Social Media sentimental analysis

CSCI461: Introduction to Big Data-2023SPRG

Dr. Khaled Mohamed Fouad

ENG. Tawfik Yasser

Ahmed Magdy

Youssef Haitham

Ahmad Sarg

Yousef Farouk

Adham ELhamzawy

Contents

Abstract

This project aims to conduct sentiment analysis on social media data to understand the overall sentiment of the public towards a specific topic, brand, or event. The goal is to use natural language processing techniques such as NLTK to analyze large volumes of social media data and distinguish between positive and negative comments. The problem is that manually analyzing social media data is time-consuming and labor-intensive. The project compares three methods for sentiment analysis to find the most accurate and efficient method for processing large volumes of social media data. The sentiment140 dataset was used, which contains 1.6 million tweets annotated as positive or negative. Spark Stream and machine learning frameworks were used for the analysis. The expected outcome is to differentiate positive tweets from negative ones.

Introduction

This project aims to conduct sentiment analysis on social media data to understand the overall sentiment of the public towards a specific topic, brand, or event. The motivation behind this project is to provide businesses, organizations, and individuals with insights into public opinion, which can be used to make informed decisions and take appropriate actions. The problem that this project aims to solve is the manual analysis of social media data, which is time-consuming and labor-intensive. With the increasing use of social media platforms, there is a growing need for automated tools that can analyze large volumes of social media data and provide insights into public opinion.

Sentiment analysis is a powerful tool that can be used to analyze social media data and determine the overall sentiment of the public towards a particular topic, brand, or event. By using natural language processing techniques such as NLTK, this project aims to distinguish between positive and negative comments on social media and provide insights into public opinion.

The project compares three different methods for sentiment analysis and evaluates their accuracy and efficiency for processing large volumes of social media data. The dataset used for this analysis is the sentiment140 dataset, which contains 1.6 million tweets annotated as positive or negative. The project uses Spark Stream and machine learning frameworks for the analysis.

Overall, this project aims to provide businesses, organizations, and individuals with a better understanding of public opinion and help them make informed decisions based on this information.

Methodology

Problem Statement: The objective of this project is to develop a sentiment analysis model to classify social media posts into positive and negative categories. The sentiment analysis model will help businesses gain insights from customer feedback and sentiment expressed on social media platforms.

Data Source: The dataset used is ‘sentiment140’ It contains 1,600,000 tweets extracted using the twitter api . The tweets have been annotated (0 = negative, 4 = positive) and they can be used to detect sentiment.

Technologies Used:

- Spark
- PySpark
- Spark MLlib
- Matplotlib
- WordCloud
- NLTK
- Matplotlib

Expected Outcomes:

A trained sentiment analysis model capable of classifying social media posts into positive and negative sentiments.

Analysis Phase

The dataset was analyzed using 3 machine learning models including the above techniques. It was analyzed based on the classification of the positive and negative tweets. Analysis phases were Applied in three phases.

First Phase SVM (Support Vector Machine) its is a machine learning model that can be used for the large datasets and wide range classification problems, and it handles linear and non-linear boundaries the SVM is applied in the project with the snowball stemming and the model is trained twice first by using stemming technique the performance resulted with a good accuracy and secondly by not applying stemming with better accuracy and shorter execution time.

Second Phase Logistic regression is a supervised learning algorithm used for classification problems. It can handle both numerical and categorical variables the logistic regression algorithm were applied the data model is trained twice first by using stemming and secondly by not applying stemming which resulted in equal accuracy and while not using stemming has better execution time.

Third Phase Naïve Bayes a supervised learning algorithm and it is particularly useful for high-dimensional data sets Unlike logistic regression, it does not learn which features are most important to differentiate between classes. The algorithm was applied to train the data. It was trained twice by applying the stemming and not applying the stemming. The performance was better in accuracy and time.

Data cleaning is the process of correcting, removing error in the data set an this was applied to the data set by handling the missing values such as incomplete data entry or null values and removing data duplicates to avoid any output bias error.

Data exploration

Examining its structure. Checked the number of rows (1600000) and columns (6) in the dataset. Each row represents a tweet, and the columns include attributes like the sentiment label, tweet id, date, username, tweet text.

Determine the distribution of sentiment labels in the dataset. Count the number of positive (label 4) and negative (label 0) tweets it is found to be balanced.

Data visualization

Plotted a bar chart to visualize sentiment distribution and word cloud to visualize the most frequent word in each sentiment category.

Data Processing and Analysis Pipeline

The data processing and analysis pipeline for this project will include the following steps:

1. Data Preprocessing:

- The text data is cleaned by removing specific patterns such as URLs, mentions, and non-alphanumeric characters using regular expressions.

- The text is converted to lowercase and leading/trailing whitespaces are trimmed.

2. Pipeline Construction:

- A pipeline is created using the PySpark ML Pipeline API to encapsulate the data preprocessing and modeling steps.

- The pipeline consists of the following stages:

- RegexTokenizer: Tokenizes the text into individual words using the specified pattern '\W' (non-word characters)

- StopWordsRemover: Removes common stop words from the tokenized words.

- CountVectorizer: Converts the filtered words into numerical features, creating a feature vector representation of the text data.

- StringIndexer: Converts the target labels from their original format to numerical format for model training.

-applying selected model

3. Data Split:

- The dataset is split into training and testing sets using a ratio of 0.8:0.2, with a fixed seed for reproducibility.

4. Model Training:

- The pipeline is trained on the training data using the `fit()` method.

5. Model Saving:

- The trained model is saved to disk using the `save ()` method with the specified file path.

6. Model Evaluation:

- Predictions are made on the testing data using the trained model.

- The model's performance is evaluated using the MulticlassClassificationEvaluator, measuring accuracy as the metric.

7. using trained model to classify new text:

-Load the saved model using the PipelineModel.load() function.

-Provide the new text to be classified.

-Create a DataFrame with the user text.

-Perform the same data cleaning and preprocessing steps as done during training.

-Predictions are made on the vectorized data using the loaded model loadedModel.transform().

Results and Analysis

The evaluation was based on accuracy, and the results showed that the Naive Bayes model without stemming had the best performance 77.21% accuracy in a runtime of 3 minutes, followed by the support vector machine model without stemming 77.16% accuracy in a runtime of 5 minutes. The logistic regression model without stemming achieved an accuracy of 75.98% in a runtime of 5minutes , while the model with stemming achieved an accuracy of 75.95% but in time of 13 minutes. Additionally, the support vector machine model with stemming achieved 76.51% accuracy in runtime of 13 minutes, while naïve bayes model with stemming achieved 77.71% accuracy in runtime of 8 minutes.

Model	Accuracy	Time
Support vector machine (with stemming)	76.51%	13 min
Support vector machine (without stemming)	77.16%	5min
Logistic regression (without stemming)	75.98%	5min
Logistic regression (with stemming)	75.95%	13min
Naïve bayes (with stemming)	76.71%	8min
Naïve bayes (without stemming)	77.21%	3min

It is worth noting that the models that incorporated stemming took longer to execute, and their accuracy was slightly lower compared to the non-stemming counterparts. Although the

differences in accuracy were minimal, this suggests that the stemming technique may not have had a significant impact on improving the models' performance in this specific sentiment analysis task.

However, it is worth noting that runtime can be a significant consideration in practical applications. The models that incorporated stemming took longer to execute, which may impact their feasibility in certain scenarios where real-time or near-real-time analysis is required.

Conclusion

The project aimed to develop a sentiment analysis model to classify social media posts into positive, or negative. The sentiment analysis model was trained on a dataset of 1,600,000 tweets extracted using the Twitter API and annotated as positive or negative. The data processing and analysis pipeline included data cleaning, data visualization, tokenization, stop words removal, vectorization, model development, and model evaluation. Three models were used: support vector machine, logistic regression, and Naive Bayes. The performance of the sentiment analysis models was evaluated based on their accuracy, and it was found that the Naive Bayes model without stemming achieved the highest accuracy, the support vector machine model without stemming came in second. On the other hand, the models that used stemming took longer to run and had slightly lower accuracy. The sentiment analysis model developed in this project provides an accurate and efficient way to classify social media posts into positive or negative sentiments, offering valuable insights into public opinion for businesses, organizations, and individuals to make informed decisions.

Future Work

Some potential future work that could be done to improve the results of the sentiment analysis model:

Use more advanced preprocessing techniques: In this project, the data was cleaned by removing special characters, links, mentions, and other noise from the tweet text, and the text was converted to lowercase. However, more advanced preprocessing techniques could be used¬, such as lemmatization, which reduces words to their base form, and part-of-speech tagging, which identifies the part of speech of each word. Use more sophisticated feature engineering techniques: In this project, the tokenized words were converted into numerical features using Count Vectorizer. Use deep learning techniques: In this project, traditional machine learning models were used, such as Naive Bayes, logistic regression, and support vector machine. However, deep learning techniques, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), could be used to improve the accuracy of the sentiment analysis model.

Use more diverse datasets: In this project, the sentiment analysis model was trained on a dataset of 1,600,000 tweets. However, using more diverse datasets, such as reviews from different sources/topics, could improve the model's ability to classify sentiment in a more general way.

References

[1] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, pp. 1-12.

Appendix

Project code and readme file:

https://github.com/AhmedMagdy002/BigData_project