Interactive Dashboard

with streamlit and python

By Abir Sait

Table of contents

01 Overview

02 Introduction

03 Course Summary

04 Project Summary

05 Problem Statement

06 Design

07 Implementation

08 Results

09 Challenges

10 Conclusion

11 References

01 Overview

Overview

The project aims to build an interactive dashboard using Streamlit, a Python library for creating web apps and dashboards with minimal code

The dashboard uses a Twitter US Airlines sentiment dataset from Kaggle, which contains tweets from the US that mentioned any of the airlines and their sentiment (positive, neutral, or negative)

The dashboard allows users to explore the dataset, visualize the data, and retrieve information from it using various widgets and plots

02 Introduction

Introduction

- This Project is based on coursera guided project that provides step-by-step instructions, code snippets, and hints - **Create Interactive Dashboards with Streamlit and Python**
- This course aims to teach us data manipulation, data visualization and creating an interactive dashboard using python, Streamlit, Plotly and WordCloud
- Sentiment analysis helps airlines to understand customer feedback and improve thrir services.

03 Course Summary

Course Summary

- This Project is based on coursera guided project that provides step-by-step instructions, code snippets, and hints - **Create Interactive Dashboards with Streamlit and Python**

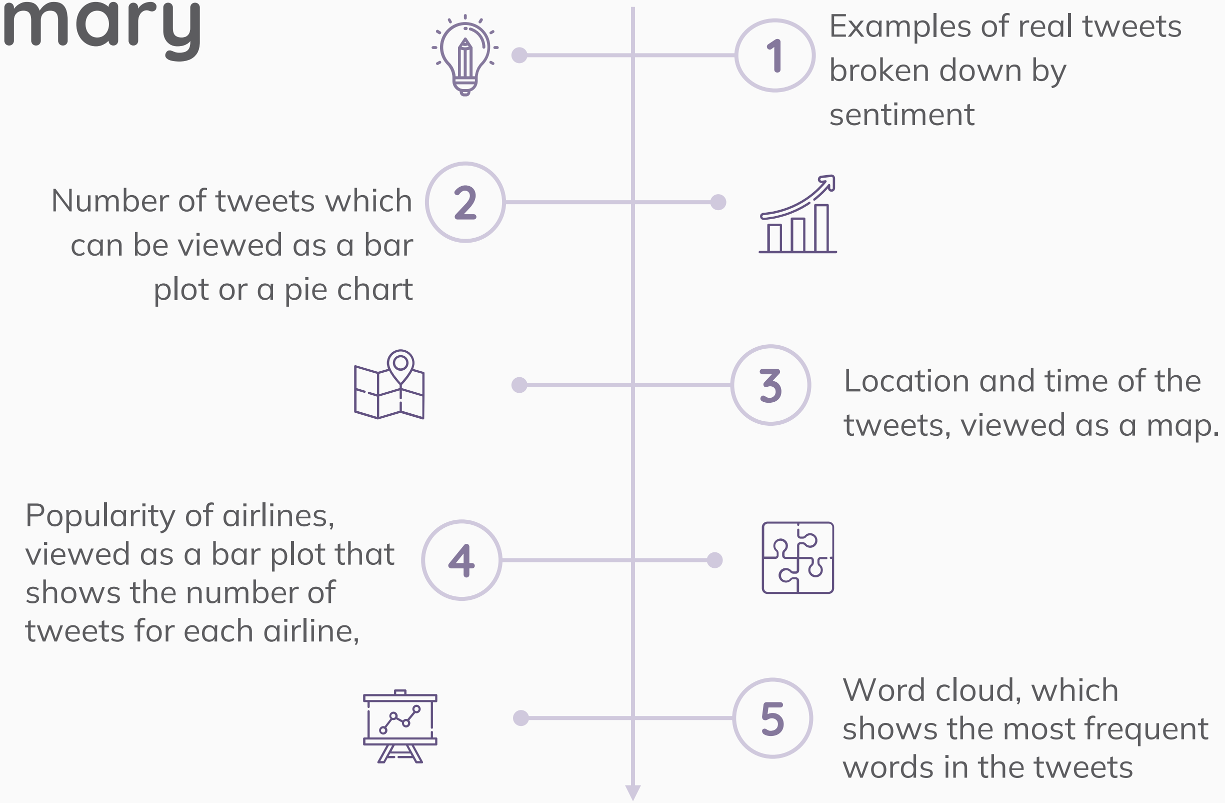
- This course helped me
  - Learn to build engaging data dashboards with Streamlit and python
  - Learned to transform raw data into actionable visualizations and dashboards such as bar plots, pie charts, maps, and word clouds for effective communication and decision-making
  - Learned how to use widgets and interactivity to filter and display the data
  - Create versatile dashboards for personal projects or workplace applications.

04 Project Summary

Project Summary

Objective:

The project demonstrates the use of Streamlit and Python to create a dashboard for analyzing the Twitter US Airlines sentiment dataset



1 Examples of real tweets broken down by sentiment

2 Number of tweets which can be viewed as a bar plot or a pie chart

3 Location and time of the tweets, viewed as a map.

4 Popularity of airlines, viewed as a bar plot that shows the number of tweets for each airline,

5 Word cloud, which shows the most frequent words in the tweets

05 Problem Statement

Problem statement

- This project aims to understand customer sentiment towards US airlines and gain insight into the aviation industry.
- Understanding customer sentiment helps airlines to improve their services and customer satisfaction.
- Sentiment analysis of customer feedback and reviews can help airlines to identify areas for improvement and address customer issues as needed.
- Effective consumer complaint management is also important to avoid negative word-of-mouth and potential brand damage.

06 ✦ Design

Design

Data Source

The dashboard is based on the Twitter US Airlines sentiment dataset from Kaggle, which contains tweet ID, sentiment, location, and the tweet text.

Visualization

The dashboard was created using Python, with charting libraries such as Plotly used for data visualizations

UI Components:

Interactive, allowing users to explore data and visualizations dynamically. Built with minimal calls to the Streamlit library, resulting in a clean and efficient design.

Hosting and Accessibility

Can be hosted on a local server, a company's internal network, or a VPN.

07 Implementation

With tools used

Tools used

python

a popular programming language for data analysis and visualization

Streamlit

Python library used for building interactive web applications and dashboards

Plotly

Plotly is a charting library used for creating interactive visualizations

Wordcloud

a Python library used for creating word clouds

Pandas

Python library used for data manipulation and analysis

VS code

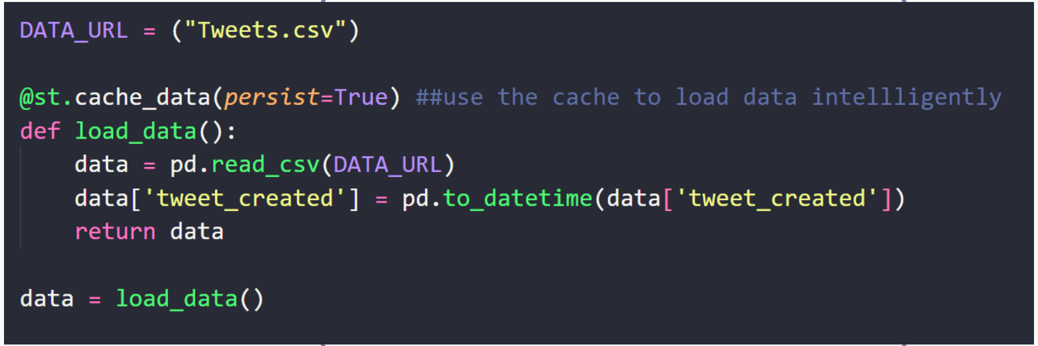
popular integrated development environment (IDE) used for writing, editing, and debugging code

Implementation

Load the Twitter US Airline Sentiment Data

This line of the code is used to Load the dataset of tweets using the pandas library to read the .csv file.

st.cache -> used to cache the output of the function and intelligently use the cache data

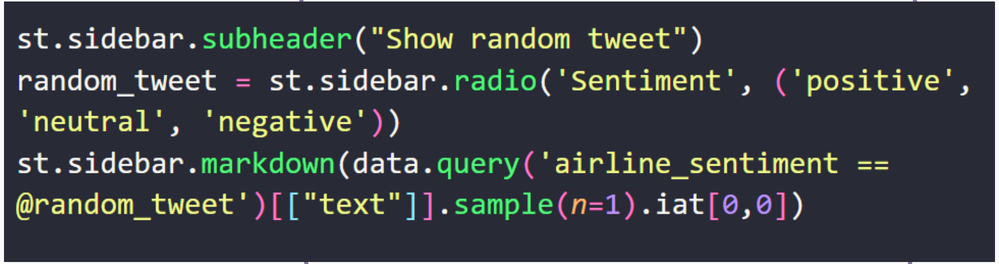


```python
DATA_URL = ("Tweets.csv")

@st.cache_data(persist=True) ##use the cache to load data intellligently
def load_data():
    data = pd.read_csv(DATA_URL)
    data['tweet_created'] = pd.to_datetime(data['tweet_created'])
    return data

data = load_data()
```

Implementation

Display Tweets in the Sidebar

This line of the code is used to display a random tweet based on the sentiment selection from the radio buttons (positive, negative, neutral) on the sidebar using pandas



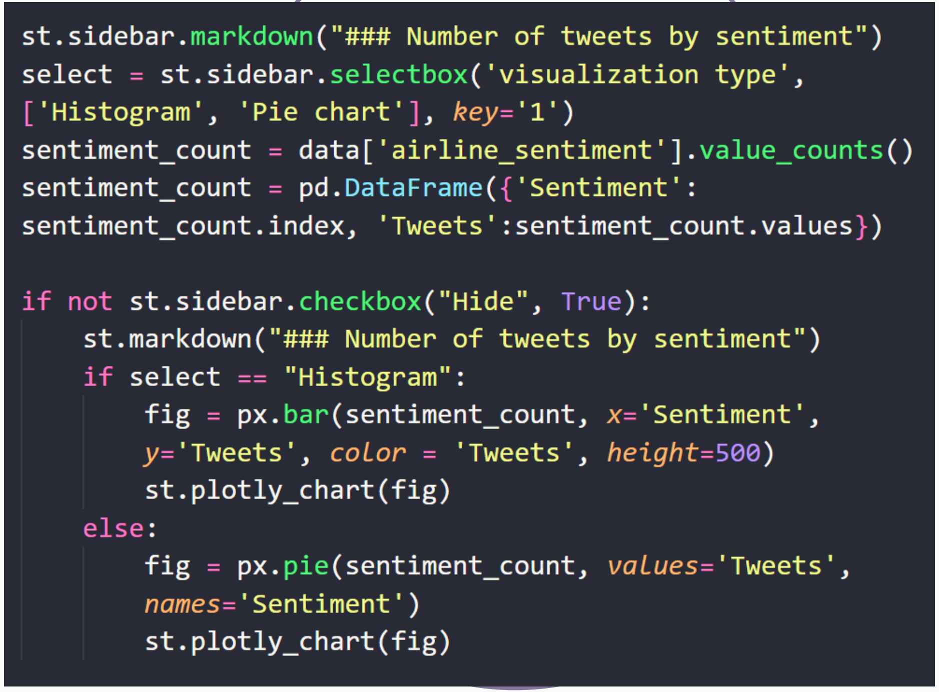
```
st.sidebar.subheader("Show random tweet")
random_tweet = st.sidebar.radio('Sentiment', ('positive',
'neutral', 'negative'))
st.sidebar.markdown(data.query('airline_sentiment ==
@random_tweet')[["text"]].sample(n=1).iat[0,0])
```

Implementation

Plot Interactive Bar Plots and Pie Charts

This line of the code is used display bar graph or pie chart based on the preference selected from the dropdown list using plotly.

A checkbox is created to hid the visuals when not needed.



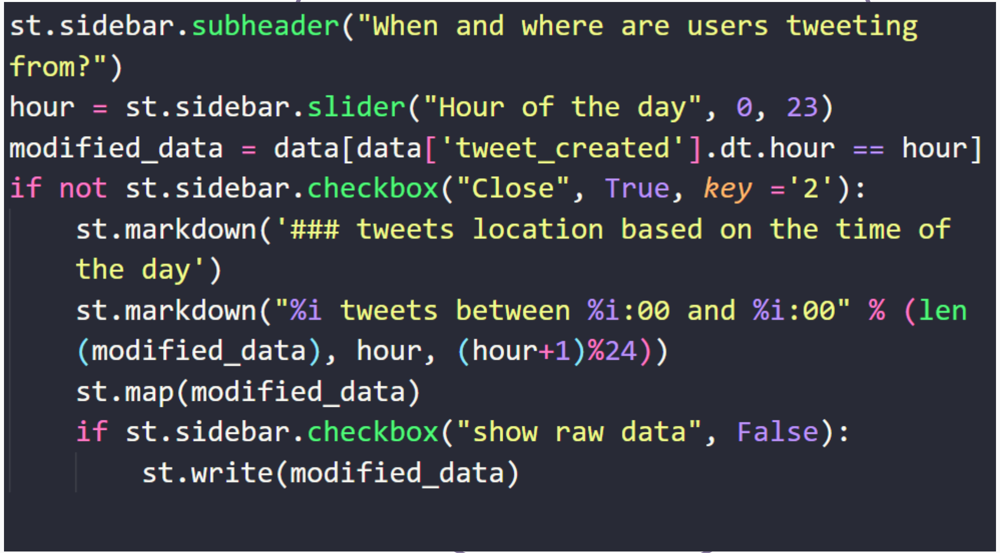
```
st.sidebar.markdown("### Number of tweets by sentiment")
select = st.sidebar.selectbox('visualization type',
['Histogram', 'Pie chart'], key='1')
sentiment_count = data['airline_sentiment'].value_counts()
sentiment_count = pd.DataFrame({'Sentiment':
sentiment_count.index, 'Tweets':sentiment_count.values})

if not st.sidebar.checkbox("Hide", True):
    st.markdown("### Number of tweets by sentiment")
    if select == "Histogram":
        fig = px.bar(sentiment_count, x='Sentiment',
        y='Tweets', color = 'Tweets', height=500)
        st.plotly_chart(fig)
    else:
        fig = px.pie(sentiment_count, values='Tweets',
        names='Sentiment')
        st.plotly_chart(fig)
```

Implementation

Plotting Location Data on an Interactive Map

This line of the code is used display an interactive map. The user can choose the hour and the map will show the location and number of tweets within that hour.

Show raw data checkbox is used to display the raw data in real time.

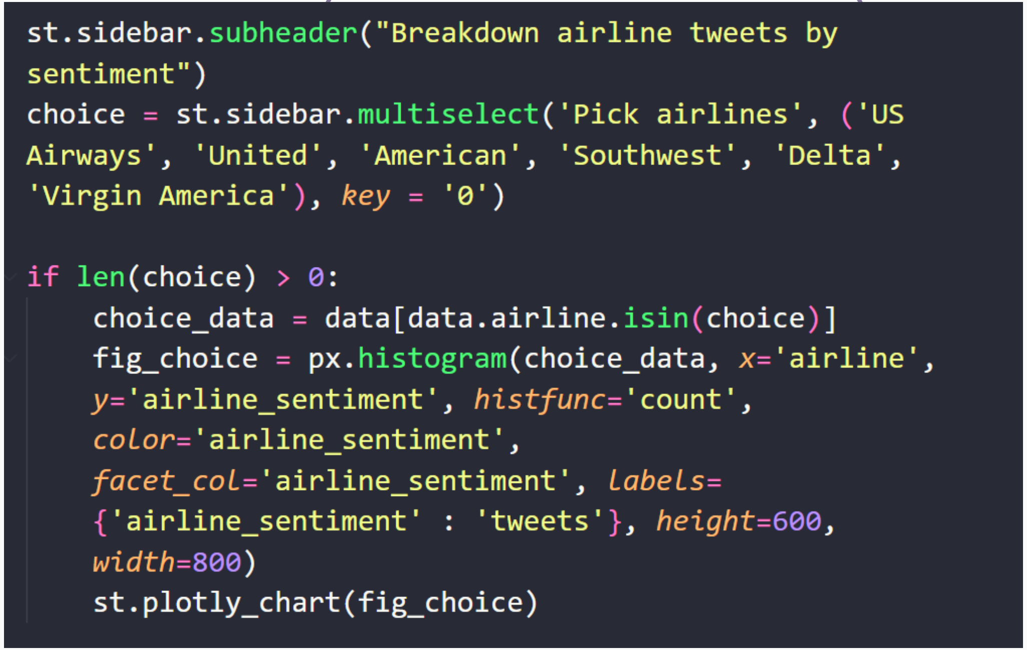


```
st.sidebar.subheader("When and where are users tweeting
from?")
hour = st.sidebar.slider("Hour of the day", 0, 23)
modified_data = data[data['tweet_created'].dt.hour == hour]
if not st.sidebar.checkbox("Close", True, key ='2'):
    st.markdown('### tweets location based on the time of
    the day')
    st.markdown("%i tweets between %i:00 and %i:00" % (len
    (modified_data), hour, (hour+1)%24))
    st.map(modified_data)
    if st.sidebar.checkbox("show raw data", False):
        st.write(modified_data)
```

Implementation

Plot Number of Tweets by Sentiment for Each Airline

This line of the code is used to plot the number of tweets broken down by sentiment using multiselect widget in and display it as a histogram.

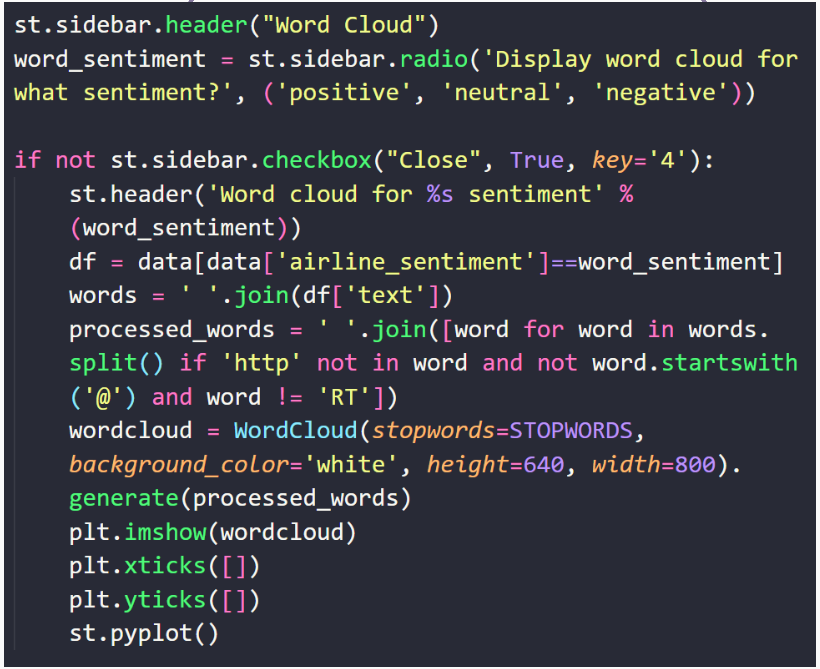


```
st.sidebar.subheader("Breakdown airline tweets by
sentiment")
choice = st.sidebar.multiselect('Pick airlines', ('US
Airways', 'United', 'American', 'Southwest', 'Delta',
'Virgin America'), key = '0')

if len(choice) > 0:
    choice_data = data[data.airline.isin(choice)]
    fig_choice = px.histogram(choice_data, x='airline',
    y='airline_sentiment', histfunc='count',
    color='airline_sentiment',
    facet_col='airline_sentiment', labels=
    {'airline_sentiment' : 'tweets'}, height=600,
    width=800)
    st.plotly_chart(fig_choice)
```

Implementation

Word Cloud for Positive, Neutral, and Negative Tweets

This line of the code is used to display a wordcloud based on the user’s choice from the radio buttons.

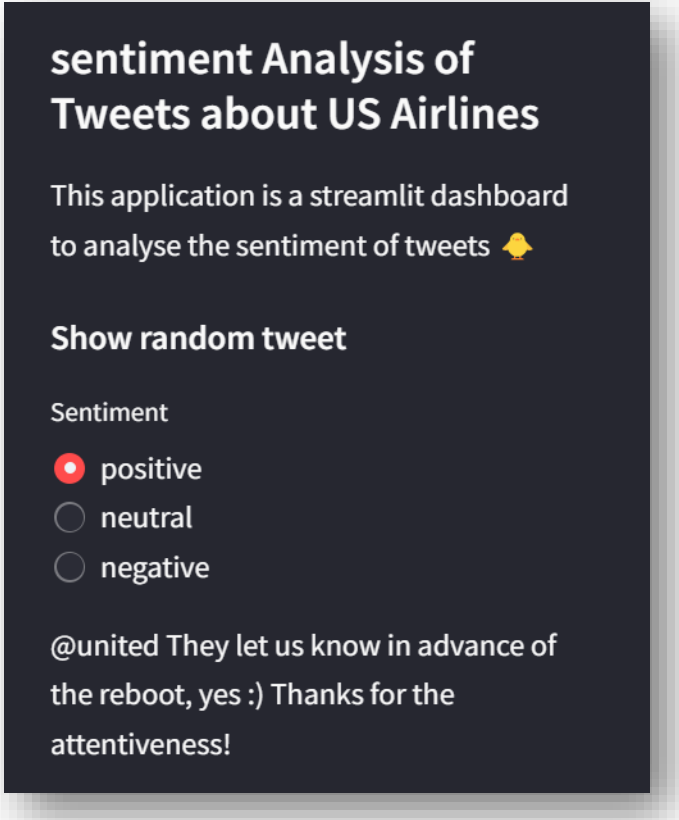


```
st.sidebar.header("Word Cloud")
word_sentiment = st.sidebar.radio('Display word cloud for
what sentiment?', ('positive', 'neutral', 'negative'))

if not st.sidebar.checkbox("Close", True, key='4'):
    st.header('Word cloud for %s sentiment' %
    (word_sentiment))
    df = data[data['airline_sentiment']==word_sentiment]
    words = ' '.join(df['text'])
    processed_words = ' '.join([word for word in words.
    split() if 'http' not in word and not word.startswith
    ('@') and word != 'RT'])
    wordcloud = WordCloud(stopwords=STOPWORDS,
    background_color='white', height=640, width=800).
    generate(processed_words)
    plt.imshow(wordcloud)
    plt.xticks([])
    plt.yticks([])
    st.pyplot()
```

08 Results

Display Tweets in the Sidebar

Examples of real tweets broken down by sentiment, which can be viewed by clicking on the buttons for positive, neutral, or negative sentiment.



sentiment Analysis of Tweets about US Airlines

This application is a streamlit dashboard to analyse the sentiment of tweets 🐥

Show random tweet

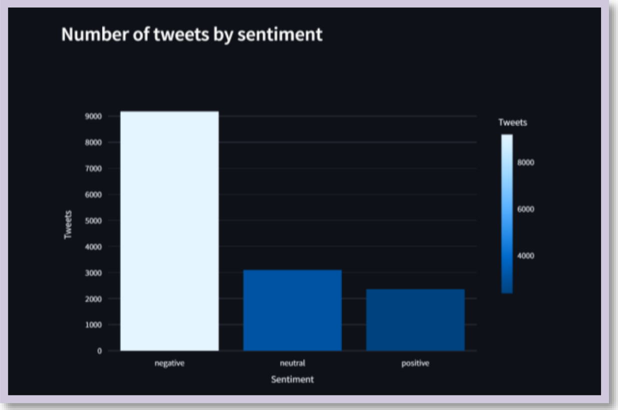
Sentiment

- ● positive
- ○ neutral
- ○ negative

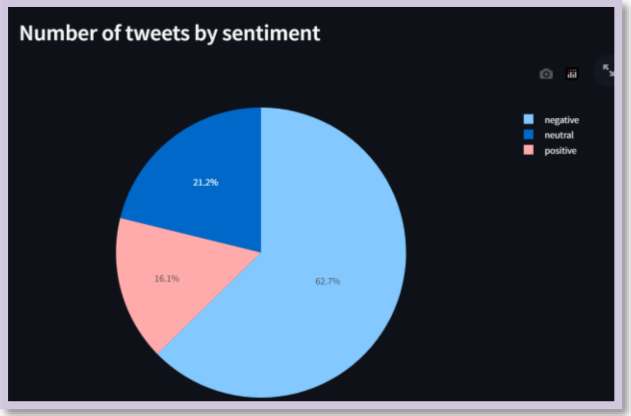
@united They let us know in advance of the reboot, yes :) Thanks for the attentiveness!

Plot Interactive Bar Plots and Pie Charts

Number of tweets broken down by sentiment, which can be viewed as a bar plot or a pie chart



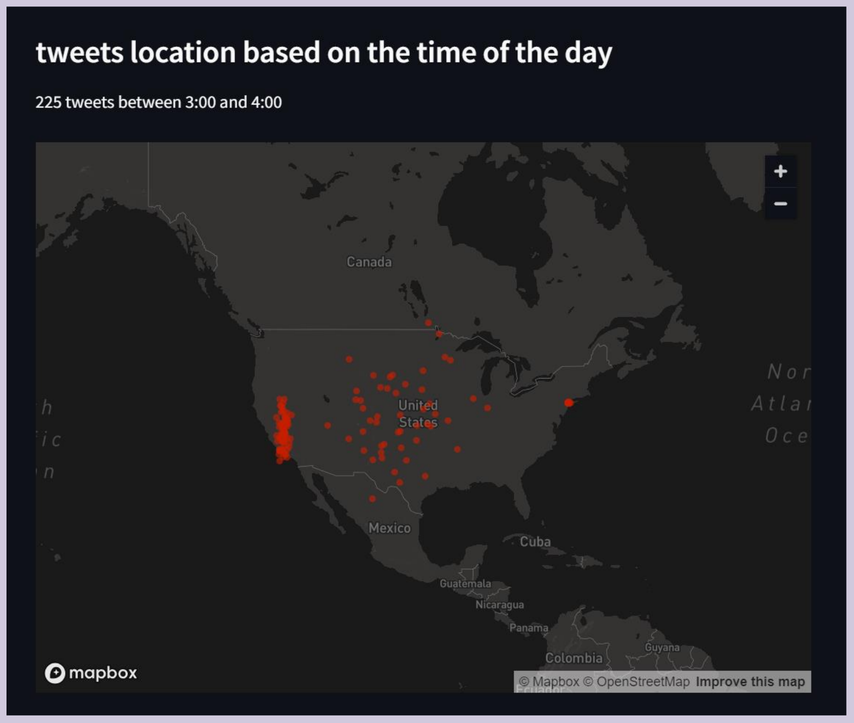
Number of tweets by sentiment



Number of tweets by sentiment

Plotting Location Data on an Interactive Map

Location and time of the tweets, which can be viewed as a map with markers indicating the tweet locations, and a slider that can filter the tweets by date and time

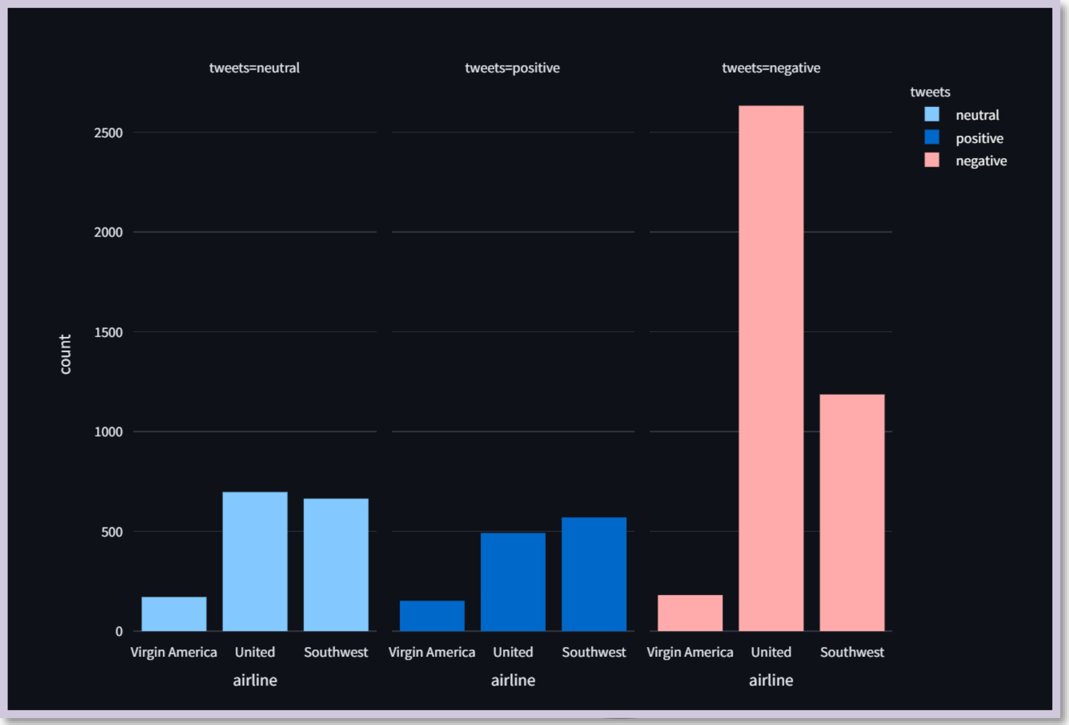


tweets location based on the time of the day

225 tweets between 3:00 and 4:00

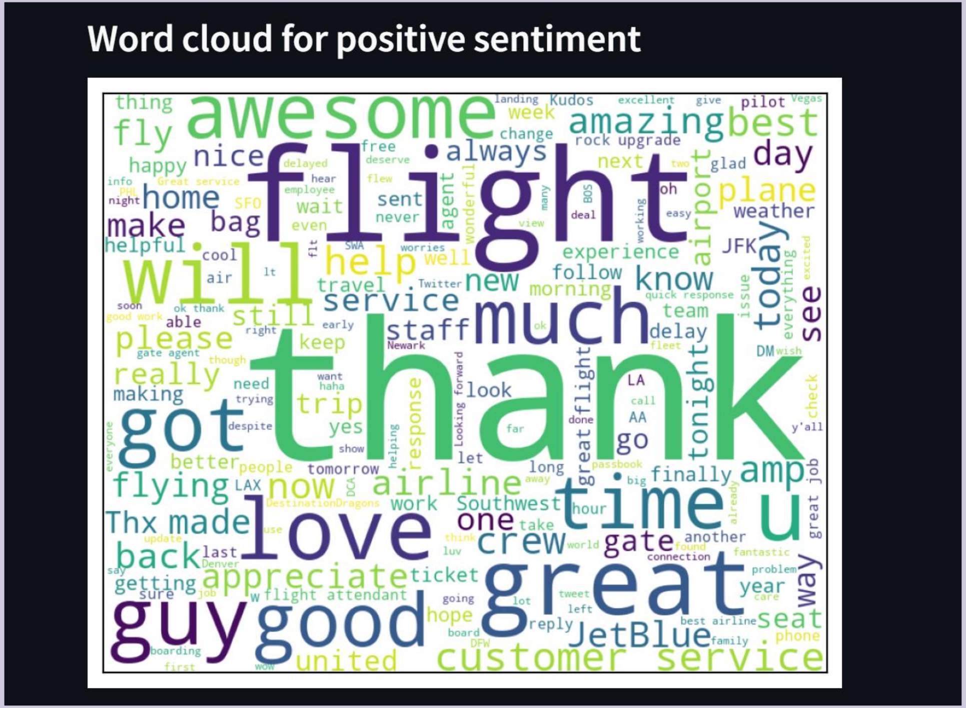
Plot Number of Tweets by Sentiment for Each Airline

Popularity of airlines, which can be viewed as a bar plot that shows the number of tweets for each airline, and can be further broken down by sentiment by selecting the airlines in the input box



Word Cloud for Positive, Neutral, and Negative Tweets

Word cloud, which shows the most frequent words in the tweets, and can be filtered by sentiment



Word cloud for positive sentiment

09 Challenges

Challenges faced

- One of the challenges faced was the difference in operating systems between the Rhyme machine used in the Coursera project and the Windows OS used by the presenter.
- Some lines of code did not work on Windows, requiring additional research to find solutions.
- This highlights the importance of being familiar with the tools and environments used in a project and being prepared to troubleshoot issues that may arise.

Challenges

1

Choosing appropriate visualizations

While Streamlit, Plotly, and WordCloud offer a wide range of visualization options, it can be challenging to choose the most appropriate ones

2

Performance optimization:

As the size of the dataset increases, the performance of the dashboard may decrease. It is important to optimize the code and use caching mechanisms to ensure that the dashboard remains responsive and interactive

3

Data cleaning and preprocessing

Dataset may require some cleaning and preprocessing before it can be used for analysis. This could involve removing duplicates, handling missing values, and converting data types.

10 Conclusion

Conclusion

The project demonstrates the use of Streamlit, Plotly, and WordCloud for data analysis and visualization, and the importance of understanding customer sentiment towards airlines

The project can help airlines to improve their services and customer satisfaction

Future work can include analyzing tweets from other countries and industries

11 References

References

- Cousera: - Create Interactive Dashboards with Streamlit and Python: https://www.coursera.org/projects/interactive-dashboards-streamlit-python
- Dataset:https://www.kaggle.com/code/parthsharma5795/comprehensive-twitter-airline-sentiment-analysis
- Streamlit documentation: https://docs.streamlit.io/en/stable/
- Plotly documentation: https://plotly.com/python/
- WordCloud documentation: https://github.com/amueller/wordcloud
- Pandas documentation: https://pandas.pydata.org/

Thanks!