Sentiment Analysis of Twitter Data:-

Name :- Rakhi Pancholi

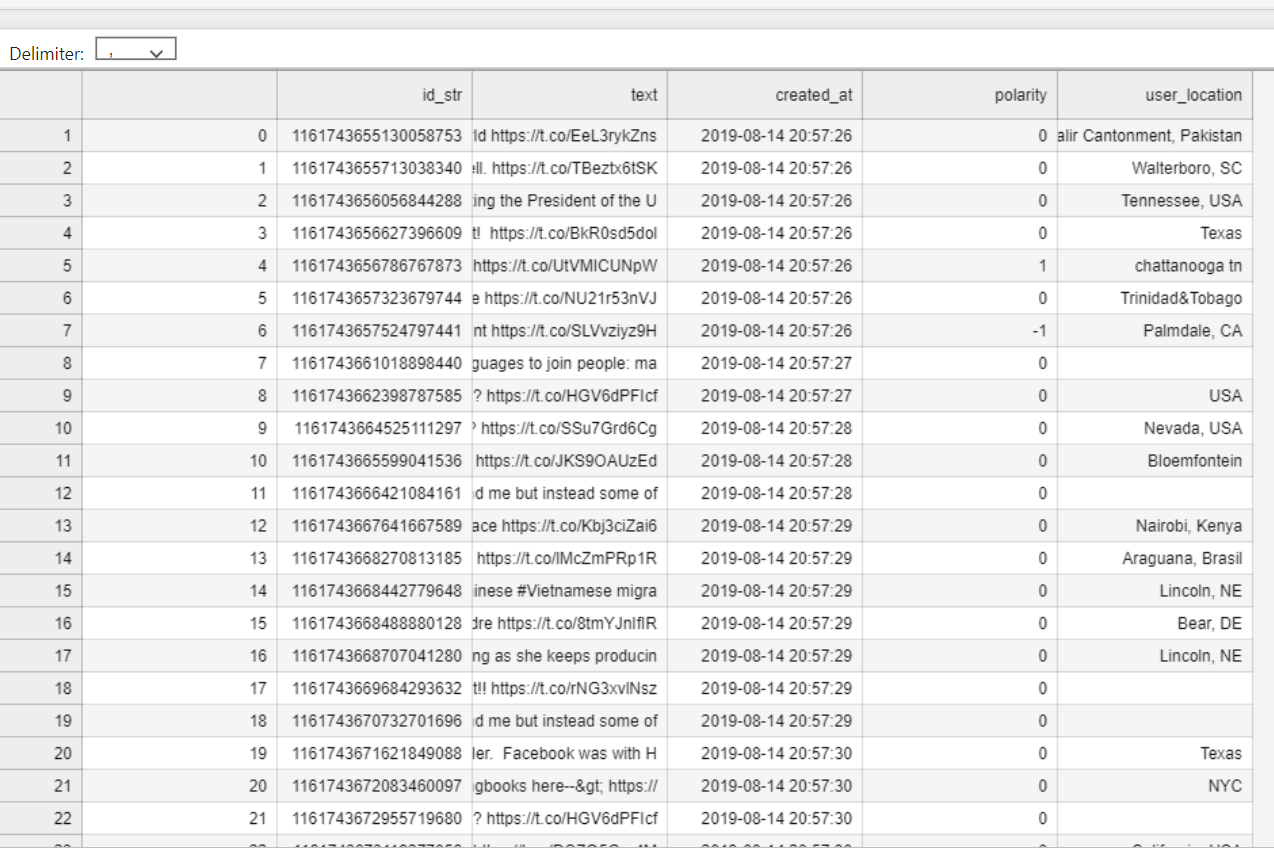
Batchno.:-3 (Sir Padampat Singhania University )

Week 1

Task: Data Collection

Work:-

* Initially, I attempted to collect data from RapidAPI. However, the dataset obtained was insufficient in terms of columns and rows.
* Consequently, I opted to use a dataset from a GitHub repository. This dataset was more comprehensive, containing user\_id, user\_location, tweet, polarity, created\_at, and other relevant columns.



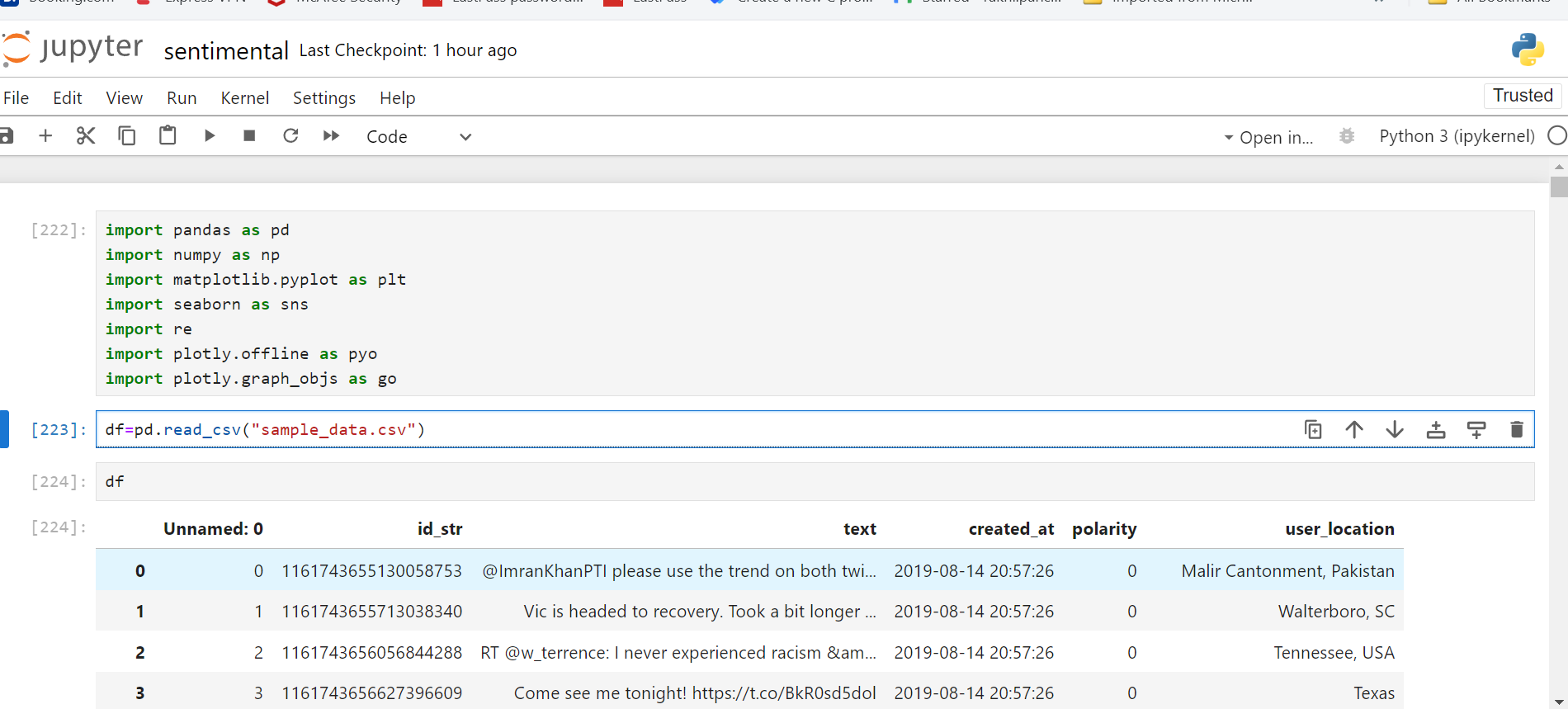
Week 2

Task: Exploratory Data Analysis (EDA) and Dashboard Creation

Work:-

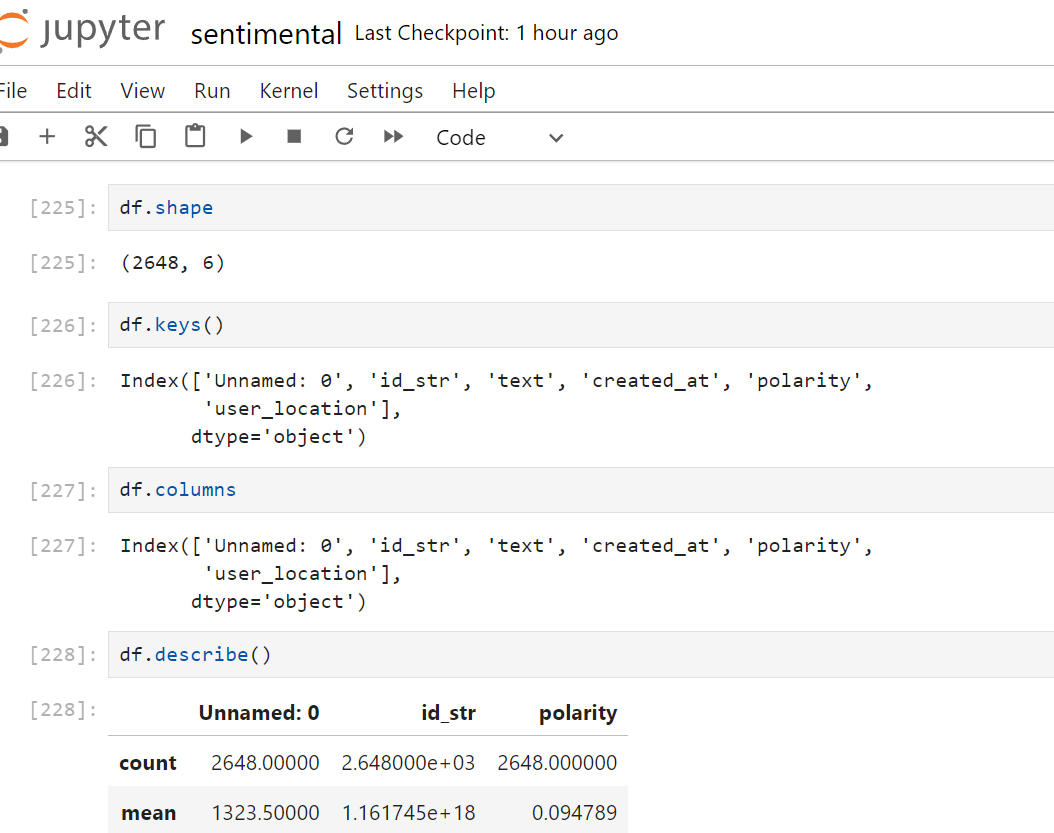
1. Library Import and Data Upload:

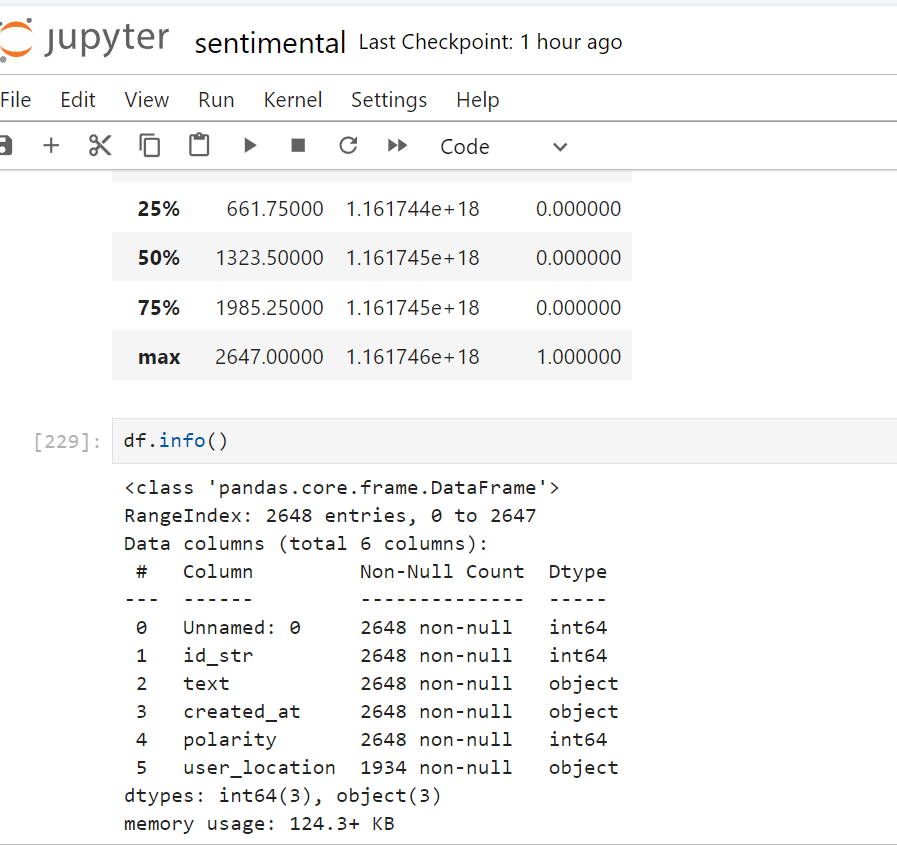
* Imported essential libraries for data analysis, such as Pandas.
* Uploaded the dataset using Pandas.



2. Initial Data Understanding:

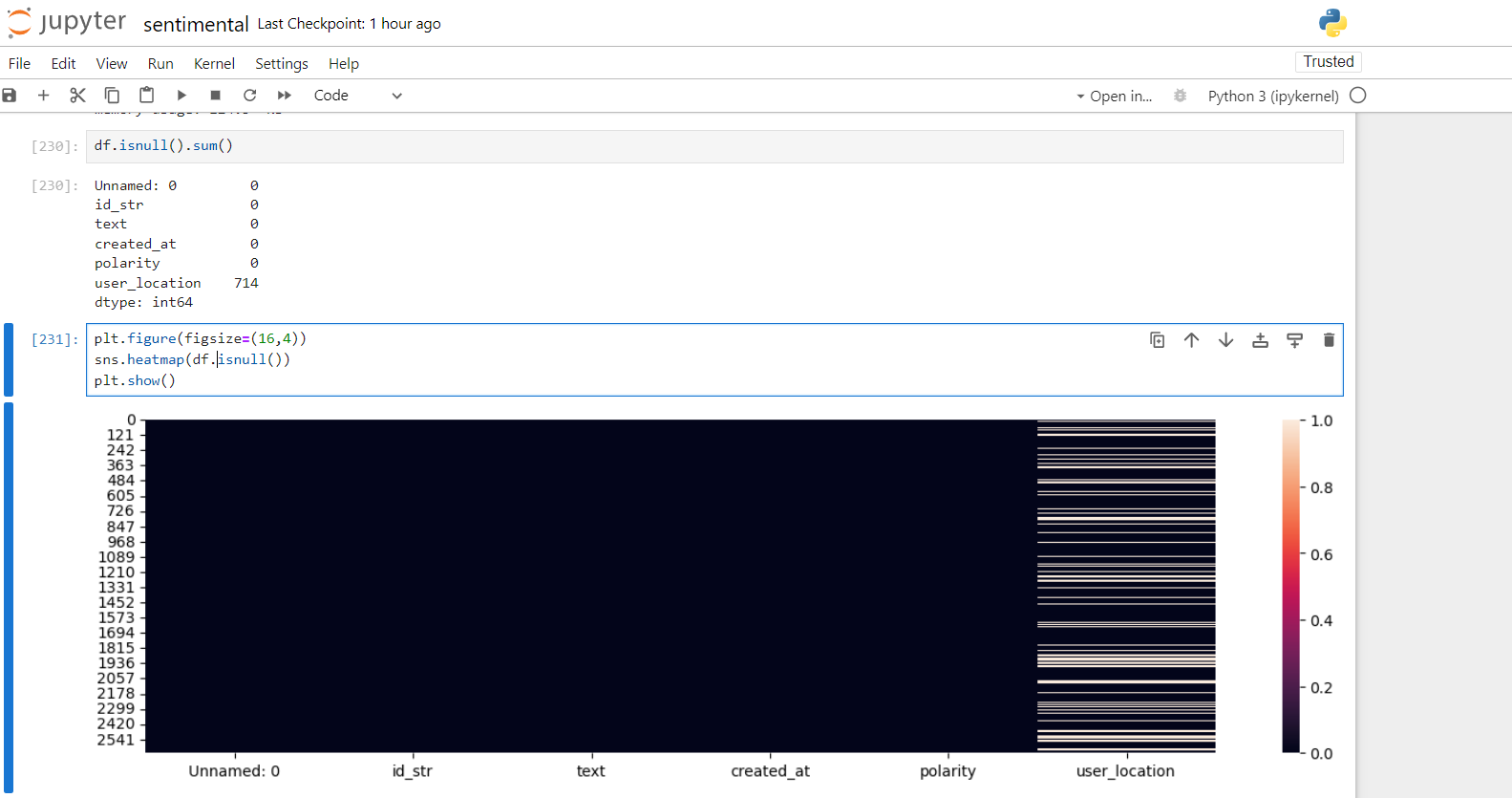
* Analyzed the dataset structure by checking its shape, row names, column names, descriptive statistics, and general information.





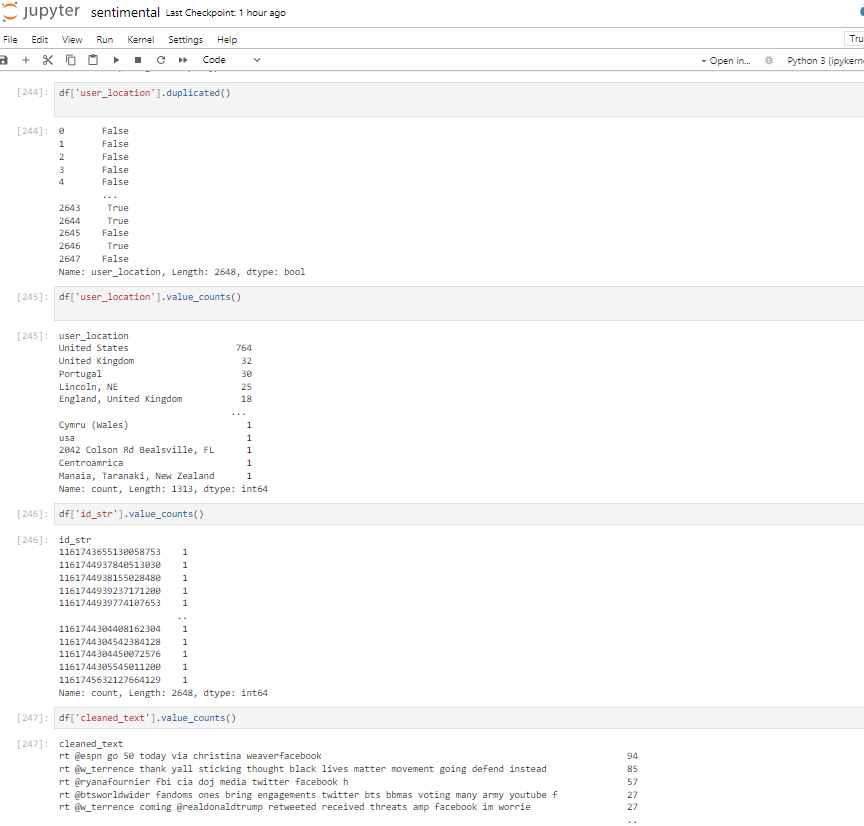
3. Handling Missing Values:

* Checked for null values in the dataset.
* Found null values in the user\_location column, which were visualized using a heatmap.
* Filled the null values in the user\_location column with the mode of the unique user\_location values using the `fillna` method.



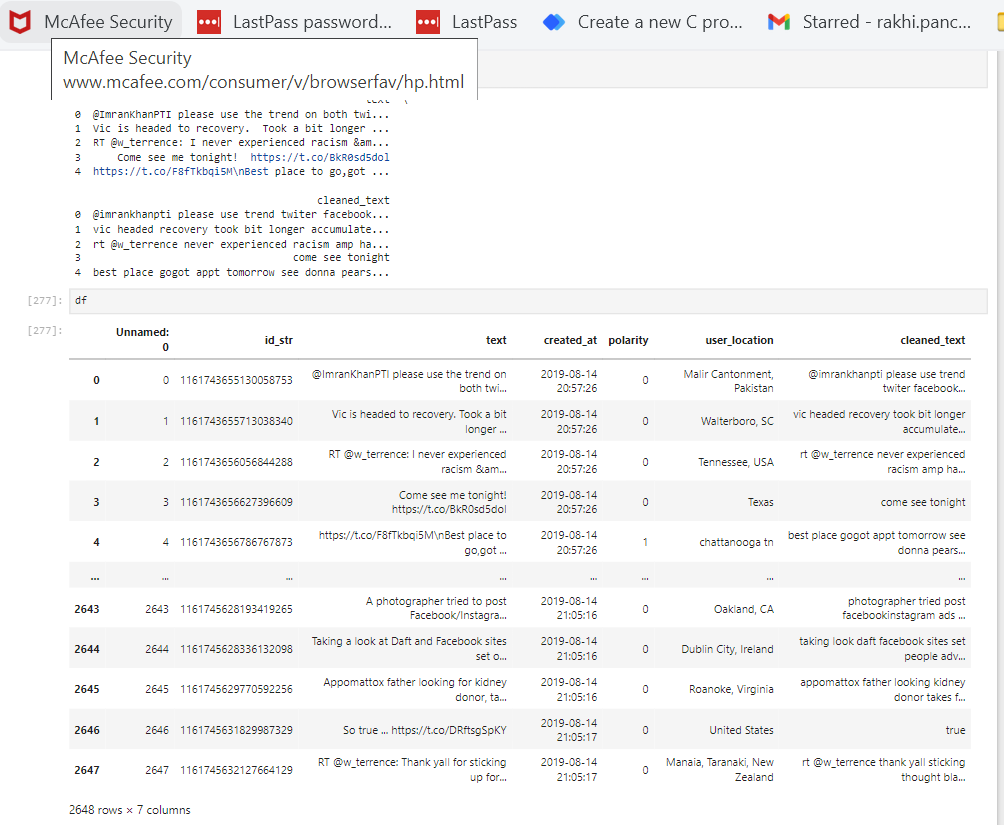
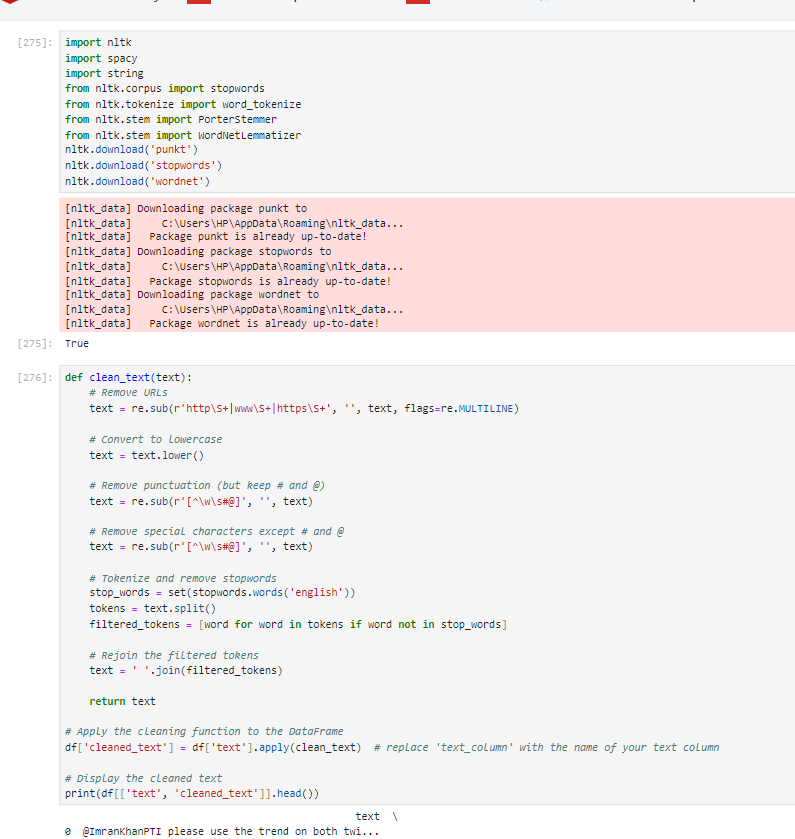
4. Handling Duplicate Values:-

* Identified and removed duplicate values in the dataset, especially focusing on the created\_at column.



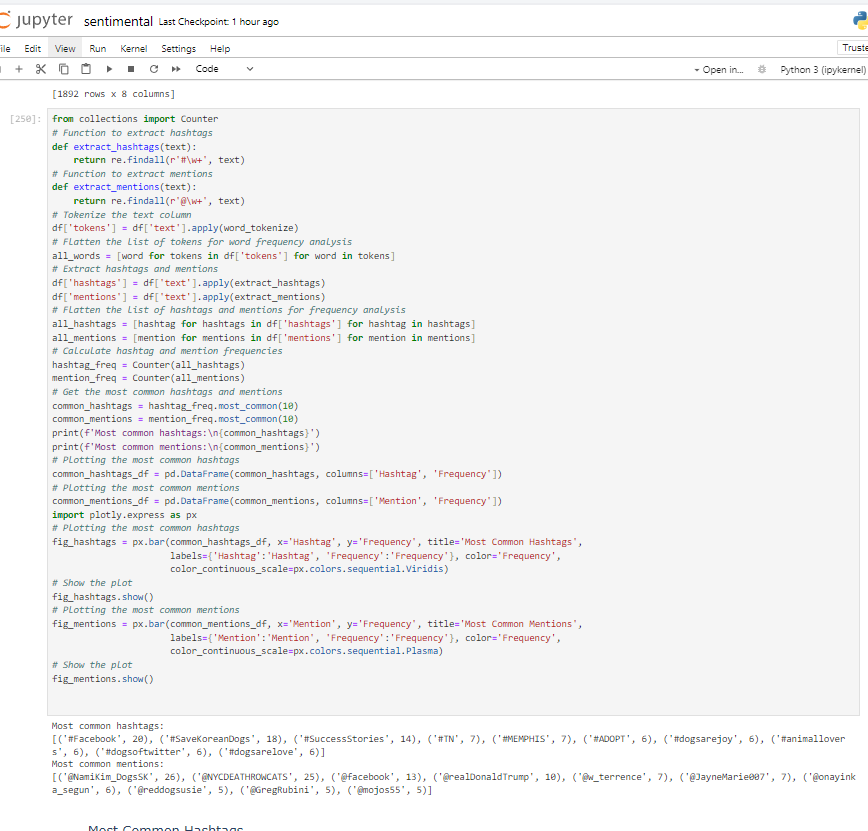
5. Data Cleaning and Preprocessing:-

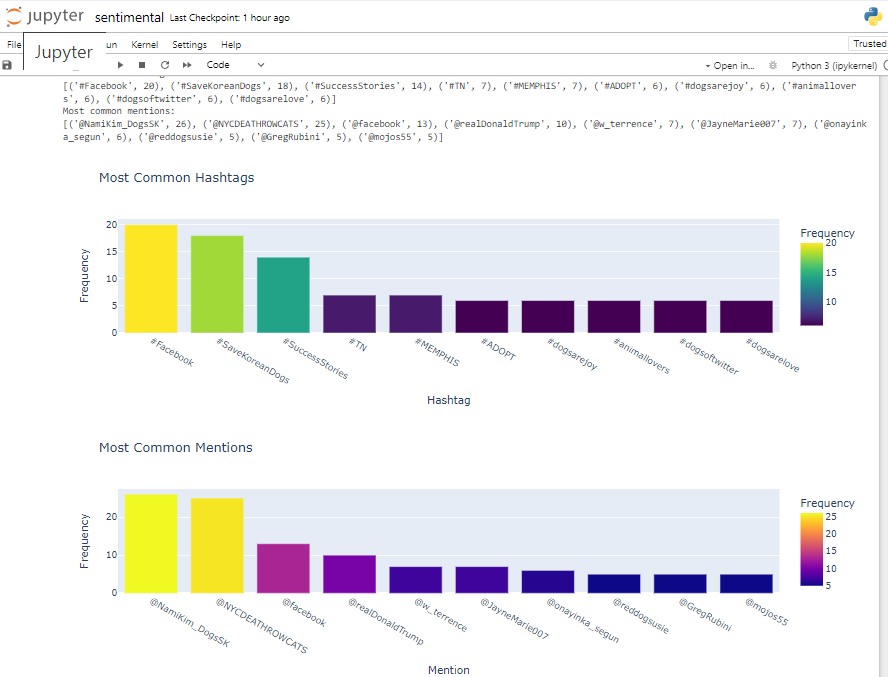
* Imported libraries such as `nltk.corpus` and `nltk.tokenize`.
* Performed tokenization, stopword removal, URL removal, special character removal, and punctuation removal (excluding # and @ symbols for further analysis).
* Created a new column, `cleaned\_text`, containing the cleaned text data.

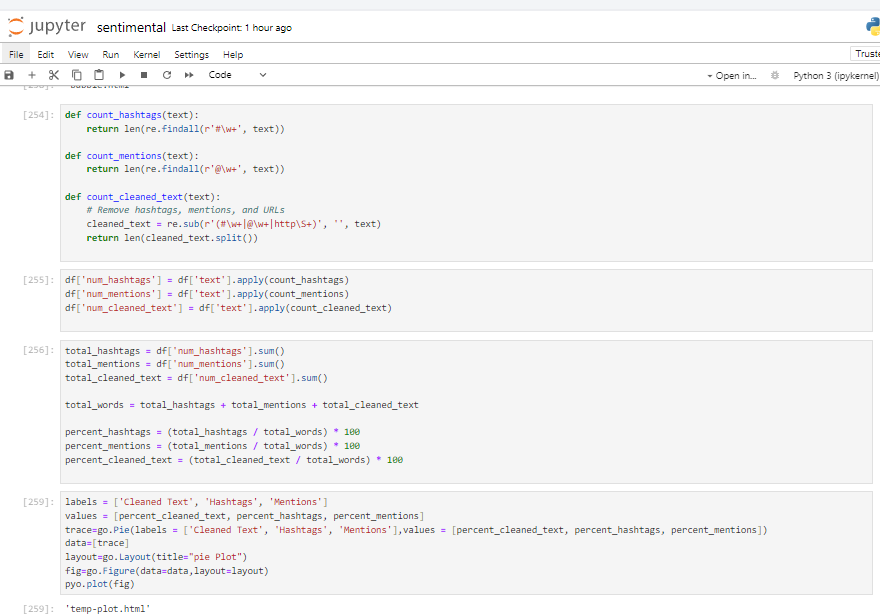


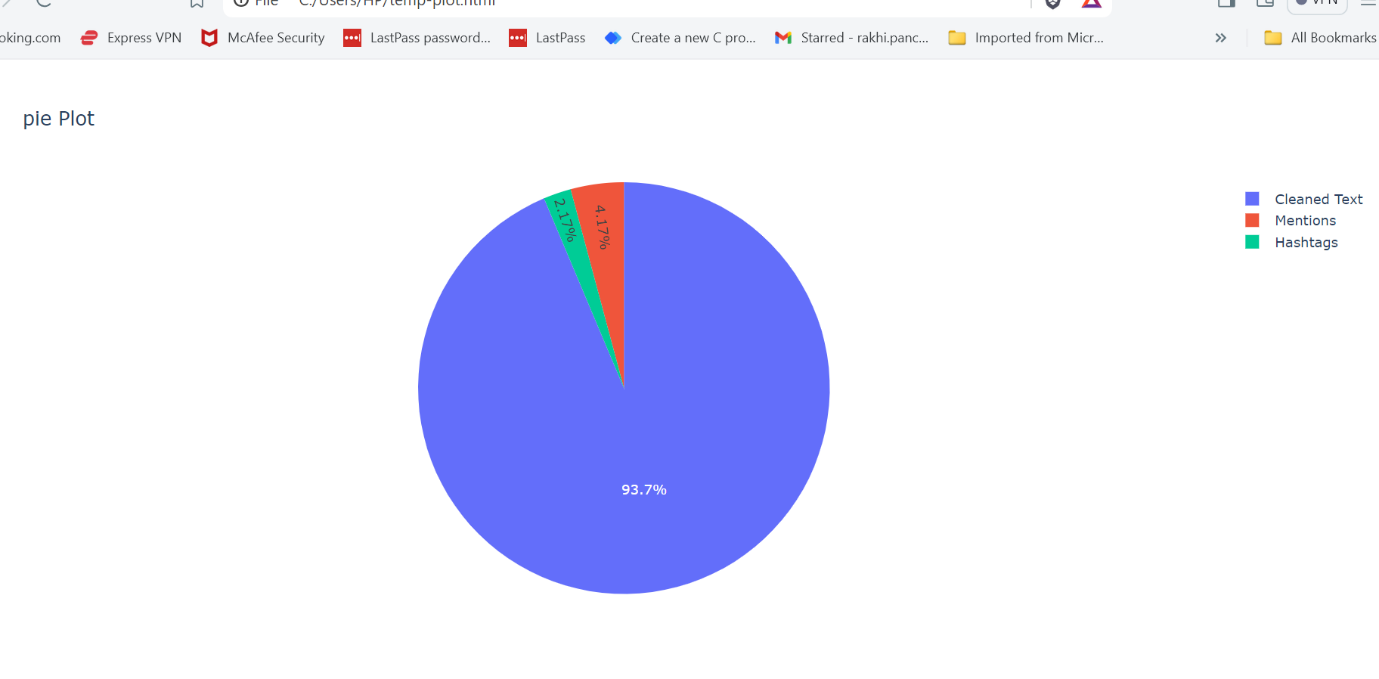
6. Data Visualization:-

* Utilized the Plotly library for creating visualizations.
* Plots Created:-
* Bar Graph: Displayed the most common hashtags in the dataset.
* Bar Graph:Showed the most common mentions in the dataset.
* Pie Chart: Illustrated the proportion of data present in cleaned\_text, mentions, and hashtags.
* Scatter Plot: Mapped `created\_at` against `user\_location` to understand the temporal and spatial distribution of tweets.

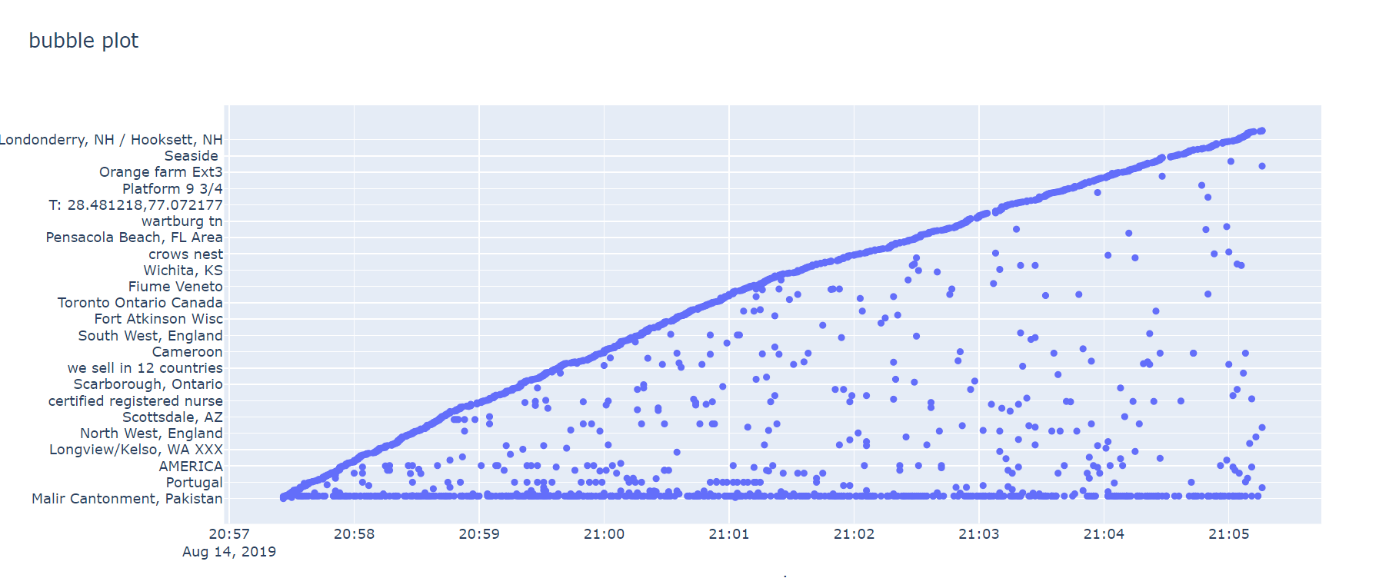












Week 3

Task:-

Deployement of the sentiment analysis model that processes Twitter tweets and classifies them into positive, negative, and neutral sentiments.

Work:-

Tools and Technologies used:-

1. Jupyter notebook.
2. Python
3. Streamlit
4. Commond prompt.
5. NLTK (Natural Language Toolkit)

Libraries:-

Installed the required libraries and tools to set up the project environment, including Streamlit, NLTK and other necessary libraries.

Code Implimentation :-

By using textblob we calculates the polarity of the text. The polarity score ranges from -1.0 to 1.0:

* **Negative Sentiment**: A score closer to -1 indicates negative sentiment.
* **Neutral Sentiment**: A score around 0 indicates neutral sentiment.
* **Positive Sentiment**: A score closer to 1 indicates positive sentiment.

By using TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical data for training a model

TF-IDF (Term Frequency-Inverse Document Frequency) is a widely used technique in text mining and information retrieval to convert text data into numerical features. It combines two key metrics:

* **Term Frequency (TF)**: Measures the frequency of a term in a sentences
* **Inverse Document Frequency (IDF)**: Measures the importance of a term by considering its occurrence across the entire sentences corpus.

Train and Test data:-

1. Split data :-

First, we need to split the data into training and testing sets. The input data consists of cleaned text, and the target data is the polarity score of the text, which indicates its sentiment. This step is crucial to ensure that the model can be evaluated on unseen data to gauge its generalization ability.

* **Cleaned Text**: This is the preprocessed text data that will be converted into numerical features using TF-IDF.
* **Polarity**: This is the target variable, representing the sentiment polarity of each text (e.g., positive, negative, or neutral).

1. Train Logistic Regression Model:-

Next, we train a Logistic Regression model using the training data. The model will learn the relationship between the TF-IDF features of the text and their corresponding polarity scores.

1. Test the Model:-

After training the model, we evaluate its performance on the testing data. This involves predicting the polarity scores for the test set and comparing them to the actual scores. We calculate the model's accuracy and other evaluation metrics to assess its performance.

1. Predict New Data:-

Finally, we use the trained Logistic Regression model to predict the polarity of new text data. This involves transforming the new text data into TF-IDF features and then using the model to predict their polarity scores.

Model Saving :-

To ensure the trained Logistic Regression model can be reused without retraining, we can save it to disk using the pickle library. Pickle is a Python module used for serializing and deserializing Python objects, making it easy to save the model to a file and load it later for predictions.

#### Steps to Save and Load the Model

1. **Train the Model**: Train the Logistic Regression model using your training data.
2. **Save the Model**: Use the pickle library to save the trained model to a file.
3. **Load the Model**: Load the saved model from the file whenever you need to make predictions.
4. **Predict Using the Loaded Model**: Use the loaded model to predict the polarity of new text data.

**Conclusion:-**

Using Jupyter Notebook, Python, Streamlit, Command Prompt, and NLTK, we set up a project environment to train a Logistic Regression model for sentiment analysis using TF-IDF and TextBlob. The model was trained, tested, and evaluated, achieving accurate predictions. We saved the model with pickle for future reuse, ensuring efficient sentiment analysis on new text data without retraining.

Week 4

Task:-

Create an interactive dashboard in Power BI to visualize the sentiment analysis results of twitter data .

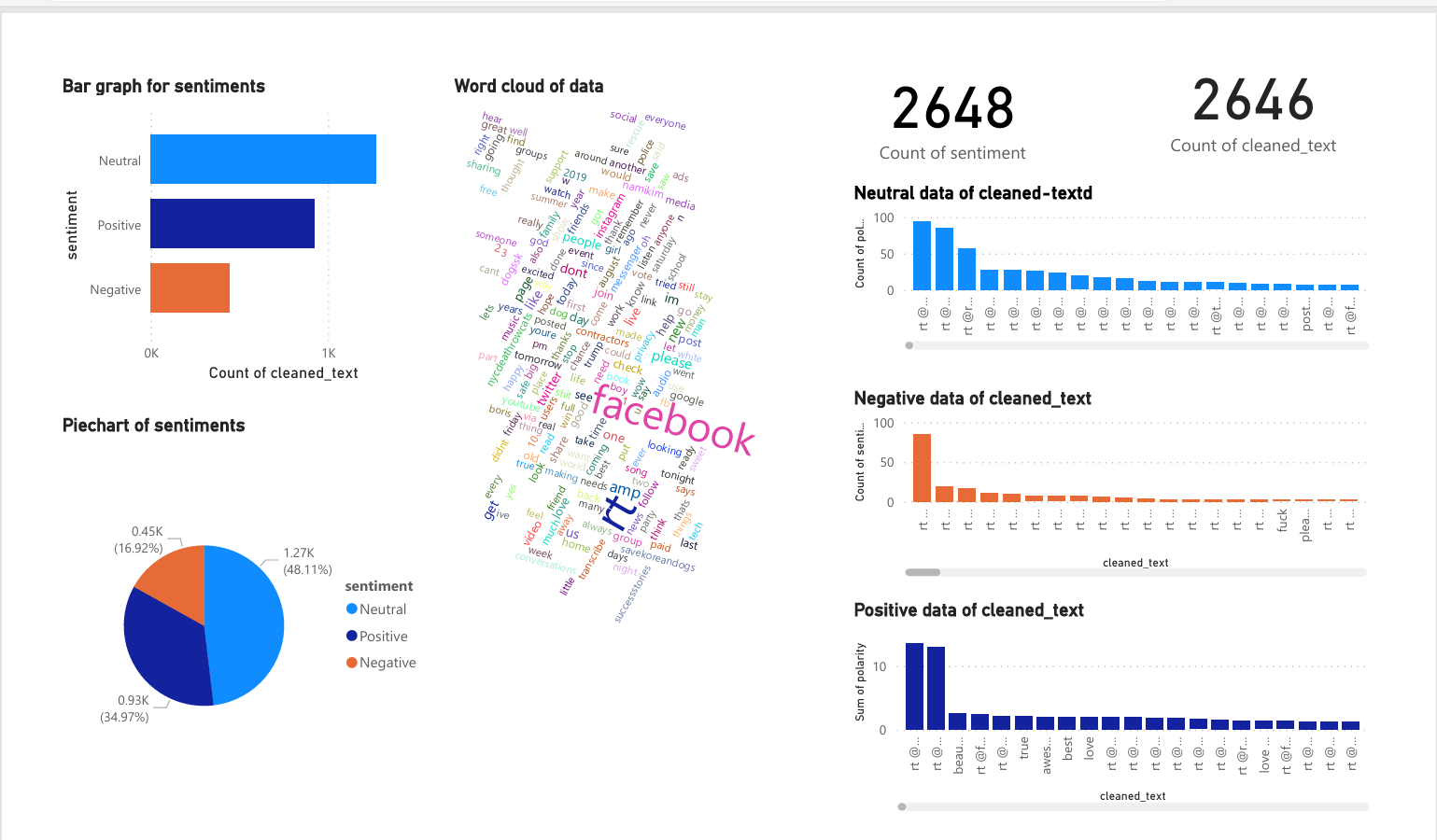
Work:-

1.Firstly we save the csv file of cleaned \_text and polarity of data which we used in the data training and test .

2.we used this csv file for creating the dashboard by using powerbi

3.we will import the csv file into the Powerbi and then we plots the various graphs and charts then we visualized that graphs for important insights.

4.Various types of plots :-



* **Bar Chart**: Utilized a bar chart to illustrate the frequency of comments across sentiment categories (positive, neutral, negative).
  + **Insight**: This visualization aids in comprehending the distribution of sentiments among the comments.
* **Pie Chart**: Constructed a pie chart to visually depict the proportion of each sentiment category.
  + **Insight**: Offers a concise overview of the distribution of sentiments.
* **Word Cloud**: Integrated a word cloud to highlight the most prevalent words used in comments, categorized by sentiment.
  + **Insight**: Provides insights into common themes or topics within each sentiment category.
* **Bar chart : 3 different types of graph.**
* Neutral Sentiments: Graph depicting the frequency or distribution of comments categorized as neutral in sentiment.
* Negative Sentiments: Graph illustrating the frequency or distribution of comments categorized as having a negative sentiment.
* Positive Sentiments: Graph showcasing the frequency or distribution of comments categorized as having a positive sentiment.

**Conclusion:-**

By employing these visualization techniques in Power BI, we gained valuable insights into sentiment patterns within the dataset, facilitating informed decision-making and deeper understanding of audience sentiments. This approach not only enhanced data interpretation but also supported effective communication of findings across stakeholders.