# Project Report: Sentiment Analysis of Airline Tweets

### 1. Introduction

This project involves analyzing airline tweets to determine the sentiment behind each tweet. Sentiment analysis classifies text data into positive, neutral, or negative categories, enabling organizations to gauge customer feedback. Using a dataset of airline-related tweets, this project demonstrates preprocessing steps, data visualization, and sentiment classification using machine learning models.

### 2. Objective

To classify airline tweets based on sentiment (positive, neutral, or negative) using various machine learning algorithms and to evaluate their performance.

### 3. Libraries Used

The following libraries were used:

- **Data manipulation**: pandas, numpy
- **Data visualization**: matplotlib, seaborn
- **Text preprocessing**: nltk, re
- **Machine learning models**: sklearn

### 4. Data Loading and Initial Exploration

The dataset Tweets.csv was loaded into a pandas DataFrame. Basic exploration revealed the following:

- **Columns**: Key columns included airline, airline_sentiment, airline_sentiment_confidence, and the tweet text.
- **Distribution**: Most tweets had negative sentiment, followed by neutral, with fewer positive sentiments. Virgin America's sentiment distribution was more balanced across categories.

### Visualization of Airline Sentiment

Pie charts were used to show:

1. Sentiment distribution across different airlines.
2. Overall sentiment distribution (negative, neutral, positive).

### 5. Data Cleaning

Text data preprocessing is crucial to improve model performance. The following steps were applied to clean and prepare the tweets:

- **Removing special characters**: To eliminate unwanted symbols.
- **Removing single characters**: These are often noise and don't contribute much to sentiment.
- **Removing prefixed characters**: Removing certain prefixed symbols.
- **Removing multiple spaces**: Replacing with single spaces.

- **Converting to lowercase**: For uniformity and to avoid case-sensitive mismatches.

**6. Feature Engineering and Text Representation**

To represent text data numerically, **TF-IDF (Term Frequency-Inverse Document Frequency)** was employed. This approach assigns weights to words based on their importance in the text and reduces the impact of commonly used words (stopwords).

- **TF-IDF Vectorization Parameters**:
    - max_features=2500: Limits the vocabulary to the 2500 most important words.
    - min_df=7 and max_df=0.8: Exclude extremely rare and overly common words.
    - stop_words=stopwords.words('english'): Removes common English stopwords.

**7. Splitting the Data**

The data was split into training and test sets using an 80-20 ratio.

- X_train, X_test: Features (TF-IDF vectors).
- y_train, y_test: Labels (sentiment).

**8. Model Training and Evaluation**

Four machine learning models were trained, and their performance was evaluated on the test set:

**8.1 Random Forest Classifier**

- **Parameters**: n_estimators=200, random_state=0
- **Evaluation**:
    - Accuracy: 75.99%
    - Precision, Recall, F1-score, and confusion matrix were calculated.

**8.2 Support Vector Machine (SVM)**

- **Parameters**: kernel='linear', C=1.0
- **Evaluation**:
    - Accuracy: 78%
    - Detailed metrics and confusion matrix were provided.

**8.3 Multinomial Naive Bayes**

- **Evaluation**:
    - Accuracy: 75.81%
    - Confusion matrix and classification report were computed.

**8.4 Logistic Regression**

- **Evaluation**:
    - Accuracy: 78.82%

  - Performance metrics indicated competitive performance with SVM.

**9. Model Comparison**

The models were evaluated based on accuracy and other performance metrics:

- **Random Forest**: 75.99% accuracy.
- **SVM**: 78% accuracy.
- **Naive Bayes**: 75.81% accuracy.
- **Logistic Regression**: 78.82% accuracy.

**Observations**

The SVM and Logistic Regression models slightly outperformed Random Forest and Naive Bayes. All models were competitive, with Logistic Regression achieving the highest accuracy.

**10. Conclusion**

This project demonstrated the use of text preprocessing and machine learning for sentiment analysis of tweets. The SVM and Logistic Regression models achieved the highest accuracy, showing that linear models can effectively classify tweet sentiment.

**11. Future Improvements**

- Experimenting with more complex feature extraction methods such as Word2Vec or BERT embeddings.
- Utilizing a larger dataset for training.
- Fine-tuning model hyperparameters further, especially in the Random Forest and SVM models.