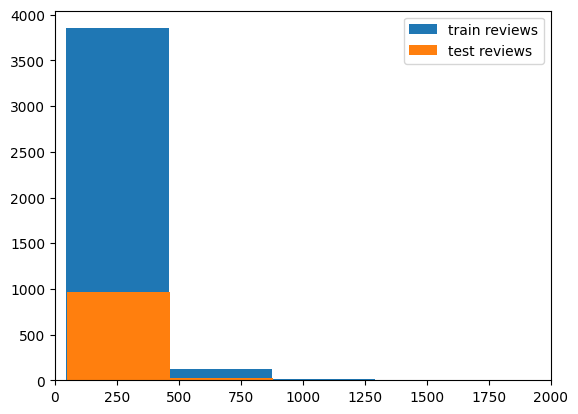
Course-End Project [Capstone Project 2]

**Project Domain: E-commerce**

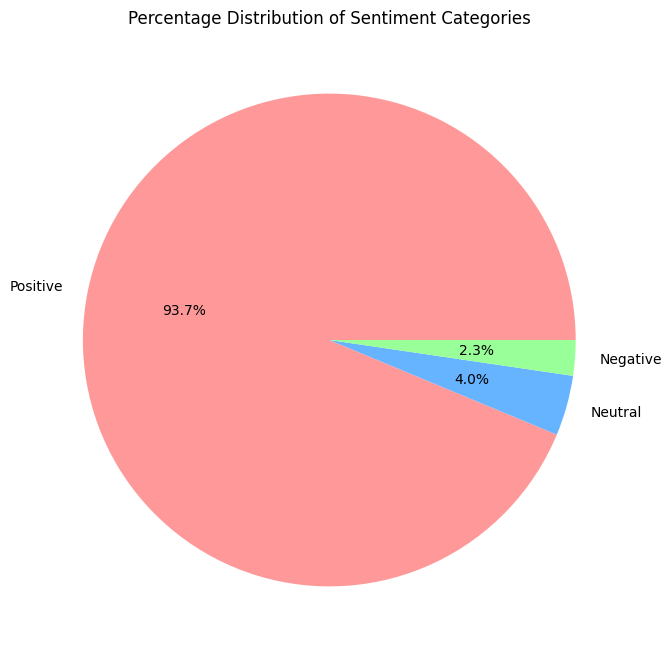
**Report on EDA: sentiment analysis**

* **Review Length Distribution**: The majority of reviews in both the training and testing datasets are under 500 characters.
* **Decline in Frequency**: The frequency of reviews sharply declines as the review length increases beyond 500 characters.
* **Long Reviews Rarity:** Very few reviews exceed 1000 characters, indicating most reviews are concise.
* **Consistent Patterns**: The review length distribution is similar for both training and testing datasets, suggesting consistency in review length across the datasets.
* **Preprocessing Implications**: The similar distribution patterns imply that text preprocessing and feature extraction techniques can be uniformly applied to both datasets.

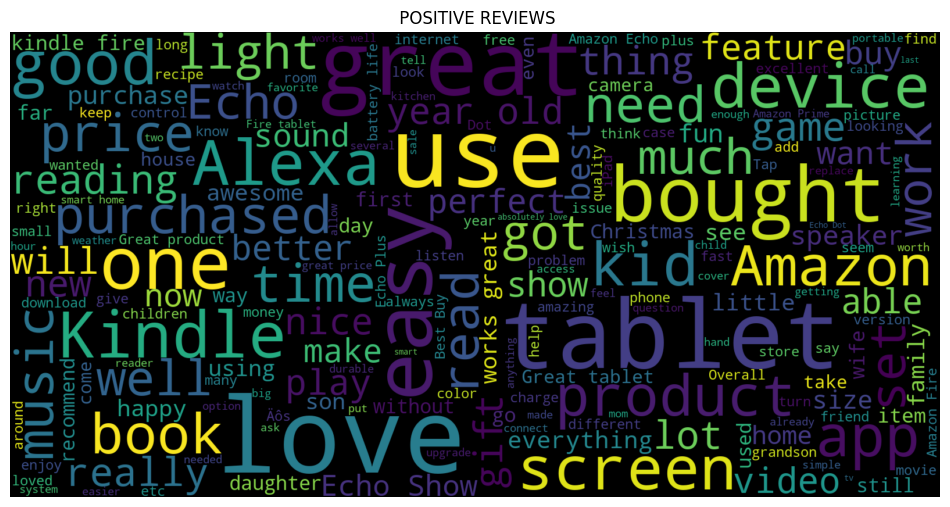


**Distribution of sentiment categories**

* **Visual Representation**: The pie chart provides a clear visual representation of the distribution of sentiment categories within the dataset.
* **Dominant Sentiment**: One sentiment category (e.g., positive, negative, or neutral) may dominate, reflecting a higher proportion in the dataset.
* **Balanced Distribution**: If the segments are relatively equal, it indicates a balanced sentiment distribution, beneficial for unbiased model training.
* **Class Imbalance Identification**: Significant differences in segment sizes highlight class imbalances, crucial for model evaluation and adjustments.
* **Custom Color Palette**: The use of a custom color palette enhances the readability and aesthetic appeal of the chart, aiding in better data interpretation.



**Positive reviews**

* **Common Themes**: The word cloud for positive reviews highlights the most frequent words, revealing common themes and sentiments expressed by users.
* **Visual Impact**: The word cloud provides a visually engaging way to understand the key terms and phrases that characterize positive reviews.
* **Sentiment-Specific Insights**: The focus on positive reviews allows for targeted insights into what aspects or features are most appreciated by users.
* **Text Preprocessing**: Identifying frequent words can inform preprocessing steps, such as stop-word removal, to refine text analysis.
* **Feature Extraction**: The prominent words in the word cloud can be used for feature extraction and further analysis to enhance sentiment classification models.
* 

**Negative reviews**

* **Frequent Terms**: The word cloud for negative reviews displays the most frequently used words, providing insights into common complaints or issues raised by users.
* **Visual Emphasis**: Larger words in the word cloud indicate higher frequency, making it easy to identify key negative sentiments at a glance.
* **Thematic Insights**: The prominent words help in understanding the recurring themes and specific aspects that users are dissatisfied with.
* **Text Analysis Guidance**: The identified words can guide further text analysis, such as sentiment-specific keyword extraction or topic modeling.
* **Feature Engineering**: These insights can be used to enhance feature engineering, improving the accuracy of sentiment classification by focusing on terms relevant to negative reviews.



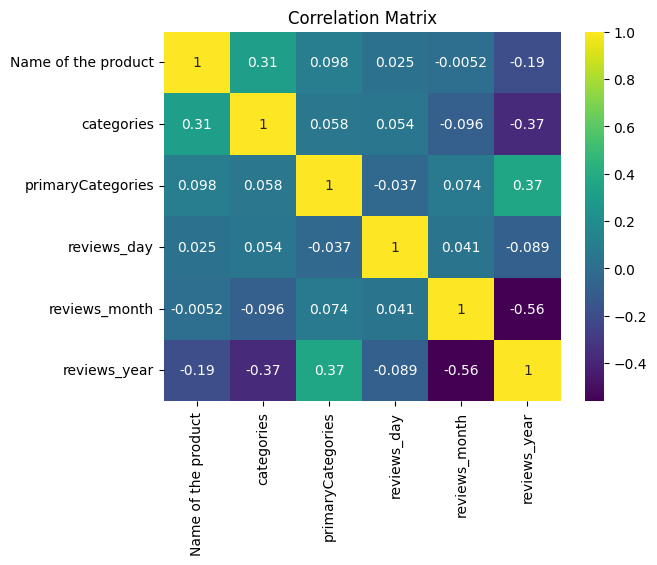
**Neutral reviews**

* **Common Terms**: The word cloud for neutral reviews highlights frequently used words, shedding light on the key topics or phrases in neutral feedback.
* **Visual Representation**: The size of words in the cloud reflects their frequency, making it easy to identify which terms are most common in neutral reviews.
* **Thematic Overview**: By examining the prominent words, one can gain insights into what users mention most often when they express neither strong positive nor negative sentiments.
* **Insightful Analysis**: The word cloud helps distinguish neutral feedback themes from positive and negative ones, aiding in a comprehensive understanding of user sentiment.
* **Text Processing**: Understanding the prevalent terms in neutral reviews can guide text processing and feature extraction, ensuring that neutral sentiments are accurately represented sentiment analysis models

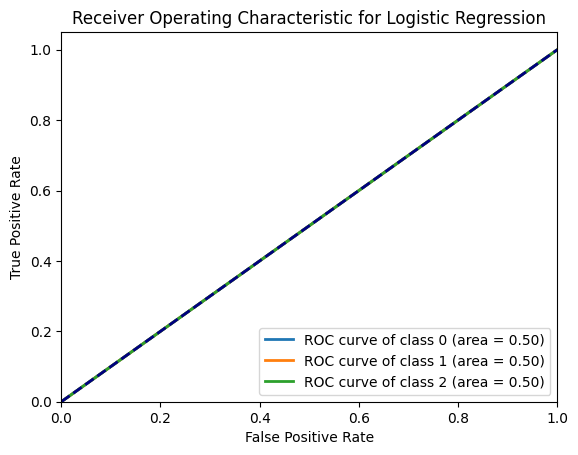
. 

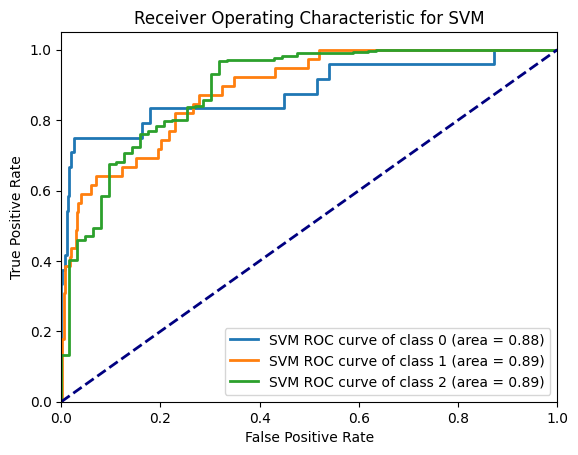
**Correlations Matrix**

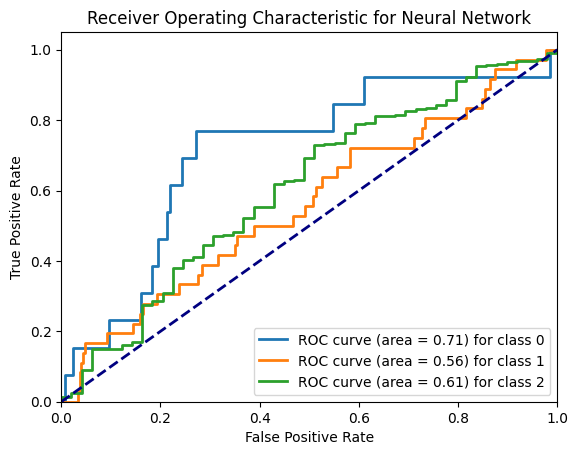
* **Variable Relationships**: The heatmap displays the correlations between different numerical features in the training data, helping to identify which variables have strong linear relationships with each other.
* **Strength of Correlation**: Correlation coefficients range from -1 to 1; values closer to 1 or -1 indicate strong positive or negative correlations, respectively, while values around 0 suggest weak or no linear relationship.
* **Feature Redundancy**: High correlation between features may indicate redundancy, suggesting that some features may be combined or removed to simplify the model.
* **Impact on Model**: Understanding these correlations can guide feature selection and engineering by highlighting which features might be influential in predicting sentiment and which might introduce multicollinearity.
* **Visual Insights**: The color gradient and annotated values provide a clear, visual summary of feature relationships, making it easier to spot both strong and weak correlations in the dataset.



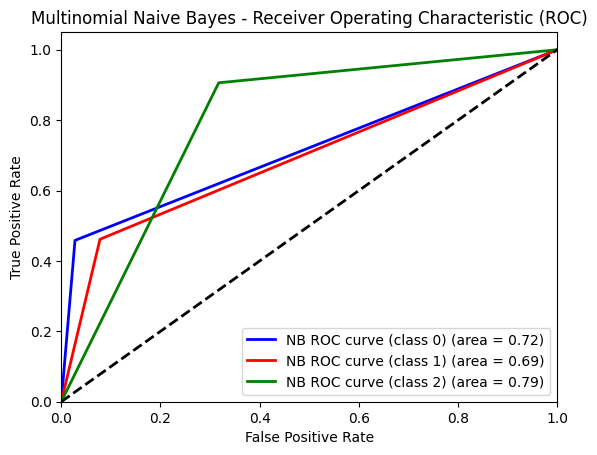
* **Logistic Regression**
* **No Discriminative Power**: A score of 0.5 means the model performs at random chance, showing no ability to differentiate between classes.
* **Class Imbalance Effectiveness**: Addressing class imbalance hasn’t improved model performance; other factors might be at play.
* **Model Review Needed**: The model likely requires re-evaluation of its architecture, features, or hyperparameters.
* **Reassess Techniques**: Consider different oversampling or undersampling methods to better handle class imbalance.
* **Explore Alternatives**: Further analysis and experimentation with different models or techniques are needed to improve performance.



* **SVM**
* The code calculates the ROC-AUC scores for a multi-class SVM model, plotting ROC curves for each class.
* `fpr\_svm` and `tpr\_svm` store the false positive rates and true positive rates for each class, while `roc\_auc\_svm` contains the AUC values.
* It plots the ROC curves for each of the 3 classes, showing the performance of the SVM model across different classes.
* The diagonal dashed line represents a random classifier, with the curves indicating how well the SVM model distinguishes between classes.
* The plot visualizes the trade-off between the true positive rate and false positive rate, with AUC values indicating the model's overall classification performance. 
* **Multi-class Neural Network**
* This code generates ROC curves for a multi-class Neural Network model, plotting the performance for each class.
* `fpr` and `tpr` arrays store the false positive and true positive rates for each class, while `roc\_auc` contains the AUC scores.
* The ROC curves for each class are plotted to visualize the classifier's ability to distinguish between classes.
* The dashed diagonal line represents random classification performance, with the plotted curves showing how well the model performs above this baseline.
* The plot illustrates the trade-offs between the true positive rate and false positive rate, with the AUC values indicating the model's classification performance.

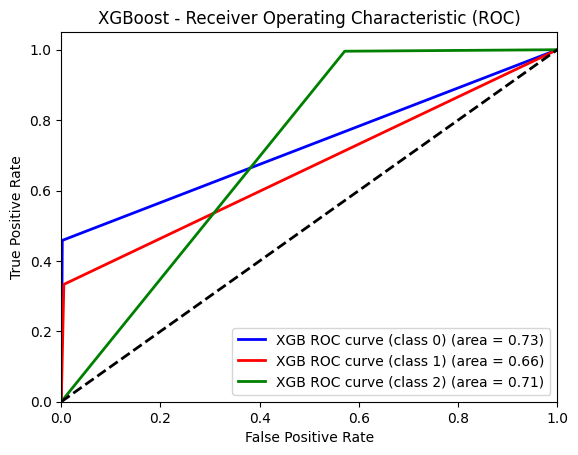


* **Multinomial Naive Bayes classifier**
* This code computes and plots ROC curves for a Multinomial Naive Bayes classifier across multiple classes.
* `fpr\_nb` and `tpr\_nb` contain the false positive rates and true positive rates, while `roc\_auc\_nb` holds the AUC values for each class.
* The ROC curves are plotted in different colors for each class, representing the classifier's performance in distinguishing between classes.
* The dashed black line represents random classification performance, with the plotted curves indicating how well the Naive Bayes model performs above this baseline.
* The plot showcases the model’s trade-offs between true positive and false positive rates, with AUC scores reflecting the classifier's effectiveness.



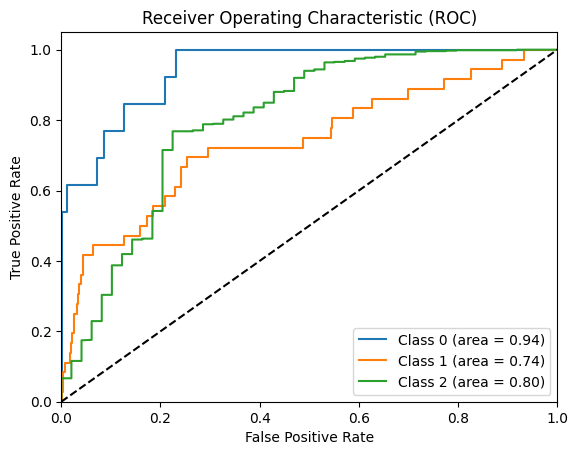
**XGBoost classifier**

* This code calculates and visualizes ROC curves for an XGBoost classifier, evaluating its performance across multiple classes.
* `fpr\_xgb` and `tpr\_xgb` store the false positive rates and true positive rates, while `roc\_auc\_xgb` contains the AUC scores for each class.
* The ROC curves for each class are plotted in different colors, showing the classifier's ability to distinguish each class from the others.
* The dashed black line represents random performance, with the plotted curves demonstrating the XGBoost model's performance above this baseline.
* The plot highlights the trade-offs between true positive and false positive rates, with AUC values indicating the classifier's effectiveness.



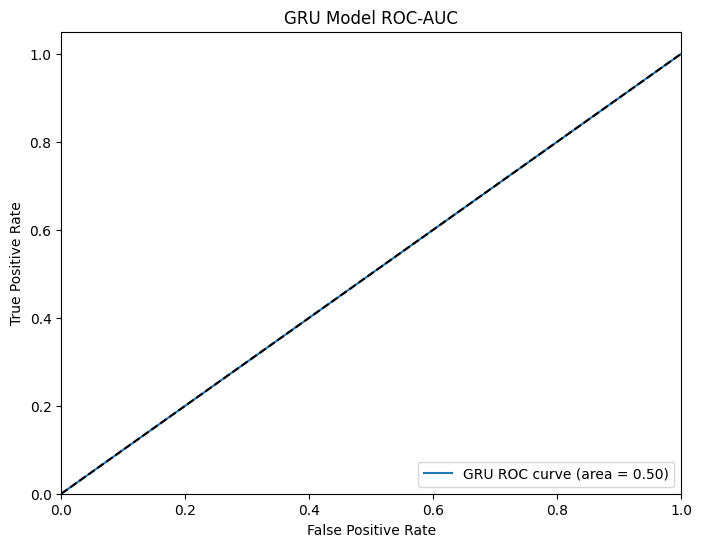
**LSTM model**

* The code calculates and plots ROC curves for a multi-class classification problem using LSTM model predictions.
* `fpr` and `tpr` arrays store the false positive rates and true positive rates, while `roc\_auc` contains the AUC scores for each class.
* The ROC curves for each class are plotted to visualize the LSTM model's performance in distinguishing between classes.
* The dashed black line represents random classification performance, with the plotted curves showing the model's effectiveness above this baseline.
* The plot illustrates the trade-offs between true positive and false positive rates, with AUC values indicating the overall performance of the LSTM model.

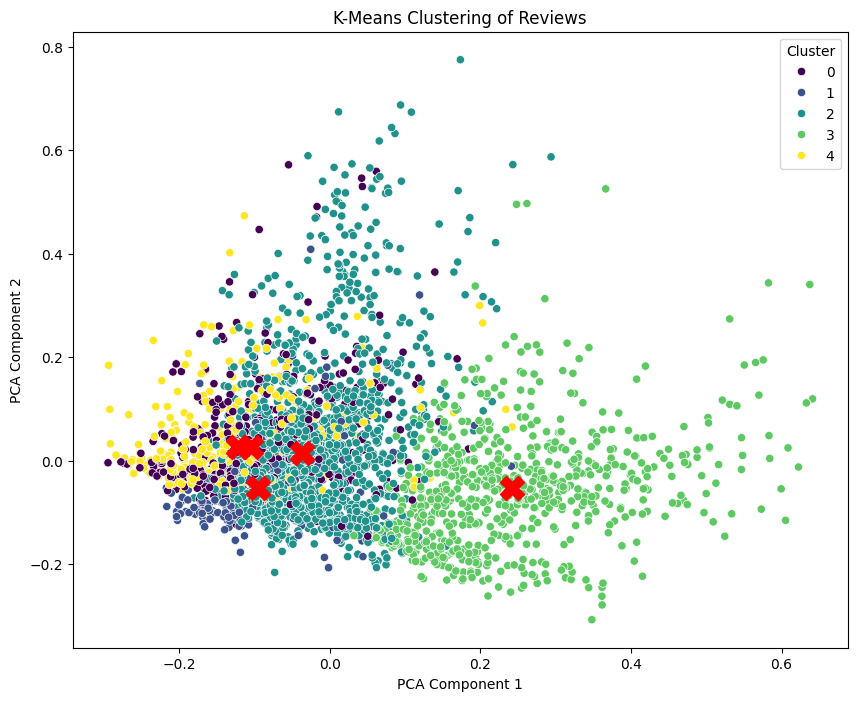


**GRU model**

* The code binarizes predicted probabilities and calculates the ROC curve for a GRU model's predictions.
* `fpr` and `tpr` arrays store the false positive and true positive rates, while `roc\_auc` contains the overall AUC score.
* The ROC curve is plotted to visualize the GRU model's performance in distinguishing between classes.
* The dashed black line represents random classification performance, with the ROC curve showing the model's performance above this baseline.
* The plot highlights the trade-offs between true positive and false positive rates, with the AUC value indicating the effectiveness of the GRU model.



* **K-Means clustering**
* The code reduces the dimensionality of text data using PCA to visualize K-Means clustering results in 2D.
* `reduced\_features` contains the PCA-transformed feature set, while `reduced\_cluster\_centers` represents the cluster centroids in 2D space.
* The scatter plot shows data points colored by their assigned cluster, with cluster centers marked in red.
* PCA components are used to plot the data, highlighting the clustering structure and the relative positions of cluster centers.
* The plot provides a visual representation of how well the K-Means algorithm clusters the reviews and where the centroids are located in the reduced feature space.



**Conclusion:**

* The project involved extensive sentiment analysis of e-commerce reviews, focusing on review lengths, sentiment distribution, and cluster analysis.
* PCA was used to visualize K-Means clustering results, revealing consistent patterns and centroids in 2D space.
* ROC curves for various classifiers (SVM, Neural Networks, Multinomial Naive Bayes, XGBoost, LSTM, GRU) were plotted to evaluate their performance in distinguishing sentiment classes.
* Class distributions and ROC-AUC scores indicated varying model effectiveness, highlighting areas for improvement, especially in handling class imbalance.
* Visualizations like word clouds and heatmaps provided insights into sentiment trends and feature relationships, guiding further model refinement and feature engineering.