**Week 3 Task Report: Modelling and Deployment on Sentiment Analysis Project**

**Enrolment:21CS002377**

**Name: Gudapelly Shrihari**

**Branch: CSE (AI & ML) [Batch 2]**

**Mentor : Sevy**

**Modelling**

The sentiment analysis project employed several machine learning models to classify the sentiment of text data into positive, negative, or neutral categories. Here’s a brief overview of the modeling process and results:

**1. Data Splitting**

The cleaned text data was split into training and testing sets, with 80% of the data used for training and 20% for testing. This ensured that the models were evaluated on unseen data to gauge their real-world performance.

**2. TF-IDF Vectorization**

To convert the text data into a numerical format suitable for machine learning, TF-IDF (Term Frequency-Inverse Document Frequency) vectorization was used. This method transformed the text into a matrix of TF-IDF features, highlighting the importance of words in each document relative to the entire dataset.

**3. Model Selection and Training**

Several machine learning models were selected for training, including:

* **Passive Aggressive Classifier**: Known for its efficiency in text classification tasks, especially in handling large and sparse data.
* **Logistic Regression**: A straightforward and interpretable model, suitable for binary and multiclass classification.
* **Random Forest Classifier**: An ensemble learning method that builds multiple decision trees and merges them for better accuracy and control overfitting.
* **Multinomial Naive Bayes**: Particularly well-suited for text classification tasks, due to its simplicity and effectiveness with large datasets.

**4. Model Evaluation**

The performance of each model was evaluated using the following metrics:

* **Accuracy**: The proportion of correctly classified instances out of the total instances.
* **Precision, Recall, F1-Score**: Detailed metrics providing insights into the performance of each sentiment category.

**Results:**

* **Passive Aggressive Classifier**: Achieved the highest accuracy of 72.73%, demonstrating robust performance in classifying sentiments.
* **Logistic Regression**: Obtained an accuracy of 63.27%, performing reasonably well but with room for improvement.
* **Multinomial Naive Bayes**: Scored an accuracy of 61.90%, showing good performance, particularly in recognizing certain sentiment categories.

**5. Hyperparameter Tuning**

For the Passive Aggressive Classifier, hyperparameter tuning was conducted using RandomizedSearchCV to find the optimal parameters. This process involved testing various combinations of hyperparameters to improve the model’s performance.

**Best Hyperparameters for Passive Aggressive Classifier:**

* C: 1.0
* fit\_intercept: True
* shuffle: False
* verbose: 0

The tuned model provided an improved accuracy and more balanced classification performance across sentiment categories.

**6. Confusion Matrix**

A confusion matrix was used to visualize the performance of the best model (Passive Aggressive Classifier) in terms of true positives, true negatives, false positives, and false negatives for each sentiment category. This helped in identifying any specific areas where the model might be underperforming.

**7. Model Deployment**

**Model Deployment on GitHub**

**1. Preparing the Repository:**

* Created a new repository on GitHub named sentiment-analysis.
* Organized the repository structure with directories for code, data, models, and documentation.

**2. Uploading the Code:**

* Uploaded all the Python scripts and Jupyter notebooks used for data preprocessing, sentiment analysis, and model training.
* Included all the necessary Python libraries for easy setup (e.g., twython, vaderSentiment, pandas, numpy, etc.).
* Added a README.md file with a project overview, setup instructions, and usage guidelines.
* Included detailed comments within the code for clarity.

**3. Uploading the Dashboard**

* Uploaded the Dashboard and Use the dashboard to upload new text data and analyze sentiments.
* Visualize the results using the interactive plots and charts.

**Links**

**Links**

**GitHub Repository**

* https://github.com/gudapellyshrihari/Xebia-Internship-Project

This deployment setup aims to provide a comprehensive and user-friendly tool for sentiment analysis, accessible to both developers and non-technical users.

**Conclusion**

The modeling phase revealed that the Passive Aggressive Classifier was the best-performing model for this sentiment analysis task. The hyperparameter tuning further enhanced its accuracy. The insights from the model evaluations and visualizations provided a comprehensive understanding of the text data's sentiment distribution, supporting the overall goal of accurately classifying sentiments in text data. Deploying the sentiment analysis model to a GitHub repository ensures accessibility and ease of collaboration. The interactive dashboard enhances user experience by providing a user-friendly interface to analyze text data and visualize results.