to the first point. Having received a developer account, it will be possible to train the model on millions of tweets from various users. This will help improve the accuracy of the model. In addition, to improve the project, we can add a web interface that will allow all people to conveniently make sentiment analysis of the user's tweets.

5. Resources

mehranshakarami. (1 March 2022 r.). *AI_Spectrum*. Received from github: https://github.com/mehranshakarami/AI_Spectrum/tree/main/2022/Sentiment_Analysis

KAZANOVA, M. M. (2018). *Sentiment140 dataset with 1.6 million tweets*. Received from kaggle: https://www.kaggle.com/datasets/kazanova/sentiment140

somvirs57. (15 May 2020 r.). *twitter_sentiment_analysis*. Received from github: https://github.com/somvirs57/twitter_sentiment_analysis

Arora, S. (7 july 2022 r.). *Sentiment Analysis Using Python*. Received from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2022/07/sentiment-analysis-using-python/