Twitter Airline Sentiment Dataset Prediction Using Text Analysis

COURSE PRESENTER

(DR. Omaima Fallatah)

Submitted by:

Name	ID
Manar Ali Alsubhi	444003523
Jana Abdulraouf Allihyani	444001382

DEPARTMENT OF (INFORMATION SCIENCE-DATA SCIENCE)

COLLEGE OF COMPUTER AND INFORMATION SYSTEMS

UMM AL-QURA UNIVERSITY

Table of content

1. Introduction

Sentiment analysis is a crucial natural language processing (NLP) task that involves determining whether a piece of text expresses a positive, negative, or neutral sentiment, building a simple Naive Bayes model with Exploratory Data Analysis (EDA), model training, evaluation, and visualization using the provided dataset.

- ## Goals of the Study

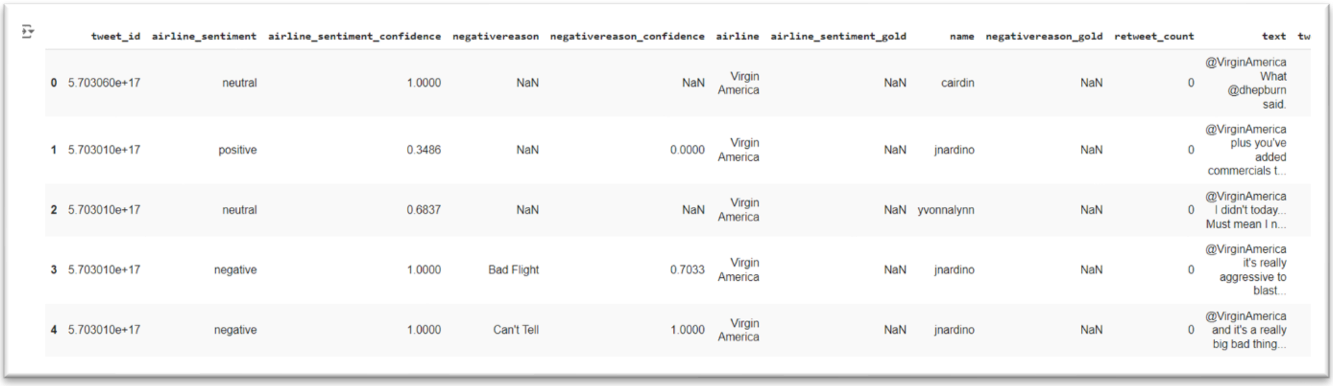
1. **Understand User Sentiment:** Analyze how customers feel about airlines based on their tweets.
2. **Predict Trends:** Use sentiment analysis to anticipate changes in customer opinions over time.
3. **Enhance Marketing Strategies:** Determine positive feedback to leverage in marketing campaigns.

2.ExploratoryDataAnalysis

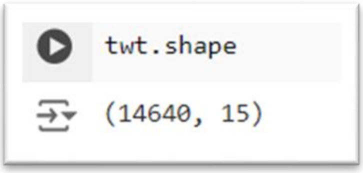
Dataset contains 15 columns.

1. tweet_id
2. airline_sentiment
3. airline_sentiment_confidence
4. negativereason
5. negativereason_confidence
6. airline
7. airline_sentiment_gold
8. name
9. negativereason_gold
10. retweet_count
11. text
12. tweet_coord
13. tweet_created
14. tweet_location
15. user_timezone

The dataset:



	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name	negativereason_gold	retweet_count	text	tw
0	5.703060e+17	neutral	1.0000	NaN	NaN	Virgin America	NaN	cairdin	NaN	0	@VirginAmerica What @dhepburn said.	
1	5.703010e+17	positive	0.3486	NaN	0.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica plus you've added commercials t...	
2	5.703010e+17	neutral	0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn	NaN	0	@VirginAmerica I didn't today... Must mean I n...	
3	5.703010e+17	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica it's really aggressive to blast...	
4	5.703010e+17	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino	NaN	0	@VirginAmerica and it's a really big bad thing...	



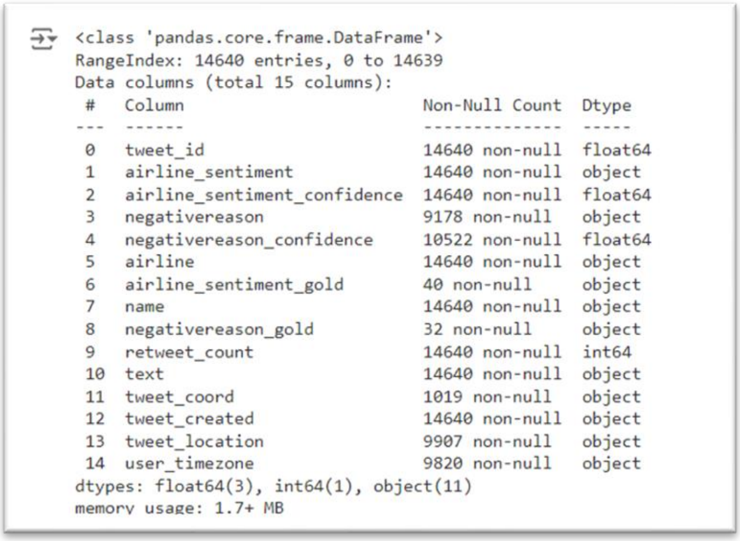
```
twt.shape
```

```
(14640, 15)
```

Number of colums:15

Number of rows:14640

- This shows how many rows have actual data (non-missing values) for each column:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   tweet_id                      14640 non-null  float64
 1   airline_sentiment             14640 non-null  object
 2   airline_sentiment_confidence  14640 non-null  float64
 3   negativereason                9178 non-null   object
 4   negativereason_confidence     10522 non-null  float64
 5   airline                       14640 non-null  object
 6   airline_sentiment_gold        40 non-null     object
 7   name                          14640 non-null  object
 8   negativereason_gold           32 non-null     object
 9   retweet_count                 14640 non-null  int64
 10  text                          14640 non-null  object
 11  tweet_coord                   1019 non-null   object
 12  tweet_created                 14640 non-null  object
 13  tweet_location                9907 non-null   object
 14  user_timezone                 9820 non-null   object
dtypes: float64(3), int64(1), object(11)
memory usage: 1.7+ MB
```

tweet_id, airline_sentiment, and text have no missing values (14,640).

negativereason has only 9,178 non-null values, which means there are missing values in that column.

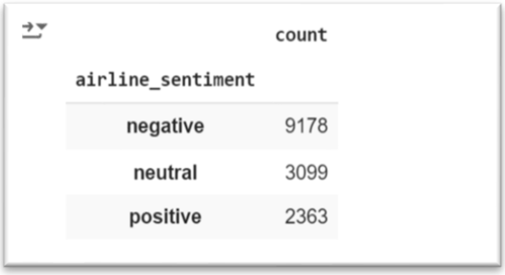
the data type of the information in each column:

- float64 is for numbers with decimals.
- int64 is for whole numbers.
- object is typically used for text data or mixed data types.

- The image shows the result of running value_counts ()

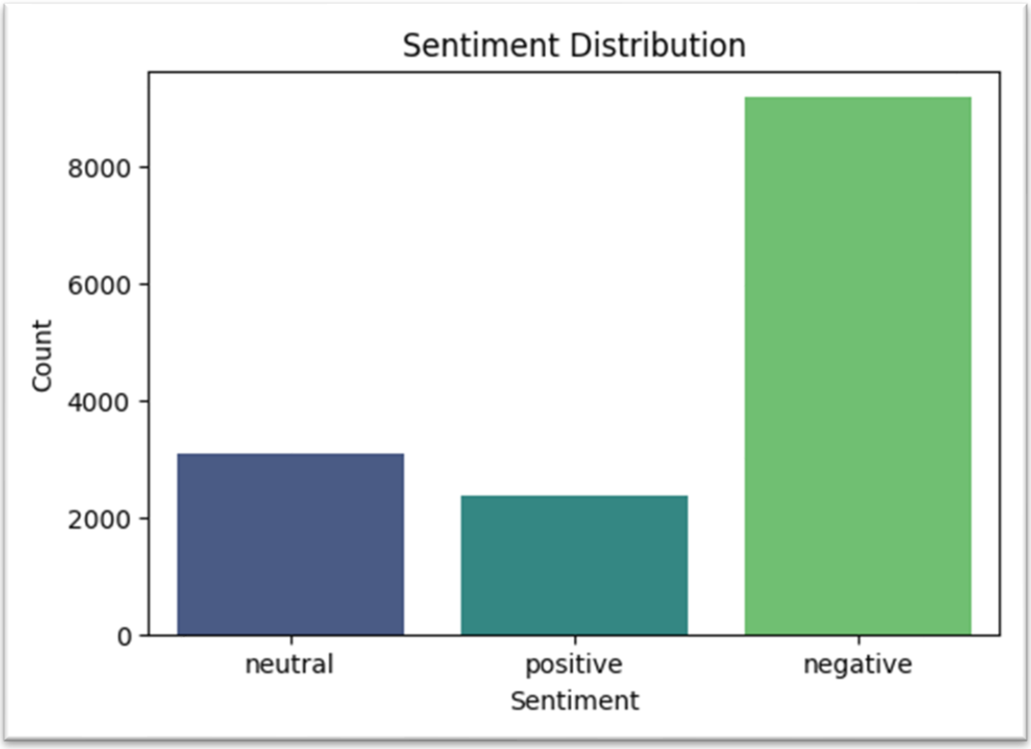
This means that most of the tweet in the dataset express

Negative sentiments about the airlines

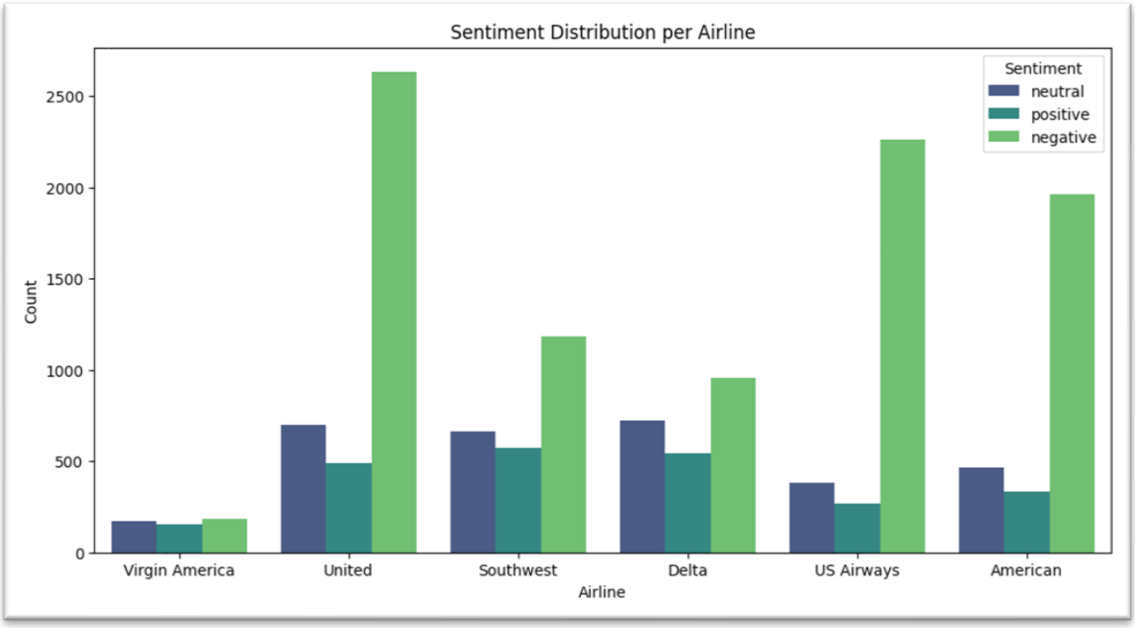


	count
airline_sentiment	
negative	9178
neutral	3099
positive	2363

Sentiment Distribution



Sentiment Distribution per Airline

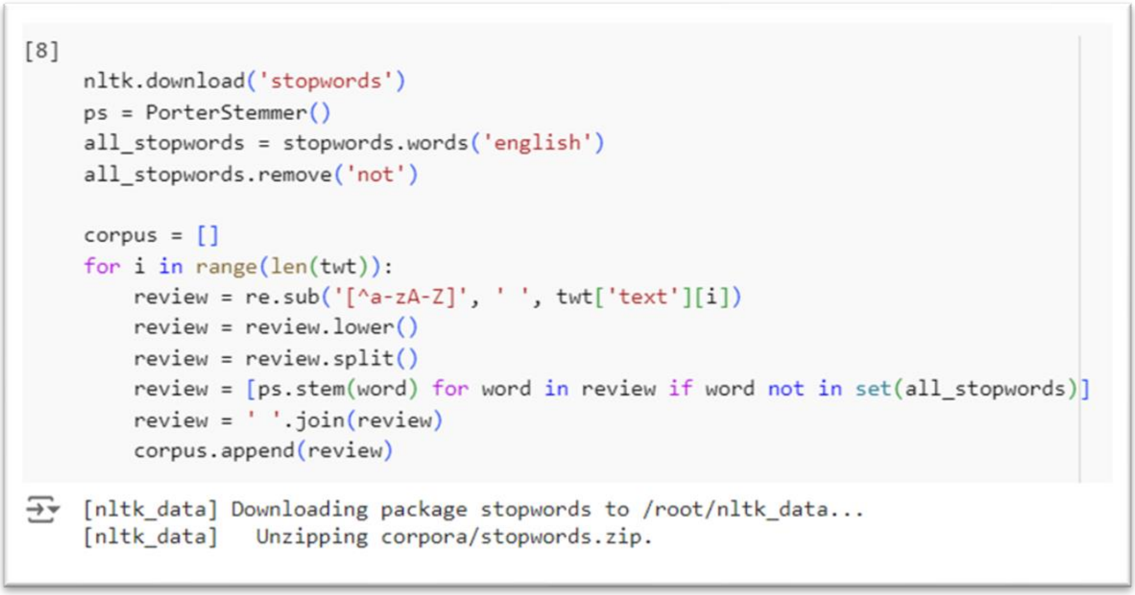


The chart helps to visually compare the sentiment for each airline

United has the highest number of negative tweets, indicating that it received a lot of criticism.

Data preprocessing:

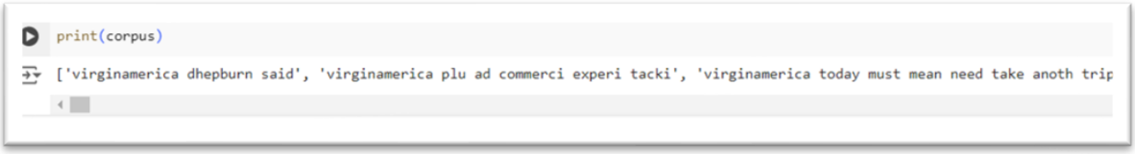
Preprocessing is crucial for improving the performance of NLP models. We'll clean the text data by removing URLs, mentions, special characters, stopwords, and perform lemmatization.



```
[8]
    nltk.download('stopwords')
    ps = PorterStemmer()
    all_stopwords = stopwords.words('english')
    all_stopwords.remove('not')

    corpus = []
    for i in range(len(twt)):
        review = re.sub('[^a-zA-Z]', ' ', twt['text'][i])
        review = review.lower()
        review = review.split()
        review = [ps.stem(word) for word in review if word not in set(all_stopwords)]
        review = ' '.join(review)
        corpus.append(review)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

This code cleans and preprocesses text data, making it easier to analyze by removing unnecessary words, converting text to lowercase, and reducing words to their root form.



```
print(corpus)
```

```
['virginamerica dhepburn said', 'virginamerica plu ad commerci experi tacki', 'virginamerica today must mean need take anoth trip
```

□ Each tweet has been converted to lowercase.

□ Stopwords (like "and," "the," "is") have been removed, except for "not," which was deliberately kept.

□ Words have been stemmed to their root form

□ Punctuation and non-alphabet characters have been removed.

Bag Of Words (BOF)

Collection of text documents into a numerical matrix. Each row of the matrix corresponds to a document, and each column represents the frequency of a specific word (from the top 1,500 words). This numerical format is often used as input for machine learning models.

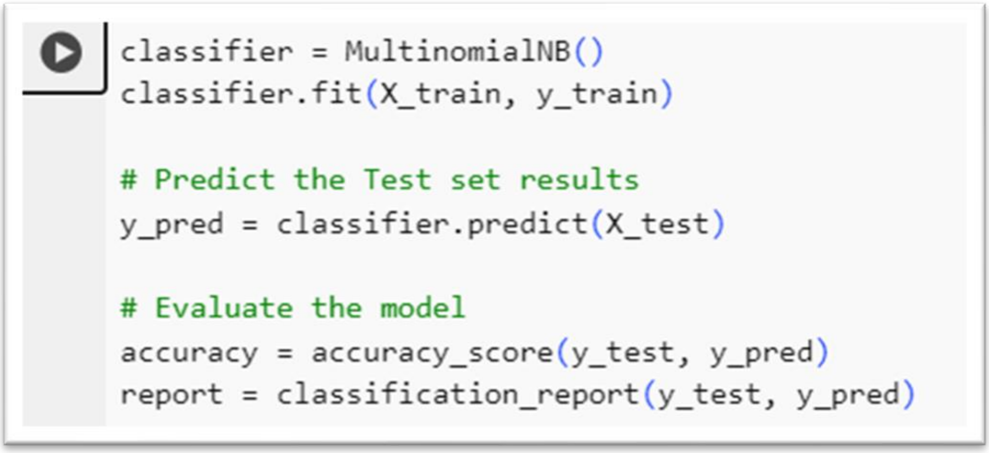
Splitting the Dataset

The cleaned text is now ready for further analysis, such as training a machine learning model

We divide the dataset into a training set to train (80% of the data) model and a testing (20% of the data). set to evaluate the model’s performance.

Model Training

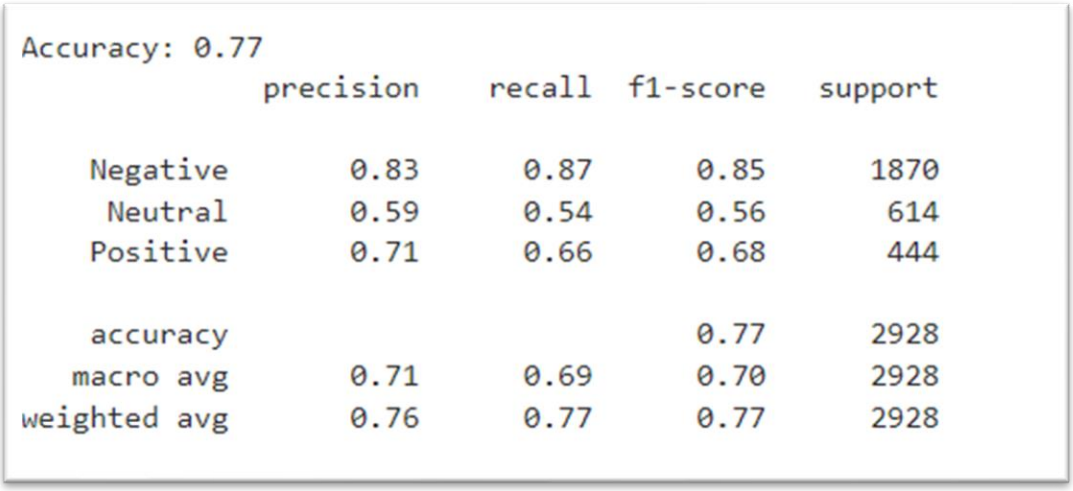
Train the Naive Bayes Model:



```
classifier = MultinomialNB()
classifier.fit(X_train, y_train)

# Predict the Test set results
y_pred = classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
```

build and evaluate a machine learning model for classification tasks using the Multinomial Naive Bayes algorithm.

Evaluate the Model

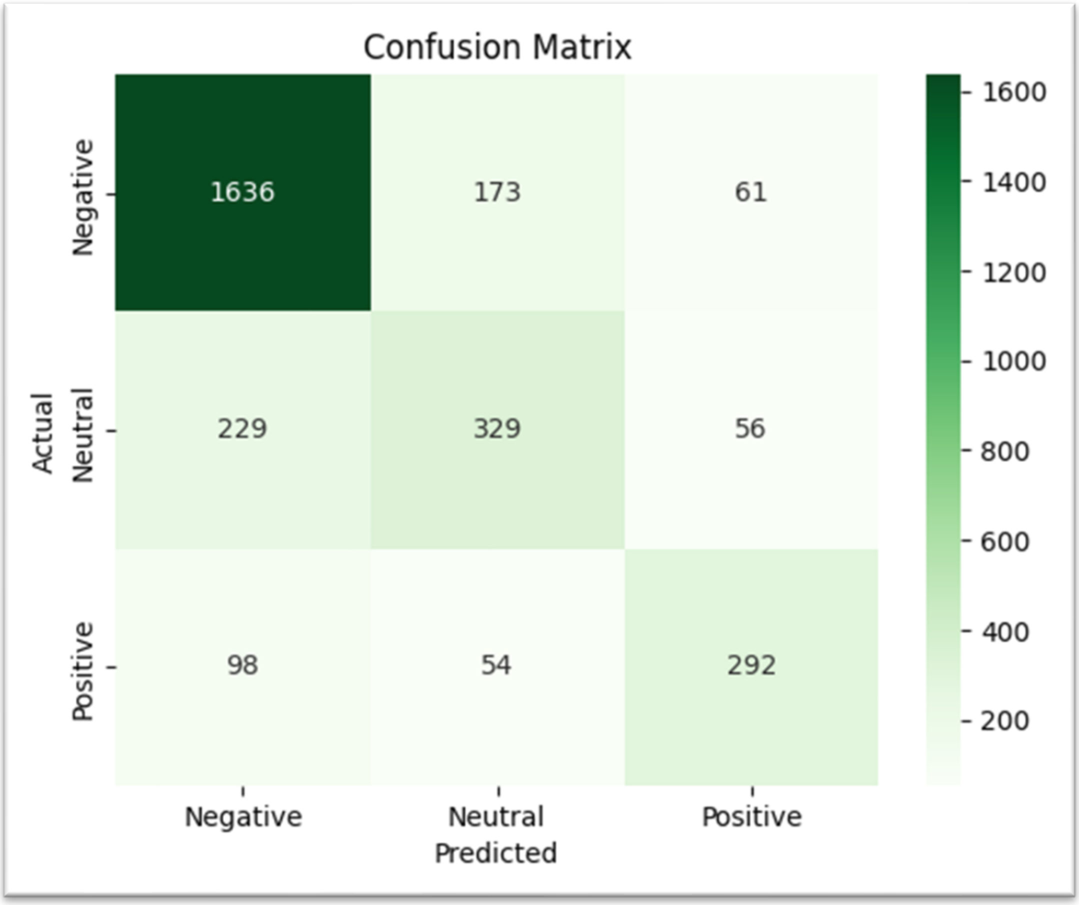


```
Accuracy: 0.77
              precision    recall  f1-score   support

    Negative       0.83      0.87      0.85      1870
     Neutral       0.59      0.54      0.56       614
    Positive       0.71      0.66      0.68       444

    accuracy                           0.77      2928
   macro avg       0.71      0.69      0.70      2928
weighted avg       0.76      0.77      0.77      2928
```

Insights

- 77% of the total predictions are correct.
- The model performs best in the "negative" class.
- Performance on the "neutral" class is lower compared to others.

Confusion Matrix Breakdown



Rows: Represent the actual classes.

Columns: Represent the predicted classes.

Negative Class: High accuracy in predicting negative samples.

Neutral Class: More confusion with the negative class, indicating it's harder to classify

Positive Class: Moderate confusion with negative samples, but better distinction from the neutral class.

Conclusion

The model performs well overall, particularly with the negative class, but there's room for improvement in distinguishing neutral samples.