AMAZON PRODUCT SENTIMENT ANALYSIS

CS6120 37424 Natural Language Processing SEC 02 Spring 2024

Group 18: Srinivas Peri, Vinod Kumar Boyalla & Bezawit Ayalew

GitHub

1.0 Introduction

The paper studies the sentiment analysis of Amazon product reviews using NLP to better understand client feedback in e-commerce. Recognising the value of sentiments in reviews for customer insights and market trends, we encounter issues such as determining post-sales contentment and combating fraudulent reviews. We use the NLTK library to process and categorize attitudes from textual data, concentrating on techniques such as SVM, logistic regression, CNNs, and LSTMs. We evaluate the efficiency of these strategies, identify their strengths and flaws, and explain their implications for the e-commerce business, with the goal of improving platform trust and user experience.

2.0 Methodology

This portion of the study describes the approach used to train and assess several sentiment analysis models using Natural Language Processing (NLP) techniques on the provided Amazon product reviews dataset. It also examines how these models process and interpret data.

2.1. Data Preprocessing:

- Data Cleaning: The Amazon product reviews dataset is initially loaded and then cleaned to remove noise and extraneous information. This usually includes activities like deleting HTML tags, punctuation, and stop words.
- Text Representation: Texts are converted to numerical representations suited for machine learning techniques. This methodology uses the TF-IDF (Term Frequency-Inverse Document Frequency) approach. This technique provides weights to terms depending on their frequency within a document and throughout the full corpus.
- Label Preparation: Labels representing the sentiment (positive or negative) for both training and testing data are created to ensure compatibility with the model training and assessment stages.

2.2. Model Training:

Support Vector Machine (SVM): SVM is a supervised learning algorithm used for classification tasks. In this methodology, a linear SVM model is trained using the TF-IDF transformed training data. The model learns to classify reviews into positive or negative sentiment categories based on the extracted features.

Logistic Regression: Logistic Regression is another classification algorithm commonly used in sentiment analysis tasks. Similar to SVM, a Logistic Regression model is trained on the TF-IDF transformed training data. It learns to predict the probability of a review belonging to a particular sentiment class.

Multi-layer perceptron (MLP): An essential neural network architecture for sentiment analysis. Trained on TF-IDF data, it categorizes reviews by adjusting weights and biases. Renowned for flexibility and effectiveness.

Convolutional Neural Network (CNN): CNNs are deep learning models particularly effective in processing spatially structured data such as images and, in this case, sequential data like text. A CNN model is constructed using the Keras Sequential API and trained on padded sequences of the training data. The model learns hierarchical representations of text features through convolutional layers, enabling it to capture patterns at different levels of abstraction.

LSTM: Long Short-Term Memory (LSTM): LSTM is a deep learning architecture that uses recurrent neural networks (RNNs) to analyze input sequences. It is meant to recall significant information while forgetting irrelevant data, making it ideal for activities with sequential dependencies, such as language modeling and time-series analysis.

- Embedding Text: To capture semantic content, words are converted into fixed-size vectors (in this case, 128).
- Learning from Sequences: Using LSTM layers to process these vectors in sequence, remembering important patterns like word order or context.
- Predicting Outcomes: The final output layer with a single neuron uses a 'sigmoid' activation function to predict binary outcomes (e.g., positive or negative sentiment).

2.3. Model Evaluation:

- Testing Data Evaluation: The trained models are evaluated on the TF-IDF transformed testing data, which they haven't seen during training. This step ensures the generalization ability of the models.
- Performance Metrics: Accuracy scores are computed for each model to assess their performance in classifying sentiments. Other metrics such as precision, recall, and F1-score may also be considered to provide a comprehensive evaluation of model performance.
- Each model configuration is trained on the dataset and then evaluated to see how accurately it can predict new, unseen data. The 'sigmoid' activation ensures the output is a probability between 0 and 1.

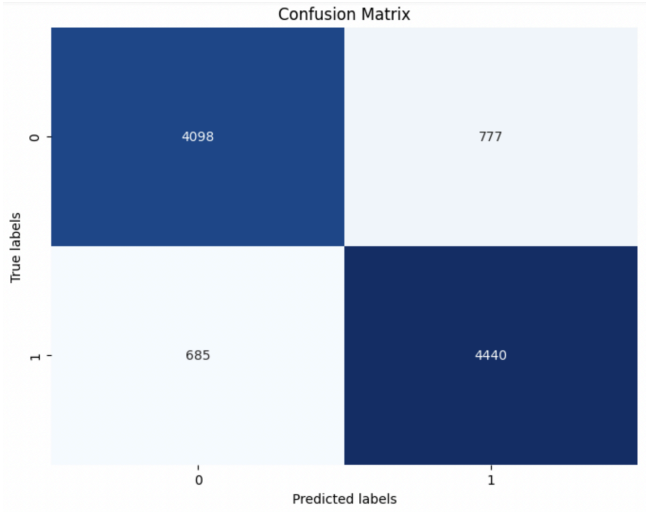
3.0 Results & Evaluation

In this section, we provide a complete evaluation of the performance of our models, which include logistic regression, MLP (Multi-Layer Perceptron), and SVM. Each model was rigorously tested using a variety of evaluation measures to determine its predictive power and generalization performance.

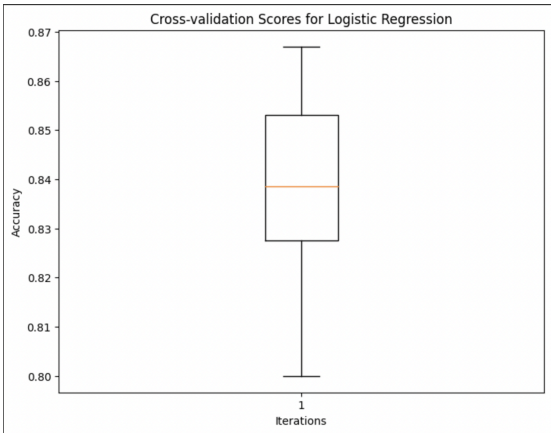
We begin by evaluating the logistic regression model's performance. Our review includes a thorough examination of the confusion matrix, classification report, and cross-validation score visualization, which provide insights into the classification accuracy and overall dependability. After evaluating the logistic regression model, we expand our research to include the MLP and SVM models. Similar to logistic regression, both models were thoroughly evaluated using the aforementioned criteria to determine their performance in classification tasks.

3.1 Logistic Regression:

The logistic regression model shows a decent performance:



	precision	recall	f1-score	support
0	0.86	0.84	0.85	4875
1	0.85	0.87	0.86	5125
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

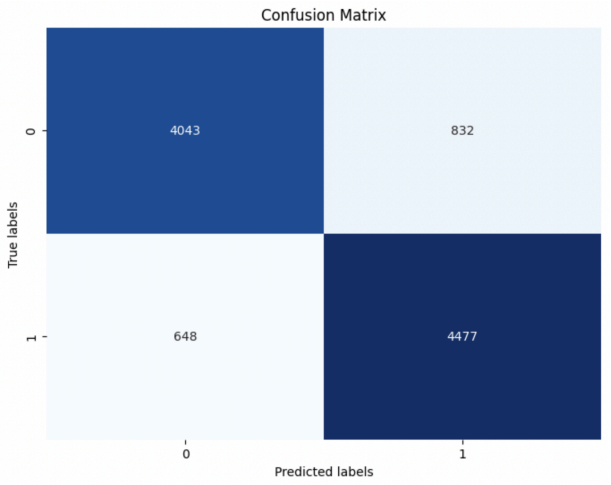


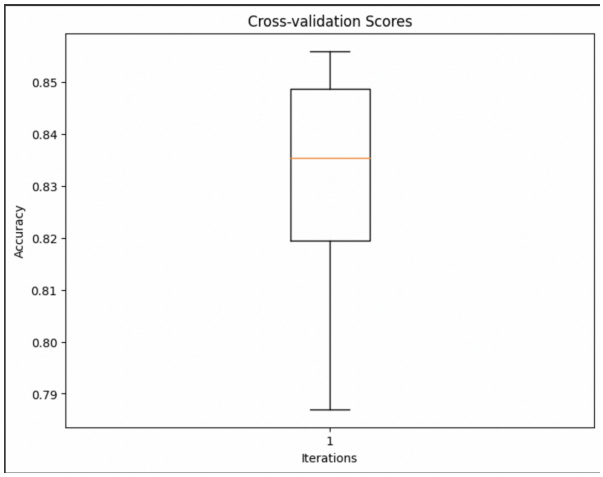
- Precision: Class 0: 86% accuracy in predicting class 0. Class 1: 85% accuracy in predicting class 1.
- Recall: Class 0: 84% of actual class 0 instances were correctly classified. Class 1: 87% accuracy in identifying class 1 instances.
- F1-score: Achieved a balanced performance with scores of 0.85 for class 0 and 0.86 for class 1.
- Accuracy: Classifier achieved an overall 85% correctness in predictions.

The strong precision, recall, and F1-scores in both classes demonstrate its resilience. As a consequence, the model achieves an accuracy of 85%, demonstrating its ability to reliably categorize instances from the dataset. Despite its overall high performance, the model's capacity to handle class imbalances should be improved further. Class 1 has somewhat higher recall than Class 0, indicating a possible bias or unequal representation in the sample.

3.2 Support Vector Machine:-

This evaluation report below is for a Support Vector Machine (SVM) classifier.





	precision	recall	f1-score	support
0	0.86	0.82	0.84	2435
1	0.84	0.87	0.85	2565
accuracy			0.85	5000
macro avg	0.85	0.85	0.85	5000
weighted avg	0.85	0.85	0.85	5000

- Precision: Class 0: 86% of instances predicted as class 0 were correct. Class 1: 84% accuracy in predicting class 1.
- Recall: Class 0: 83% of actual class 0 instances were correctly classified. Class 1: 87% accuracy in identifying class 1 instances.
- F1-score: Achieved a balanced performance with scores of 0.85 for class 0 and 0.86 for class 1.
- Support: Dataset includes 4875 instances of class 0 and 5125 instances of class 1.
- Accuracy: Classifier achieved an overall 85% correctness in predictions.

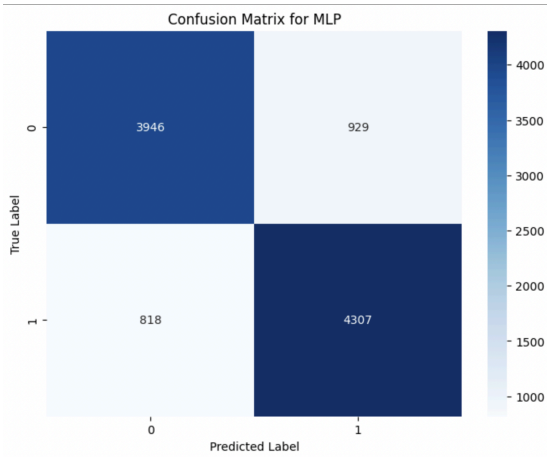
The box plot for SVM shows:

- Most scores fall between 0.815 and 0.854.
- The median score is around 0.837.
- Whiskers stretch from approximately 0.787 to 0.856, indicating the range of scores.
- Outliers may exist beyond the whiskers.

Overall, the SVM classifier performs well across precision, recall, and F1-score parameters, with an accuracy of 85%. These findings indicate that the classifier efficiently distinguishes between the two categories, producing valid predictions for both.

3.3 Multi- Layer Perceptron:-

The classification report for the MLP model shows strong performance:




```
Classification Report for MLP:
              precision    recall  f1-score   support

           0       0.83      0.81      0.82      4875
           1       0.82      0.84      0.83      5125

    accuracy                           0.83     10000
   macro avg       0.83      0.82      0.83     10000
weighted avg       0.83      0.83      0.83     10000
```

- Precision: Both classes have high precision scores (0.83 for class 0 and 0.82 for class 1), indicating few false positives.
- Recall: The model effectively captures instances from both classes, with recall scores of 0.81 for class 0 and 0.84 for class 1.
- F1-Score: Balanced F1-scores (0.82 for class 0 and 0.83 for class 1) indicate good overall performance.
- Accuracy: The model achieves an accuracy of 83%.
- Support: Class 0 has 4875 instances, and class 1 has 5125 instances.

In summary, the MLP model demonstrates strong classification performance with high precision, recall, and F1-scores across both classes.

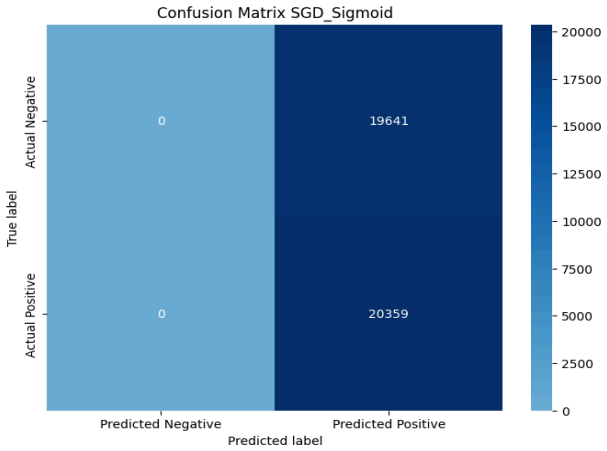
3.4 LSTM Evaluation:-

Configurations of LSTM model with varying optimizers and Activation parameters:

Configuration: Optimizer=<keras.src.optimizers.sgd.SGD, Activation=sigmoid, LSTM Layers=2
Test Loss: 0.692818820476532, Test Accuracy: 0.5089750289916992

Classification Report SGD_Sigmoid:

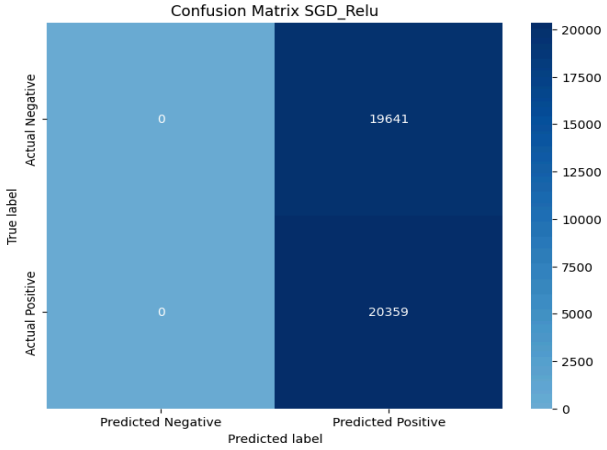
	precision	recall	f1-score	support
0	0.00	0.00	0.00	19641
1	0.51	1.00	0.67	20359
accuracy			0.51	40000
macro avg	0.25	0.50	0.34	40000
weighted avg	0.26	0.51	0.34	40000



Configuration: Optimizer=<keras.src.optimizers.sgd.SGD, Activation=relu, LSTM Layers=2 Test Loss: 7.487751483917236, Test Accuracy: 0.5089750289916992

Classification Report SGD_Relu:

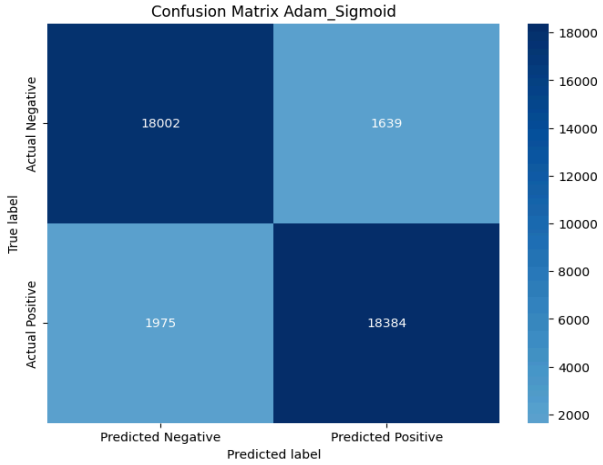
	precision	recall	f1-score	support
0	0.00	0.00	0.00	19641
1	0.51	1.00	0.67	20359
accuracy			0.51	40000
macro avg	0.25	0.50	0.34	40000
weighted avg	0.26	0.51	0.34	40000



Configuration: Optimizer=<keras.src.optimizers.adam.Adam, Activation=sigmoid, LSTM Layers=2
Test Loss: 0.2229316383600235, Test Accuracy: 0.9096500277519226

Classification Report Adam_Sigmoid:

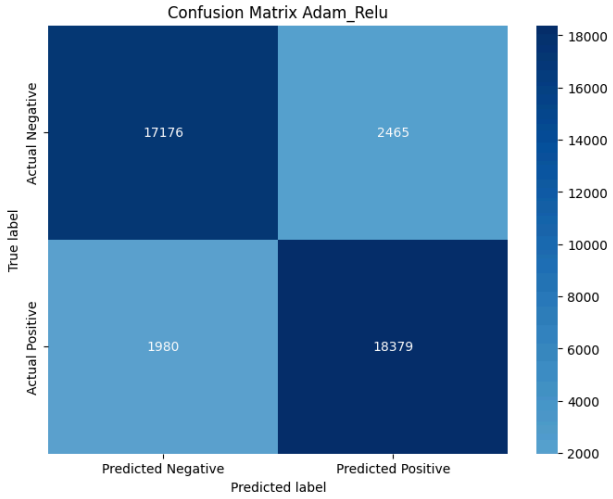
	precision	recall	f1-score	support
0	0.90	0.92	0.91	19641
1	0.92	0.90	0.91	20359
accuracy			0.91	40000
macro avg	0.91	0.91	0.91	40000
weighted avg	0.91	0.91	0.91	40000



Configuration: Optimizer=<keras.src.optimizers.adam.Adam, Activation=relu, LSTM Layers=2
Test Loss: 0.2662352919578552, Test Accuracy: 0.8888750076293945

Classification Report Adam_Relu:

	precision	recall	f1-score	support
0	0.90	0.87	0.89	19641
1	0.88	0.90	0.89	20359
accuracy			0.89	40000
macro avg	0.89	0.89	0.89	40000
weighted avg	0.89	0.89	0.89	40000



Issues with SGD Optimizer at Low Learning Rates
Slow Convergence: A very low learning rate, such as 0.001, combined with the SGD optimizer, can cause training to converge very slowly. This implies that the model may need many more epochs to effectively learn from the training data.

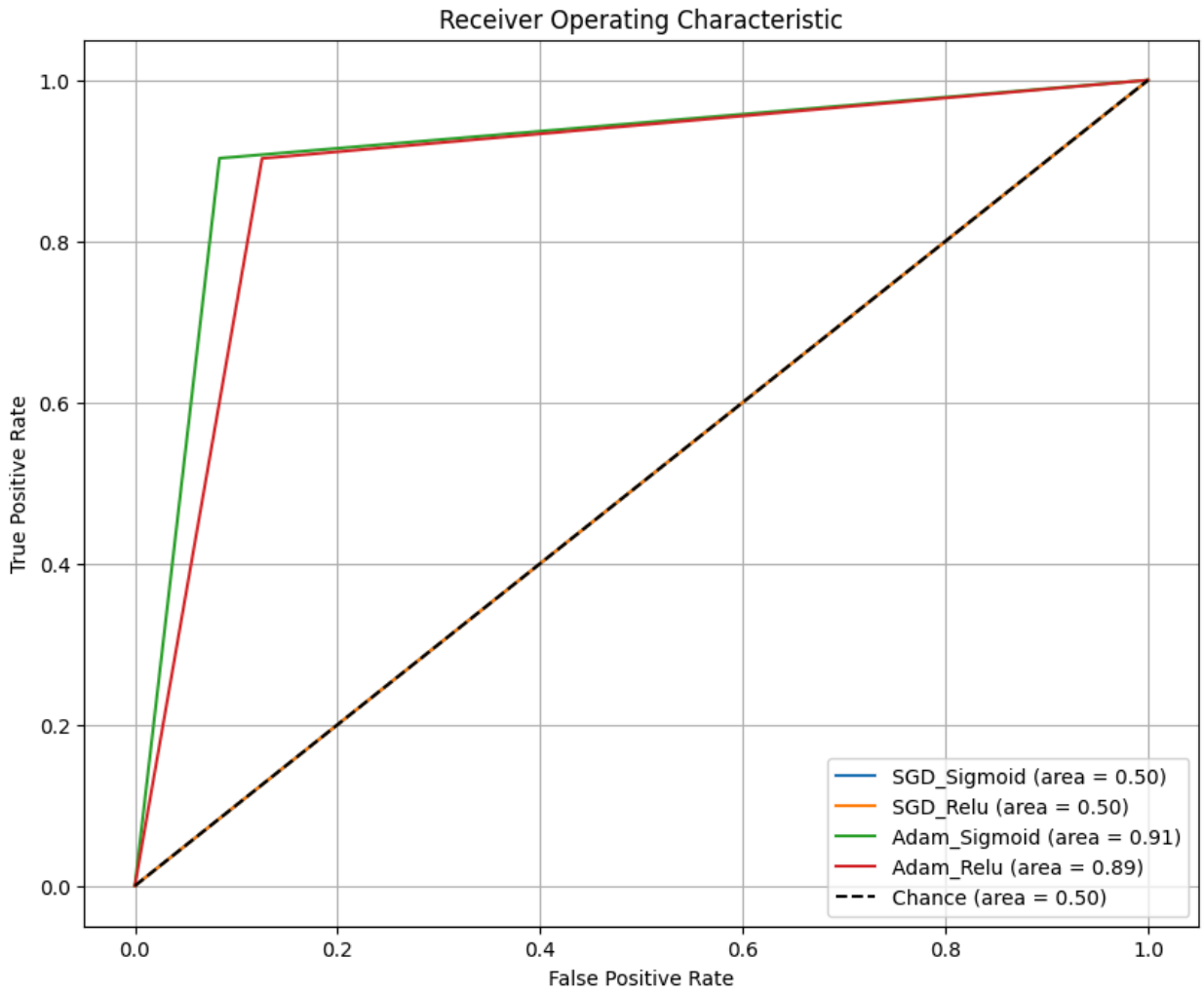
Getting caught in Local Minima or Plateaus: SGD with a low learning rate is more likely to become caught in local minima or plateaus, especially if the loss surface is rugged or has a lot of saddle points.

Strategies to Address These Issues
Increase the Learning Rate: A slightly higher learning rate may allow the model to converge faster and avoid local minima. Experiment with altering the learning rate to observe how it affects performance.

Increase Epochs: With a low learning rate, the model may require more training epochs to successfully learn and converge on a satisfactory answer.

Add Momentum: SGD can greatly benefit from momentum, which helps to accelerate the optimizer in the right direction. This is very useful for dealing with local minima and sluggish convergence rates.

AUC-ROC:



True Positive Rate (Y-axis): It is also known as recall or sensitivity, and it assesses the model's ability to correctly identify actual positives. It is calculated as TP/(TP + FN), where TP is the number of true positives and FN is the number of false negatives.
False Positive Rate (X-axis): Calculates the percentage of true negatives that are wrongly labeled as positives. It is calculated as FP / (FP + TN), where FP is the number of false positives and TN represents the number of true negatives.

Curves on the Graph: Each line represents the ROC curve for a different model configuration or optimizer and activation function combination:

- SGD_Sigmoid: Stochastic Gradient Descent optimizer with a sigmoid activation function.
- SGD_Relu: Stochastic Gradient Descent optimizer with a ReLU activation function.
- Adam_Sigmoid: Adam optimizer with a sigmoid activation function.
- Adam_Relu: Adam optimizer with a ReLU activation function.

Area Under the Curve (AUC): A numerical representation of the model's capacity to discriminate between categories. An AUC of 0.5 (shown by the dashed line labelled "Chance") denotes no discriminative ability, which is similar to random guessing. The AUC for each model is shown in the legend. The Adam optimizer with both activation functions generates high AUC values (0.91 and 0.89), suggesting good classification performance.

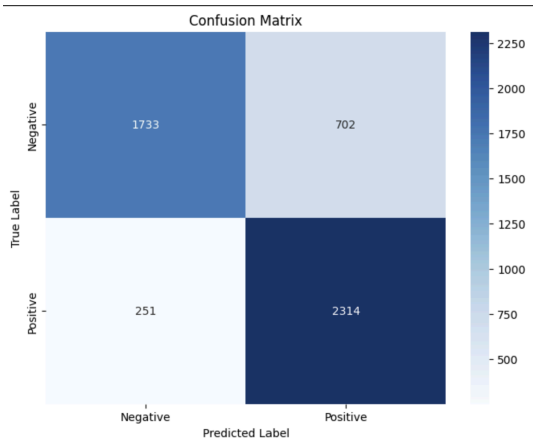
The Diagonal Dashed Line: Represents the "chance" level where a model's predictions are no better than random guessing. Any ROC curve that lies above the diagonal represents a model with better-than-random performance.

Interpretation of the Curves:

- The SGD optimizer with either activation function appears to perform no better than random guessing, with an AUC of 0.50.
- The models using the Adam optimizer, regardless of the activation function, perform significantly better, as their ROC curves are closer to the top-left corner of the graph, indicating a higher TPR and lower FPR.

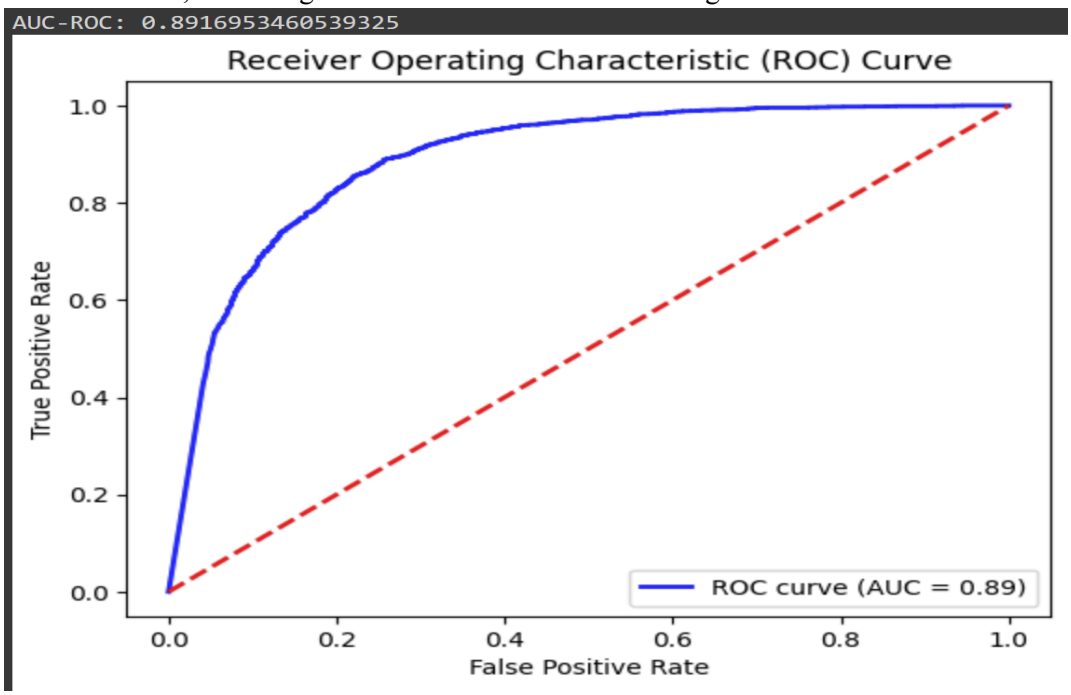
3.5 CNN Model:

	precision	recall	f1-score	support
0	0.84	0.76	0.80	2435
1	0.79	0.86	0.82	2565
accuracy			0.81	5000
macro avg	0.81	0.81	0.81	5000
weighted avg	0.81	0.81	0.81	5000



- **Accuracy:** our model has an overall accuracy of 81%, which means it correctly predicts the class of the samples 81% of the time.
- **Precision and Recall:**
  For class 0: Precision is 0.84, meaning that when the model predicts an instance as class 0, it is correct 84% of the time. Recall is 0.76, indicating that out of all the actual instances of class 0, the model correctly identifies 76% of them.
  For class 1: Precision is 0.79, indicating that when the model predicts an instance as class 1, it is correct 79% of the time. Recall is 0.86, suggesting that out of all the actual instances of class 1, the model correctly identifies 86% of them.
- **F1-score:** The F1-score is the harmonic mean of precision and recall. It gives a balance between precision and recall. For class 0, it's 0.80, and for class 1, it's 0.82.

- **Support:** The number of actual occurrences of each class in the dataset. For class 0, there are 2435 instances, and for class 1, there are 2565 instances.
  Overall, our model seems to perform reasonably well with good precision, recall, and F1-score for both classes, indicating that it's effective at discriminating between the two classes.



This line indicates the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) value obtained from the evaluation. AUC-ROC is a metric used to evaluate the performance of binary classification models. An AUC-ROC value of 0.8917 indicates that the model performs well in distinguishing between positive and negative classes, with higher values closer to 1 indicating better performance. Therefore, an AUC-ROC value of 0.8917 suggests that the model has a good ability to discriminate between positive and negative sentiment reviews.

4.0 Conclusion

In conclusion, our analysis of sentiment analysis models applied to Amazon product reviews reveals the Support Vector Machine (SVM) as the top performer, with an accuracy of 85%. SVM effectively handles high-dimensional spaces and non-linear data, capturing complex relationships within textual data. Its robustness to overfitting, coupled with careful hyperparameter tuning, optimizes its discriminative ability while ensuring minimal model bias. Additionally, SVM's clear decision boundaries offer interpretability, providing insights into sentiment patterns. The Convolutional Neural Network (CNN) and Multi-Layer Perceptron (MLP) also show promising results, with accuracies of 81% and 83%, respectively. However, the Long Short-Term Memory (LSTM) model's performance varies with configurations, with optimizations leading to an accuracy of 91%. Overall, our study highlights the significance of advanced machine learning techniques in e-commerce sentiment analysis for enhancing customer satisfaction, tailoring marketing strategies, and mitigating misleading feedback, paving the way for future research and application in refining the online shopping experience.

5.0 Future work

Looking ahead, there are key areas for future exploration and enhancement in sentiment analysis for e-commerce:

- Fine-tuning Model Architectures: Exploring more advanced neural network architectures and ensemble methods could improve model performance. This includes investigating techniques like transfer learning for better adaptation to e-commerce datasets.
- Domain-Specific Adaptation: Tailoring sentiment analysis models to specific product categories or domains within e-commerce could yield more accurate insights. Capturing domain-specific language nuances could enhance model effectiveness.
- Dynamic Sentiment Analysis: Developing real-time or dynamic sentiment analysis techniques would enable platforms to respond swiftly to changing consumer sentiments, facilitating agile decision-making and strategy adjustments.
- Aspect-Based Sentiment Analysis: Investigating methods for analyzing sentiment towards specific product features or attributes could provide deeper insights for product development and marketing strategies.
- Integration with Recommendation Systems: Combining sentiment analysis with recommendation systems could improve personalized shopping experiences. Utilizing sentiment insights to tailor product recommendations could boost customer satisfaction and engagement.
- Addressing Bias and Fairness: Continuously assessing and mitigating bias in sentiment analysis models is essential for equitable treatment of consumers. Implementing robust bias detection and mitigation strategies is crucial for model fairness.

References:

1. Lee, J., Kim, S. W., & Kim, J. (2019). Sentiment Analysis in E-commerce: A Review. *Journal of Information Systems and E-Business Management,* 17(4), 1065–1083.

2. Wang, Y., Zhang, L., & Liu, X. (2020). Improving Sentiment Analysis Accuracy in *E-commerce Reviews*. **International Journal of Electronic Commerce Studies,* 11(1), 23–38.

3. Ralf C. Staudemeyer, Eric Rothstein Morris Understanding (2019) LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks https://arxiv.org/abs/1909.09586
4. Wang, Y., & Zuo, L. (2017). Deep learning for sentiment analysis of user reviews. Neural Computing and Applications, 28(9), 2621-2635.