# INSY 5377-001 Web and Social Analytics

**Project Report – Dec 12, 2023**

**Twitter Airline Sentiment Analysis**

**Professor**

**Riyaz Sikora**

## Team Members


Rafia Fasih

Varun Milind Joshi

Safoor Naureen

Rishitha Reddy Chinnareddy gari

# Contents

# Table of Figures

## 1.Background:

In today's digital age, where social media plays an increasingly prominent role in shaping public opinion, it is crucial for businesses to stay abreast of online conversations about their brands. Twitter, with its vast user base and real-time nature. Twitter has become a valuable source of data for businesses, including airlines, to understand customer sentiment. By implementing Twitter airline sentiment analysis, businesses can gain valuable insights into how customers perceive their services, identify areas for improvement, and proactively address potential issues.

Sentiment analysis is a technique used to extract and classify opinions from text, and it can be applied to Twitter data to gauge how customers feel about an airline's products, services, and overall experience.

Twitter sentiment analysis provides businesses with a comprehensive overview of customer sentiment, enabling them to understand the overall perception of their brand. This includes identifying trending topics, analyzing common feedback, and gauging customer satisfaction levels. By tracking sentiment over time, businesses can monitor the impact of new initiatives, marketing campaigns, and customer service interactions.

- **Operational Efficiency:** Understanding customer sentiments allows airlines to optimize their operations by identifying main points and streamlining processes, leading to increased operational efficiency and cost-effectiveness.

- **Crisis Preparedness:** sentiment analysis as a tool for helping airlines anticipate & manage potential PR crises by identifying emerging negative sentiments & allowing for timely and effective communication strategies.

- **Brand Reputation Management:** Actively manage and respond to sentiments on social media for maintaining a positive public image.

- **Data-Driven Decision-Making:** Support business decisions with quantitative insights from sentiment analysis for continuous improvement and innovation.

- **Customer Experience Enhancement:** Performing sentiment analysis on airline tweets enables businesses to gain valuable insights into customer sentiments, allowing them to proactively address issues, enhance overall customer experience, and build loyalty.

- **Competitive Positioning:** Leveraging sentiment analysis insights provides a competitive advantage by enabling airlines to differentiate themselves based on customer satisfaction, attracting new customers, and retaining existing ones in a highly competitive industry.

## 2.Data Description:

The dataset comprises tweets pertaining to airline experiences, classified into three sentiment classes: positive, negative, and neutral. It includes key information such as tweet text, sentiment labels, and details about the respective airlines. Notably, the tweets in this dataset date back to 2015, raising considerations about the relevance of sentiment analysis due to potential shifts in the airline industry's sentiments over time. It has 14640 rows and 15 columns.

| Sno | Column | Description |
|---|---|---|
| 1 | Tweet id | Unique identifier for each tweet in dataset. |
| 2 | Airline sentiment | The sentiment classification of the tweet, indicating whether it is positive, negative, or neutral. |
| 3 | Airline sentiment confidence | The confidence level associated with the sentiment classification |
| 4 | Negative reason | If the sentiment is negative, this column provides the reason for the negativity. |
| 5 | Negative reason confidence | The confidence level associated with the negative reason classification. |
| 6 | Airline | The name of the airline mentioned in the tweet. |
| 7 | Airline sentiment gold | Additional sentiment information, possibly used for gold standard labelling or annotation. |
| 8 | Name | The name of user who posted the tweet. |
| 9 | Negative reason gold | Additional information related to the negative reason, possibly used for gold standard labelling or annotation. |
| 10 | Retweet count | The count of retweets for the tweet. |
| 11 | Text | The actual text context of the tweet. |
| 12 | Tweet coord | Coordinates associated with the tweets if possible. |
| 13 | Tweet created | The timestamp indicating when the tweet was created. |
| 14 | Tweet location | The location information provided by the user in their twitter profile. |
| 15 | User Time zone | The time zone of the user who posted the tweet. |

| | |
|---|---|
| tweet_id | 5.68265E+17 |
| airline_sentiment | negative |
| airline_sentiment_confidence | 0.924 |
| negativereason | Late Flight |
| negativereason_confidence | 0.4904 |
| airline | American |
| airline_sentiment_gold | negative |
| name | beaubertke |
| negativereason_gold | Late Flight |
| retweet_count | 0 |
| text | @AmericanAir Okay, I think 1565 has waited long enough for a gate at DFW... |
| tweet_coord | |
| tweet_created | 2/18/2015 20:25 |
| tweet_location | Texas |
| user_timezone | Central Time (US & Canada) |

*Figure 1*

## 3.Research Questions:

- 1.**Sentiment Distribution:** What is the distribution of sentiment (positive, negative, neutral) within tweets discussing airline experiences?

- 2.**Airline Performance:** How do different airlines compare in terms of sentiment, revealing patterns of consistently positive or negative customer feedback?

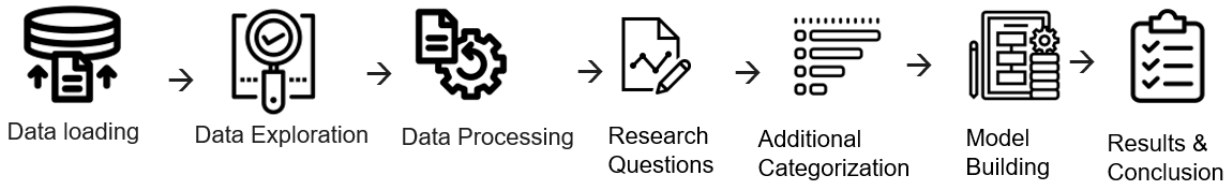
- 3.**Temporal Trends:** Are there discernible temporal trends in sentiment, such as seasonality or evolving sentiments over time, offering insights into changing customer perceptions?

- 4.**Key Topics:** What are the predominant topics or issues highlighted in negative tweets concerning airlines, shedding light on the common concerns among customers?

- 5.**Sentiment Impact:** To what extent does sentiment correlate with key engagement factors like retweets, providing an understanding of the impact of sentiment on social media engagement?

- 6.**Geographic Variation:** Is there noticeable geographic variation in sentiment, with specific regions demonstrating more positive or negative sentiments toward airlines, offering insights into regional perceptions and experiences?

## 4.Methodology:




## 5. Data Loading and Data Preprocessing:

### i. Data loading:

In the initial stages of our data analysis and machine learning project, we commenced by importing essential Python libraries to facilitate various aspects of our investigation. These included Pandas, a versatile library for data manipulation and analysis, Seaborn for visually appealing statistical graphics, NumPy for efficient numerical operations, scikit-learn for

machine learning tools, and NLTK, the Natural Language Toolkit, which is specifically designed for processing and analyzing human language data.

Following the library imports, we proceeded to load our dataset using the read_csv() function from Pandas, a crucial step in preparing the groundwork for our analysis. It is imperative to replace the placeholder 'your_dataset.csv' with the actual filename or URL of our dataset, and additional parameters within the read_csv() function were adjusted based on the specific format of our data.

To provide a preliminary understanding of the dataset, we displayed the initial rows using the head() function. This allowed us to inspect the structure and contents of the data, gaining insights into its composition. Subsequently, we gathered informative details about the dataset, such as data types and potential missing values, utilizing the info() function.

Descriptive statistics of the numerical columns were computed to gain a comprehensive summary of central tendencies, dispersion, and distribution shapes.

As a part of data quality assessment, we conducted an examination for missing values in the dataset using the isnull().sum() function. Identifying and addressing missing values is crucial for ensuring the reliability of subsequent analyses and modeling processes.

data loading

```
df = pd.read_csv('Tweets.csv') #February of 2015 data
print(df.shape)
df.head()
```

```
(14640, 15)
```

| | tweet_id | airline_sentiment | airline_sentiment_confidence | negativereason | negativereason_confidence | airline | airline_sentiment_gold | name | negativereason_gold | retweet_count |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 570306133677760513 | neutral | 1.0000 | NaN | NaN | Virgin America | NaN | cairdin | NaN | 0 |
| 1 | 570301130888122368 | positive | 0.3486 | NaN | 0.0000 | Virgin America | NaN | jnardino | NaN | 0 |
| 2 | 570301083672813571 | neutral | 0.6837 | NaN | NaN | Virgin America | NaN | yvonnalynn | NaN | 0 |
| 3 | 570301031407624196 | negative | 1.0000 | Bad Flight | 0.7033 | Virgin America | NaN | jnardino | NaN | 0 |
| 4 | 570300817074462722 | negative | 1.0000 | Can't Tell | 1.0000 | Virgin America | NaN | jnardino | NaN | 0 |

*Figure 2*

## ii. Data processing:

In the process of refining our dataset for more focused analysis, we systematically eliminated extraneous columns that did not significantly contribute to our investigative goals. These columns were identified based on their limited correlation with other variables, as determined through a correlation matrix analysis. This strategic removal streamlines our dataset, ensuring that only relevant and influential features are retained for subsequent analyses, thereby enhancing the efficiency and interpretability of our findings.

Moreover, to enhance the consistency and relevance of our location data, we undertook a modification of the 'tweet_location' column. This involved the removal of prefixes and suffixes, resulting in the retention of only the core location information. This step aims to standardize the representation of locations, facilitating a more uniform and meaningful analysis.

Addressing missing values is a critical aspect of data preprocessing. In our dataset, specifically for the 'tweet_location' and 'user_timezone' columns, we opted for an imputation strategy by assigning

the value 'Unknown' to instances where data was initially absent. This imputation ensures that these variables remain usable in subsequent analyses, preventing the loss of potentially valuable information.

Furthermore, to enrich our temporal analysis, we derived two new columns, namely 'tweet_hour' and 'tweet_date', from the existing 'tweet_created' column. Extracting the hour and date components allows for a more granular examination of temporal patterns within the dataset. This transformation provides valuable insights into the temporal dynamics of the tweets, enabling us to uncover trends or patterns associated with specific times of the day and dates.

## 6.Data Visualization and Research Questions Solutions:

### i. Data visualization using PowerBI


*Figure 3*

- Jet Blue News was the user with highest tweets in the dataset with 63 tweets.

- Most tweets were negative i.e., around 63% of total tweets.
- 95% of the users did not retweet about their airline experiences, they only shared once.
- Customer Service issue was the most highlighted negative issue among the tweets.
- Highest number of tweets were about United Airlines.


*Figure 4*

American Airlines: 71% of the tweets on this airlines were negative with around 2-3% of users trying to retweet. The main reasons for negative tweets being customer service issues followed by Late flights and cancelled flights with highest number of tweets from Pekin.


*Figure 5*

Delta Airlines: Only 43% of the tweets on this airlines were negative with around 5% of users trying to retweet. The main reasons for negative tweets being Late flights followed by customer service issues.

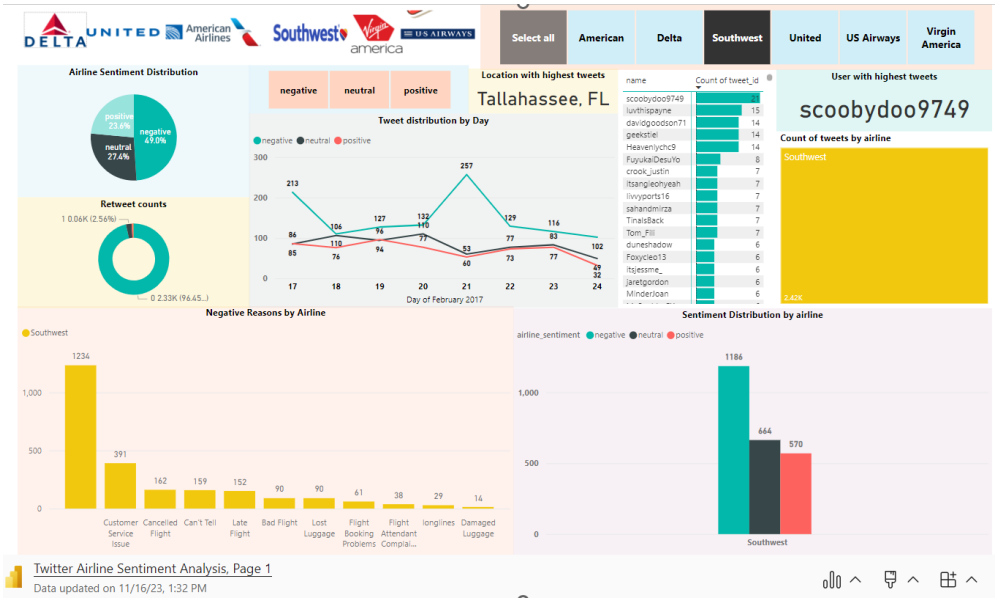



*Figure 6*

Southwest Airlines: almost half (49%) of the tweets on this airlines were negative with around 4% of users trying to retweet. The main reasons for negative tweets being customer service issues followed by Cancelled Flights, with highest number of tweets from Tallahassee, FL.


*Figure 7*

United Airlines: 69% of the tweets on this airlines were negative with around 10% of users trying to retweet. The main reasons for negative tweets being customer service issues followed by Late Flights.

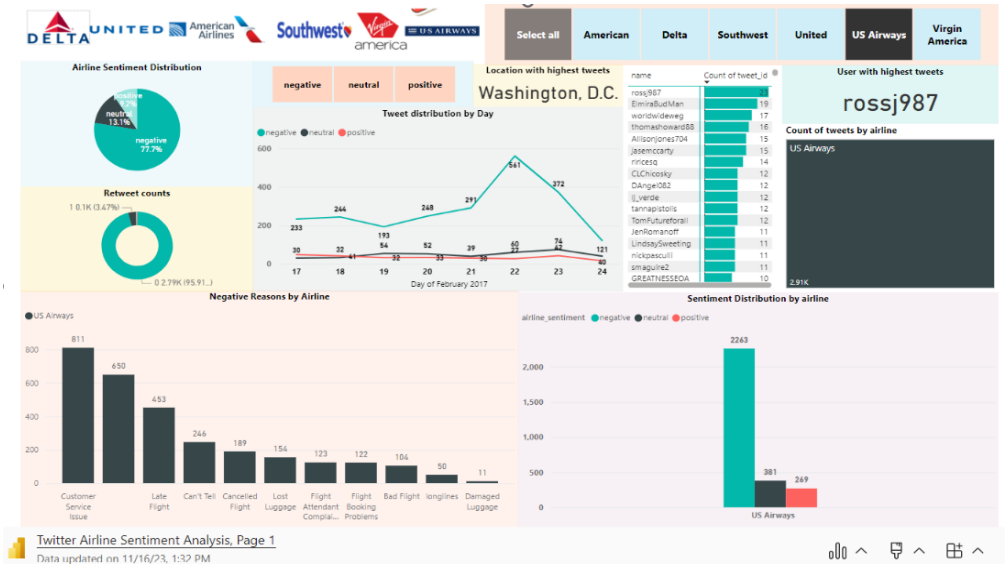



*Figure 8*

US Airways: 77.7% of the tweets on this airlines were negative with around 5% of users trying to retweet. The main reasons for negative tweets being customer service issues followed by Late Flights with highest number of tweets from Washington DC.


*Figure 9*

Virgin America Airlines: 36% of the tweets on this airlines were negative with around 4% of users trying to retweet. The main reasons for negative tweets being customer service issues followed by Flight booking problems.

**ii.Research Questions Solutions:**

- 1.**Sentiment Distribution:** What is the distribution of sentiment (positive, negative, neutral) within tweets discussing airline experiences? And 2. **Airline Performance:** How do different airlines compare in terms of sentiment, revealing patterns of consistently positive or negative customer feedback?

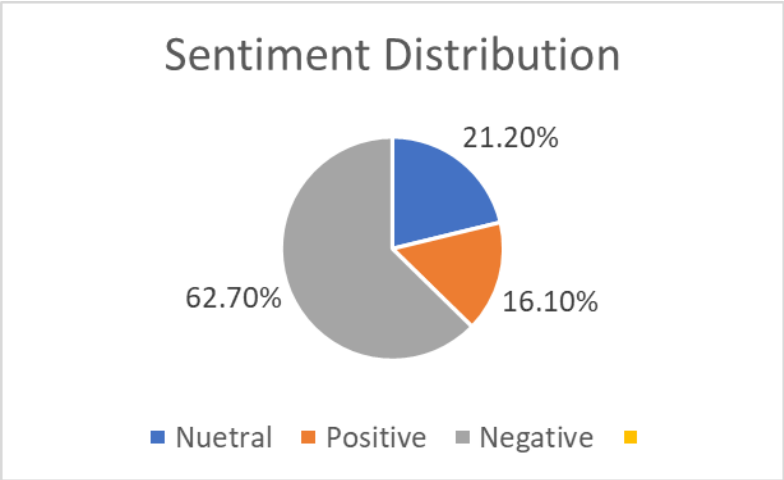



*Figure 10*

Users tried to share their negative experiences more, 62.7% of total tweets were negative, followed by 21.2% neutral and 16.1% of positive tweets.

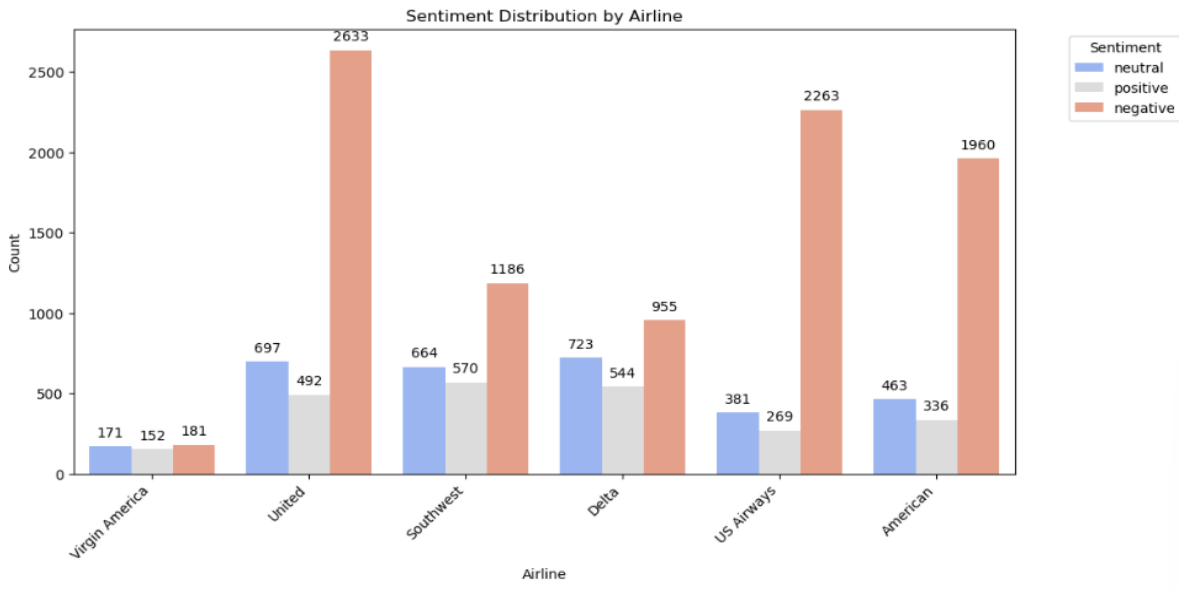



*Figure 11*

For all the airlines, negativev airlines were highly reported with United airlines(2633) having the highest number of Negative tweets followed US airways(2263). Southwest airlines(570) followed by Delta Airlines(544) had the highest number of positive tweets.

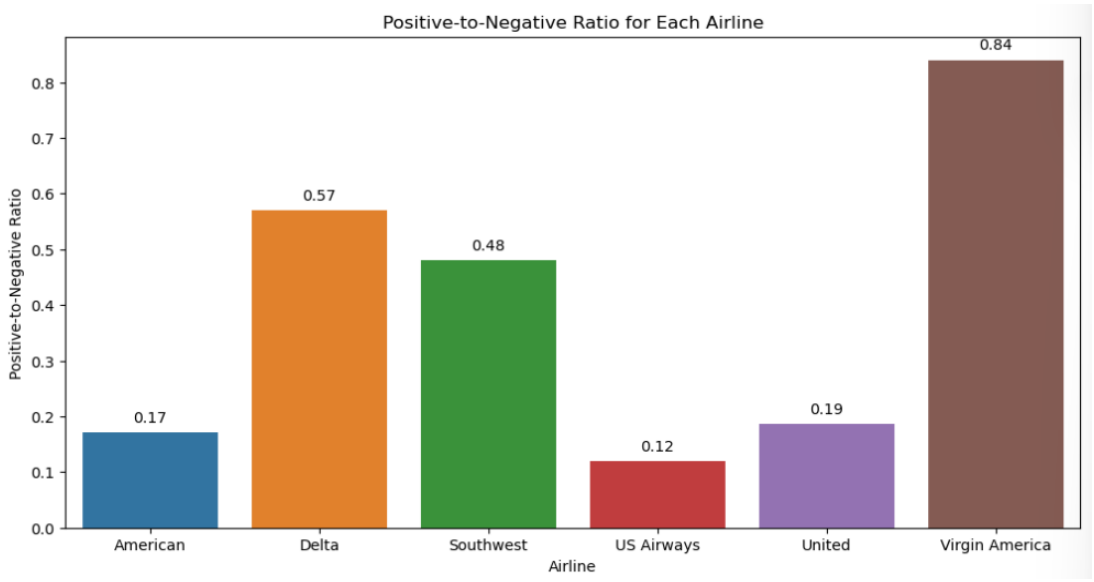



*Figure 12*

Virgin America had the highest positive to negative tweet ratio. US Airways had the lowest positive-to-negative tweet ratio.

- 3.**Temporal Trends:** Are there discernible temporal trends in sentiment, such as seasonality or evolving sentiments over time, offering insights into changing customer perceptions?

Daily:

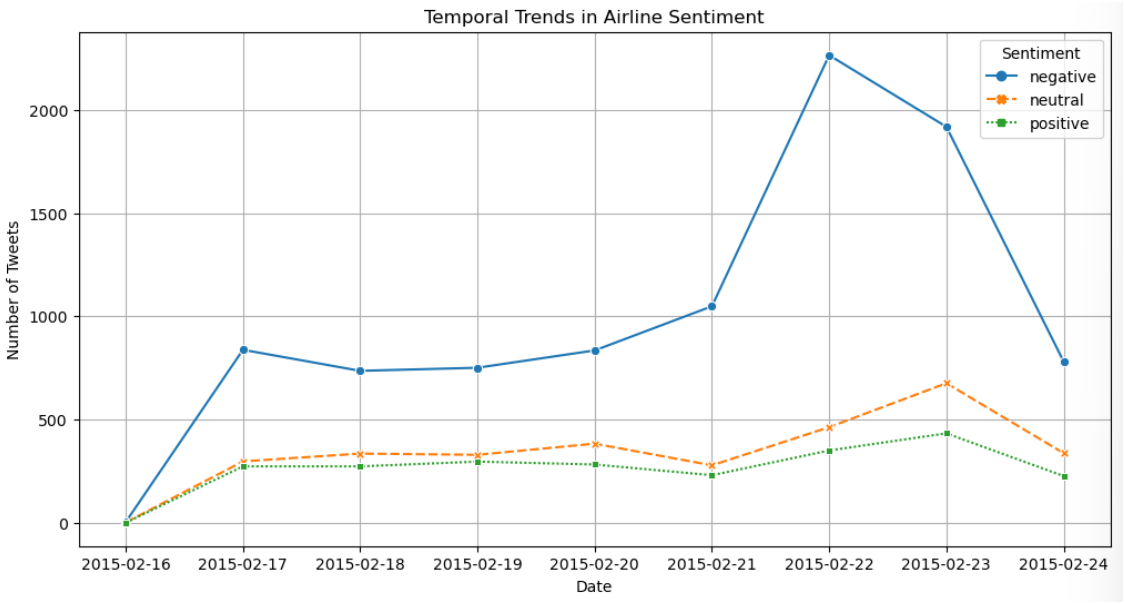



*Figure 13*

22nd and 23rd of February reported the highest number of tweets posted and these dates were on the weekend, which shows more tweets were posted on the weekend.

Hourly:

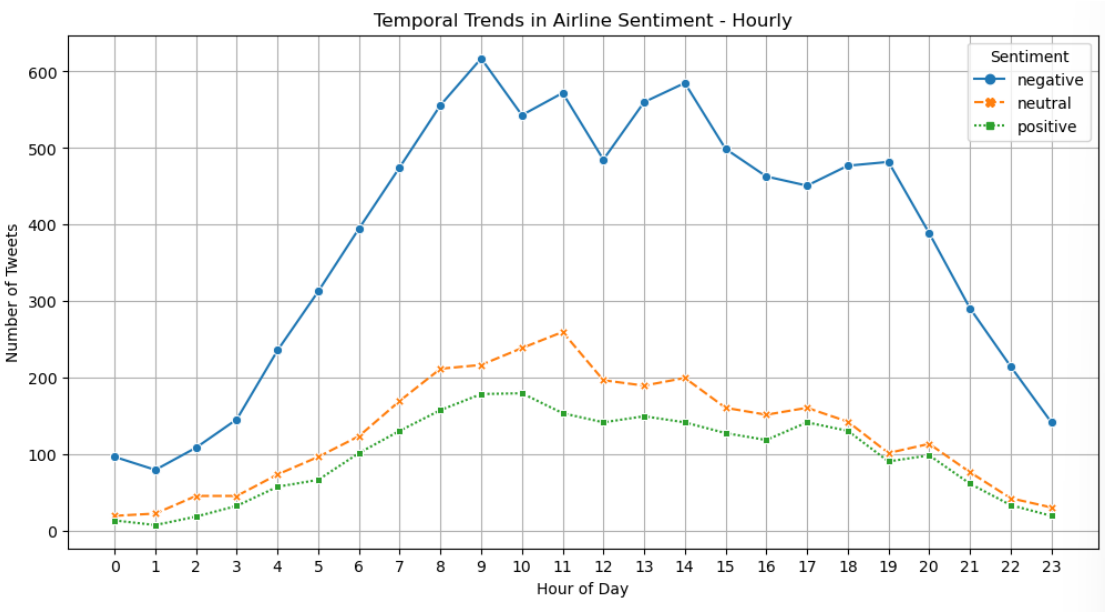



*Figure 14*

Users tried to post more tweets during day time, being more active 9:00AM to 2:00PM

4.**Key Topics:** What are the predominant topics or issues highlighted in negative tweets concerning airlines, shedding light on the common concerns among customers?


*Figure 15*

Users tried to report various issues, mainly customer service issues, Late Flights, Lost Luggage, Booking Problems.

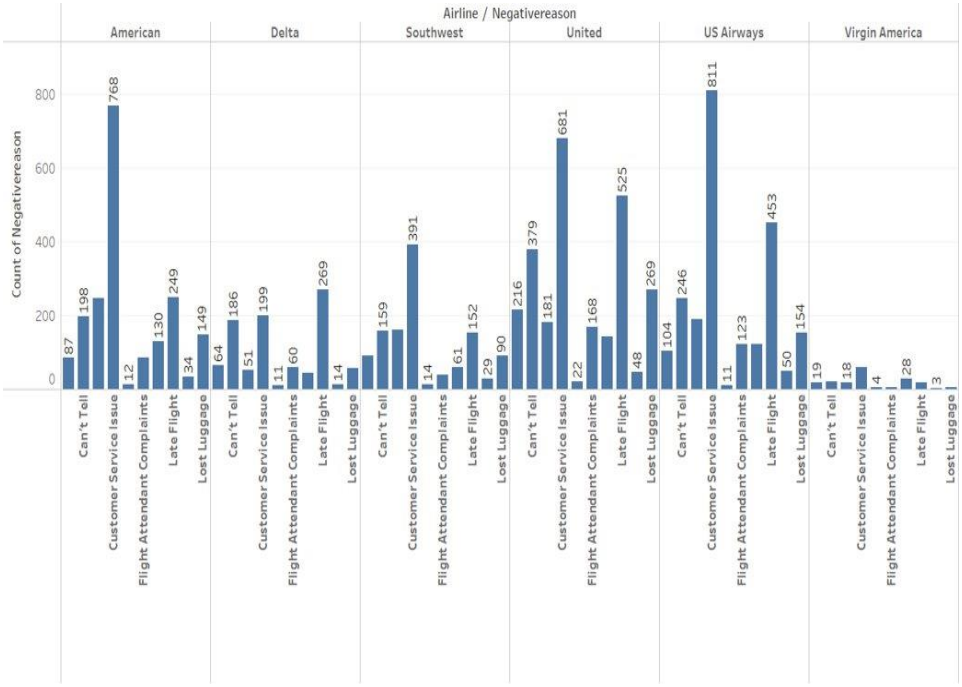



*Figure 16*

All most all the airlines as per the tweets commonly had customer service issues, followed by late flights and cancelled flights.

Retweet Analysis:

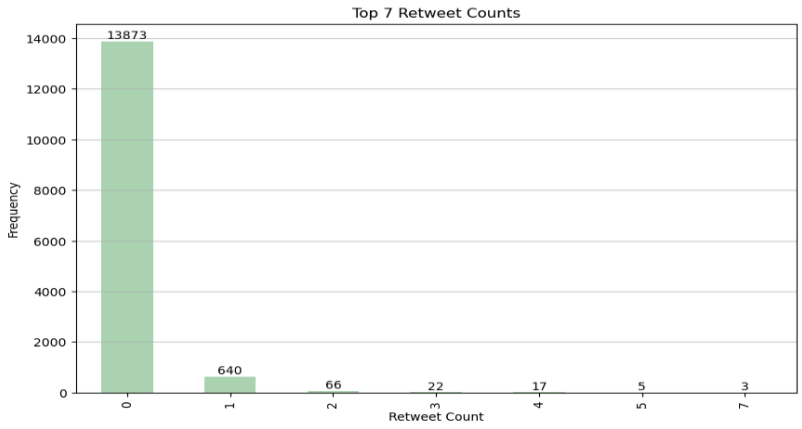



*Figure 17*

Most of the users only shared their experience once, with very less users around 800 were retweeting their experiences.

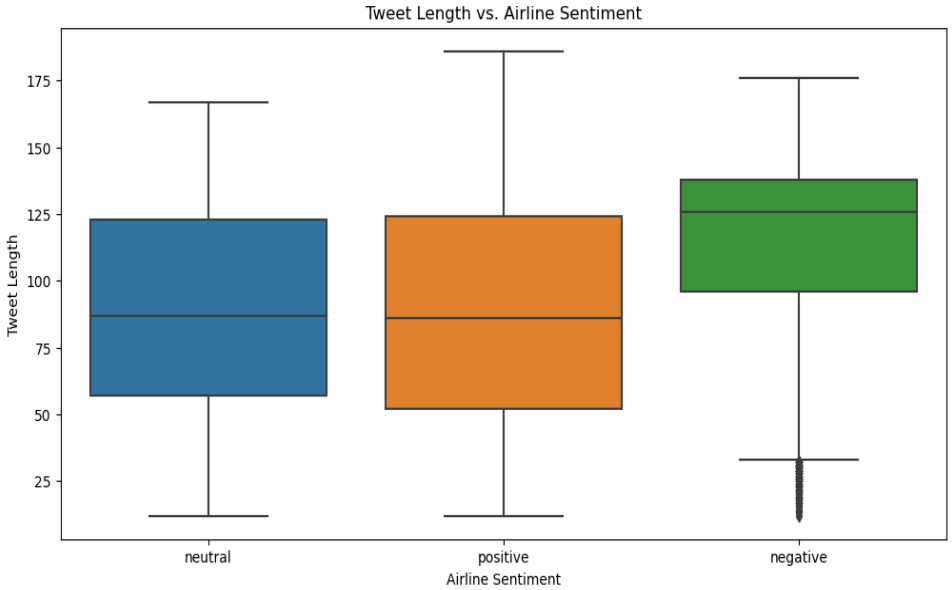
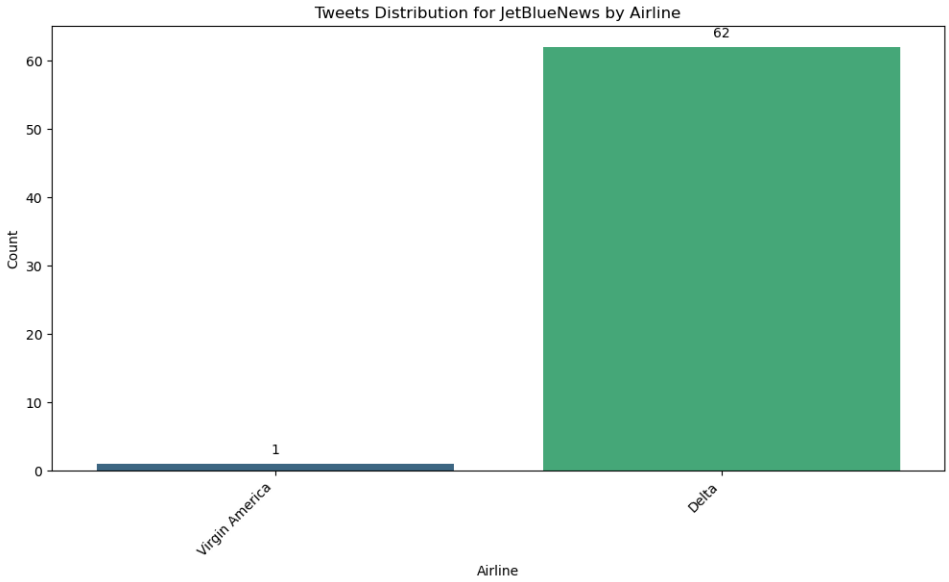



*Figure 18*

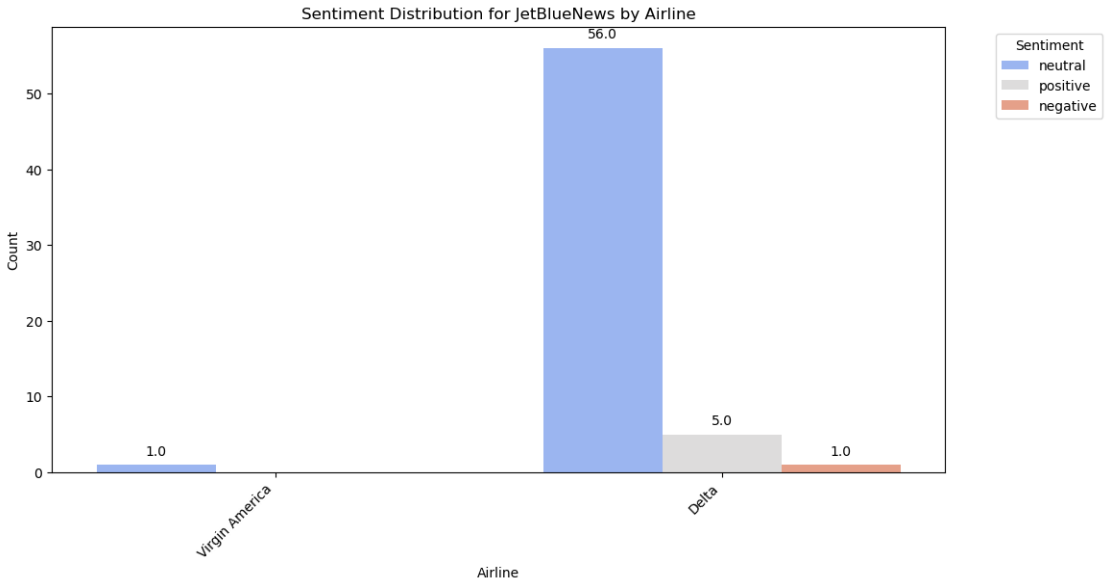
Negative sentiment tweets were comparatively longer, which means users tried to explain their negative experiences in more words/detail.

User with highest number of tweets:




The user with highest tweet count was JetBlueNews which mostly covered updates on Delta Airlines.

Distribution of tweets of user with the highest tweet count:



Being a media outlet, most of it tweets were neutral as seen from the bar chart above.

- 6.**Geographic Variation:** Is there noticeable geographic variation in sentiment, with specific regions demonstrating more positive or negative sentiments toward airlines, offering insights into regional perceptions and experiences?

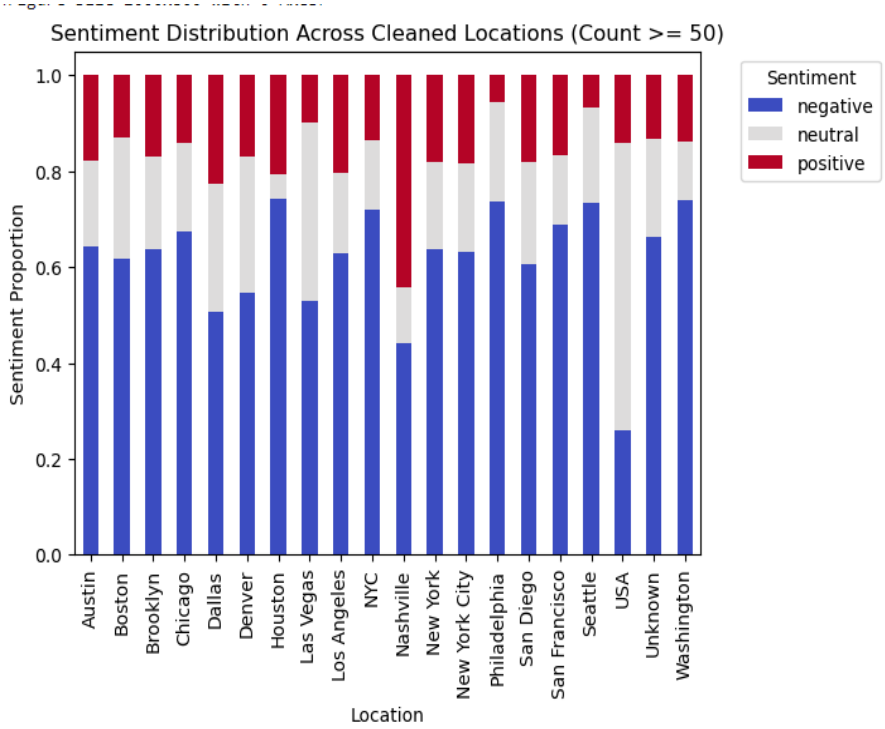



*Figure 19*

Highest number of negative tweets were from Washington, Houston, Philadelphia and Seattle. Highest number of Positive tweets were from Nashville.

- 5.**Sentiment Impact:** To what extent does sentiment correlate with key engagement factors like retweets, providing an understanding of the impact of sentiment on social media engagement?

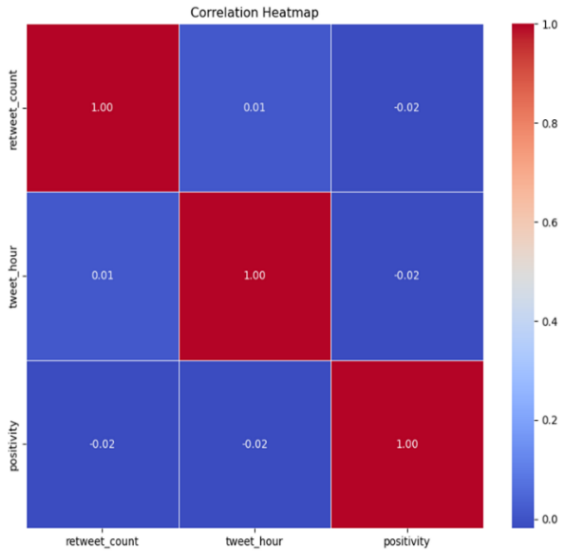



*Figure 20*

Sentiment of the tweet were not much correlated to retweets or time of the tweets.

**Additional Categorization:** Initially the dataset had three categories of sentiments(Positive, Negative and Neutral), we have diversified the sentiments to 5 categories with respect to the initial sentiment and confidence of that sentiment. We made the below 5 conditions to recategorize the tweets to 5 sentiments.

```
# Define conditions
conditions = [
    (df['airline_sentiment'] == 'negative') & (df['airline_sentiment_confidence'] >= 0.8),
    (df['airline_sentiment'] == 'negative') & (df['airline_sentiment_confidence'] < 0.8),
    (df['airline_sentiment'] == 'neutral'),
    (df['airline_sentiment'] == 'positive') & (df['airline_sentiment_confidence'] < 0.8),
    (df['airline_sentiment'] == 'positive') & (df['airline_sentiment_confidence'] >= 0.8)
]

# Define values for each condition
values = [1, 2, 3, 4, 5]

# Create 'positivity' column based on conditions
df['positivity'] = np.select(conditions, values, default=np.nan)
df['positivity'].value_counts()
```

*Figure 21*

| | | | |
|---|---|---|---|
| 1.0 | 7392 | 3.0 | 846 |
| 3.0 | 3099 | 4.0 | 846 |
| 2.0 | 1786 | 5.0 | 846 |
| 5.0 | 1517 | 1.0 | 846 |
| 4.0 | 846 | 2.0 | 846 |

*Figure 22*

As most of the tweets were of negative sentiment, to have an unbiased dataset for sentimental analysis we took equal number of tweets in all the 5 categories.

## 7.Predictive Analytics:

### Model Building for Sentimental Analysis:

***i.Removing Stop words:*** We have removed the unnecessary symbols and stop words from the tweets to have the cleaned tweets for analysis.

```
def tweet_to_words(tweet):
    letters_only = re.sub("[^a-zA-Z]", " ", tweet)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(" ".join( meaningful_words ))
```

```
df_model["clean_tweet"] = df_model["text"].apply(lambda x: tweet_to_words(x))
df_model.head()
```

| | text | positivity | clean_tweet |
|---|---|---|---|
| 0 | @JetBlue marks 15th birthday with @Airbus #A32... | 3.0 | jetblue marks th birthday airbus painted bluma... |
| 1 | alright @JetBlue.... done! alternatively, if y... | 4.0 | alright jetblue done alternatively like charte... |
| 2 | @united appreciate the sentiment and you were ... | 4.0 | united appreciate sentiment able get ground st... |
| 3 | @united Confirmation number: NJV4BP - All I ne... | 3.0 | united confirmation number njv bp need email c... |
| 4 | @JetBlue Utah, I think. And thanks! | 5.0 | jetblue utah think thanks |

*Figure 23*

***ii.Splitting the dataset**:* we spilt the dataset into 80% training and 20% testing set.

```
1 print(X.shape, y.shape)
```

```
(4230,) (4230,)
```

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2, random_state = 42)
```

*Figure 24*

***ii.NLP Methods:***

**Count Vectorizer:**

Count Vectorizer is a text feature extraction technique commonly used in sentiment analysis. It converts text documents into numerical vectors that represent the occurrence of words in each document. This conversion allows sentiment analysis algorithms to process and analyze text data effectively.

How Count Vectorizer Works:

Count Vectorizer follows a three-step process:

- Preprocessing: The text data is preprocessed to clean and prepare it for tokenization. This may involve removing punctuation, converting text to lowercase, and removing stop words (common words that don't add much meaning to the text).

- Tokenization: The text is broken down into individual words or tokens.

- Vocabulary Building: A vocabulary of unique words is created based on the tokens. Each word is assigned a unique index.

- Document-Term Matrix: Count Vectorizer creates a document-term matrix, where each row represents a document and each column represents a word in the vocabulary. The value in each cell corresponds to the number of times the corresponding word appears in the corresponding document.

```
1 vect = CountVectorizer()
2 vect.fit(X_train)
```

```
CountVectorizer()
```

```
1 X_train_dtm = vect.transform(X_train)
2 X_test_dtm = vect.transform(X_test)
```

```
1 vect_tunned = CountVectorizer(stop_words = "english", ngram_range = (1, 2), min_df = 0.1, max_df = 0.7, max_features = 100)
2 vect_tunned
```

```
CountVectorizer(max_df=0.7, max_features=100, min_df=0.1, ngram_range=(1, 2),
                stop_words='english')
```

*Figure 25*

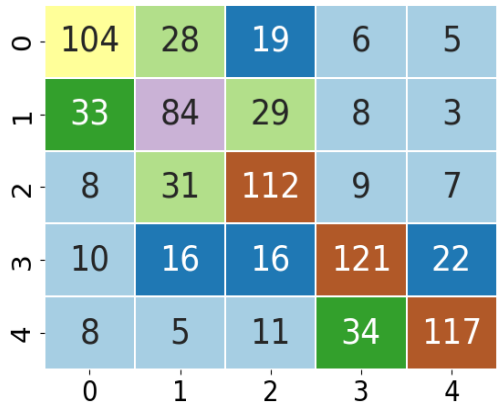



*Figure 26*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1.0 | 0.64 | 0.64 | 0.64 | 162 |
| 2.0 | 0.51 | 0.54 | 0.52 | 157 |
| 3.0 | 0.60 | 0.67 | 0.63 | 167 |
| 4.0 | 0.68 | 0.65 | 0.67 | 185 |
| 5.0 | 0.76 | 0.67 | 0.71 | 175 |
| accuracy | | | 0.64 | 846 |
| macro avg | 0.64 | 0.63 | 0.63 | 846 |
| weighted avg | 0.64 | 0.64 | 0.64 | 846 |

*Figure 27*

Count vectorizer model showed 64% accuracy.

**TF-IDF:**

Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer is a popular text feature extraction technique commonly used in sentiment analysis. It converts text documents into numerical vectors that represent the importance of words in each document. This conversion allows sentiment analysis algorithms to process and analyze text data effectively.

How TF-IDF Vectorizer Works:

TF-IDF Vectorizer follows a three-step process:

1. Preprocessing: The text data is preprocessed to clean and prepare it for tokenization. This may involve removing punctuation, converting text to lowercase, and removing stop words (common words that don't add much meaning to the text).

2. Tokenization: The text is broken down into individual words or tokens.

3. Vocabulary Building: A vocabulary of unique words is created based on the tokens. Each word is assigned a unique index.

4. Document-Term Matrix: TF-IDF Vectorizer creates a document-term matrix, where each row represents a document and each column represents a word in the vocabulary. The value in each cell corresponds to the TF-IDF weight of the corresponding word in the corresponding document.

TF-IDF Formula

The TF-IDF weight for a word in a document is calculated as follows:

TF-IDF = TF * IDF

where:

- TF (Term Frequency) is the number of times the word appears in the document.

- IDF (Inverse Document Frequency) is a measure of how important the word is to the entire corpus. It is calculated as follows:

Cross-validation: Provides a reliable way to evaluate the performance of a model with different hyperparameter settings.

GridSearchCV: Uses cross-validation to systematically evaluate a grid of hyperparameter combinations and identify the best performing set.

By combining these two techniques, you can effectively find the best hyperparameters for your machine learning model and improve its performance.

```python
from sklearn.feature_extraction.text import TfidfVectorizer

# Create and fit TF-IDF vectorizer on training data
tfidf_vect = TfidfVectorizer(stop_words='english', ngram_range=(1, 2), max_features=5000)
X_train_tfidf = tfidf_vect.fit_transform(X_train)
X_test_tfidf = tfidf_vect.transform(X_test)
```

```python
from sklearn.model_selection import GridSearchCV

# Define parameter grid for SVM
param_grid = {'C': [1], 'kernel': ['linear', 'rbf']}

# Perform grid search
grid_search = GridSearchCV(SVC(random_state=10), param_grid, cv=3, scoring='accuracy')
grid_search.fit(X_train_tfidf, y_train)

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Train SVM with the best parameters
best_model = SVC(**best_params, random_state=10)
best_model.fit(X_train_tfidf, y_train)

# Predictions
pred_tfidf = best_model.predict(X_test_tfidf)
```

```
Best Parameters: {'C': 1, 'kernel': 'rbf'}
```

```python
best_model
```

```
SVC(C=1, random_state=10)
```

*Figure 28*

```python
from sklearn.model_selection import cross_val_score

# Perform cross-validation
cv_scores = cross_val_score(best_model, X_train_tfidf, y_train, cv=5, scoring='accuracy')
print("Cross-Validation Scores:", cv_scores)
```

```
Cross-Validation Scores: [0.59231905 0.59231905 0.58493353 0.61152142 0.6183432 ]
```

*Figure 29*

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1.0 | 0.59 | 0.74 | 0.66 | 162 |
| 2.0 | 0.48 | 0.46 | 0.47 | 157 |
| 3.0 | 0.56 | 0.66 | 0.60 | 167 |
| 4.0 | 0.74 | 0.60 | 0.66 | 185 |
| 5.0 | 0.76 | 0.64 | 0.70 | 175 |
| accuracy | | | 0.62 | 846 |
| macro avg | 0.63 | 0.62 | 0.62 | 846 |
| weighted avg | 0.63 | 0.62 | 0.62 | 846 |

*Figure 30*

TF-IDF model showed 62% accuracy.

**GloVe:**

- GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm that generates word embeddings, which are vector representations of words that capture their semantic relationships. Word embeddings have become a powerful tool in natural language processing (NLP) tasks, including sentiment analysis.

How GloVe Vectorizer Works:

- GloVe Vectorizer first trains on a large corpus of text to learn the relationships between words. It then uses this information to generate vector representations for each word. These vectors are designed to capture the semantic meaning of words, such as their synonyms, antonyms, and their overall context in the corpus.

```python
# Word Embeddings (using pre-trained GloVe embeddings)
from sklearn.feature_extraction.text import TfidfVectorizer

# Load pre-trained GloVe embeddings
glove_vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(1,3), max_features=10000)
X_train_glove = glove_vectorizer.fit_transform(X_train)
X_test_glove = glove_vectorizer.transform(X_test)

# SVM with RBF kernel and tuned parameters
svm_glove = SVC(C=1, kernel='rbf', random_state=10)
svm_glove.fit(X_train_glove, y_train)

# Predictions
pred_glove = svm_glove.predict(X_test_glove)

```

```python
print("Accuracy Score: ", accuracy_score(y_test,pred_glove) * 100)
```

```
Accuracy Score:  62.4113475177305
```

*Figure 31*

GloVe model showed 62.4% accuracy.

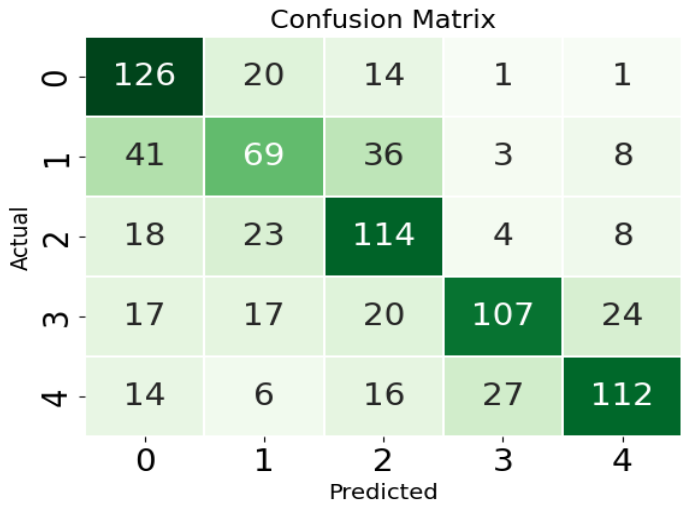



*Figure 32*

***iii.Predictions:***

```
# Sample text for prediction
sample_text = ["I really enjoyed the flight, great experience!"]

# Transform the sample text using the same vectorizer
sample_text_glove = glove_vectorizer.transform(sample_text)

# Predict positivity for the sample text
predicted_positivity = svm_glove.predict(sample_text_glove)

# Print the predicted positivity value
print("Predicted Positivity:", predicted_positivity)
```

```
Predicted Positivity: [5.]
```

```
# Sample text for prediction
sample_text = ["the flight was okay"]

# Transform the sample text using the same vectorizer
sample_text_glove = glove_vectorizer.transform(sample_text)

# Predict positivity for the sample text
predicted_positivity = svm_glove.predict(sample_text_glove)

# Print the predicted positivity value
print("Predicted Positivity:", predicted_positivity)
```

```
Predicted Positivity: [2.]
```

```
# Sample text for prediction
sample_text = ["poor service and luggage handling"]

# Transform the sample text using the same vectorizer
sample_text_glove = glove_vectorizer.transform(sample_text)

# Predict positivity for the sample text
predicted_positivity = svm_glove.predict(sample_text_glove)

# Print the predicted positivity value
print("Predicted Positivity:", predicted_positivity)
```

```
Predicted Positivity: [1.]
```

*Figure 33*

**Future work done based on suggestions:**

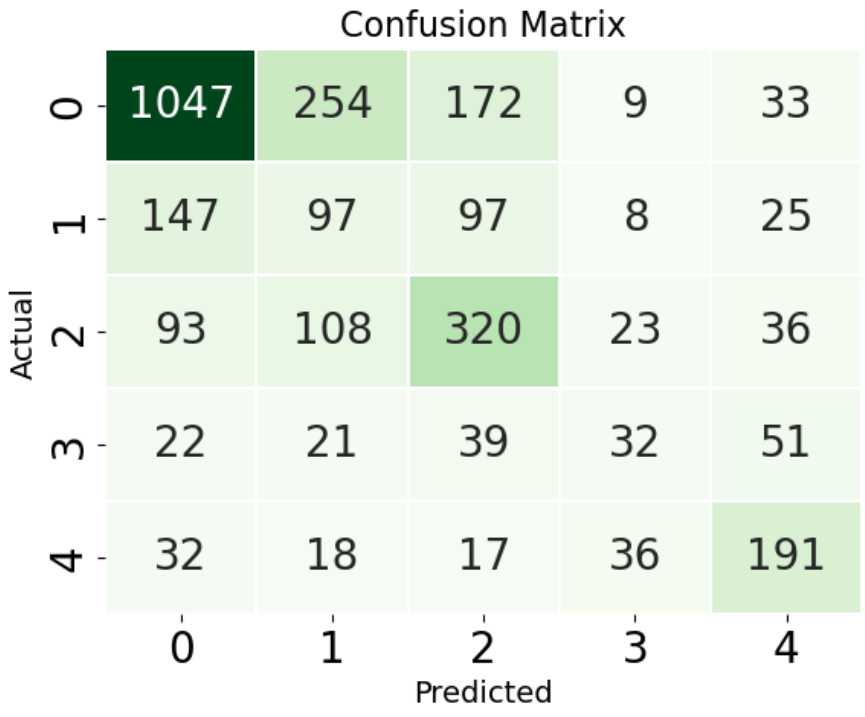
Instead of balancing the dataset and splitting for testing and training purposes, we split the data into test and train first, and then modelled the training data. That gave us the accuracy of 57.6% using the GloVe Vectorisor.

```python
# Word Embeddings (using pre-trained GloVe embeddings)
from sklearn.feature_extraction.text import TfidfVectorizer

# Load pre-trained GloVe embeddings
glove_vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(1,3), max_features=10000)
X_train_glove = glove_vectorizer.fit_transform(X_train_balanced)
X_test_glove = glove_vectorizer.transform(X_test)

# SVM with RBF kernel and tuned parameters
svm_glove = SVC(C=1, kernel='rbf', random_state=10)
svm_glove.fit(X_train_glove, y_train_balanced)

# Predictions
pred_glove = svm_glove.predict(X_test_glove)
```

```python
print("Accuracy Score: ", accuracy_score(y_test,pred_glove) * 100)
```

```
Accuracy Score:  57.61612021857923
```

And the confusion matrix for the same is as follows.



## 8.Results and Conclusion:

- Dominance of Negative Sentiments: Individuals expressing their opinions online, particularly on platforms like Twitter, tend to share negative reviews more frequently than neutral or positive ones.

- Negative sentiment ratio is higher for US Airways and American Airlines despite the higher count for United Airlines

- Enhancing positive sentiments can be leveraged for marketing and brand-building

- Addressing negative sentiments in real-time can lead to improved customer satisfaction and loyalty.

- Limitations: The data is of 2015 which cannot be applicable in the Post COVID Era; Balanced dataset also limited the size of it in entirety.

- Recommendations: Based on our findings, airlines should actively engage with customers on social media, promptly addressing concerns and showcasing positive experiences. Utilizing the developed real-time sentiment analysis tool can enhance targeted marketing efforts, leading to an improved brand perception.

## 9.Future Recommendations for Sentimental Analysis:

Improved NLP Techniques:

- Utilizing Contextual Embeddings: Traditional word embedding models like Word2Vec and GloVe may not capture the full context of words in tweets. Newer contextual embedding models like BERT, RoBERTa, and XLNet can better understand the nuances of language and provide more accurate sentiment analysis.
- Exploring Multilingual Models: Airlines operate globally, and handling diverse languages is crucial. Multilingual models like M-BERT and XLM-R can effectively analyze tweets in different languages, providing airlines with a broader understanding of customer sentiment across various regions.
- Leveraging Explainable AI (XAI): XAI techniques can help explain the reasoning behind model decisions and provide insights into why certain tweets are classified as positive, negative, or neutral. This can be valuable for airlines to understand the factors influencing customer sentiment and address specific concerns.

## 10.Acknowledgements and References:


I. **Dataset**:https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment

II. **PowerBI:** https://learn.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-types-for-reports-and-q-and-a

III. **Sklearn:** https://scikit-learn.org/stable/model_selection.html#model-selection

IV. **NLP:** https://www.kaggle.com/code/andreshg/nlp-glove-bert-tf-idf-lstm-explained

V. **Future Reccomendations:** https://www.researchgate.net/publication/341231839_Contextualized_Embeddings_based_Transformer_Encoder_for_Sentence_Similarity_Modeling_in_Answer_Selection_Task

https://medium.com/@aman.anand54321/cross-lingual-models-xlm-r-7d557302698b