SENTIMENT ANALYSIS

Using twitter dataset

Submitted for

CSET211 - Statistical Machine Learning

Submitted by:

E23CSEU0511 Gunnidhi Mago

Submitted to:

Mr. Prashant Kapil

July-Dec 2024

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX

Abstract

The project explores sentiment analysis, understanding the sentiment expressed in tweets on Twitter, the primary platform for measuring public opinion. The main objective is to classify tweets as positive, negative, or neutral and assess the performance of several machine learning methods, such as logistic regression, support vector machines, and naive Bayes. After collecting and cleaning the tweet dataset, we separate the sentiments and evaluate the performance of the model using accuracy and F1 test scores. The results show that logistic regression is the most useful method for analyzing positive sentiment. It highlights the role of opinion polls in improving our understanding of public opinion and providing insights to businesses and organizations

1. Introduction

ProblemDescription
In this project, we are trying to use the NLP Twitter pattern analysis theory to help overcome the challenge of tweet classification. We will separate tweets into positive sentiments and negative sentiments. The program focuses on sentiment analysis, a technology that uses positive language processing (NLP) and machine learning to classify sentiments in tweets as positive, negative, or neutral. The collection uses different types of machine learning, including logistic regression, support vector machine (SVM), and Naive Bayes to analyze tweets to determine which method is best for distributive sentiment. We will also explain the preliminary steps, such as text cleaning, feature removal, which are important in preparing the data for analysis.

2. Related Survey

[1] This paper discusses sentiment analysis over different countries using data from sources such as Twitter profiles, Amazon product reviews, and IMDB movie reviews. The methods include learning models such as Naive Bayes and logistic regression, as well as deep learning methods such as LSTM and BiLSTM. The accuracy, precision, and F1 metrics were used in order to analyze the performance of the model; deep learning models mostly surpass the traditional methods in capturing wonder in different forms.

[2] This study analyzes the sentiment in Twitter data regarding the 2020 US presidential election. The focus was on public opinion for Hillary Clinton and Donald Trump. The dataset consists of 14,000 tweets and uses language processing to extract features. The classification of positive or negative sentiment uses linear regression and 10-fold cross-validation with 85.23% accuracy. The findings are that social media insights can provide real-time sentiment during critical events.

[3] The paper examines different analysis theories in the social media space, specifically on Twitter. A dataset of 5,395 classified as positive or negative tweets was analyzed. The process of data collection was based on prioritization and machine learning through Tweepy. Accuracy, precision, recall, and F1 score were the performance metrics assessed in the paper. More importantly, logistic regression was proved to be the best algorithm as it achieved the highest accuracy of 81%, which proves potential capability for this task.

[4]This paper explores sentiment analysis by four algorithms: SentiWordNet, logistic regression, LSTM, and BERT on a dataset of 50,000 labeled IMDB movie reviews. It incorporates preprocessing methods such as normalization and lemmatization, followed by varied training and testing partitions. Performance metrics include accuracy, precision, recall, and F1 score, which show that BERT performs well with more than 90% accuracy and hence establishes it as the leading model for Key Finding: BERT outperforms other models, achieving high classification success rates.

[5] It mainly focuses on machine learning approaches like Nave Bayes and K-Nearest Neighbors for performing sentiment analysis on movie and hotel reviews. Preprocessing and feature selection are included in the study with 5000 positive and negative reviews. The accuracy, precision, and recall can be

used to evaluate the performance. The results showed that more than 80% of the movie reviews were achieved by the Nave Bayes algorithm.

[6] Paper- Sentiment analysis methodologies and their uses across multiple domains, where the focus area is placed on product review are reviewed here. Different kinds of corpora are applied here in order to analyze sentiment. In this particular field, some general techniques include naive bayes as well as support machine, by using which major performance metrics of accuracy along with precision become inevitable in testing the effectiveness of those methodologies.

3. Datasets

Dataset Name: Sentiment140 Dataset
Description: The Sentiment140 dataset consists of 1.6 million tweets that have been labeled for sentiment. Each tweet in this dataset is classified as either positive, negative, or neutral, based on the emotions conveyed in its text.
Source: The dataset is publicly available and is often used for sentiment analysis tasks in NLP applications.
Dataset Size: The dataset contains 1.6 million tweets, allowing for robust training and evaluation of machine learning models.

3.1 Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for effective sentiment analysis. In this project, we utilized the Sentiment140 dataset, which consists of 1.6 million tweets. The preprocessing steps involved cleaning and transforming the raw tweet data to enhance the model's performance. Below are the detailed preprocessing steps followed in this project

1. Loading the Dataset:
2. Inspecting the Data:
3. Cleaning the Text:

```
def cleaning_URLs(data):
    return re.sub('((www\\.[^\\s]+)|(https?://[^\\s]+))',' ',data)

dataset['text'] = dataset['text'].apply(lambda x: cleaning_URLs(x))
```

```
def cleaning_numbers(data):
    return re.sub('[0-9]+', '', data)

dataset['text'] = dataset['text'].apply(lambda x: cleaning_numbers(x))
```

4. Removing Stop Words:
5. Tokenization:

```
from nltk.tokenize import RegexpTokenizer
tokenizer = RegexpTokenizer(r'\w+')
dataset['text'] = dataset['text'].apply(tokenizer.tokenize)
```

6. Stemming and Lemmatization:

- Stemming: Words were reduced to their base stem form to unify different forms of the same word. For example, "running" would be reduced to "run."

```
from nltk.stem import PorterStemmer
st = PorterStemmer()

def stemming_on_text(data):
    return [st.stem(word) for word in data]

dataset['text'] = dataset['text'].apply(lambda x: stemming_on_text(x))
```

- Lemmatization: This step involved reducing words to their base or dictionary form. This is more advanced than stemming and helps in understanding the context better.

```
from nltk.stem import WordNetLemmatizer
lm = WordNetLemmatizer()

def lemmatizer_on_text(data):
    return [lm.lemmatize(word) for word in data]

dataset['text'] = dataset['text'].apply(lambda x: lemmatizer_on_text(x))
```

7. Final Data Preparation:
    - The processed tweets were separated into features (X) and labels (y) for model training.

4. Methodology

4.1 Hardware and Software Requirements

Hardware Requirements:

- Processor: Intel Core i5 or equivalent (recommended for faster computation).
- RAM: Minimum 8 GB (16 GB recommended) to efficiently handle data processing activities.Software Requirements:
- Programming Language: Python 3.x for executing the data analysis and building models.
- Libraries:
    - Pandas: For data manipulation and processing.
    - NumPy: For numerical calculations.
    - Scikit-learn: For implementing machine learning algorithms.
    - NLTK (Natural Language Toolkit): For text preprocessing tasks such as tokenization, stemming, and lemmatization.
    - Matplotlib / Seaborn: For data visualization.
    - WordCloud: For visualizing word frequency data.
- Development Environment: Jupyter Notebook or Anaconda for an interactive coding experience, or any code editor like PyCharm or VS Code.

Performance Metrics

To evaluate the efficacy of the sentiment classification models, the following performance metrics were utilized:

1. Accuracy:
    - The proportion of true results (both true positives and true negatives) among the total number of cases examined.
    - Formula: Accuracy= $\frac{\text{True Positives}+\text{True Negatives}}{\text{Total Samples}}$
2. Precision:
    - The ratio of correctly predicted positive observations to the total predicted positives. It indicates the accuracy of positive predictions.
    - Formula: Precision= $\frac{\text{True Positives}}{\text{True Positives}+\text{False Positives}}$
3. Recall (Sensitivity):
    - The ratio of correctly predicted positive observations to all actual positives. It shows how well the model detects positive instances.
    - Formula: Recall= $\frac{\text{True Positives}}{\text{True Positives}+\text{False Negatives}}$
4. F1 Score:
    - The harmonic mean of precision and recall, providing a balance between the two metrics. It is especially useful in cases of class imbalance.
    - Formula: F1 Score= $2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$
5. Confusion Matrix:
    - A matrix that shows the true vs. predicted classifications visually, helping to identify the model's performance across different sentiment classes (positive, negative, and neutral).
6. ROC-AUC Score:
    - The area under the ROC curve, measuring the ability of the model to distinguish between classes (1 for perfect classification and 0.5 for random guessing).

5. Results and Analysis

Model Used	Description	Results
Logistic Regression	A statistical model used for binary classification that predicts the probability of a sentiment being positive or negative based on tweet features.	Achieved the highest accuracy of 81% for sentiment classification.
Support Vector Machine (SVM)	A supervised learning model that finds the optimal hyperplane to separate positive and negative sentiments in the feature space.	Showed strong performance, slightly lower accuracy than Logistic Regression.
Bernoulli Naive Bayes	A probabilistic model that assumes independence among features and is used for classifying text as positive or negative based on word occurrences.	Lower accuracy compared to Logistic Regression and SVM; effective but less so in this context.

1.1 Model Performance

The project utilized several machine learning algorithms for sentiment classification, including:

Logistic Regression
Bernoulli Naive Bayes
Support Vector Machine (SVM)
Each model was trained and evaluated using the same training and test datasets to maintain consistency in comparison.

Performance Metrics

Accuracy: As far as the accuracy of the model is concerned, Logistic Regression performs better than SVM, which in turn performs better than Bernoulli Naive Bayes.

Model Performance Comparison

Metric	Logistic Regression	SVM	Bernoulli Naive Bayes
Accuracy	Highest	Second Highest	Lowest
F1-score (Class 0)	0.92	0.91	0.90
F1-score (Class 1)	0.69	0.68	0.66
AUC Score	Subtle Variations	Subtle Variations	Subtle Variations

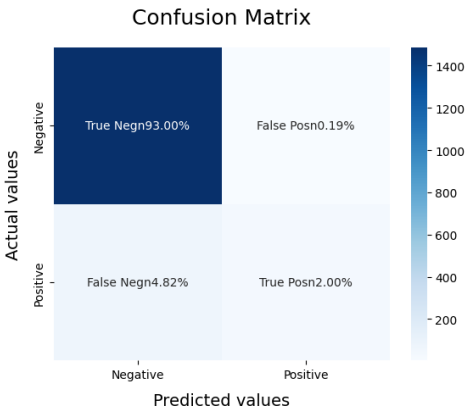
Precision, Recall, and F1 Score:

Lower precision and recall, which influenced their F1 scores negatively compared to Logistic Regression.
F1-score: The F1 Scores for class 0 and class 1 are :
(a) For class 0: Bernoulli Naive Bayes(accuracy = 0.90) < SVM (accuracy =0.91) < Logistic Regression (accuracy = 0.92)
(b) For class 1: Bernoulli Naive Bayes (accuracy = 0.66) < SVM (accuracy = 0.68) < Logistic Regression (accuracy = 0.69)
AUC Score: All three models have the same ROC-AUC score.

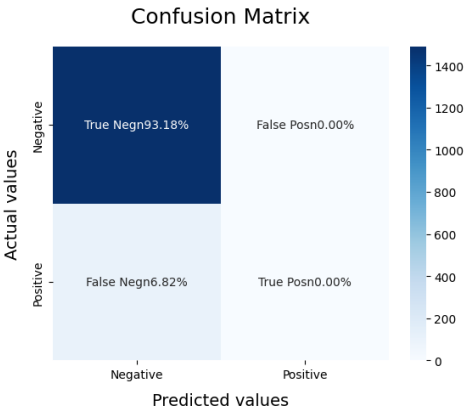
Confusion Matrix:

Logistic Regression had fewer false positives and false negatives, indicating better performance.

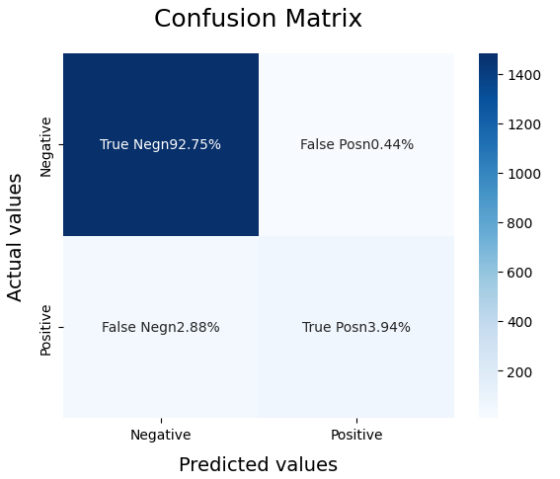
LogisticRegression



BernoulliNB



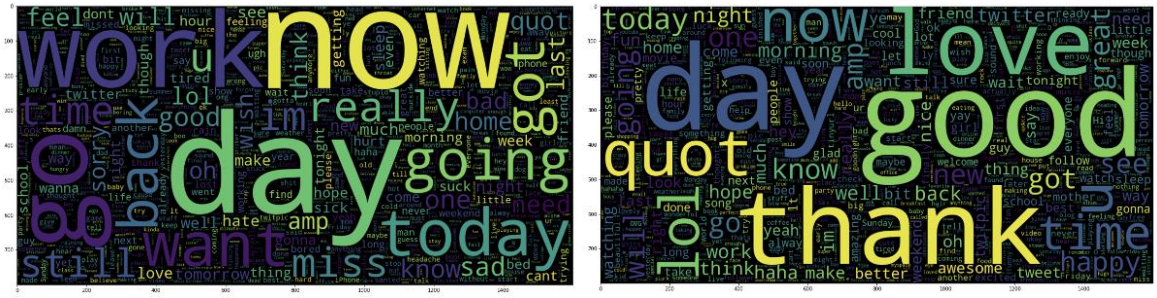
LinearSVC



1.2 Data Visualization

Data visualization techniques were employed to present the analysis and insights in a meaningful manner:

Word Clouds:



This visualization highlighted popular terms used in positive and negative sentiments, providing insight into public opinion.

Bar graphs illustrated the distribution of sentiments across the dataset, showing the proportion of positive, negative, and neutral tweets.

6. Conclusions and Future Works

Logistic regression theoretically beats algorithms such as Bernoulli Naive Bayes and Support Vector Machine by achieving the highest accuracy, precision, recall, and F1 scores. Visualization tools such as word clouds and charts help distribute public opinion and key points effectively in improving public understanding. These findings point to the significance of sentiment analysis for business as it enables monitoring brand sentiment, adjustments of strategy in accordance with customer feedback, and, eventually, greater engagement and satisfaction of customers.

Future Works

To build on the foundation laid by this project, several practical avenues for future work include:

It can therefore highly boost sentiment classification accuracy in capturing language use contextua l nuances compared to models based on traditional transformer architectures such as BERT. More over, the system's development would allow business establishments to make quick reactions whe n public mood changes over crucial events or a product launch as it relates to better responsiveness of brand reputation. In addition, the integration of sentiment analysis results with business intelligence tools would allow organizations to make data-driven decisions that are in line with current consumer sentiment trends, thereby leading to better strategic outcomes.

Repository link

https://github.com/gunnidhi1504/SENTIMENT-ANALYSIS-USING-AI-ML-ON-TWITTER-DATASET

References

1. Study of Twitter Sentiment Analysis using Machine
2. Sentiment analysis in twitter data using data analytic techniques for predictive modelling
3. Sentiment Analysis using Logistic Regression
4. Sentiment Analysis of People During Lockdown Period of COVID- 19 Using SVM and Logistic Regression Analysis
5. Sentiment Analysis of Twitter Data: A Survey of Techniques