Twitter Sentiment Analysis

BY VANSHAY KULKARNI

Outline

- Introduction
- Dataset used
- Required tools
- Steps performed
- Conclusion

Introduction

The Twitter Sentiment Analysis project involves using natural language processing and machine learning techniques to assess the sentiments (positive or neutral) expressed in tweets. By collecting and labeling a dataset of tweets, the project aims to train models that can automatically categorize the sentiment of new tweets.

This analysis is valuable for businesses and researchers to understand public opinion, monitor brand perception, and track sentiment trends on the Twitter platform.

Required tools

The tools used for this project are:

- Jupyter Notebook
- Kaggle
- MS PowerPoint

Dataset used

The dataset used in this project is 'Sentiment140 dataset with 1.6 million tweets' from Kaggle.

Link: https://www.kaggle.com/datasets/kazanova/sentiment140

This dataset contains 1.6 million tweets in which, 800k tweets are labeled as **positive** and the remaining 800k tweets are labeled as **negative**.

Steps Performed

Following are the steps performed in this project:

1. Importing the libraries
2. Data pre-processing
3. Training and Testing the data
4. Model building
5. Saving the model

Importing the libraries

In this step the required libraries are imported.

```
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [3]:

```
import nltk
nltk.download('stopwords')
```

Data pre-processing

In this step the following processes are performed:

1. Importing the dataset
2. Cleaning the dataset
3. Organizing the dataset
4. Stemming

Training and Testing the data

After pre-processing the data, we need to train and test it so that it can be used to feed a model.

For training and testing, 80/20 split of the data is used here.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, stratify = y,
random_state = 2)
```

In [23]:

```
print(x.shape, x_train.shape, x_test.shape)
```

```
(1600000,) (1280000,) (320000,)
```

In [24]:

```
vectorizer = TfidfVectorizer()
x_train = vectorizer.fit_transform(x_train)
x_test = vectorizer.transform(x_test)
```

Model building

Logistic Regression model is used here as it is suitable for the given dataset.

The accuracy of the model is measured using 'Accuracy Score'.

Accuracy Score of the training data: **81.02%**

Accuracy Score of the testing data: **77.80%**

Saving the model

The model is saved for future use by using the following commands:

```
import pickle
```

```
In [35]:
```

```
filename = 'twitter_trained_model.sav'
pickle.dump(model, open(filename, 'wb'))
```

Conclusion

The Twitter Sentiment Analysis project provides valuable insights into public sentiments expressed on the platform. By leveraging natural language processing and machine learning, the project successfully categorizes tweets as positive or negative. This analysis proves beneficial for businesses and researchers in understanding public opinion, evaluating brand perception, and monitoring sentiment trends over time.

As social media continues to play a significant role in shaping public discourse, the insights gained from this project contribute to a deeper understanding of the evolving landscape of opinions and attitudes on Twitter.