Social Media Sentiment Analysis Project 🌐

In this project, I conducted a comprehensive sentiment analysis on tweets related to airlines, utilising various data science methodologies and techniques. Below, I detail the key steps undertaken during the project, the libraries used, and the results obtained.

1. Import Libraries and Set Up 📚

I started by importing essential libraries for data manipulation and analysis. Key libraries included:

- **Pandas**: For data handling and analysis.
- **NumPy**: For numerical operations.
- **Matplotlib & Seaborn**: For visualisation.
- **NLTK**: For natural language processing tasks, such as tokenisation and stemming.
- **SQLite**: For connecting and querying the SQLite database.

2. Load & Merge Data 🔄

I loaded the datasets from both a SQLite database and a CSV file using the **sqlite3** and **pandas** libraries. After merging the two datasets on the tweet_id, I saved the combined data to a CSV file for further analysis. This step was crucial to ensure that I had a comprehensive dataset that contained relevant features for sentiment analysis.

Key Steps:

- Established a database connection and executed SQL queries to extract tweet data.
- Merged datasets on the tweet_id column, enhancing the dataset with richer context.

3. Data Cleaning and Preprocessing 🧹

The merged dataset underwent cleaning to ensure data quality. This involved:

- **Dropping duplicates**: Ensured there were no repeated entries.
- **Handling missing values**: Filled missing values or dropped rows based on a defined threshold (10%).
- **Datetime conversion**: Converted tweet creation times to the proper datetime format for analysis.

- **Dropping unnecessary columns**: Removed irrelevant columns to simplify the dataset.

I achieved a clean dataset ready for exploratory analysis.

4. Exploratory Data Analysis and Visualization 📊

Exploratory Data Analysis (EDA) allowed me to gain insights into the dataset's characteristics:

- **Sentiment Scores Distribution**: Visualised the distribution of sentiment confidence scores using a histogram.
- **Correlation Heatmap**: Created a heatmap to identify relationships between numeric features, revealing key insights into how different features correlate with sentiment.
- **Bar Plots**: Utilised count plots to show the distribution of sentiments across different airlines, providing insights into public perceptions.

Results:

- The mean sentiment confidence score was indicated with a red line in the distribution plot, helping identify the general sentiment trend.

5. Feature Selection and Engineering ⚙️

I selected key features to train my sentiment classification model. This involved creating new features, such as the length of tweets, which could impact sentiment analysis.

I performed text preprocessing, including:

- **Tokenisation**: Broke down tweets into individual words.
- **Stop Word Removal**: Eliminated common words that may not contribute to sentiment (e.g., "the", "is").
- **Stemming and Lemmatization**: Normalised words to their base forms to reduce dimensionality in text data.
- **POS Tagging**: Identified parts of speech to enhance the contextual understanding of words.

6. Model Training 🏋️

I utilised a **Random Forest Classifier** for sentiment prediction. The model was trained using the processed features, and a training-test split (70-30%) was implemented to ensure robust evaluation.

Model Training Results:

- The model was trained successfully, setting the stage for evaluation and tuning.

7. Model Evaluation 📈

I evaluated the model using various metrics:

- **Accuracy**: Achieved an accuracy of **85.75%**, indicating a high level of correctness in predicting sentiments.
- **Confusion Matrix**: Visualised true vs. predicted classifications to assess model performance.
- **Classification Report**: Detailed precision, recall, and F1-scores for each sentiment class.

Interpretation of Results:

- High accuracy signifies that the model generalises well to unseen data, but precision and recall were variable across sentiment classes, highlighting the need for further tuning.

8. Hyperparameter Tuning 🔧

To optimise model performance, I implemented hyperparameter tuning using **GridSearchCV**. After testing various combinations of parameters, I found that the best estimator improved the model's accuracy to **88.5%**.

9. Save the Model 💾

I saved the trained model using **joblib**, making it available for future use in predicting sentiments of new tweets.

10. Visualize Sentiment Distribution 🎨

Finally, I visualised the sentiment distribution across all tweets using bar and pie charts. This visualisation provided a clear summary of public sentiment towards different airlines, highlighting overall trends and patterns.

Key Takeaways:

This project not only reinforced my data cleaning, preprocessing, and analysis skills but also deepened my understanding of natural language processing techniques in a practical context. I enhanced my capabilities in model training and evaluation, ultimately leading to a well-performing sentiment analysis model.

Skills Acquired:

Data Manipulation, Data Visualisation, Natural Language Processing, Machine Learning, Model Evaluation, Feature Engineering

Hashtags:

#DataScience #SentimentAnalysis #NLP #MachineLearning #DataCleaning #DataVisualisation #Python #RandomForest #EDA #ModelEvaluation #FeatureEngineering #Pandas #NumPy #Matplotlib #Seaborn #NLTK #DataAnalysis #GridSearchCV #BigData #AI