Task 3: Sentiment Analysis

Sharlene Moetjie – Data Science Intern

Objective:

- The objective of this sentiment analysis task is to classify movie reviews as either positive or negative. By leveraging machine learning techniques, the goal is to develop a model that accurately classifies and predicts the sentiment of a review based on its textual content. This involves training and evaluating the model to ensure it effectively distinguishes between positive and negative sentiments.

1.Project Selection

- The aim of this project is to build a sentiment analysis model that can automatically classify movie reviews into positive or negative categories. The process involves text preprocessing, vectorization, model training, and evaluation to achieve accurate sentiment classification.

2.Data Selection

- The dataset, sourced from Kaggle, contains movie reviews, each labelled with a sentiment of either POSITIVE or NEGATIVE. The data consists of two columns: the textual content of the reviews and their corresponding sentiment labels. The dataset is well-balanced (containing 5000 textual content of each sentiment label), allowing for effective training and evaluation of the model.

3. Model Development

- I used a systematic process approach that included: data preprocessing, feature extraction, model training and evaluation. The implementations are as follows:
  - **Data Collection and Preprocessing:**
    1. **Text Cleaning**: The raw text data was cleaned by converting to lowercase, removing URLs, punctuation, digits, and non-alphabetic characters, and eliminating any HTML tags or unnecessary spaces.Used statistical measures to compare legitimate and fraudulent transactions.
    2. **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets to evaluate the model's performance on both seen and unseen data.
  - **Train_Test_Split:** To split the dataset into training and testing sets to evaluate the model's performance on seen and unseen data.
  - **Vectorization**:
    - The text data was vectorized using TF-IDF (Term Frequency-Inverse Document Frequency), which transformed the text into numerical features suitable for model input. Stop words were removed, and the maximum document frequency

was set to 0.7 to filter out common terms that might not contribute to the sentiment prediction.

- **Model Training:** A Logistic Regression model was trained on the vectorized training data. The model was selected for its simplicity and effectiveness in binary classification tasks.
- **Model Evaluation:** The model was evaluated using the training data to assess its performance metrics such as accuracy, precision, recall, and F1-score.

5.Training and Evaluation

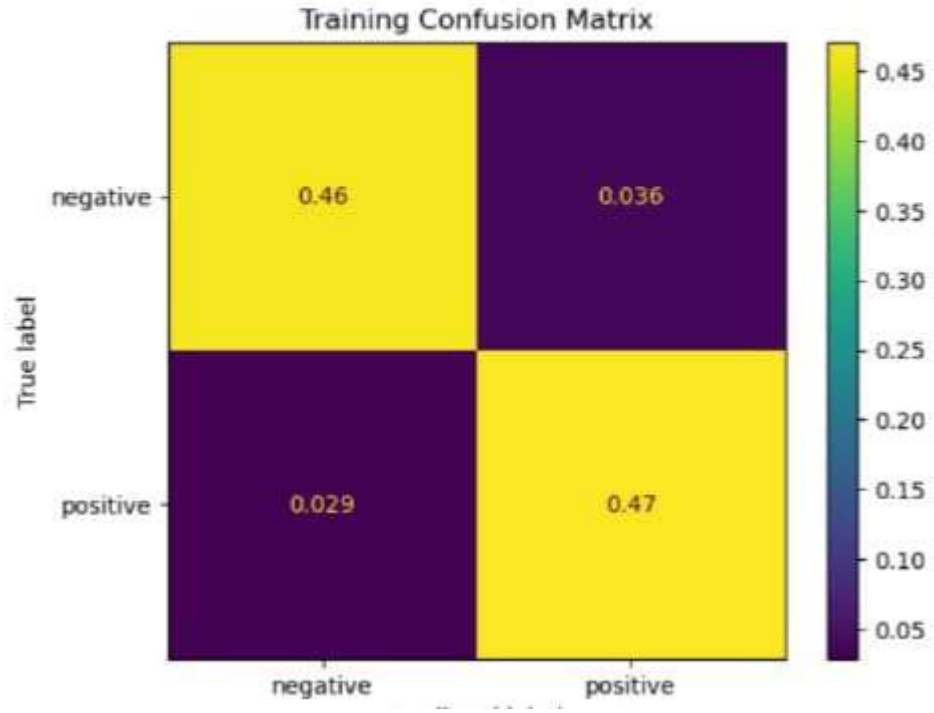
- The Logistic Regression model achieved a training accuracy of **93.5%**. The precision, recall, and F1-score for both positive and negative classes were balanced, indicating that the model was well-fitted to the training data. The model demonstrated strong performance on the testing set, with a testing accuracy of **93.5%**. The precision, recall, and F1-score metrics remained consistent with the training phase, confirming the model's ability to generalize well to new, unseen data.

Results Presentation

CLASSIFICATION REPORT ON TRAINING (SEEN) DATA:

Training Accuracy: 0.93495

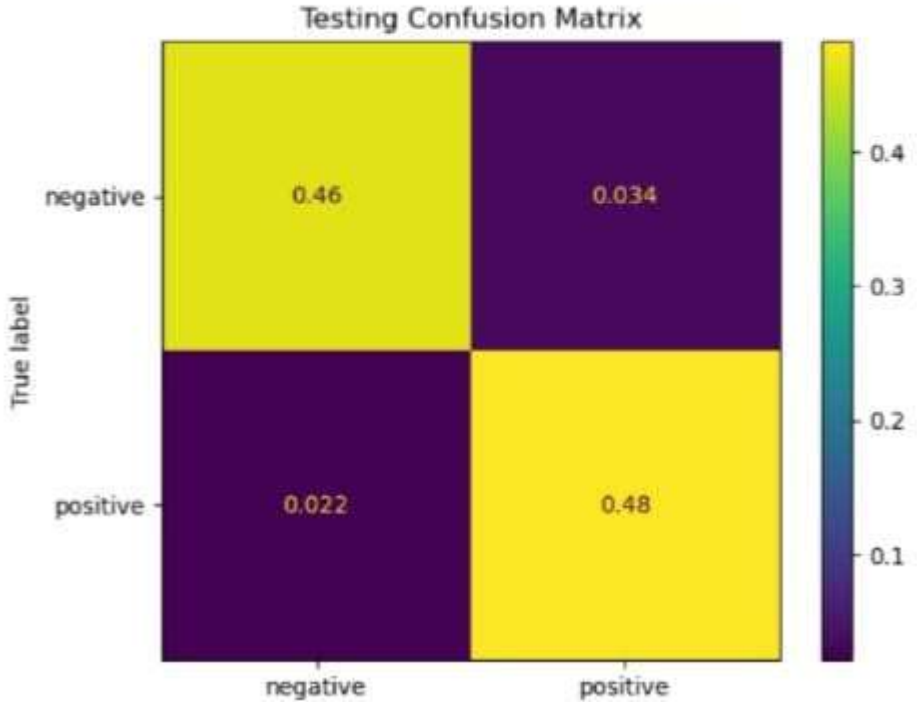
	precision	recall	f1-score	support
negative	0.94	0.93	0.93	20039
positive	0.93	0.94	0.94	19961
accuracy			0.93	40000
macro avg	0.94	0.93	0.93	40000
weighted avg	0.94	0.93	0.93	40000



CLASSIFICATION REPORT ON TESTING (UNSEEN) DATA:

Testing Accuracy : 0.93495

	precision	recall	f1-score	support
negative	0.95	0.93	0.94	4961
positive	0.93	0.96	0.95	5039
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000



Summary

- The results indicate that the sentiment analysis model performs effectively on both the training and testing datasets. With a high accuracy of 93.5% on both sets, the model shows a strong ability to generalize to new, unseen data. The consistency between training and testing metrics suggests that the model is well-balanced and not overfitting.

References

- Great Learning. Sentiment Analysis In 10 Minutes | Sentiment Analysis Using Python | Great Learning. [video]. YouTube. [Sentiment Analysis In 10 Minutes | Sentiment Analysis Using Python | Great Learning (youtube.com)]

- NeuralNine. Simple Sentiment Analysis in Python. [video]. YouTube. [What is ClickUp? VA (youtube.com)]
- DataCamp. Sentiment analysis and prediction in Python | Live Code-Along. [video]. YouTube. [Sentiment analysis and prediction in Python | Live Code-Along - YouTube]