Homework 4

Ishika Kulkarni
ikulkar1@ucsc.edu

Abstract

This report addresses sentiment analysis using the Twitter US Airline Sentiment dataset. We do comprehensive data analysis, text processing, tokenization, and SVM for sentiment analysis.

1 Introduction

Sentiment analysis is essential for understanding how individuals feel and behave, particularly in the service industry. This study focuses on the Twitter US Airline Sentiment dataset to examine how passengers feel about major US airlines. We aim to classify tweets into three categories: positive, negative, or neutral.

To execute this, we start with data exploration and use techniques like tokenization and cleaning to prepare our dataset. We've developed a custom tokenizer and applied various preprocessing methods to prepare the dataset for training a support vector machine (SVM) classifier.

Throughout the process, we dealt particularly with data cleaning and feature extraction, and we explored how these factors influenced the classifier's performance. Additionally, we conducted an ablation study to see how individual cleaning steps impacted the model's accuracy.

2 Dataset

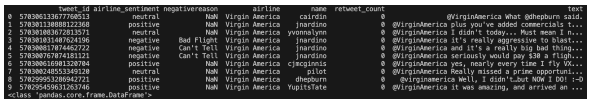
As mentioned earlier, the dataset is the Twitter US Airline Sentiment dataset, which we downloaded from Kaggle.

The dataset contains the following columns:

- tweet_id: A unique identifier for each tweet.
- airline_sentiment: The tweet's sentiment (positive, neutral, or negative).
- airline_sentiment_confidence: A confidence score indicating the certainty of the sentiment classification.
- negativereason: A detailed reason for negative sentiment (e.g., "Late Flight").
- negativereason_confidence‘: A confidence score for the classification of the negative reason.
- airline: The airline mentioned in the tweet.
- airline_sentiment_gold: Annotated sentiment for gold-standard evaluation (sparsely populated).
- name: The username of the person who tweeted.
- negativereason_gold: Annotated negative reason for gold-standard evaluation (sparsely populated).
- retweet_count: The number of retweets the tweet has received.
- tweet_coord: Geographical coordinates from where the tweet was sent (if available).
- tweet_created: The timestamp indicates when the tweet was created.
- user_timezone: The timezone of the user who tweeted.
- text: The actual text of the tweet.
- tweet_location: The location mentioned by the user in their profile.

But obviously, we did not use them all; below are the ones that we used for the task and how the dataset looks:

```
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            14640 non-null  int64
 1   airline_sentiment   14640 non-null  object
 2   negativereason      9178 non-null   object
 3   airline             14640 non-null  object
 4   name                14640 non-null  object
 5   retweet_count       14640 non-null  int64
 6   text                14640 non-null  object
dtypes: int64(2), object(5)
memory usage: 800.8+ KB
```



```
             tweet_id airline_sentiment negativereason      airline     name  retweet_count                                               text
0  570306133677760513           neutral            NaN  Virgin America   cairdin              0                 @VirginAmerica What @dhepburn said.
1  570301130888122368          positive            NaN  Virgin America  jnardino              0  @VirginAmerica plus you've added commercials t...
2  570301083672813571           neutral            NaN  Virgin America  yvonnalynn              0  @VirginAmerica I didn't today... Must mean I m...
3  570301031407624196          negative     Bad Flight  Virgin America  jnardino              0  @VirginAmerica it's really aggressive to blast...
4  570300817074462722          negative     Can't Tell  Virgin America  jnardino              0  @VirginAmerica and it's a really big bad thing...
5  570300767074181121          negative     Can't Tell  Virgin America  jnardino              0  @VirginAmerica seriously would pay $30 a fligh...
6  570300616901320704          positive            NaN  Virgin America  cjmcginnis              0  @VirginAmerica yes, nearly every time I fly VX...
7  570300248553349120           neutral            NaN  Virgin America      pilot              0  @VirginAmerica Really missed a prime opportuni...
8  570299953280942721          positive            NaN  Virgin America   dhepburn              0    @virginamerica Well, I didn't…but NOW I DO! :-D
9  570295459631263746          positive            NaN  Virgin America  YupitsTate              0  @VirginAmerica it was amazing, and arrived an ...
<class 'pandas.core.frame.DataFrame'>
```

3 Part 1

We use the relevant columns and analyze the basic data in this part.

3.1 Part A

3.2 Overall Analysis

- Total number of data samples: 14640
- Overall Unique values in airline sentiment: {neutral, positive, negative}
- Overall Unique values in negative reason: {nan, 'Bad Flight', 'Can't Tell', 'Late Flight', 'Customer Service Issue', 'Flight Booking Problems', 'Lost Luggage', 'Flight Attendant Complaints', 'Cancelled Flight', 'Damaged Luggage', 'longlines'}
- Overall Most frequent value in airline sentiment: negative
- Overall Most frequent value in negative reason: Customer Service Issue
- Overall Frequency of most frequent value in airline sentiment: 9178
- Overall Frequency of most frequent value in negative reason: 2910

3.3 Analysis for Airline: American

- Total number of data samples: 2759
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 1960)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Customer Service Issue (Frequency: 768)
- Shortest tweet length: 17
- Longest tweet length: 167

3.4 Analysis for Airline: Delta

- Total number of data samples: 2222
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 955)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Late Flight (Frequency: 269)
- Shortest tweet length: 13
- Longest tweet length: 167

3.5 Analysis for Airline: Southwest

- Total number of data samples: 2420
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 1186)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Customer Service Issue (Frequency: 391)
- Shortest tweet length: 18
- Longest tweet length: 165

3.6 Analysis for Airline: US Airways

- Total number of data samples: 2913
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 2263)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Customer Service Issue (Frequency: 811)
- Shortest tweet length: 15
- Longest tweet length: 186

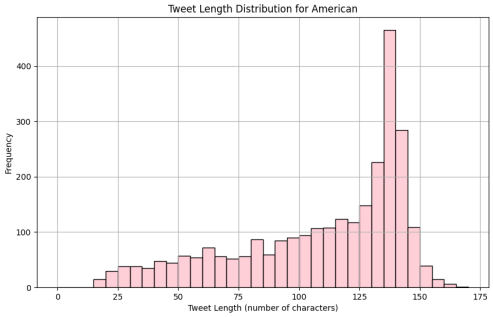
3.7 Analysis for Airline: United

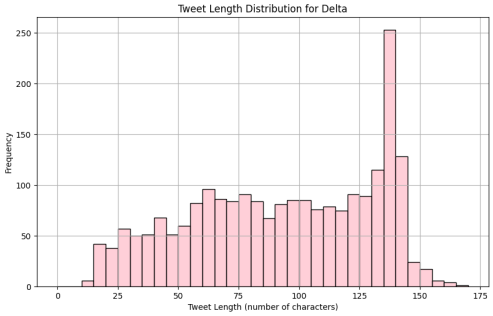
- Total number of data samples: 3822
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 2633)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Customer Service Issue (Frequency: 681)
- Shortest tweet length: 12
- Longest tweet length: 165

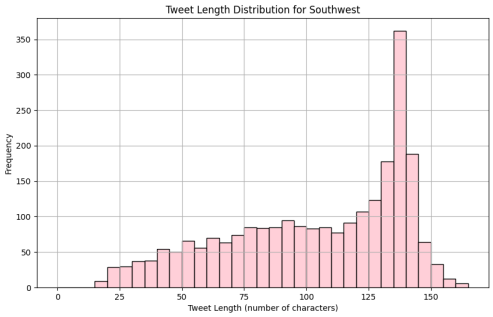
3.8 Analysis for Airline: Virgin America

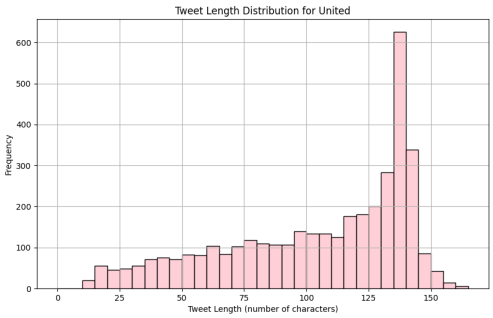
- Total number of data samples: 504
- Unique values in airline sentiment: 3
- Most frequent value in airline sentiment: negative (Frequency: 181)
- Unique values in negative reason: 10
- Most frequent value in negative reason: Customer Service Issue (Frequency: 60)
- Shortest tweet length: 22
- Longest tweet length: 159

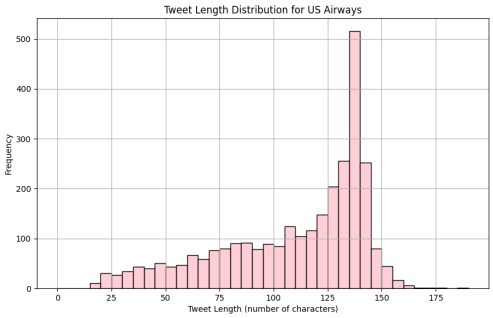
3.9 Tweet Length Distribution

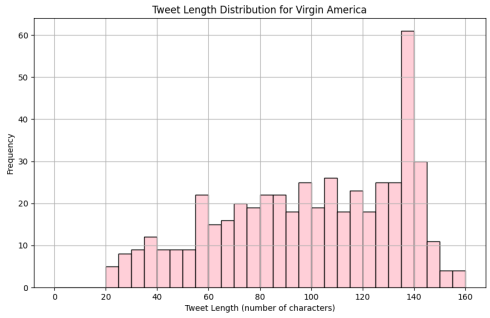


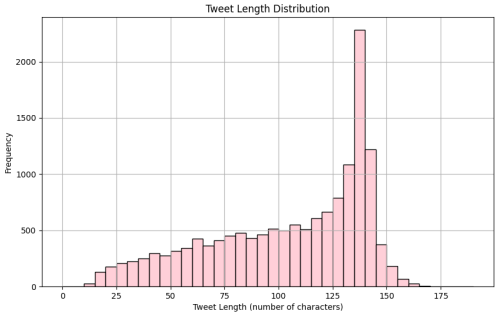


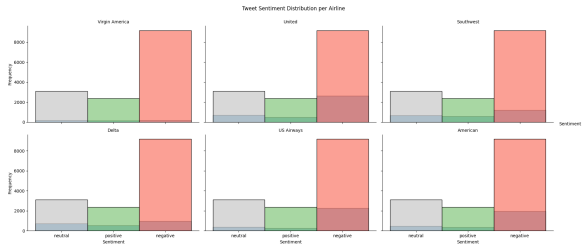







Tweet Length Distribution for Virgin America
Frequency
Tweet Length (number of characters)
0
20
40
60
80
100
120
140
160
0
10
20
30
40
50
60


Tweet Length Distribution
Frequency
Tweet Length (number of characters)
0
25
50
75
100
125
150
175
0
500
1000
1500
2000



3.10 Part B

3.11 Part C

For this task, the tokenizer was implemented as follows:

- **URLs:** URLs are captured as a single token. This ensures that the entire URL is treated as one unit.
- **Contractions:** The tokenizer preserves contractions (e.g., `don't`, `it's`, `you're`) as single tokens.
- **Numbers:** The tokenizer captures whole and decimal numbers as single tokens.
- **Hashtags and Mentions:** The tokenizer treats hashtags (e.g., #NLP) and mentions (e.g., user) as single tokens, preserving their integrity.
- **Punctuation:** Punctuation marks (e.g., `.,!?;:"'()[]`) are separated from words and treated as individual tokens.

3.12 Part D

When comparing the custom tokenizer to NLTK's tokenizer, we uncovered some essential differences in how they handle social media text. The custom tokenizer works well in preserving key elements like URLs, mentions, hashtags, and contractions, treating them as single units. This feature is vital for effectively analyzing tweets. Conversely, NLTK's tokenizer tends to split up contractions and punctuation mistakenly. It often struggles with URLs and hashtags, leading to treating these elements as separate tokens or even leaving them out entirely.

The custom tokenizer's ability to accurately manage these elements makes it the better choice for processing Twitter data, which is filled with unique components.

This is how the tokenizers performed:

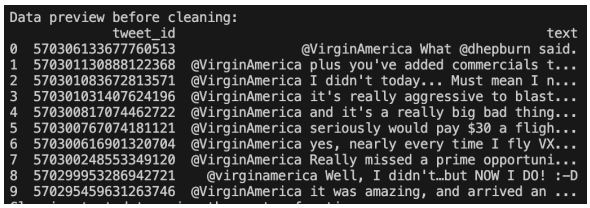
- **URL Handling:** The custom tokenizer preserves URLs as a single token, while NLTK may split them into multiple tokens or not capture them correctly.
  - **Custom Tokenizer:** `['http://mywebsite.com']`
  - **NLTK Tokenizer:** `['http', ':', '/', '/', 'mywebsite', '.', 'com']`

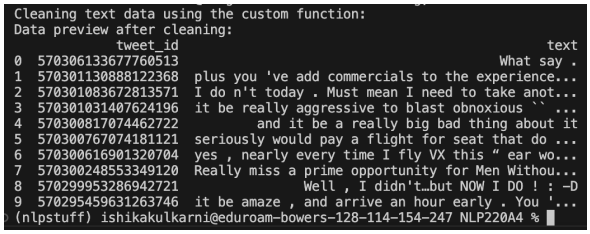
- **Contractions:** The custom tokenizer treats contractions like `"don't"` and `"it's"` as single tokens, while NLTK breaks them into two tokens, splitting the contraction (e.g., `"don't"` becomes `"do"` and `"n't"`).
  - **Custom Tokenizer:** `["don't"]`
  - **NLTK Tokenizer:** `["do", "n't"]`
- **Hashtags and Mentions:** The custom tokenizer correctly identifies hashtags and mentions as single tokens, whereas NLTK might split them or treat them inconsistently.
  - **Custom Tokenizer:** `['@user', '#NLP']`
  - **NLTK Tokenizer:** `['@', 'user', '#', 'NLP']`
- **Punctuation Handling:** The custom tokenizer isolates punctuation marks (e.g., `"!"`, `"?"`, `"'"`) as individual tokens. In contrast, NLTK sometimes attaches punctuation marks to the adjacent words or splits them incorrectly.
  - **Custom Tokenizer:** `['Hello', '!', "How", "'", "s", 'it', 'going', '?']`
  - **NLTK Tokenizer:** `['Hello', '!', 'How', "'", 's', 'it', 'going', '?']`
- **Ellipses (Multiple Periods):** The custom tokenizer treats ellipses (three or more consecutive periods) as a single token, while NLTK splits them into individual periods or omits them.
  - **Custom Tokenizer:** `['...', 'tacky']`
  - **NLTK Tokenizer:** `['...', 'tacky']` (But in some cases, NLTK would split them as `['.', '.', '.']`)

4 Part 2

The texts were cleaned using regular expressions.

- Remove Mentions: All mentions, such as "@united," were removed. This ensures that the dataset does not contain irrelevant user references.
- Handle Currency: Currency values were removed. This helps eliminate financial data that is irrelevant to sentiment analysis.
- Remove Email Addresses: Email addresses like "jane.doe@email.com" were removed to eliminate private information.
- Convert Emojis to Text: Emojis were converted to textual descriptions using the `emoji` library. The placeholder `[EMOJI]` was used for unrecognized emojis to indicate their presence.
- Decode HTML Escaped Characters: HTML escaped characters were decoded to restore the correct symbols and characters in the dataset.
- Normalize Punctuation: Repeated punctuation marks (e.g., "!!!!!!", "?!?", "....") were normalized to a single punctuation mark to reduce noise in the dataset.
- Normalize Times and Dates: Times and dates (e.g., "2/24 2:10pm", "6/30", "7:00 AM") were replaced with the token `[TIME/DATE]` to standardize these expressions.
- Remove URLs: URLs (e.g., "http://t.co/NfAQHhr09j", "https://t.co/caf2cx3gfi") were removed since they do not contribute to sentiment analysis.
- Verb Lemmatization: Verbs were lemmatized to their base form using the `WordNetLemmatizer` from the `nltk` library. For example, "running" becomes "run".
- Remove Duplicate Rows: Duplicate rows (where both the cleaned tweet and sentiment are the same) were removed to eliminate redundant entries.
- Remove Empty Tweets: Empty tweets were removed, which may result from cleaning processes that strip all characters or symbols.





- Contraction Expansion: Common contractions, such as "don’t," were expanded to their complete forms, e.g., "do not." This was achieved using a dictionary of ordinary contractions.

5 Part 3

In this part, we use SVM to classify sentiments and determine the impact of preprocessing techniques.

The dataset contains 14,266 tweets labeled with their respective sentiments: negative, neutral, and positive.

Some of the important columns in the dataset are:

- **airline_sentiment:** The sentiment label (negative, neutral, or positive).
- **text:** The tweet content.
- **Other metadata:** Includes tweet ID, sentiment confidence, and airline.

The preprocessing was as mentioned in the previous part.

We split the dataset into a training set (90%) and a test set (10%) using the train_test_split method from Scikit-learn, ensuring that the data was shuffled to avoid any bias. The distribution of the classes in the training and test sets is as follows:

Training Set:

- **Negative:** 8,157 tweets
- **Neutral:** 2,686 tweets
- **Positive:** 1,996 tweets

Test Set:

- **Negative:** 910 tweets
- **Neutral:** 308 tweets
- **Positive:** 209 tweets

This class distribution was printed to verify that the dataset is balanced.

We used the `SGDClassifier` with a hinge loss (linear SVM) and L2 regularization for model training. The model hyperparameters were set as follows:

- **Loss:** Hinge (for linear SVM)
- **Penalty:** L2 regularization
- **Alpha:** 1×10^{-4} (regularization strength)
- **Max Iterations:** 100
- **Tolerance:** None (no stopping criteria based on convergence)

- **Shuffle:** True (shuffling during training)
- **Random State:** 3 (for reproducibility)

An ablation study was conducted to assess the effect of different preprocessing combinations on model performance. The following preprocessing combinations were tested:

- **Original Preprocessing (All steps):** Full cleaning (mentions, URLs, emojis, etc.)
- **Lemmatization only:** Only tokenization and lemmatization.
- **Emoji Handling only:** Only emoji-related preprocessing.
- **Lemmatization + Emoji Handling:** Combination of lemmatization and emoji handling.
- **Emoji Handling + Lemmatization:** Emoji handling followed by lemmatization.

Preprocessing Method	Accuracy
Original Preprocessing	0.8005
Lemmatization only	0.8000
Emoji Handling only	0.8005
Lemmatization + Emoji Handling	0.8005
Emoji Handling + Lemmatization	0.8000

Table 1: Accuracies for different preprocessing methods

The Original Preprocessing combination achieved the highest accuracy, making it the best choice for training the final model.

Based on the ablation study results, the Original Preprocessing combination was chosen for the final model. The SGDClassifier was trained on the data preprocessed with all steps and evaluated on the test set.

Metric	Score
Test Set Accuracy	0.8017
Macro Average	
Precision	0.77
Recall	0.70
F1-score	0.73
Weighted Average	
Precision	0.79
Recall	0.80
F1-score	0.79

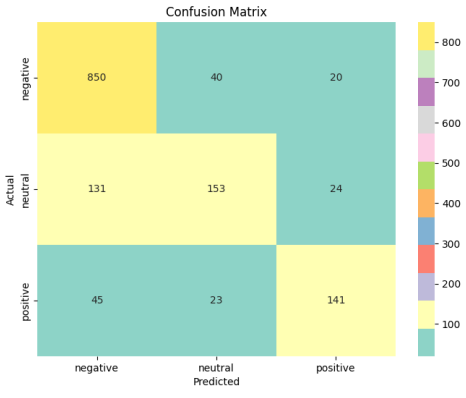
Table 2: Test Set Accuracy and Performance Metrics

```
Training the model using Original preprocessing steps.
Test Set Accuracy: 0.8017

Classification Report:
              precision    recall  f1-score   support

    negative       0.83      0.93      0.88       910
     neutral       0.71      0.50      0.58       308
    positive       0.76      0.67      0.72       209

    accuracy                           0.80      1427
   macro avg       0.77      0.70      0.73      1427
weighted avg       0.79      0.80      0.79      1427
```



6 Part 5

6.1 Unique Users

The number of unique users is determined by identifying distinct entries in the name column, which represents the users of the dataset.

Here the number of unique users is 7,701.

6.2 Active users

The top-5 words for the first 5 users are as follows:

- **0504Traveller:** http, sfjduahx9z, southwestair, usatoday, virginamerica
- **09202010:** 699, baggages, la, rdu, usairways
- **0veranalyser:** aircraft, americanair, cancelled, flightled, flight
- **0xjared:** depends, fair, getting, jetblue, vibe
- **10Eshaa:** 4llwi5oxvo, fleek, fleet, flight, hook

6.3 Active users by airlines

Virgin America

- **wmrrock:** Tweets about seat recline issues and cool pictures.
- **AirlineFuel:** Tweets about Virgin America's stock performance.
- **GunsNDip:** Tweets about business travel and flight quality.
- **total_janarchy:** Tweets about a bad experience with the airline.
- **ChrysiChrysic:** Tweets about issues with other passengers.

United Airlines

- **throthra:** Complaints about missing luggage and poor service.
- **patrick_maness:** Tweets about flight delays and service issues.
- **ColtSTaylor:** Tweets about flight updates and lost luggage.
- **Evan_Flay:** Tweets expressing frustration with flight services.
- **urno12:** Tweets thanking the airline for follow-up on issues.

Southwest Airlines

- **scoobydoo9749:** Tweets about reimbursement issues and lost luggage.
- **luvthispayne:** Tweets expressing gratitude but pointing out service flaws.
- **Heavenlychc9:** Tweets about flight delays and customer service complaints.
- **davidgoodson71:** Tweets about airline's failure to follow through.
- **geekstiel:** Tweets thanking the airline for customer service.

Delta Airlines

- **JetBlueNews:** Tweets about JetBlue's services (appears to be mixed with JetBlue's news).
- **kbosspotter:** Tweets regarding JetBlue flights and direct messaging.
- **ThatJasonEaton:** Tweets criticizing JetBlue's customer service.
- **SMHillman:** Tweets about JetBlue's Mint service and flight experiences.
- **heyheyman:** Tweets about JetBlue's loyalty and service issues.

US Airways

- **rossj987:** Tweets about personal grievances with US Airways.
- **ElmiraBudMan:** Tweets criticizing US Airways' customer service and delays.
- **worldwideweg:** Tweets about dissatisfaction with service.
- **thomashoward88:** Tweets about flight delays and bad experiences.
- **jasemccarty:** Tweets about miscommunication with US Airways.

American Airlines

- **otisday:** Tweets about understaffing and delayed services.
- **flemmingerin:** Tweets requesting assistance from American Airlines.

- **chagaga2013:** Tweets about cost issues and poor service.
- **_mhertz:** Tweets about research and seeking further information from American Airlines.
- **georgetietjen:** Tweets about flight details and scheduling issues.

6.4 Missing values

tweet_location: 4,733 missing entries tweet_coord: 13,621 missing entries user_timezone: 4,820 missing entries

6.5 Date time parsing

- Used `pd.to_datetime()` to parse the `tweet_created` field into a `datetime64` object, with `errors='coerce'` to handle invalid entries.
- Ensured the timestamps were time zone aware (`UTC-08:00`), allowing for accurate time-based operations.
- Verified the successful conversion by checking the column's data type with `df_cleaned.dtypes`.

6.6 Philadelphia

Total number of tweets from Philadelphia: 70

Different spellings of Philadelphia: ['Philadelphia, Pa' 'Los Angeles, CA (via Philly)' 'Philadelphia, PA' 'Philadelphia PA ' 'Philly' 'Philadelphia/Cali' 'Philadelphia' 'Philadelphia, PA USA' 'Philly Yo' 'Philly Area' 'Philly, Chicago, MSP, Vegas' 'Philly to NY/NJ' 'Philadelphia, pa' 'Philadelphia Suburbs']

- Identified tweets where the `tweet_location` field contains references to Philadelphia, considering variations and misspellings.
- Used a case-insensitive search with `str.contains()` to capture different forms of "Philadelphia" in the dataset.
- Discovered multiple variations, such as "Philly," "Philadelphia, PA," and "Philadelphia/Cali."
- Replaced common misspellings and standardized these variations using the `replace()` function in Pandas.
- Retrieved all unique spellings of Philadelphia using `unique()` and displayed them for verification.
- Counted a total of 70 tweets originating from locations related to Philadelphia.
- This approach ensures a more accurate analysis by accounting for diverse ways users refer to Philadelphia.

6.7 airline_sentiment_confidence

Attached CSV