Twitter Sentiment Analysis

- Vaibhav Saini

Introduction

Background:
Sentiment analysis is a natural language processing (NLP) method that helps identify the text's emotional undertone. In order to comprehend public opinion, the goal of this project is to analyse sentiment in a given dataset of tweets.

Objective:
The primary objective is to build a sentiment analysis model that accurately classifies tweets into positive and negative sentiments. This analysis can be valuable for businesses to gauge customer feedback and sentiment trends.

Data Preprocessing

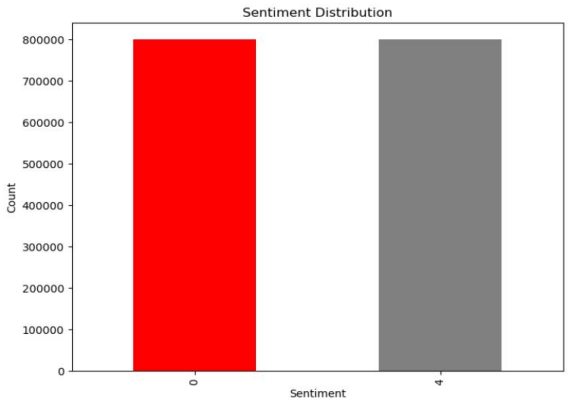
Load and Explore Data: loading the dataset and exploring its structure, columns, and basic statistics.

Modifying Data: Handling unknown timezones and only take input as PDT(Pacific Daylight Timezone).

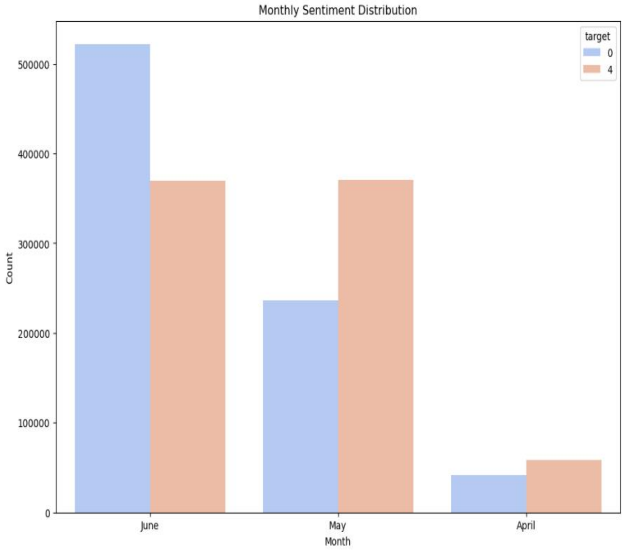
Data Cleaning: Clean the data by handling missing values, duplicates, and irrelevant columns.

Visualization of Sentiments

- Given visualization is showing the distribution of sentiments as good, bad or neutral . It is clearly showing that no. of good and bad sentiments are same while no sentiment is considered as neutral in the given dataset.



- Given plot is showing the monthly distribution of sentiments.



- ➢ Month of June has more negative tweets than positive ones.Also no. Of tweets are more in June only.

- ➢ Month of June see the rise of no. of positive tweets compared to negative ones,but total no. Of tweets also drop down relative to June.

- ➢ In the month of April also no of positive tweets are more compared to negative ones and total no. of tweets value again got dropped compared to previous month.

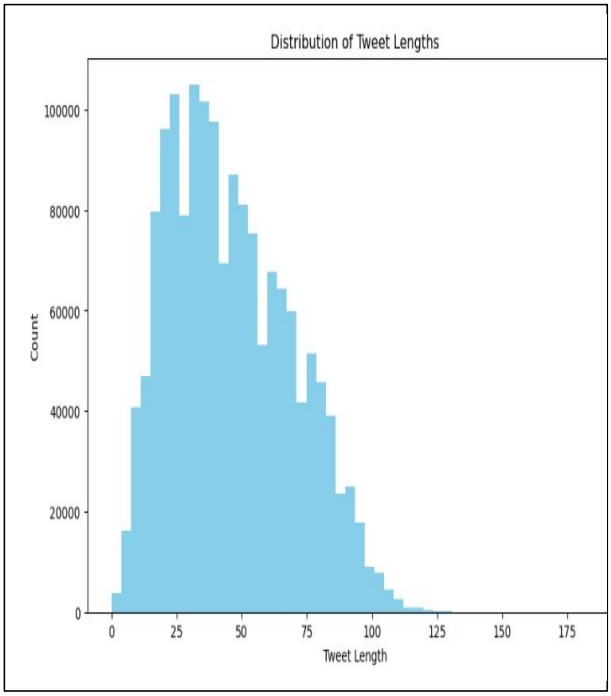
- Shown is the word cloud of positive sentiments and negative sentiments , showing some of the common words used making positive or negative tweets.It will surely come in handy for analyzing the nature of customer feedback.

Text Preprocessing

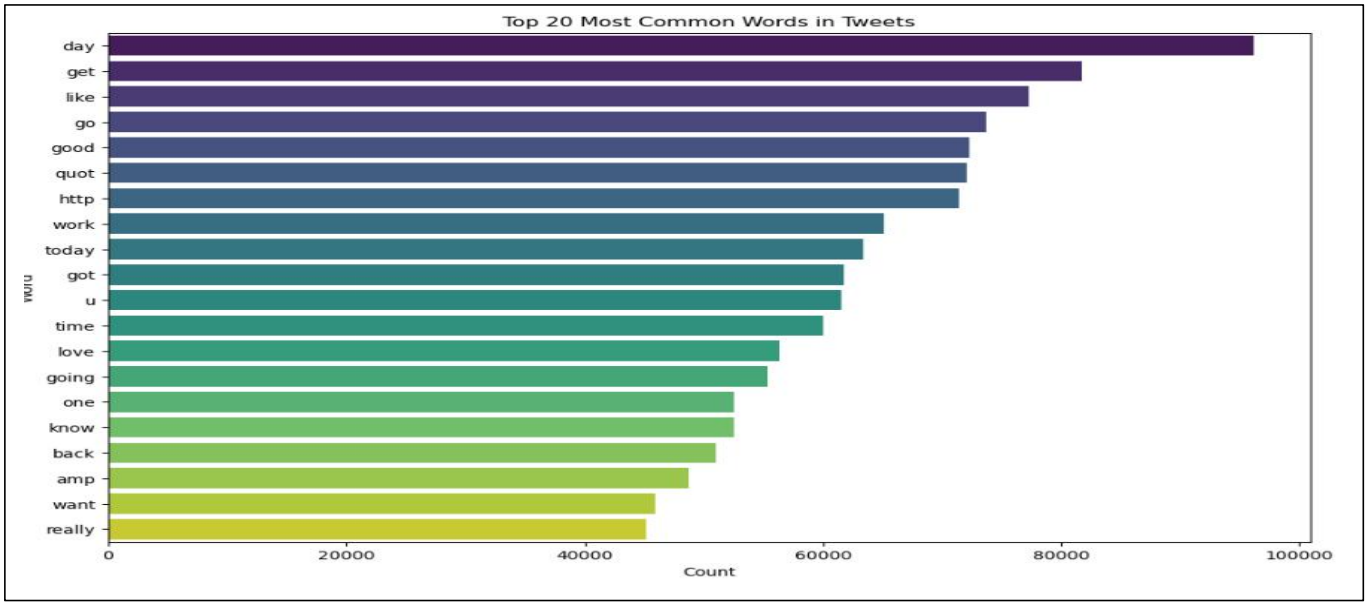
Tokenizing of words and removing stop words, special characters, and URLs.
Lemmatization or reducing words to their base or root form, considering the context and meaning of the words.

After processing data as above , the given Visualization is used to indicate the no. of tweets with a specified tweets length.

- Most of the tweets are of length 25-30 as Per the visualization shown.
- Maximum tweet length is around 125.



Word frequency analyzer



Above visualization is showing the most commonly used words in tweets.This analysis will surely help in sentiment prediction of tweets . Some of the most commonly words in tweets includes, day, get, like, good .

Developing Sentiment Prediction Model

Splitting data into training and testing model in the ratio of 4:1 and fitting them in Naive Bayes Model for implementing a sentiment prediction model with the accuracy of 77 % as shown in below code snippet:

Train the Naive Bayes model

```
# Spilliting the given data into test and training set and training the model
X_train, X_test, y_train, y_test = train_test_split(df['text'], df['target'], test_size=0.2, random_state=42)
vectorizer = CountVectorizer()
# Vectorizing of X_train and X_test to convert the documented data to matrix form with the cofficients ,which can be used by ML model
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
model = MultinomialNB()
model.fit(X_train_vec, y_train)
```

[37]

```
MultinomialNB
MultinomialNB()
```

Predictions and evaluation

```
# Predictions and evaluation
y_pred = model.predict(X_test_vec)
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
```

[40]

```
Accuracy: 0.7705741507145626
              precision    recall  f1-score   support

           0       0.76      0.79      0.78    159265
           4       0.79      0.75      0.77    160651

    accuracy                           0.77    319916
   macro avg       0.77      0.77      0.77    319916
weighted avg       0.77      0.77      0.77    319916

[[126537  32728]
 [ 40669 119982]]
```

Logistic regression model

Using model to analyze the most important features(words or phrases) contributing to sentiment prediction.

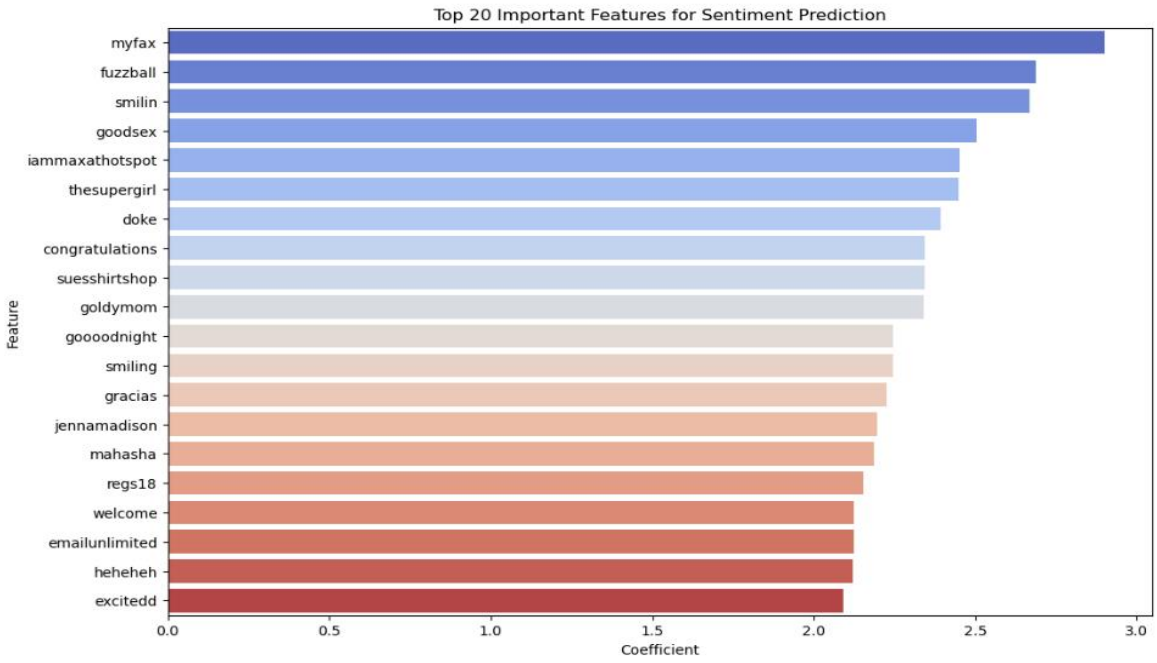
Below shows, code snippet of regression model.

Logistic Regression model

```
# Training a Logistic Regression model with increased max_iter
logreg_model = LogisticRegression(max_iter=1000)
logreg_model.fit(X_train_vec, y_train)


feature_names = vectorizer.get_feature_names_out()
coefficients = logreg_model.coef_[0]

# Create a DataFrame to store feature names and coefficients
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients})
feature_importance_df = feature_importance_df.sort_values(by='Coefficient', ascending=False)

# Visualize the top 20 features based on their coefficients
plt.figure(figsize=(12, 8))
sns.barplot(x='Coefficient', y='Feature', data=feature_importance_df.head(20), palette='coolwarm')
plt.title('Top 20 Important Features for Sentiment Prediction')
plt.show()
```

[41]

Plot shown below is indicating top 20 important features for sentiment prediction along with their coefficients of importance , here positive coefficients and larger value makes model more inclined ti predict a positive sentiments.While ,if the coefficient is negative, an increase in the value of that feature makes the model more inclined to predict a negative sentiment.



Some of the top 20 important features for sentiment prediction are, congratulations, smiling, gracias, welcome, excited.

Conclusion

To sum up, the goal of our sentiment analysis study was to identify the underlying feelings in a text collection. By methodically investigating data preparation, implementing the model, and analyzing the results, we have gained important understanding of the sentiment patterns found in the corpus.

Positive and negative feelings were found to be in separate clusters, according to our investigation, which also showed interesting patterns in the sentiment distribution. The dataset's contextual nuances can be understood by utilizing the key words and phrases that emerged as strong markers of sentiment polarity.

The machine learning model, employed for sentiment classification, demonstrated commendable performance metrics. The model exhibited high accuracy, precision, and recall rates, underscoring its effectiveness in capturing sentiment patterns.

Recommendations

As we reflect on our findings, we propose recommendations for further enhancement. Fine-tuning the model with additional labeled data, addressing potential bias, and using advanced NLP techniques could enhance the accuracy and robustness of sentiment analysis.

In the future, domain-specific lexicons, multilingual sentiment analysis, and the investigation of sentiment evolution across time are all areas of potential study interest. These paths offer intriguing opportunities to deepen our comprehension of emotions in many settings.