CSML1010 – Project Proposal

Group 11 – Alex Fung and Patrick Osborne
Instructor – Dr. En-Shiun Annie Lee

Twitter US Airline – Sentiment Analysis

- **Dataset from Twitter**
  - 14,640 thousand tweets
  - 15 Features
  - 6 US Airlines
  - Positive | Neutral | Negative

- **Aligns with a business problem**
  - How to stop negative tweet trends before they go out of control?
  - Identify the early tweets accurately -> Sentiment Analysis!

- **Presents an interesting ML/Data challenge**
  - Data is messy (Twitter character limit, slang, emojis, @ and #)

Loading Data

- **Loading Data**
  - Available as CSV and SQL
  - UTF-8 encoding to preserve emojis 👍

- **Check for NULL Values**

Column 'negativereason' has a few missing nulls, but they are null only if 'airline_sentiment' is positive or neutral.

```
df_tweets['negativereason'].isnull().sum()
```

```
5462
```

```
df_tweets['airline_sentiment'].loc[df_tweets['negativereason'].isnull()].unique()
```

```
array(['neutral', 'positive'], dtype=object)
```

Cleaning Data

- **Twitter data notoriously unclean**
  - Character limit forces non-standard English
- **Remove noise for sentiment analysis**
  - @ Mentions, **RT** re-tweets
- **Retain value-adds**
  - Emojis 💻, hashtags #
  - Add important context and content

- **Clean text itself**
  - Remove HTML, translate slang, remove STOP words and grammar, lemmatize

Data Cleaning Challenges

– **How to extract emojis?**

  - **emot 2.1**

    ```
    pip install emot
    ```

  - ```
    18    True
    Name: emojis_flag, dtype: object
    18    [❤, ☺, 👍]
    Name: emojis, dtype: object
    18    I ❤️ flying @VirginAmerica. ☺👍
    Name: text, dtype: object
    18    I ❤️ flying . ☺👍
    Name: text_cleaned, dtype: object
    18    I  flying .
    Name: text_cleaned_without_emojis_emoticons, dtype: object
    ```
    ```

Data Cleaning Challenges

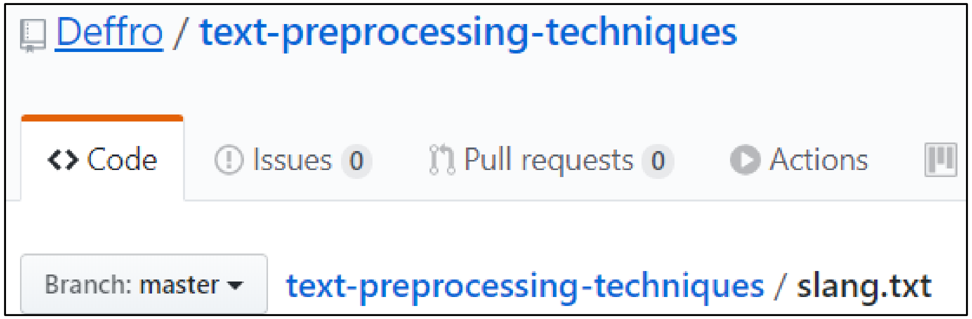
- **Extract Hashtags?**

```
def extract_and_remove_hashtags(df_tweets):
    regex_to_replace= r'#(\w+)'
    replace_value= ''
    df_tweets['hashtags'] = ''
    df_tweets['text_cleaned_without_emojis_emoticons_hashtags'] = ''

    for i, row in df_tweets.iterrows():
        df_tweets.at[i, 'hashtags'] = re.findall(regex_to_replace, df_tweets.at[i, 'text_cleaned'])
        df_tweets.at[i, 'text_cleaned_without_emojis_emoticons_hashtags'] = re.sub(regex_to_replace, replace_value,  df_tweets.a
t[i, 'text_cleaned'])
```

- **Text to lowercase**

- **Convert Slang**
  - Find and replace based on an existing dictionary



Text Cleaning

- **Remove stop words**
  - “The”, “and”, “You”, etc. -> noise

- **Remove punctuation & numbers**

- **Lemmatize**

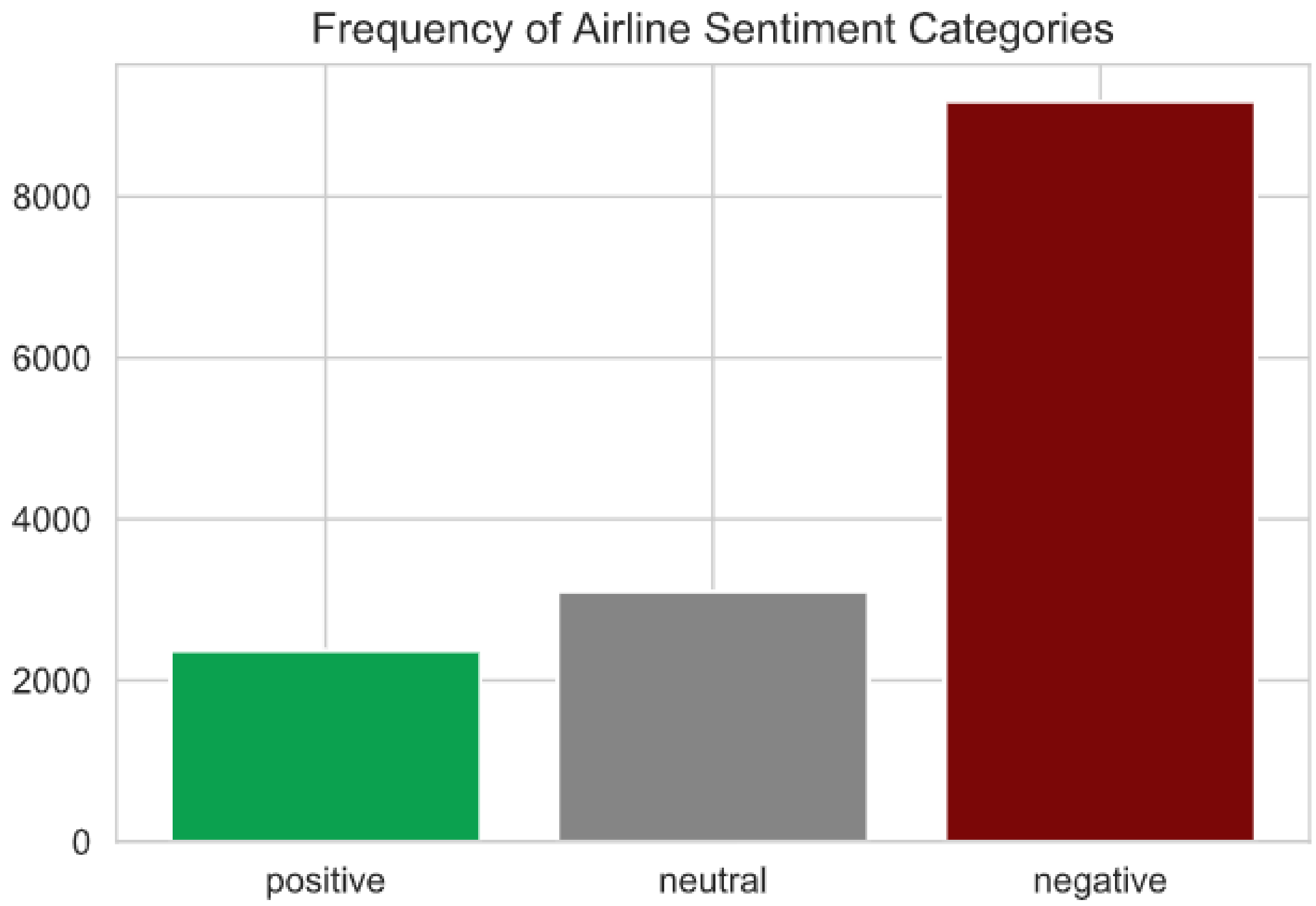
- **Tokenize**

Data Exploration

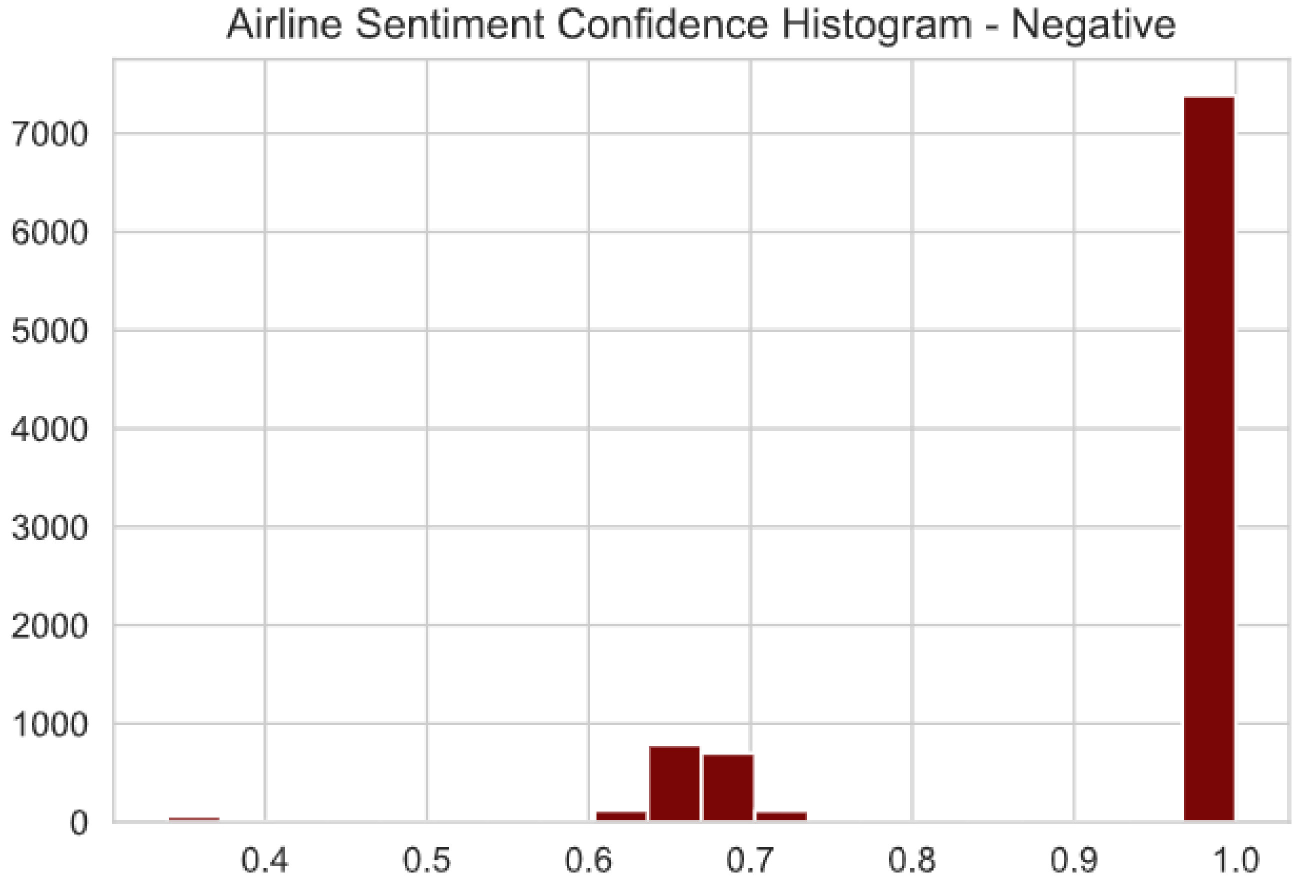
- **15 columns**
  - 6 main features that interest us
    1. airline_sentiment
    2. airline_sentiment confidence
    3. negativereason
    4. negativereason_confidence
    5. airline
    6. text

- **Engineered Features**
    1. emojis
    2. emoticons
    3. cleaned text (lemmas)

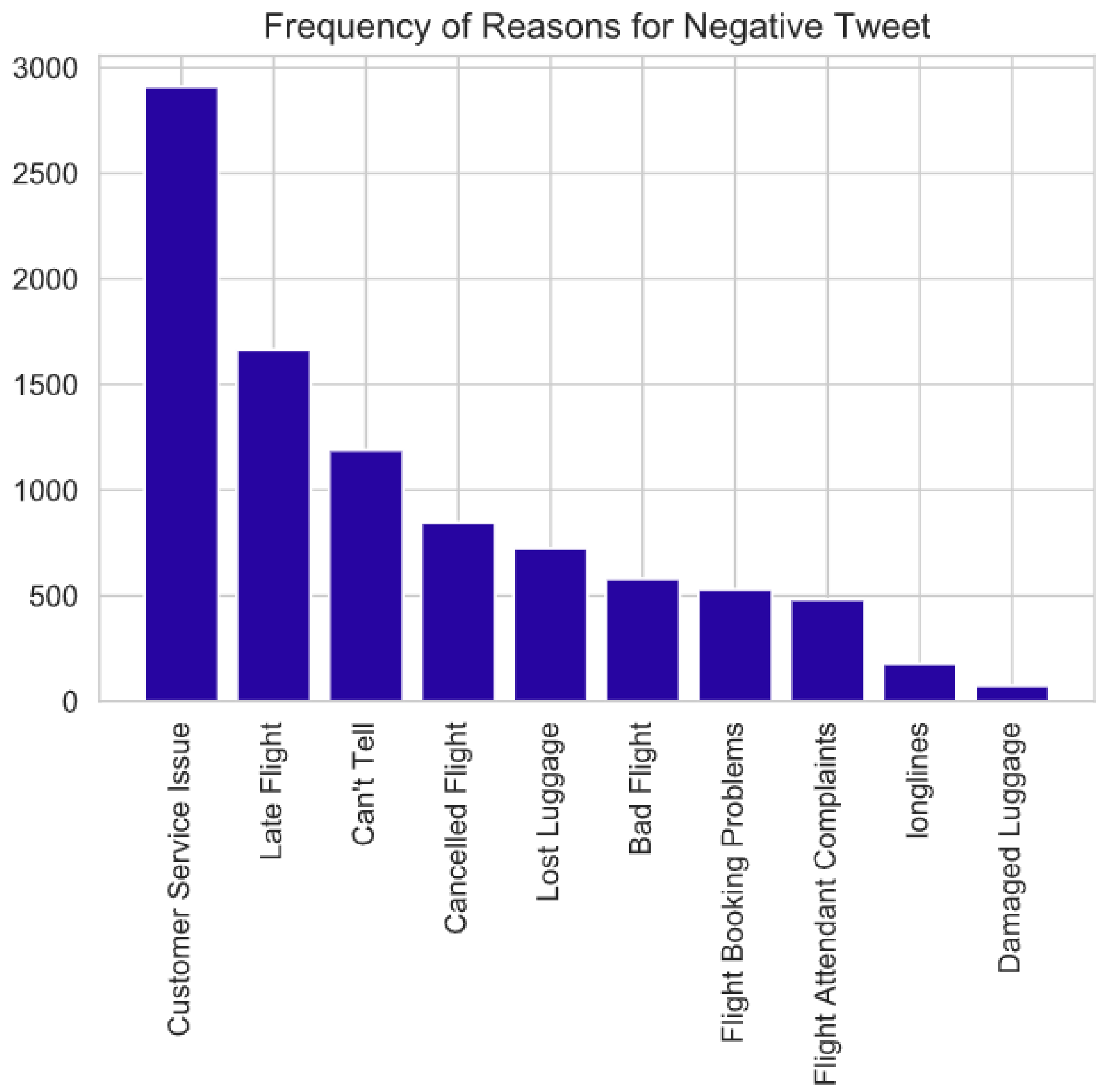
Data Exploration



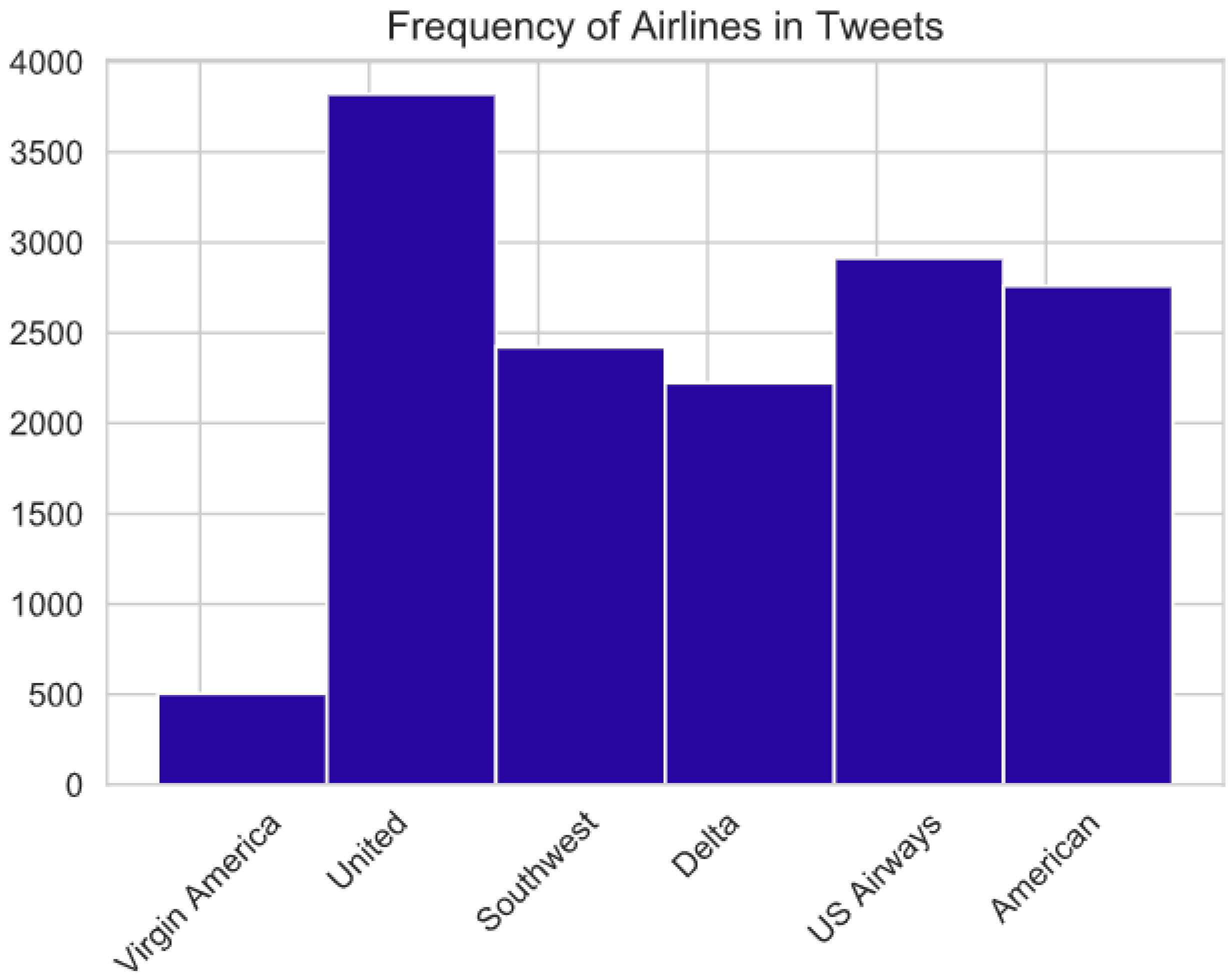
Data Exploration



Data Exploration



Data Exploration

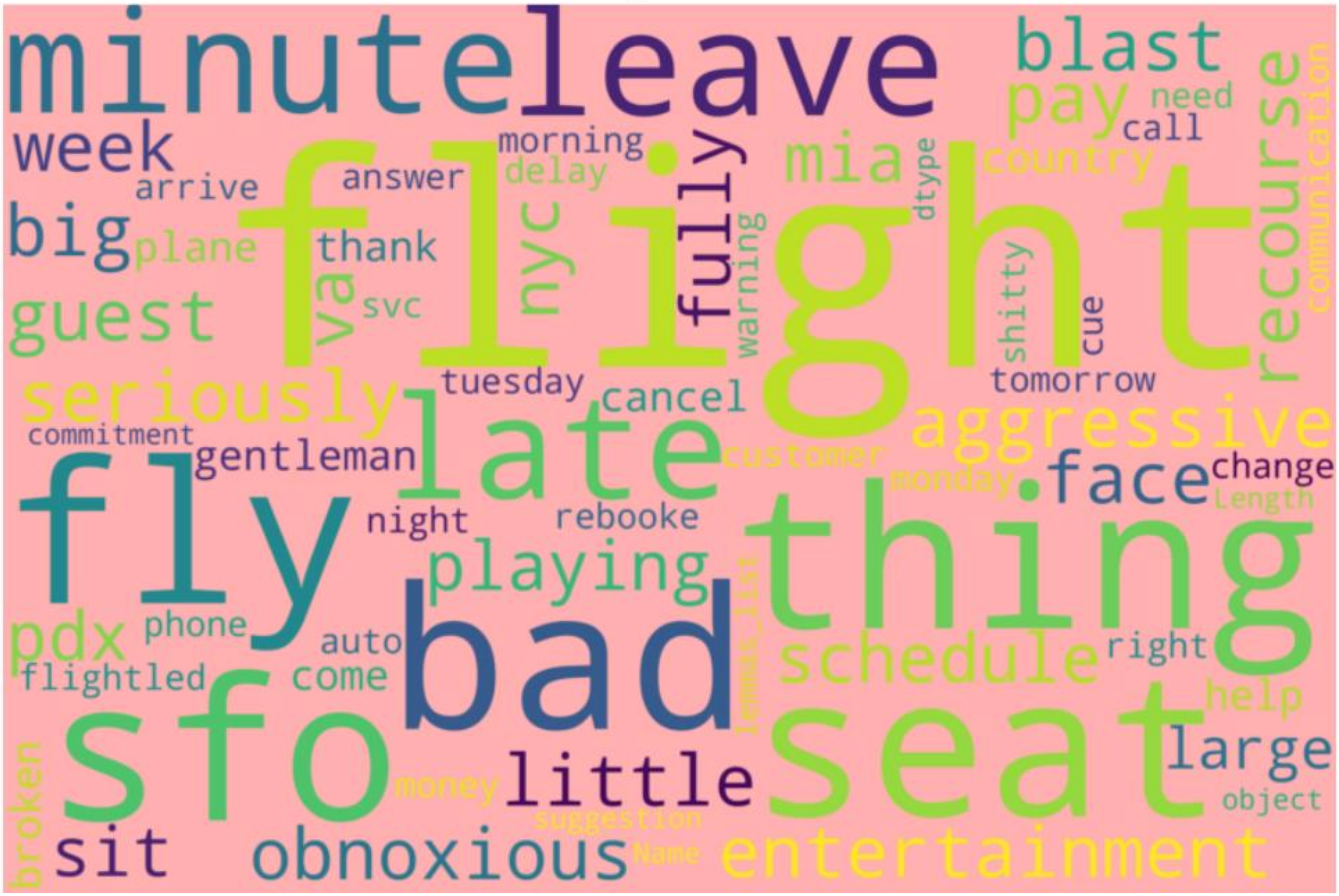


Data Exploration

Word Cloud for Positive Sentiment Tweets

Data Exploration

Word Cloud for Negative Sentiment Tweets



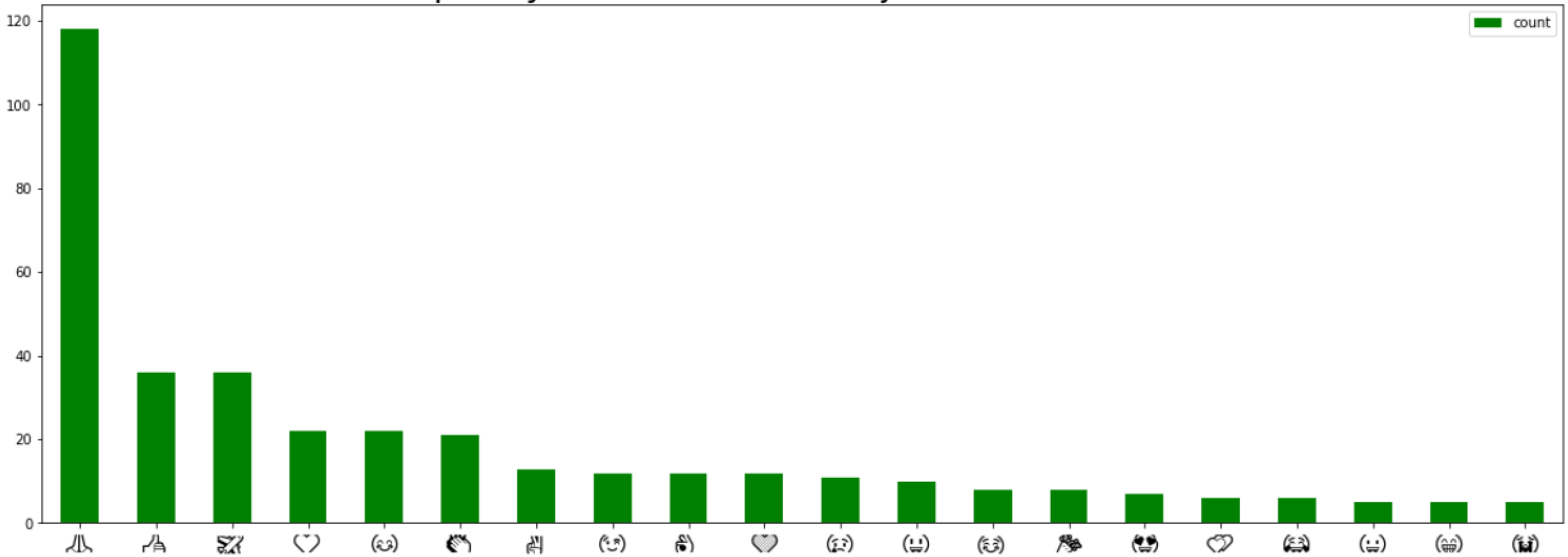
Data Exploration – Emojis & Hashtags

- **Is there value in the extracted emojis and hashtags?**

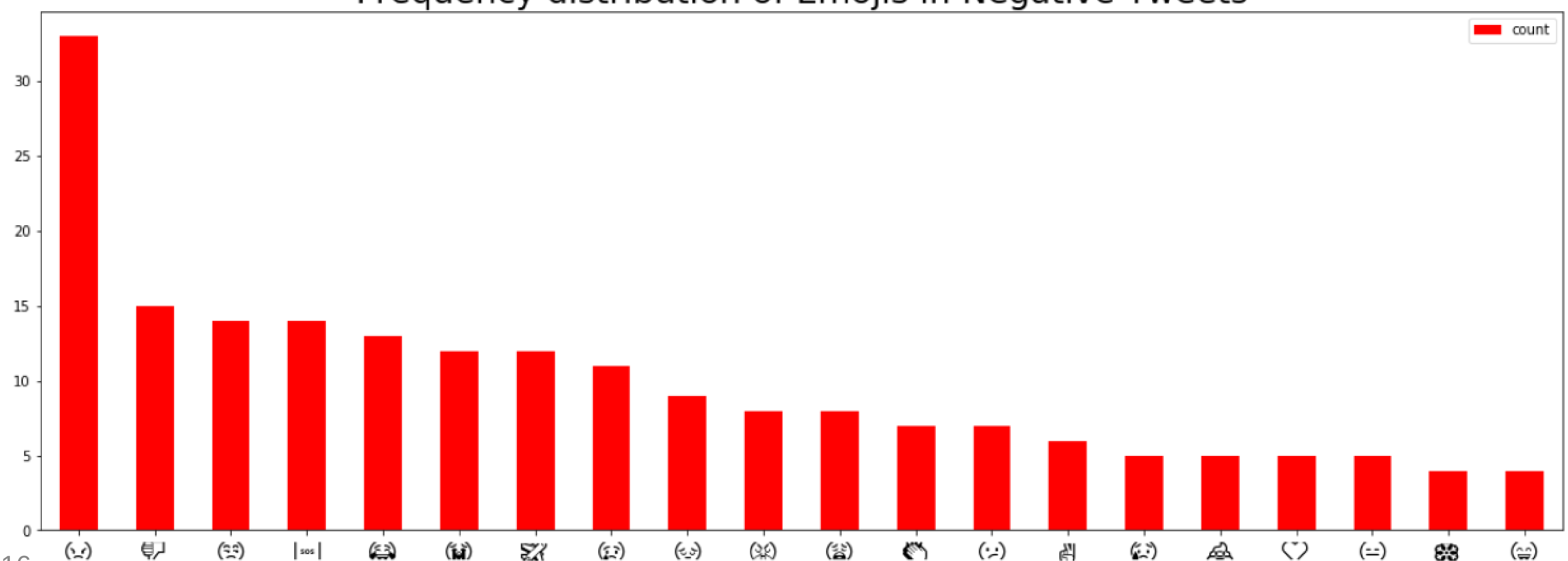
- **Method:**
  - Plot frequency by sentiment class (positive, neutral, negative)

  - If there is a clear difference – emojis & hashtags are differentiators -> add context

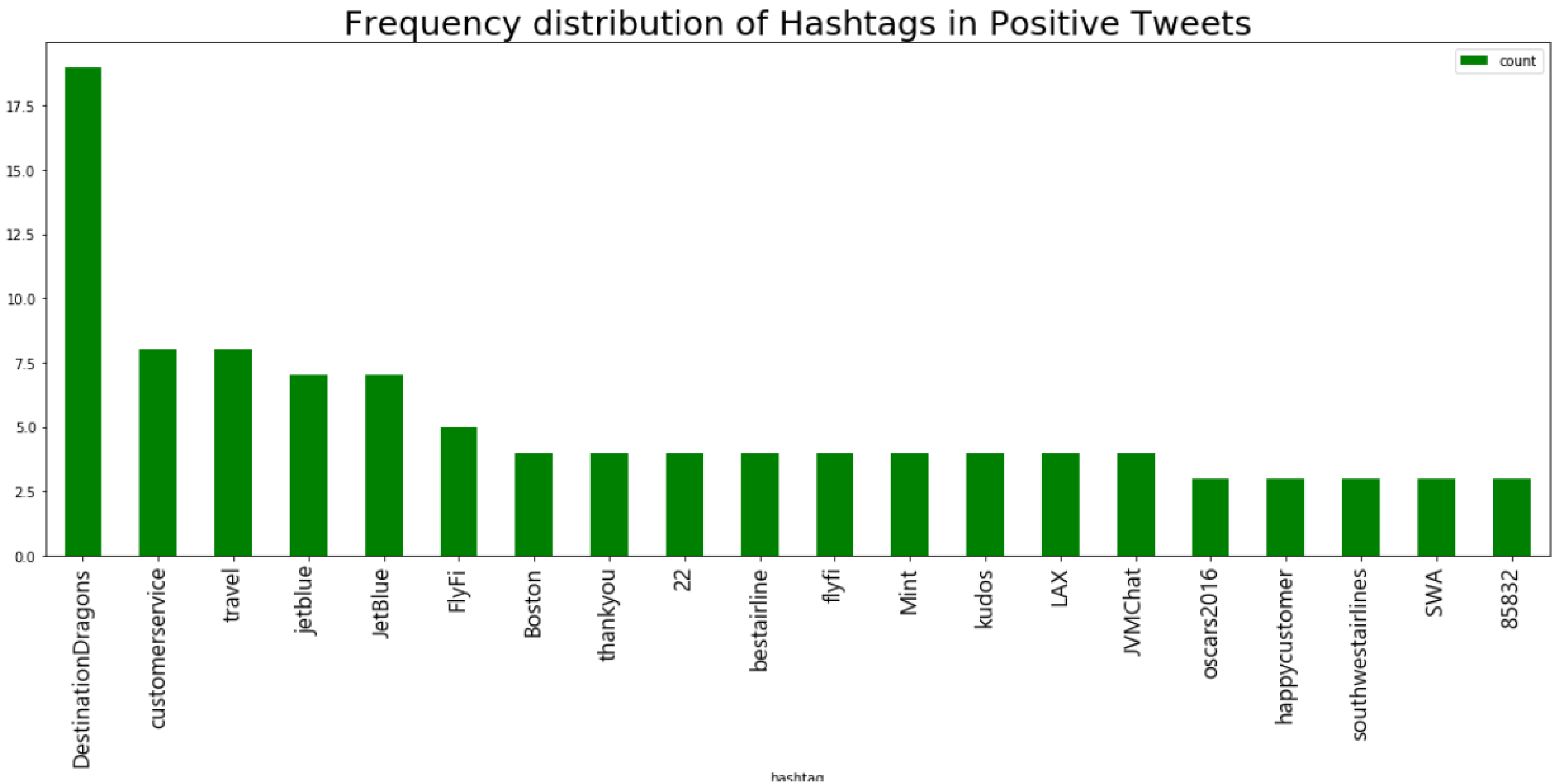
Frequency distribution of Emojis in Positive Tweets



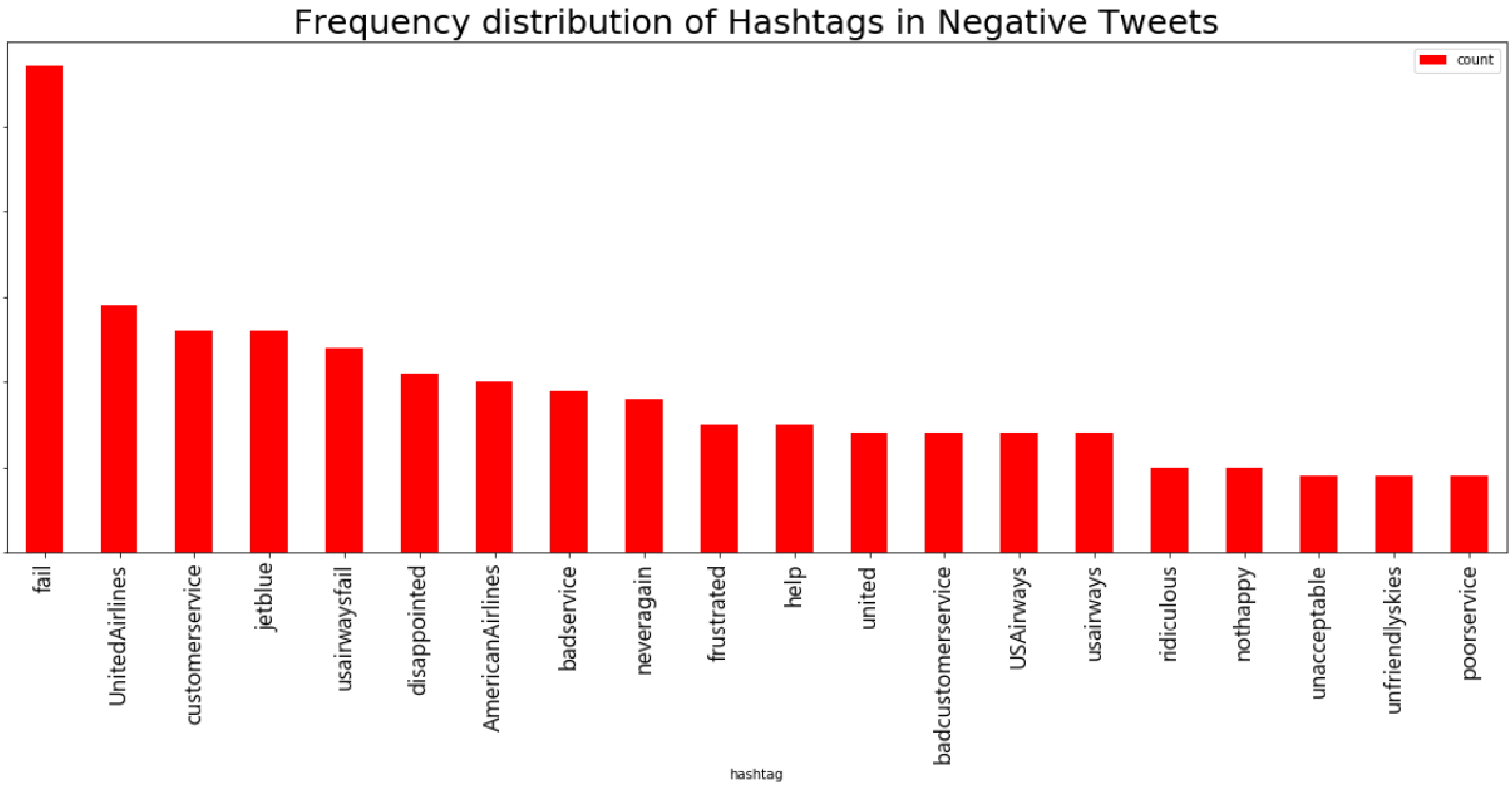
Frequency distribution of Emojis in Negative Tweets



Data Exploration – Emojis & Hashtags



Data Exploration – Emojis & Hashtags



Data Exploration – Emojis & Hashtags

- **Issues with our emoji, emoticon and hashtag extraction**
  - Extraction process confuses non-emoticon string of characters for an emoticon
    - E.g. time of day, airport codes
  - Hashtags could be cleaned further
    - Reduce to lowercase, remove stop words, lemmatize
    - Challenging as hashtags are not pure English

YORK UNIVERSITÉ UNIVERSITY

Thank You!