**PROJECT: US AIRLINE TWEETS SENTIMENT ANALYSIS**

**CONTENTS:**

1.Objective

2.Data Description

3.Performing Basic Analysis and Inferences

4.Data Pre-Processing:

* Removing HTML Tags
* Removing Numbers
* Tokenization
* Removing stop words
* Lemmatization

5.Vectorization using Count Vectorizer and TF-IDF Vectorization

6.Model Building

7.Model Evaluation

8.Hyper Parameter Tuning of the Model

**1. OBJECTIVE:**

* The main objective of this project is to classify the sentiment of tweets directed at major U.S. airlines into three distinct categories: positive, neutral, and negative.
* This sentiment analysis aims to uncover the public perception and problems associated with different airlines, as expressed through tweets.
* By analysing customer sentiment, airlines can gain valuable insights into their operational strengths and weaknesses, thereby enabling them to improve customer satisfaction and overall service quality.
* Social media platforms like Twitter have become powerful tools for customers to share their experiences, concerns, and feedback about various services, including airline operations.
* This data-rich environment offers an opportunity to perform sentiment analysis using natural language processing (NLP) techniques. Specifically, this project focuses on understanding and quantifying customer sentiment through systematic preprocessing, feature extraction, and classification modelling.

**To achieve this goal, the project follows a structured approach:**

1. **Data Collection:** The dataset for this analysis is obtained from the Kaggle project site: [Twitter Airline Sentiment Dataset](https://www.kaggle.com/crowdflower/twitter-airline-sentiment). It consists of tweets directed at U.S. airlines, tagged with sentiments (positive, neutral, or negative).
2. **Data Preprocessing:** Textual data in its raw form contains noise and inconsistencies that must be addressed before analysis. The following preprocessing steps were performed:
   * Removal of numbers and HTML tags.
   * Lowercasing of text to ensure uniformity.
   * Tokenization to break down text into individual words or tokens.
   * Removal of stop words (e.g., "and," "the") to focus on meaningful words.
   * Vectorization using techniques such as TF-IDF or Count Vectorizer to convert text into numerical features for model training.
3. **Exploratory Data Analysis (EDA):** Several analyses were conducted to understand the distribution and characteristics of the data:
   * Visualization of the number of tweets per airline and the sentiment associated with each.
   * Creation of word clouds to highlight frequently occurring words in positive, neutral, and negative tweets.
   * Analysis of airline tweets by geographical distribution to understand sentiment trends across countries.
4. **Model Development:** A Random Forest Classifier was implemented to classify tweets into the predefined sentiment categories. The model's hyperparameters were tuned using grid search or randomized search techniques to enhance its accuracy and performance. The classifier was evaluated based on metrics such as accuracy, precision, recall, and F1-score.

**2. DATA DESCRIPTION:**

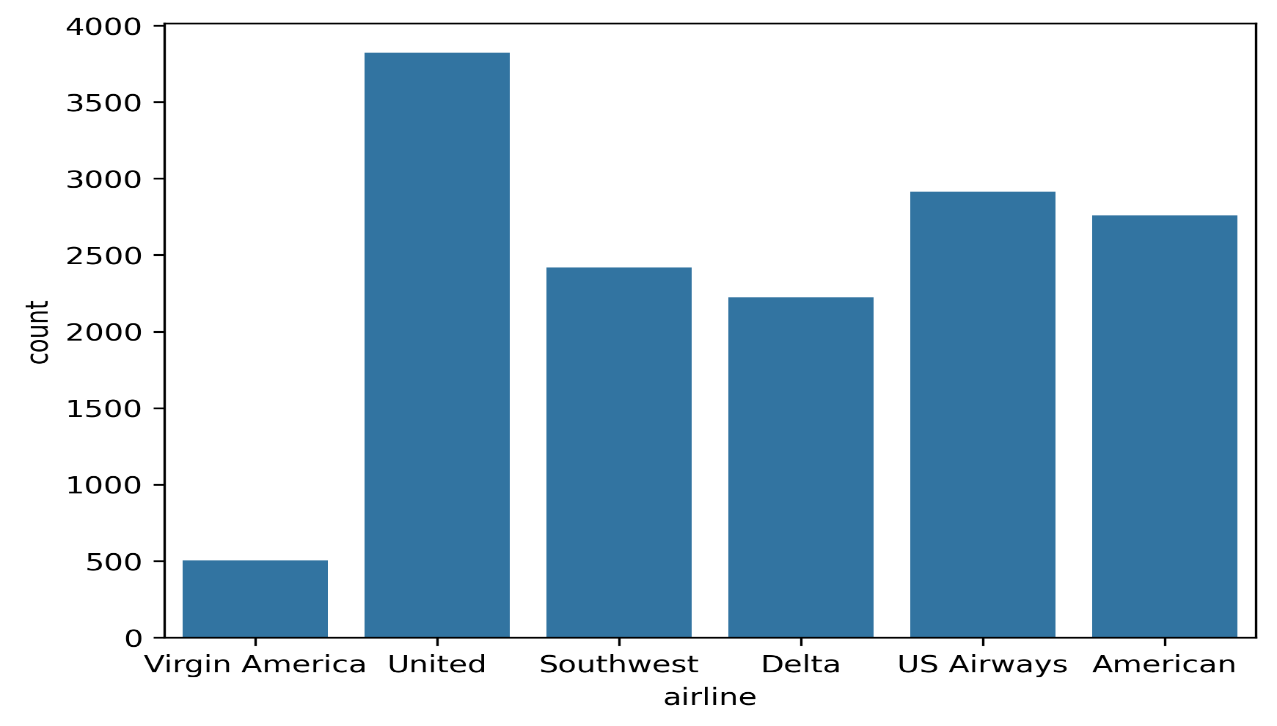
The dataset used in this project is the "Twitter Airline Sentiment" dataset available on Kaggle. Below are the key details of the dataset:

1. **Source:**
   * Kaggle project site: [**Twitter Airline Sentiment Dataset**](https://www.kaggle.com/crowdflower/twitter-airline-sentiment)**.**
   * The dataset is curated by CrowdFlower and contains tweets about major U.S. airlines.
2. **Data Structure:**
   * The dataset contains a total of approximately 14,640 tweets.
   * Each tweet is labelled with one of three sentiment categories: positive, neutral, or negative.
3. **Key Features:**
   * **tweet\_id:** Unique identifier for each tweet.
   * **airline\_sentiment:** The sentiment label (positive, neutral, or negative).
   * **airline\_sentiment\_confidence:** Confidence score of the sentiment label.
   * **airline:** Name of the airline to which the tweet is directed.
   * **name:** Name of the user who tweeted.
   * **text:** Content of the tweet.
   * **tweet\_created:** Timestamp indicating when the tweet was posted.
   * **user\_timezone:** Timezone of the user.
4. **Data Characteristics:**
   * The dataset is imbalanced, with a higher proportion of negative tweets compared to neutral and positive tweets.
   * Tweets are written in English and exhibit informal language, abbreviations, and emoticons.
5. **Preprocessing and Challenges:**
   * Tweets often include noise such as URLs, hashtags, and mentions, which were addressed during preprocessing.
   * The dataset's imbalanced nature required careful handling during model training to ensure unbiased classification.
6. **Significance of the Data:**
   * The dataset offers a rich source of textual data that reflects real-world customer experiences and opinions about airline services.
   * It provides a practical scenario for applying NLP techniques to solve sentiment classification problems.

The insights and models developed from this dataset can help airlines identify common pain points, address customer complaints effectively, and enhance their service offerings. Moreover, the dataset serves as an excellent resource for demonstrating the applications of NLP and machine learning in analyzing social media data.

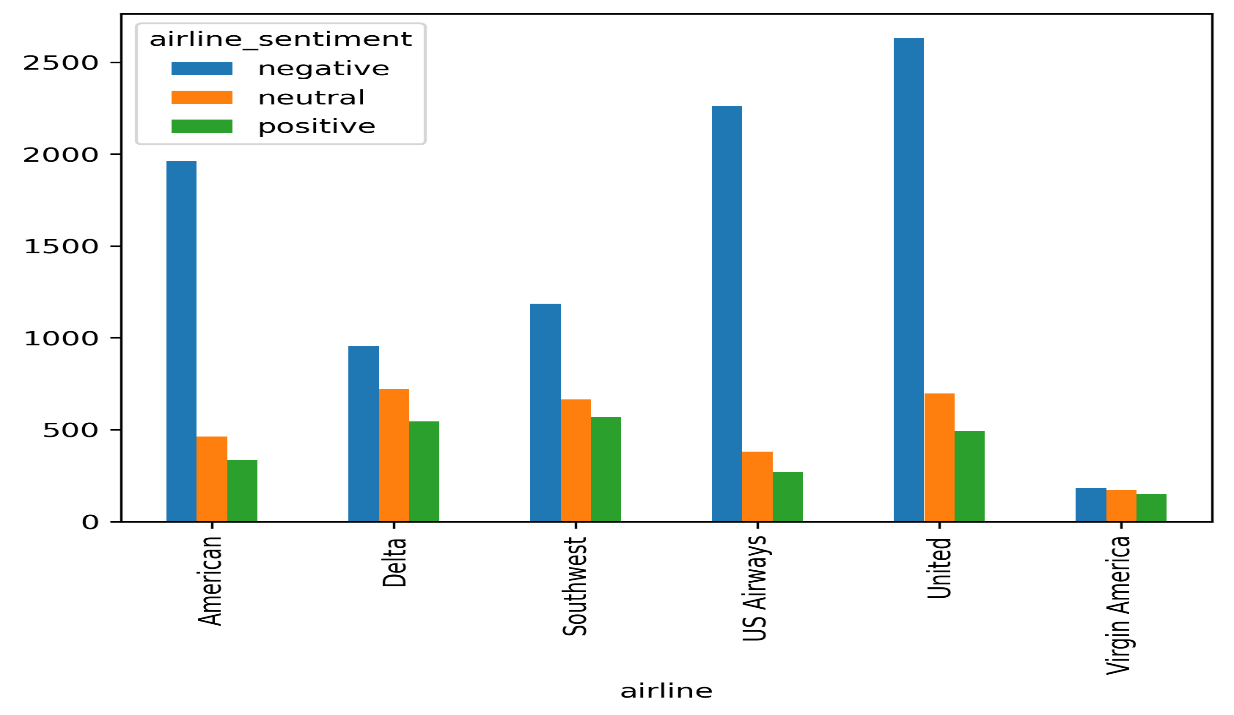
**3.ANALYTICAL INFERENCES:**

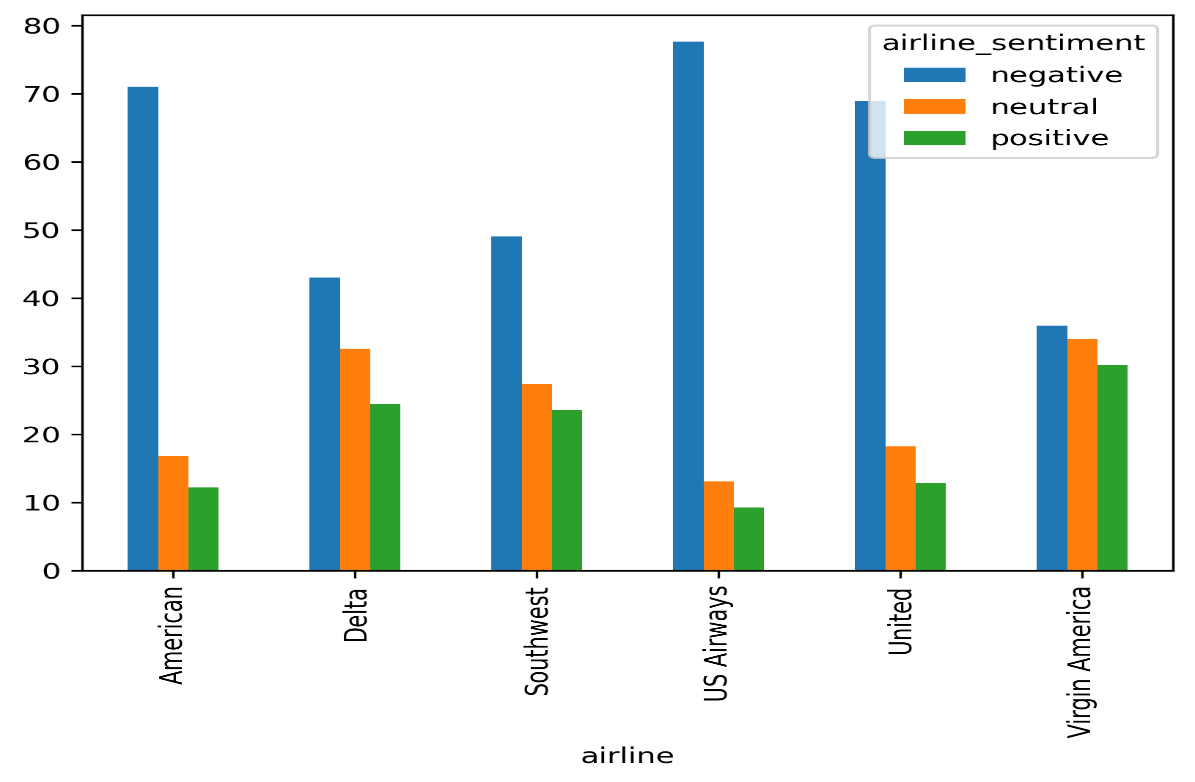
**1. Number of Tweets per Airline:**

****

* From the count plot of tweets for each airline, it was observed that **United Airlines** has the highest number of tweets, followed by **US Airways**.
* This indicates that these airlines are the most frequently discussed on Twitter, which may be due to their larger customer base or a higher volume of operational feedback.

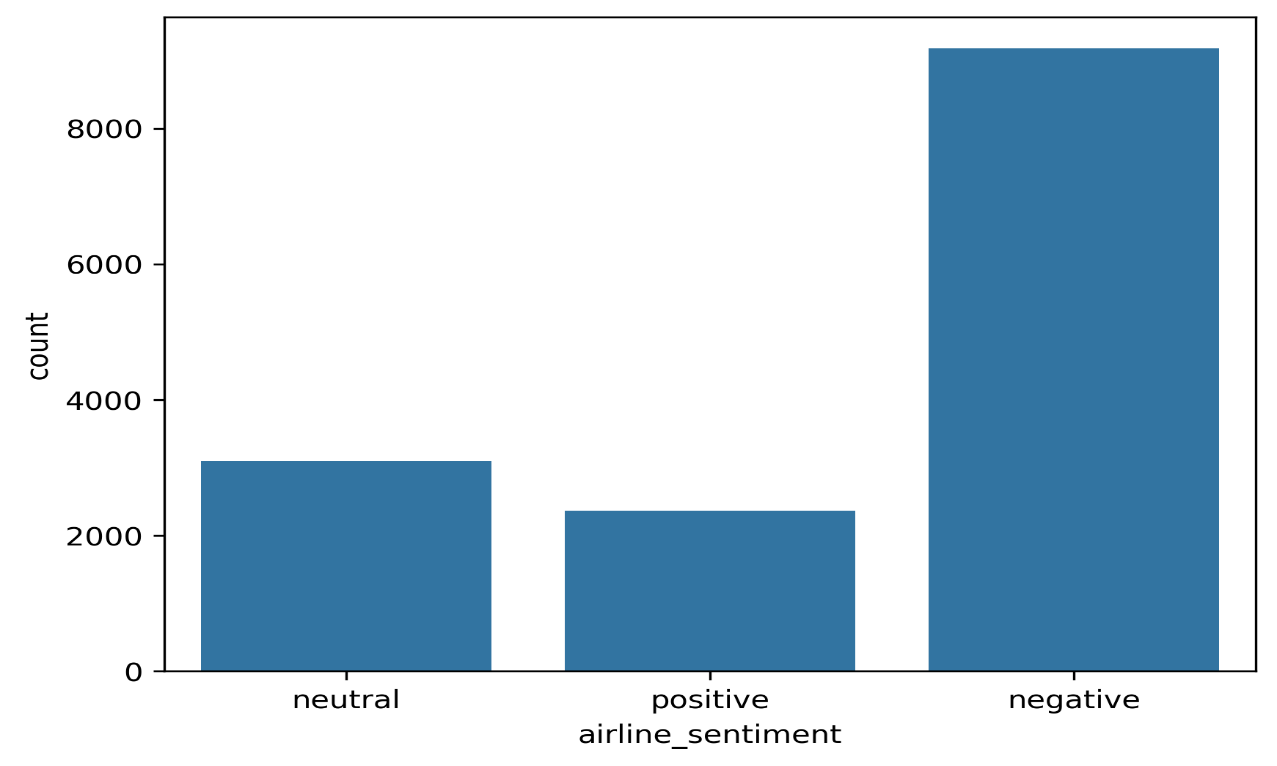
**2.Distribution of Sentiment Across All Tweets**:





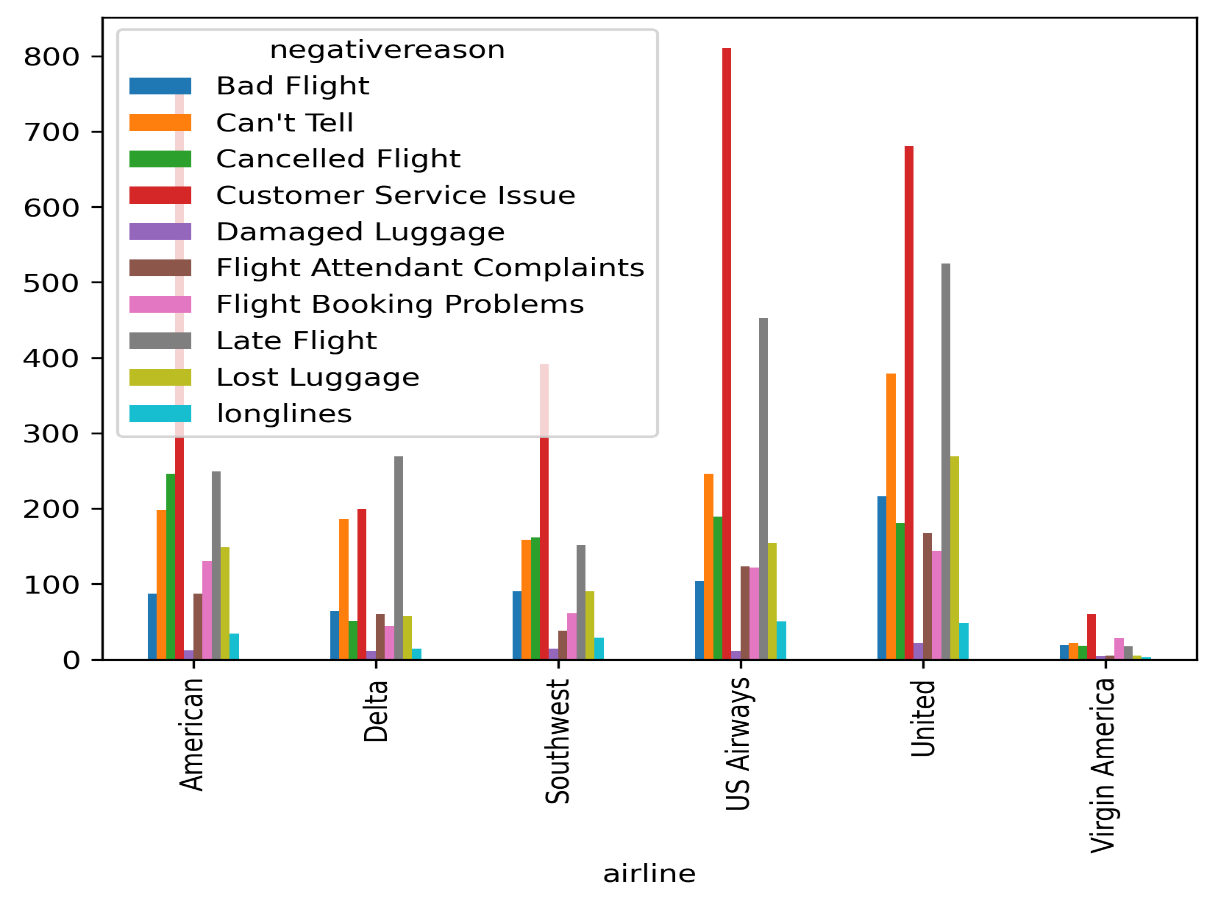
* The count plot for airline\_sentiment revealed that the majority of tweets across all airlines are **negative**, followed by **neutral** tweets, with **positive** tweets being the least frequent.
* This suggests that users are more inclined to express dissatisfaction or criticism on social media platforms like Twitter.

3. **Sentiment Distribution for Each Airline**:



* A grouped bar plot showing sentiment distribution across airlines highlighted that most airlines received a significantly higher number of negative reviews compared to neutral or positive ones.
* However, Virgin America stood out, showing a relatively balanced sentiment distribution with nearly equal proportions of positive, neutral, and negative tweets.

**4.Negative Reasons Across Airlines**:



* An analysis of the reasons for negative sentiment showed that **customer service** issues were the most frequently cited concern, followed by **late flights** and other operational challenges.
* This indicates that improving customer service could have the most significant impact on reducing negative feedback.

5. **Word Cloud Analysis**:

* **Negative Sentiment**:  
  The word cloud for negative sentiment tweets showed frequent use of words like **"cancelled," "customer," "hour,"** and **"flight"**, reflecting common complaints related to flight cancellations, customer service delays, and long waiting times.
* **Positive Sentiment**:  
  The word cloud for positive sentiment tweets featured words such as **"great," "thank,"** and **"love,"** indicating satisfaction with specific services or commendable experiences.

**4.DATA PRE-PROCESSING:**

* Data preprocessing is a crucial step in preparing textual data for analysis, particularly for natural language processing (NLP) tasks.
* The raw text often contains noise, inconsistencies, and irrelevant information that must be addressed to improve the quality of the data and ensure better performance of machine learning models.
* Below are the steps carried out for preprocessing the text data in this project:

**1. HTML Tag Removal**

* **Why**: HTML tags are often present in textual data sourced from websites or online platforms. These tags do not carry meaningful information for sentiment analysis and can hinder the processing of text.
* **How**: HTML tags were removed using regular expressions or libraries like BeautifulSoup. This step ensures the textual content is free of any markup tags and consists only of the core text.
* **Necessity**: Removing HTML tags helps clean the data and reduces noise, making the text more relevant and easier to process during subsequent steps.

**2. Number Removal**

* **Why**: Numbers in text data, unless they hold specific meaning (e.g., dates or quantities), are usually irrelevant for sentiment analysis. They can act as noise, especially in tasks like word vectorization.
* **How**: Numbers were removed using regular expressions that identify numeric patterns in the text.
* **Necessity**: Removing numbers ensures the focus remains on the words that contribute to understanding the sentiment, thereby improving the quality of feature extraction.

**3. Tokenization**

* **Why**: Tokenization is the process of breaking down text into individual words or tokens. It is essential for transforming raw text into manageable units for analysis.
* **How**: Tokenization was performed using libraries like NLTK or spaCy, which split sentences into words based on spaces and punctuation.
* **Necessity**: Tokenization is a foundational step for NLP tasks as it allows further processing like removing stop words and lemmatization to operate on individual words rather than the entire text.

**4. Stop word Removal**

* **Why**: Stop words (e.g., "and," "is," "the") are common words that do not add significant meaning to the analysis. Removing them helps focus on the meaningful words that contribute to the sentiment.
* **How**: Stop words were removed using predefined lists provided by libraries like NLTK or spaCy.
* **Necessity**: Eliminating stop words reduces the dimensionality of the text data and improves the computational efficiency of vectorization techniques.

**5. Lemmatization**

* **Why**: Lemmatization reduces words to their base or root form (e.g., "running" becomes "run"), ensuring consistency and reducing redundancy in the text data.
* **How**: Lemmatization was performed using libraries like NLTK or spaCy, which consider the context of the word to generate its lemma.
* **Necessity**: Lemmatization ensures that words with the same meaning but different forms are treated as the same, thereby reducing dimensionality and improving model performance.

**5.VECTORIZATION TECHNIQUES:**

After preprocessing, the textual data must be converted into numerical form for machine learning algorithms to process it. Vectorization techniques like Count Vectorizer and TF-IDF Vectorizer are used for this purpose.

**1. Count Vectorizer**

* **What**:  
  Count Vectorizer converts text data into a sparse matrix of token counts. Each word is assigned a unique index in a vocabulary, and the matrix represents the frequency of each word in a given document.
* **How**:
  + The text is tokenized, and each unique word is indexed.
  + A sparse matrix is generated where each row corresponds to a document and each column corresponds to a word, with the cell value representing the word’s count in the document.
  + This can be implemented using libraries like scikit-learn.
* **Necessity**:
  + Simple and interpretable representation of text data.
  + Useful when word frequencies are directly correlated with the target variable (e.g., certain words are highly indicative of sentiment).
  + However, it does not consider the importance of words across the entire dataset, leading to limitations in handling very common or rare words.

**2. TF-IDF Vectorization**

* **What**:  
  Term Frequency-Inverse Document Frequency (TF-IDF) is an improvement over Count Vectorizer. It evaluates the importance of a word in a document relative to the entire dataset.
  + **Term Frequency (TF)**: Measures how frequently a term appears in a document.
  + **Inverse Document Frequency (IDF)**: Measures how unique or rare a term is across all documents.

**How**:

* Similar to Count Vectorizer, but includes the IDF weighting.
* Implemented using libraries like scikit-learn.

 **Necessity**:

* Balances the influence of frequent and rare words by reducing the weight of common terms and emphasizing the importance of unique terms.
* Ideal for datasets where some words (e.g., “flight,” “airline”) occur frequently but are not necessarily indicative of sentiment.
* Produces a more nuanced representation of text compared to raw word counts.

**Importance of Pre-processing and Vectorization:**

1. **Data Preprocessing:**
   * Ensures that the textual data is clean, consistent, and relevant.
   * Reduces noise and redundancy, improving the quality of features extracted for model training.
2. **Vectorization:**
   * Converts text into numerical form, which is essential for machine learning algorithms.
   * Enables algorithms to process and analyse textual data efficiently.
   * Techniques like TF-IDF improve the representation of text by capturing the significance of words in the context of the dataset.

By performing these steps systematically, the raw text data is transformed into a format suitable for building accurate and robust sentiment classification models.

**MODEL BUILDING, EVALUATION AND HYPER PARAMETER TUNING:**

**Random Forest Classifier for Sentiment Analysis:**

After preprocessing the textual data and transforming it into numerical vectors using Count Vectorizer and TF-IDF Vectorizer, a Random Forest Classifier was employed to classify the tweets into three sentiment categories: positive, neutral, and negative. Below is a detailed explanation of the model implementation, reasoning, and outcomes.

**Why Random Forest?**

1. **Multi-Class Classification**:
   * Random Forest is inherently capable of handling multi-class classification tasks. Since the target variable in this project involves three classes (positive, neutral, negative), Random Forest is a natural choice.
2. **Ensemble Learning**:
   * Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their outputs to make a final prediction. This approach reduces the risk of overfitting and enhances the model’s generalization capabilities.
3. **Robustness**:
   * It is robust to outliers and noise, making it well-suited for text data, where the input can be sparse and noisy.
4. **Interpretability**:
   * While complex, Random Forest provides mechanisms to understand feature importance, which can offer insights into the most influential words for predicting sentiments.
5. **Performance**:
   * Random Forest delivers high accuracy and precision due to its ensemble nature, which is essential for tasks like sentiment analysis where prediction quality is critical.

**Model Implementation:**

1. **Separate Models for Count Vectorizer and TF-IDF Vectorizer**:
   * **Count Vectorizer Model**:  
     A Random Forest Classifier was trained on the feature matrix generated using Count Vectorizer. This matrix represents the frequency of words in each tweet.
   * **TF-IDF Vectorizer Model**:  
     Another Random Forest Classifier was trained on the feature matrix produced by TF-IDF Vectorizer, where the word importance was adjusted using the term frequency-inverse document frequency metric.
2. **Hyperparameter Tuning**:
   * Hyperparameters like the number of decision trees (n\_estimators), the maximum depth of each tree (max\_depth), and the minimum samples required to split a node (min\_samples\_split) were tuned to optimize the model's performance.
   * Grid Search Cross-Validation was used to identify the best combination of hyperparameters for both models.
3. **Prediction and Evaluation**:
   * Predictions were made on the test dataset using both models.
   * Metrics such as **accuracy**, **precision**, **recall**, and **F1-score** were calculated to assess the model's performance.
   * The confusion matrix was used to visualize the model's ability to classify each sentiment accurately.

**Results and Observations**

1. **Performance on Count Vectorizer**:
   * The Random Forest Classifier trained on the Count Vectorizer matrix showed good accuracy.
   * Commonly occurring words in negative, positive, and neutral tweets contributed significantly to the predictions.
2. **Performance on TF-IDF Vectorizer**:
   * The model trained on the TF-IDF feature matrix achieved slightly better performance than the Count Vectorizer model.
   * This improvement is attributed to the TF-IDF’s ability to emphasize rare yet significant words and reduce the impact of overly frequent but uninformative words.
3. **Feature Importance**:
   * The Random Forest model provided insights into feature importance, highlighting the words that were most predictive of each sentiment.
   * Words like "cancelled," "delayed," and "customer" were among the top contributors to negative sentiment, whereas words like "great" and "thank" were indicative of positive sentiment.
4. **Class Imbalance**:
   * Since the dataset had a higher proportion of negative tweets, the model performed best for this class. Balancing techniques like class weights or sampling could be explored for future improvements.

**Conclusion:**

The use of a Random Forest Classifier for sentiment analysis proved to be effective for both Count Vectorizer and TF-IDF Vectorizer feature representations. The following conclusions can be drawn:

1. **TF-IDF Vectorizer**: Slightly outperformed Count Vectorizer due to its ability to balance frequent and rare words.
2. **Model Strength**: Random Forest demonstrated robust performance, handling the multi-class nature of the problem effectively.
3. **Feature Insights**: The ability to extract feature importance helped understand which words were most predictive of sentiment, providing actionable insights for further analysis.

This approach successfully classified tweets into positive, neutral, and negative sentiments while addressing the challenges of text preprocessing and multi-class classification.