**Project Report Xebia**

**Title**: Sentiment Analysis on social media data

**Student Name**: M.ShivaKumar [Batch-3 SPSU,Udaipur] **Mentor Name**: Sevy Singh

**GitLink*-*** *https://github.com/shiva90140/Sentiment-Analysis-on-Social-media-data*

* **Week-1**:

For my Week-1 task, I focused on Data Collection. Specifically, I gathered Twitter comments data from a particular post using an API called RapidAPI.

Application Programming Interfaces (APIs) are powerful tools that enable direct access to data from a website's database. They facilitate the extraction of specific information without the need for manual data scraping, thus saving time and ensuring accuracy.

Using RapidAPI, I was able to seamlessly collect a rich dataset of comments from the Twitter post. This included details such as the comment text, user information, timestamps, and other relevant metadata. The collected data was then stored in a JSON file, which is a versatile and widely-used format for storing structured data. This JSON file will be crucial for subsequent data analysis tasks, allowing for easy manipulation and extraction of insights.

By leveraging the capabilities of APIs, the data collection process was efficient and scalable, setting a strong foundation for further analysis and application in future tasks.

**Code:**

import http.client

conn = http.client.HTTPSConnection("Twitter-scraper-api2.p.rapidapi.com")

headers = {

'x-rapidapi-key': "30af47468emsh513de8a20194acap1c22a1jsnd81896a2b9d1",

'x-rapidapi-host': "Twitter-scraper-api2.p.rapidapi.com"

}

conn.request("GET", "/v1/likes?code\_or\_id\_or\_url=CxYQJO8xuC6", headers=headers)

res = conn.getresponse()

data = res.read()

import json

# Save data to a JSON file

with open("Twitter\_data.json", "w") as f:

json.dump(data, f, indent=4)

This Python code snippet demonstrates how to collect data from Twitter using the RapidAPI service. Specifically, This code illustrates a typical workflow for making an API call, handling the response, and saving the retrieved data for later use

* **Week-2:**

For Week-2, I am focusing on Data Storing, Preprocessing, and preparing a Dashboard using Streamlit for the data collected in Week-1. Here's the details of these tasks:

**1.Data Storing**:

After collecting the Twitter comments data using RapidAPI in Week-1, the next step is to ensure the data is securely and efficiently stored. The collected data, initially saved in a JSON file, needs to be loaded and possibly stored in a more accessible format, such as a database or a structured file format like CSV. This step ensures that the data is well-organized and easily retrievable for further analysis.

**2.Data Preprocessing**:

Preprocessing the data is a crucial step before any analysis or visualization. This involves:

***Cleaning the Data***: Removing any irrelevant or duplicate entries, handling missing values, and correcting any inconsistencies in the data.

***Transforming the Data***: Converting data types, normalizing text (e.g., converting to lowercase), and extracting relevant features from the raw data (e.g., timestamps, user metadata).

***Structuring the Data***: Organizing the data into a format suitable for analysis, such as data frames or tables, which can be easily manipulated using libraries like Pandas.

**3.Preparing Dashboard using Streamlit**:

Streamlit is an open-source app framework used to create interactive web applications for data science and machine learning projects. For this task:

***Setting Up Streamlit***: Install and set up Streamlit in the working environment.

***Loading the Data***: Load the preprocessed data into the Streamlit app.

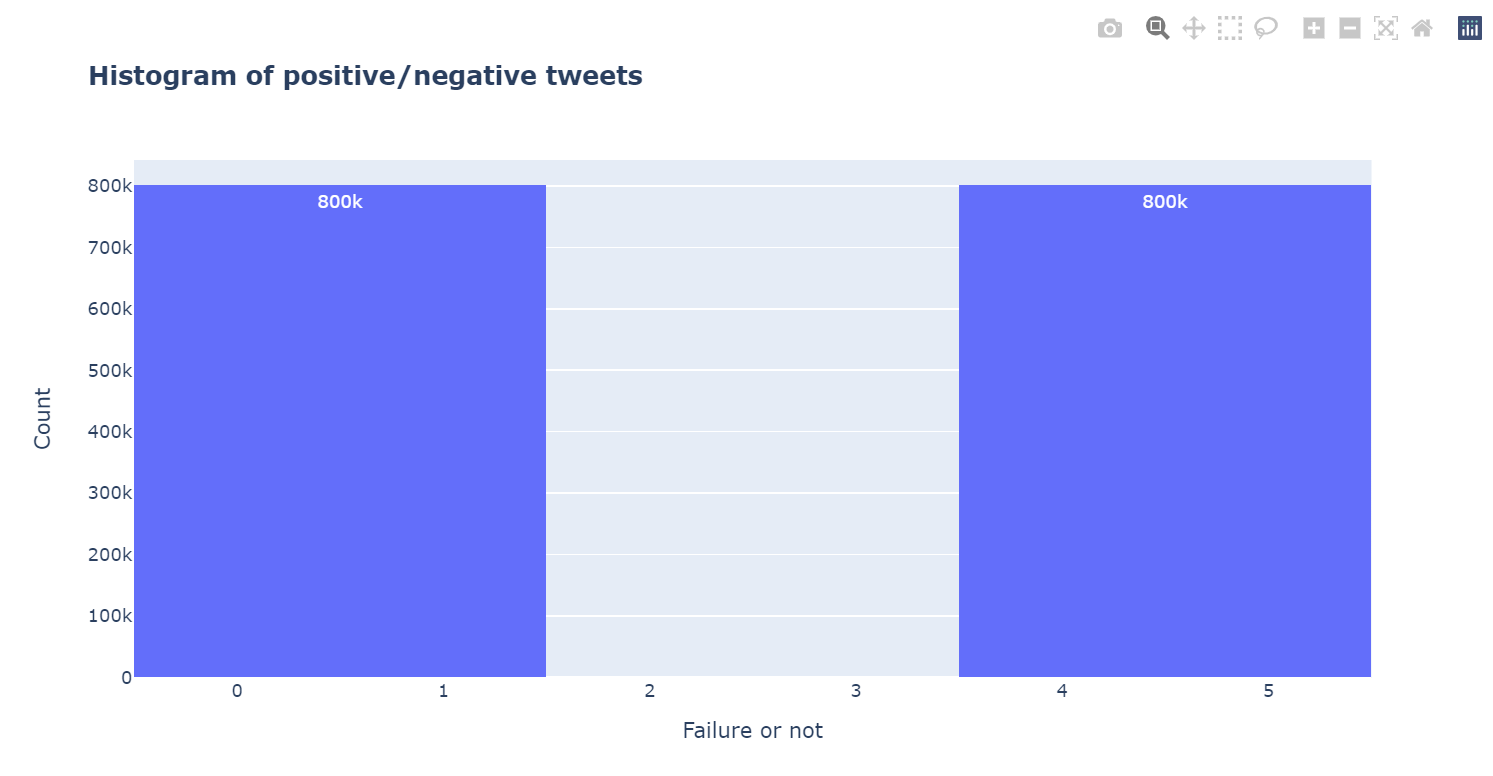
***Designing the Dashboard***: Create an interactive dashboard that visualizes the Twitter comments data. This might include:

***Summary Statistics***: Display basic metrics such as the number of comments, most frequent commenters, and average length of comments.

***Interactive Visualizations***: Use charts and graphs to represent data trends, such as comment frequency over time, word clouds of the most common words, and sentiment analysis of the comments.

***User Interactivity***: Allow users to filter and explore the data dynamically, providing tools to drill down into specific aspects of the data.

This comprehensive approach ensures that the data collected in Week-1 is effectively stored, cleaned, and transformed for meaningful analysis. The interactive dashboard created using Streamlit will provide valuable insights and allow for intuitive exploration of the Twitter comments data.



***Word cloud***

******

* **Week-3**

For Week-3, my primary focus is on Data Analysis and Visualization, building upon the data collected and preprocessed in the previous weeks. Here's an in-depth overview of the tasks involved:

**Data Analysis:**

Data analysis involves examining the preprocessed Twitter comments data to extract meaningful insights. This step includes:

1. **Exploratory Data Analysis (EDA):**
   * ***Descriptive Statistics*:** Calculate basic statistical measures such as mean, median, mode, and standard deviation for the data. This helps in understanding the overall distribution and central tendencies.
   * ***Data Distribution*:** Analyze the distribution of various attributes such as comment lengths, number of comments per user, and frequency of comments over time.
   * ***Correlation Analysis*:** Identify any correlations between different features of the data. For instance, examine if there's a relationship between the time of day and the number of comments received.
2. **Text Analysis:**
   * ***Sentiment Analysis*:** Perform sentiment analysis to classify comments as positive, negative, or neutral. This helps in understanding the overall sentiment of the audience towards the post.
   * ***Keyword Extraction*:** Use techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings to extract significant keywords and phrases from the comments.
   * ***Topic Modeling*:** Apply topic modeling algorithms like LDA (Latent Dirichlet Allocation) to identify common themes and topics discussed in the comments.

**Data Visualization:**

Visualization plays a critical role in presenting the insights derived from the data analysis in an intuitive and engaging manner. For this, I will be using Streamlit to create interactive visualizations. Key visualizations include:

1. **Time Series Analysis:**

* ***Comment Frequency Over Time*:** Plot a time series graph to show the number of comments received over different time periods, highlighting peaks and trends.
* ***Engagement Patterns*:** Visualize patterns in user engagement, such as the times of day or days of the week when comments are most frequent.

1. **Sentiment Distribution:**

* ***Sentiment Pie Chart*:** Create a pie chart to show the proportion of positive, negative, and neutral comments.
* ***Sentiment Over Time*:** Plot the sentiment scores over time to observe how the sentiment has changed during the period under analysis.

1. **Keyword and Topic Visualization:**

* ***Word Clouds:*** Generate word clouds to visually represent the most frequently used words in the comments.
* ***Topic Distribution*:** Visualize the distribution of topics identified in the topic modeling step, showing the prevalence of each topic.

1. **User Interaction:**

* ***Interactive Filters*:** Provide interactive filters in the Streamlit dashboard to allow users to explore the data based on specific criteria, such as filtering comments by sentiment or time period.
* ***Detailed Drill-downs*:** Enable users to click on visual elements to see more detailed information, such as viewing all comments related to a specific topic or keyword.

**Deliverables:**

By the end of Week-3, the goal is to have a comprehensive Streamlit dashboard that not only presents the analyzed data in a visually appealing manner but also allows for interactive exploration. This dashboard will serve as a powerful tool to derive actionable insights from the Twitter comments data.

* **Week-4**

For Week-4, my primary focus is on Model Deployment and Evaluation. This involves deploying machine learning models and assessing their performance using the Twitter comments data. Specifically, I selected two algorithms for this task: Naive Bayes and Logistic Regression. Here’s a detailed overview:

**Model Selection:**

For this project, I chose Naive Bayes and Logistic Regression algorithms due to their effectiveness in text classification tasks, particularly sentiment analysis.

1. **Naive Bayes:**
   * Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence between features. It is particularly well-suited for text data due to its simplicity and effectiveness in handling high-dimensional datasets.
   * The Naive Bayes model achieved an accuracy rate of 74%.
2. **Logistic Regression:**
   * Logistic Regression is a linear model used for binary classification tasks. It predicts the probability of a binary outcome and can be extended to multi-class classification.
   * The Logistic Regression model achieved an accuracy rate of 78%.

**Model Evaluation:**

Evaluating the performance of the models is crucial to understand their effectiveness and reliability. The key evaluation metrics include:

1. **Accuracy:**
   * ***Naive Bayes*:** 74%
   * ***Logistic Regression*:** 78%
   * Accuracy is the proportion of correctly predicted instances out of the total instances.
2. **Confusion Matrix:**
   * Analyze the confusion matrix to assess the number of true positives, true negatives, false positives, and false negatives for each model.
   * This helps in understanding the types of errors made by the models.
3. **Precision, Recall, and F1-Score:**
   * ***Precision*:** The proportion of true positive predictions out of all positive predictions.
   * ***Recall:*** The proportion of true positive predictions out of all actual positives.
   * ***F1-Score*:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
4. **ROC-AUC Curve:**
   * Plot the ROC-AUC curve to evaluate the models' ability to discriminate between classes. A higher AUC indicates better model performance.

**Model Deployment:**

Once the models are evaluated, the next step is deployment. This involves integrating the trained models into the Streamlit dashboard to provide real-time predictions and insights based on new data.

1. **Integrating Models with Streamlit:**
   * Embed the Naive Bayes and Logistic Regression models into the Streamlit app.
   * Provide an interface for users to input new Twitter comments and receive sentiment predictions in real-time.
2. **User Interactivity:**
   * Allow users to switch between the two models to compare their predictions.
   * Display confidence scores and other relevant metrics for each prediction to provide transparency.
3. **Scalability and Maintenance:**
   * Ensure the deployed models are scalable and can handle a large volume of input data.
   * Plan for regular updates and maintenance to improve model accuracy and adapt to new data trends.

* **Conclusion:**

By the end of Week-4, the goal is to have a fully functional and interactive Streamlit dashboard that not only visualizes the analyzed data but also provides real-time sentiment predictions using the deployed Naive Bayes and Logistic Regression models. The evaluation metrics indicate that Logistic Regression performs slightly better with a 78% accuracy rate compared to Naive Bayes' 74%, but both models offer valuable insights for sentiment analysis of Twitter comments