DATASET REPORT

**Week – 2 Task Report : Modelling and Deployment on Sentiment Analysis**

**Enrolment Number : 21CS002362**

**Name : Bairagoni Manikanth**

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

**Mentor : Sevy**

**Modelling Progress:**

This week, we made significant progress in modelling. We experimented with three different classification models: Logistic Regression, Decision Tree Classifier, and Random Forest Classifier. We trained and tested each model on our preprocessed dataset, and evaluated their performance using metrics such as precision, AUC score, and F1 score. Our results showed that the Random Forest Classifier outperformed the other two models, achieving an F1 score of 0.992425352244649 on the training dataset and 0.736053044545597 on the test dataset.

**Modelling:**

This week, we experimented with three different classification models: Logistic Regression, Decision Tree Classifier, and Random Forest Classifier. We trained and tested each model on our preprocessed dataset, and evaluated their performance using metrics such as precision, AUC score, and F1 score.

**Logistic Regression:**

We trained a Logistic Regression model with a penalty of L2 and a tolerance of 0.001. The model achieved a precision score of 0.619 on the test dataset and an AUC score of 0.619. We also performed hyperparameter tuning for the Logistic Regression model, tuning the maximum number of iterations and the regularization parameter. The best hyperparameters were found to be a maximum number of iterations of 200 and a regularization parameter of 0.01.

**Decision Tree Classifier:**

We trained a Decision Tree Classifier with a maximum depth of 13 and a minimum sample size of 3. The model achieved a precision score of 0.631 on the test dataset and an AUC score of 0.631. We also performed hyperparameter tuning for the Decision Tree Classifier, tuning the maximum depth of the tree and the criterion for splitting. The best hyperparameters were found to be a maximum depth of 13 and a criterion of Gini.

**Random Forest Classifier:**

We trained a Random Forest Classifier with 100 estimators and a maximum depth of 13. The model achieved a precision score of 0.724 on the test dataset and an AUC score of 0.823. We also performed hyperparameter tuning for the Random Forest Classifier, tuning the number of estimators, the maximum depth of the tree, and the criterion for splitting. The best hyperparameters were found to be 100 estimators, a maximum depth of 13, and a criterion of Gini.

**Hyperparameter Tuning:**

We also performed hyperparameter tuning for each model to optimize their performance. For the Logistic Regression model, we tuned the maximum number of iterations and the regularization parameter. For the Decision Tree Classifier, we tuned the maximum depth of the tree and the criterion for splitting. For the Random Forest Classifier, we tuned the number of estimators, the maximum depth of the tree, and the criterion for splitting. Our results showed that hyperparameter tuning significantly improved the performance of each model.

We used GridSearchCV to perform hyperparameter tuning for each model. GridSearchCV is a method of performing hyperparameter tuning that involves training and evaluating a model on multiple combinations of hyperparameters, and selecting the combination that results in the best performance.

**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/Manikanth0107/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.