CSE 587: Data Intensive Computing

Project Phase #3

Sentiment Analysis Across Social Media

S.No.	Team Member	UB Name
1.	Aishwarya Nayak	anayak
2.	Rama Krishna Reddy Molakaseema	rmolakas
3.	Sarah Chothiakadavil Abraham	schothia

Problem Statement Recap:

Social media has defined how people share and perceive the news today. The platforms widely used for this purpose include Facebook, GooglePlus, and LinkedIn and are becoming increasingly popular amongst audiences for news transmission as compared to the traditional print and broadcast media. Thus, it becomes important to understand how news trends and popularity evolve on these digital platforms.

To do so, we have utilized a dataset that tracks the popularity of news across various social media platforms and examines the sentiment of the news titles and headlines and provides valuable insights into the changing patterns of news consumption and the impact of news articles in the digital/virtual age.

Overall, this project will help publishers and content producers tailor their content more effectively, support strategic decision-making in the news industry, and contribute to academic and practical advancements in media studies. Furthermore, data analysts, research scientists and other educationalists can leverage these findings and help businesses create content that is positively received by the audience.

Phase 1 Recap:

- **Data Collection:** The dataset was utilized from Kaggle covering approximately 100,000 news items across four topics (economy, Microsoft, Obama, and Palestine) from November 2015 to July 2016. The dataset includes various features such as news item ID, title, headline, source, topic, publication date, and popularity metrics on Facebook, GooglePlus, and LinkedIn.
- **Data Cleaning & Processing:** The initial dataset required extensive cleaning and preprocessing to ensure the quality and reliability of our analyses:
    - Null Value Treatment: Replaced 175 null values in the 'Source' feature with the most common source, 'Bloomberg'.
    - Date and Time Extraction: Separated the publication date and time into distinct columns for more straightforward analysis.
    - Text Normalization: Converted all entries in the 'Source' feature to lowercase to avoid duplicates due to case sensitivity.
    - Data Filtering: Removed news items outside the specified date range and checked for duplicates, confirming none were present.
    - Outlier Treatment: Applied the Quantile Method to manage outliers in the dataset.
    - Sentiment Categorization: Classified the sentiment scores of titles and headlines into positive, negative, and neutral categories.
    - Feature Engineering: Transformed the 'Title' and 'Headline' text into numerical values using CountVectorizer and performed standardization on numerical features to prepare for modeling.
- **Exploratory Data Analysis (EDA):** We conducted thorough EDA to understand the distribution and characteristics of our data:
    - Visualized the distribution of news popularity across social media platforms.
    - Analyzed sentiment distribution in news titles and headlines.
    - Investigated the trends in news popularity, identifying peak times for user engagement on different platforms.

Phase 2 Recap:

- **Feature Selection:** We began by employing an Extra Trees Regressor model to perform feature importance analysis. This step helped us to identify the most important factors in predicting social media engagement across different platforms such as Facebook, LinkedIn, and GooglePlus. The analysis highlighted the headline and publishing hour as significant predictors, while source type showed minimal impact, guiding our model refinement process.
- **Machine Learning Models Used:** We applied six different machine learning algorithms, each chosen for its specific strengths and suitability for our data characteristics:
    - Decision Trees: Known for their simplicity and interpretability. We used Halving Randomized Search CV for hyperparameter tuning to reduce overfitting and improve generalization. Post-tuning, the model showed more realistic performance metrics, suggesting improved model validity.
    - CatBoost: Ideal for handling categorical features, it showed high performance from the outset. The tuning confirmed the model's robustness, though the exceptionally high initial scores suggested potential overfitting.
    - LightGBM: Chosen for its efficiency on large datasets, it underwent similar tuning processes, showing only minor performance changes after tuning, indicating a strong initial fit.
    - Gradient Boosting Regressor: This model demonstrated high flexibility and effectiveness in handling regression tasks. The tuning helped in achieving a balance between fit and generalization.
    - KNN (K-Nearest Neighbors): It provided high accuracy both pre and post-tuning. The model's simplicity was balanced against computational demands, which escalated with increased dataset size.
    - Random Forest: As an ensemble method, it offered robustness and high accuracy, maintaining exceptional performance throughout the tuning process.

- **Model Tuning and Evaluation:** Each model was tuned using Halving Randomized Search CV, focusing on key hyperparameters like max_depth, min_samples_leaf, and learning rate, among others. The effectiveness of tuning was assessed through changes in performance metrics such as MSE, RMSE, R2, and Adjusted R2, which generally showed improvements or confirmed the models' capabilities.
- **Visualization:**
    - **Feature Importance:** Bar chart visualization was done using Extra Trees Regressor output which highlighted the headline and publishing hour as key influencers, guiding us to focus on structural features for model tuning.
    - **Hyper-parameter Tuning:** Parallel coordinate plots and multi-dimensional visualizations for each model's tuning were obtained which helped in identification of optimal parameter combinations, crucial for enhancing model performance.

Phase 3 (Data Product Development): This phase leverages the LightGBM model, chosen for its effectiveness in handling large datasets and its efficiency in training, to predict social media engagement for news content. The objective of this phase is to design a user interface using Streamlit that allows users to dynamically adjust hyperparameters, evaluate model performance, and download the configured model as a .pkl file for their chosen social media platform along with visualization of sentiment distribution.

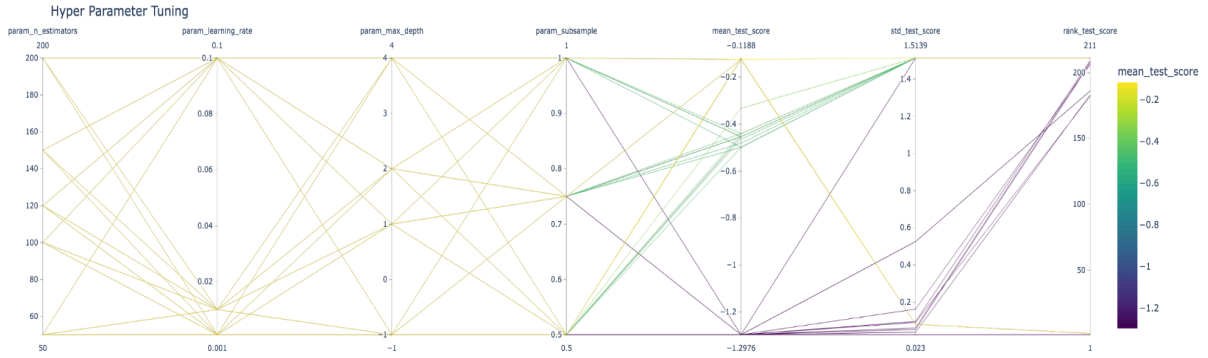
- **Choice of LightGBM:** From our extensive model evaluation in Phase 2, LightGBM stood out due to its superior performance and scalability. Its gradient-based learning and efficient handling of categorical features make it well-suited for our dataset, which involves diverse variables from various social media platforms. LightGBM's ability to provide quick results without compromising on accuracy is especially valuable for an interactive application where response time is crucial.
- The model underwent hyperparameter tuning with Halving Randomized Search CV. This method is known for being efficient with large parameter spaces, as it narrows the search space in each iteration by discarding less promising parameter combinations. The hyperparameters included in the tuning process are:

- **n_estimators:** The number of boosting stages to be run, essentially the number of trees in the forest.
- **learning_rate:** The step size shrinkage used to prevent overfitting.
- **max_depth:** The maximum depth of the trees.
- **subsample:** The fraction of samples to be used for fitting the individual base learners.

- The best parameters obtained can thus be found in the screenshots below.

```
        Tuning Model: Grid_Search_CV

LGBMRegressor for Facebook

[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001357 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 262
[LightGBM] [Info] Number of data points in the train set: 11795, number of used features: 19
[LightGBM] [Info] Start training from score 0.002401

Evaluation of LGBMRegressor before tuning:
------------------------------------------------
        MSE      RMSE  R2_Score  Adjusted_R2_Score
0  0.000002  0.001231  0.999998           0.999998
```

```
Best Score for LGBMRegressor : -4.741022665888433e-05
Best Parameters for LGBMRegressor : {'learning_rate': 0.1, 'max_depth': -1, 'n_estimators': 120, 'random_state': 25, 'subsample': 0.5}


Evaluation of LGBMRegressor after tuning:
------------------------------------------------
       MSE     RMSE  R2_Score  Adjusted_R2_Score
0  0.000002  0.001248  0.999998           0.999998
```



- **Development of Streamlit UI:** The Streamlit framework was chosen for its simplicity and effectiveness in building interactive apps directly from Python scripts. The UI allows users to:
  - Select a Social Media Platform: Users can choose from Facebook, LinkedIn, or GooglePlus, determining the dataset the model will predict on.
  - Configure Hyperparameters: Through intuitive sliders and dropdown menus, users can adjust key hyperparameters like n_estimators,

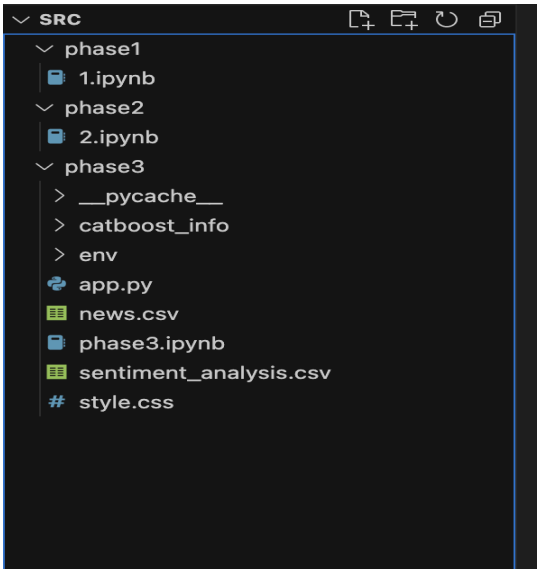
learning_rate, max_depth, and subsample. These controls make the model tuning accessible to users without deep technical knowledge of machine learning.

- View Evaluation Metrics: After running the model with selected parameters, users can immediately see key metrics such as MSE, RMSE, R2, and Adjusted R2. This instant feedback helps in understanding the impact of their configurations.
- View Pie Chart: Users can also view the sentiment distribution for the chosen social media platform in the form of a pie chart thereby denoting each sentiment with a specific color i.e., Positive with green, Neutral with blue and Negative with red.
- Download Configured Model: Once satisfied with the settings, users can download the trained model as a .pkl file, enabling them to deploy or further analyze it offline.

- **Utility and Target Audience:** This interactive data product is designed to serve multiple user groups:
  - **Understanding Model Behaviour:** Data Scientists and Analysts can experiment with different model configurations to find optimal settings for specific datasets. The tool allows for the visualization of sentiments through pie charts, providing insights into the distribution of different sentiments within the data.
  - **Sentiment Analysis:**
    - Educators and Students in data science can use this tool to see how sentiments are classified and visualized and can use this further for practical learning.
    - Non-technical Stakeholders such as media strategists and content producers can understand how different factors influence the visibility and engagement of news content on various platforms, aiding in more informed decision-making.

Product Usage Instructions:

- **System Setup:**
  - Required Files:
    - Ensure that you have the app.py, news.csv, sentiment_analysis.csv, phase3.ipynb and style.css files stored in the phase 3 folder.
    - Also, for using VSCode for the first time to run the python files, select the appropriate kernel before running the programs.

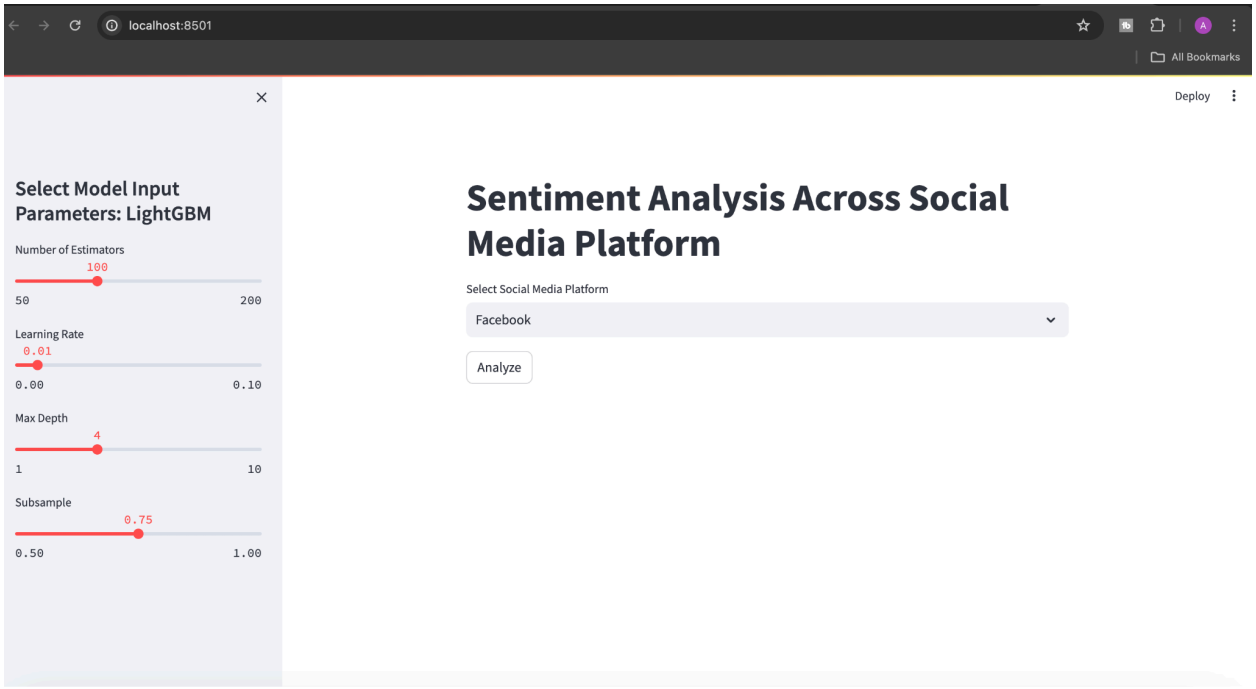


  - Python Environment Setup:
    - Create a new Python environment specific for this project to avoid any conflicts with existing libraries.
    - Make sure you are in the 'phase3' folder.
    - Activate the Python environment using the command in the screenshot below.

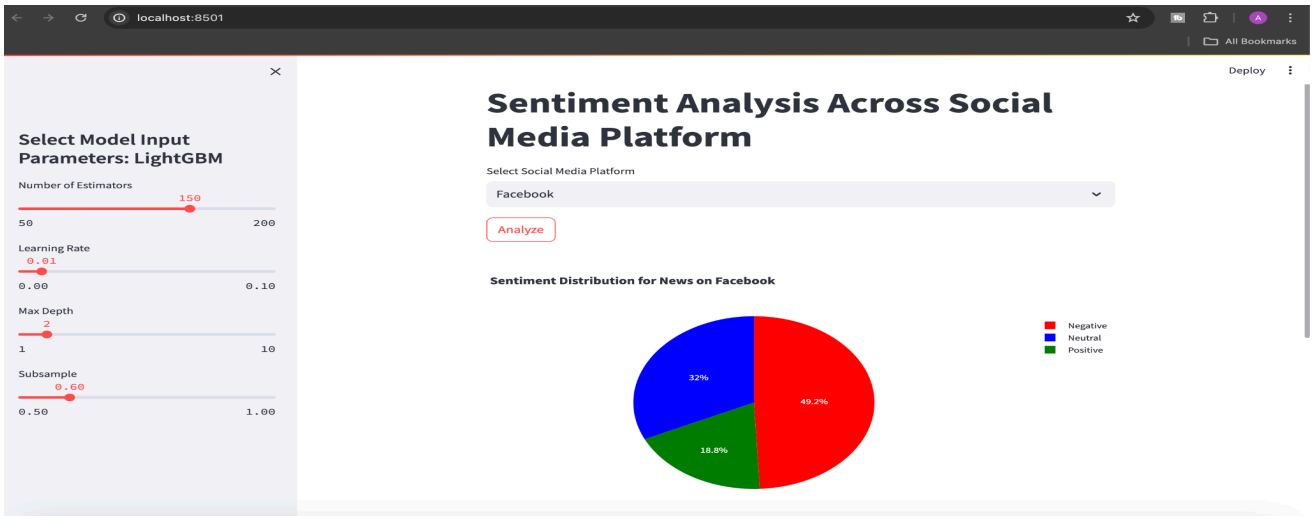
```
(base) aish@Aishwaryas-MacBook-Air-3 phase3 % python3 -m venv env && source ./env/bin/activate
```

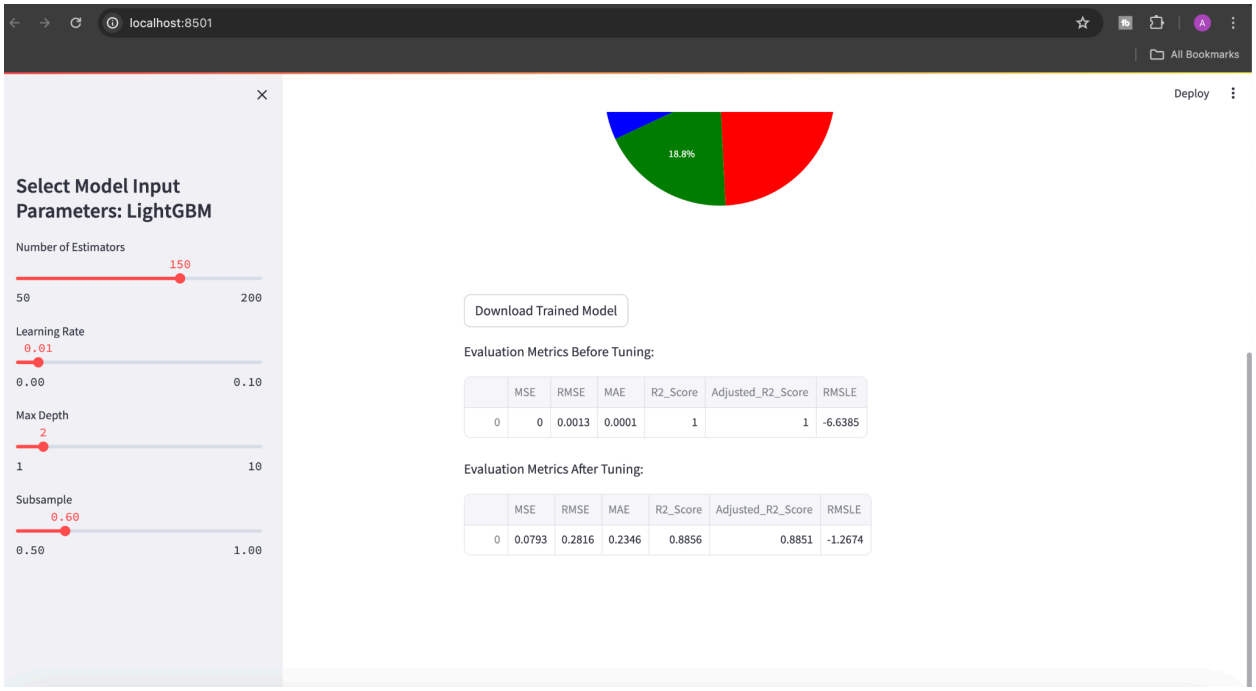
  - Installation:
    - Install Streamlit in your Python environment using the following command: **pip install streamlit**
    - Check if additional packages are required for the Streamlit app to function correctly and install or update them as needed.

- **Running the Streamlit App:**
  - Launch the App:
    - Navigate to the directory containing app.py if not already.
    - Run the following command in your terminal: **streamlit run app.py**
    - This command will start the Streamlit server, and it will automatically open your default web browser to localhost:8051 where the app is hosted.

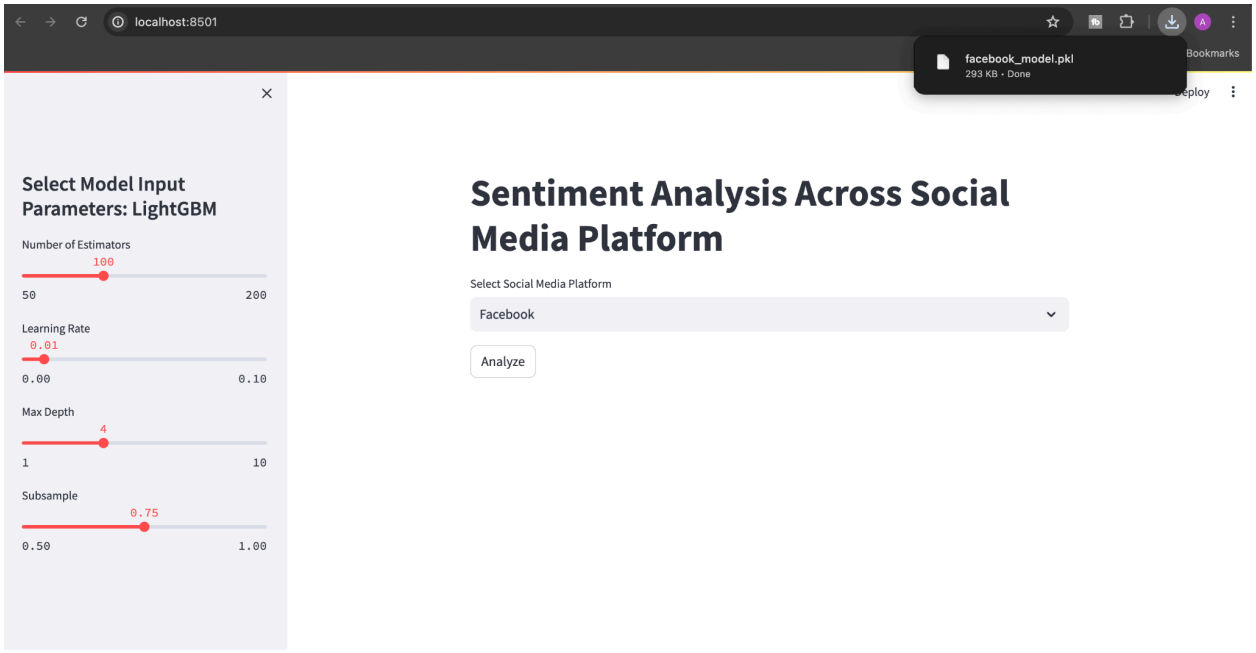


  - Interacting with the UI:
    - The Streamlit interface allows you to select hyperparameters, choose the social media platform for which you want to predict engagement, and view the performance of the model along with a plot for sentiment distribution.
    - After configuring the settings, you can run the model to see the evaluation metrics directly in the browser.

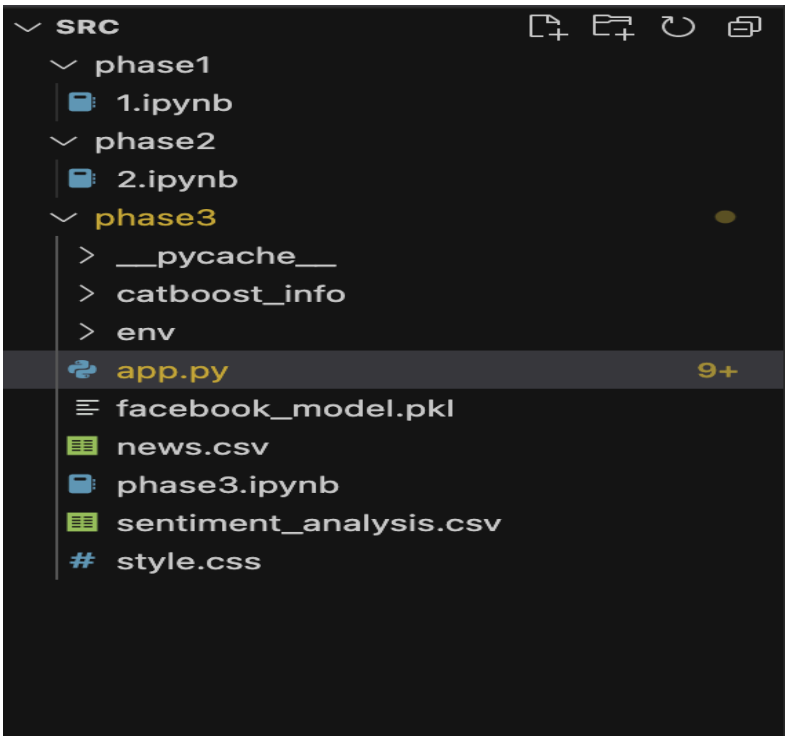




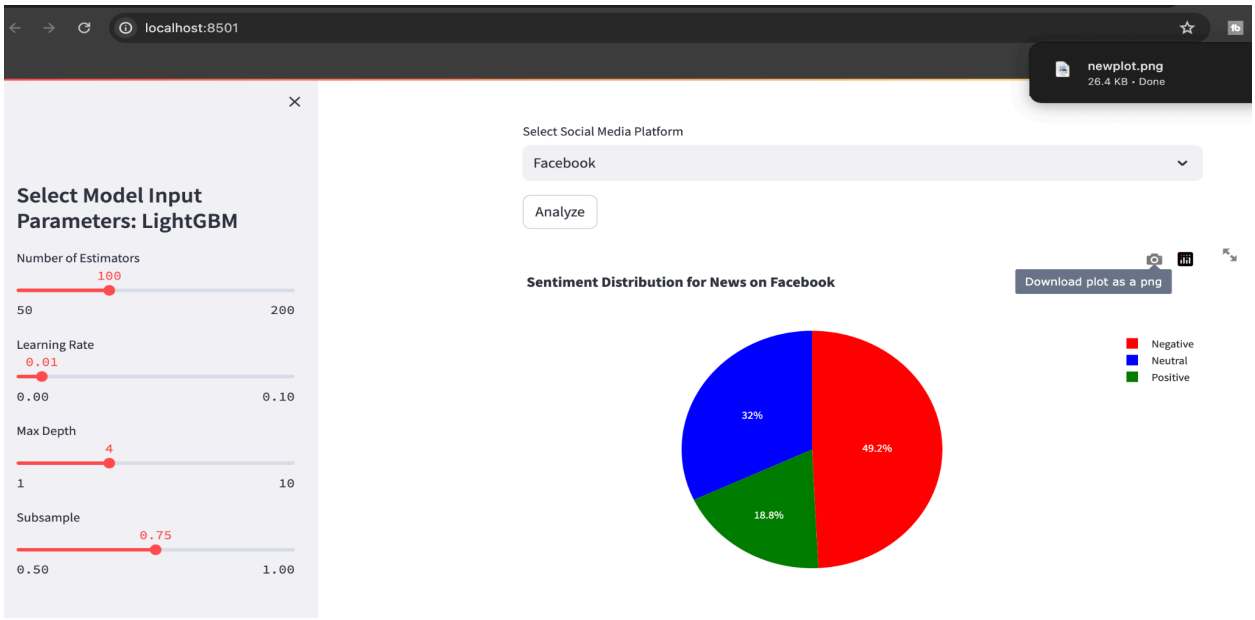
- Downloading Models:
  - If you are satisfied with the model's performance, you can download the trained model as a .pkl file by clicking the download button provided in the UI.



- The .pkl file will also be saved in the phase 3 directory. If the file already exists, it will be overwritten with the new model based on the latest user interaction, ensuring that you always have the most up-to-date model configuration.



- You can even save the pie-chart for the chosen social media platform as a png file.



- **Troubleshooting:**
  - Package Installation: Sometimes, specific Python packages may require updates or reinstallation. If you encounter any errors related to package dependencies, try updating the package using:
    pip install --upgrade [package-name]
  - Environment Issues: If the Streamlit app doesn't run as expected, ensure that all required packages are correctly installed in the active Python environment and that there are no conflicting software versions. Also, consider restarting the session if needed.
- **Note:**
  - The sentiment_analysis.csv file is preprocessed (phase3.ipynb) and included in the project folder to streamline the data loading process. This approach eliminates the need for users to upload datasets and allows the Streamlit app to quickly load and display the data without additional preprocessing steps.
  - For the development of this phase, we have utilized VSCode, which might have led to some very minor code changes in the name of python code compatibility wherein the logic will still stand intact.

Recommendations/ Scope of Future Work: To extend our project's functionality in future, we can leverage the following ideas:

- Posting Time Optimization: We can analyze the engagement trends to identify the best times to post on each platform to maximize visibility and interaction.
- Applying Advanced NLP Techniques: We can use sophisticated natural language processing (NLP) techniques to analyze the sentiment and emotional tone of content more accurately.
- User Behavior Analysis: We can further incorporate user demographic and behavior analysis to tailor content and posting strategies.
- AI-Driven Content Generation: We can explore the use of AI to suggest or even generate optimized headlines and content based on historical data and trends. etc.

Conclusion: In summary, our project "Sentiment Analysis Across Social Media" involved undergoing different stages of the development lifecycle and was finally utilized to design a user-friendly Streamlit application that enables dynamic interaction with the model. There is still scope to enhance this project further by utilizing more sophisticated NLP techniques and AI-driven content strategies which could significantly expand the project's impact.

References:

- News Popularity Dataset on Kaggle
- News Popularity in Multiple Social Media Platforms - UCI ML Repository
- News Popularity Prediction in Social Media Capstone Project (CatBoost, LightGBM, Gradient Boosting)
- Decision Trees and Random Forest: Data Science from Scratch by Joel Grus, An Introduction to Statistical learning in R