G4 Report: Project NLP | Business Case: Automated Customer Reviews

Team Members: Jaime Velez, Latif Calderon, Juan Andujar & Larry W. Davila

Executive Summary

This business case outlines the development of an NLP model to automate the processing of customer feedback for a retail company. The goal is to classify customer reviews into positive, negative, or neutral categories to help the company improve its products and services. Additionally, the project leverages GenerativeAI to summarize reviews based on review scores (0-5) and product categories, and creates a dynamic visualization dashboard using Plotly.

Problem Statement

The company receives thousands of text reviews every month, making it challenging to manually categorize, analyze, and visualize them. An automated system can save time, reduce costs, and provide real-time insights into customer sentiment.

Project Goals

1. **Classify Customer Reviews**: Classify customer reviews (textual content) into positive, neutral, or negative.
2. **Summarize Reviews**: Summarize reviews for each product category broken down by star rating.
3. **Handle Multiple Product Categories**: Manage a feasible number of product categories, e.g., top 10 or top 50.
4. **Create a Dynamic Dashboard**: Develop an interactive visualization dashboard to present insights.

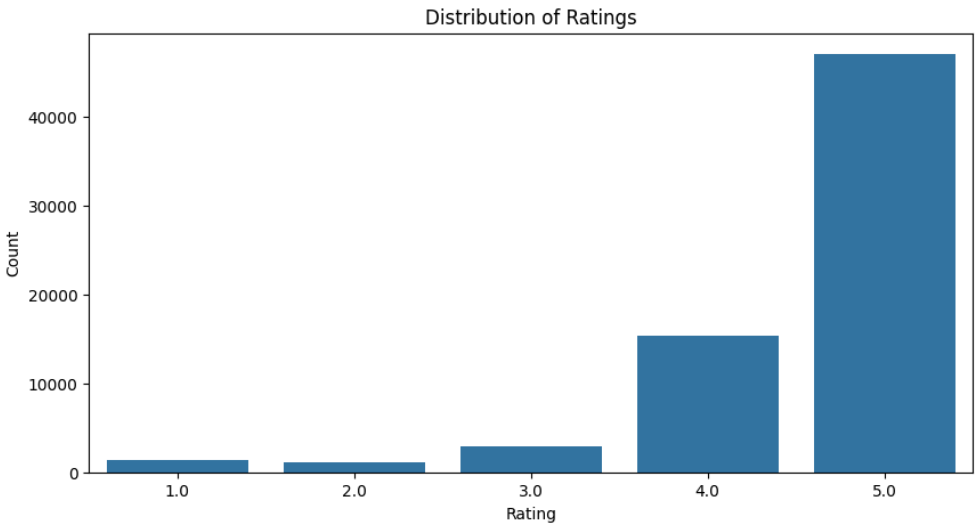
Data Collection

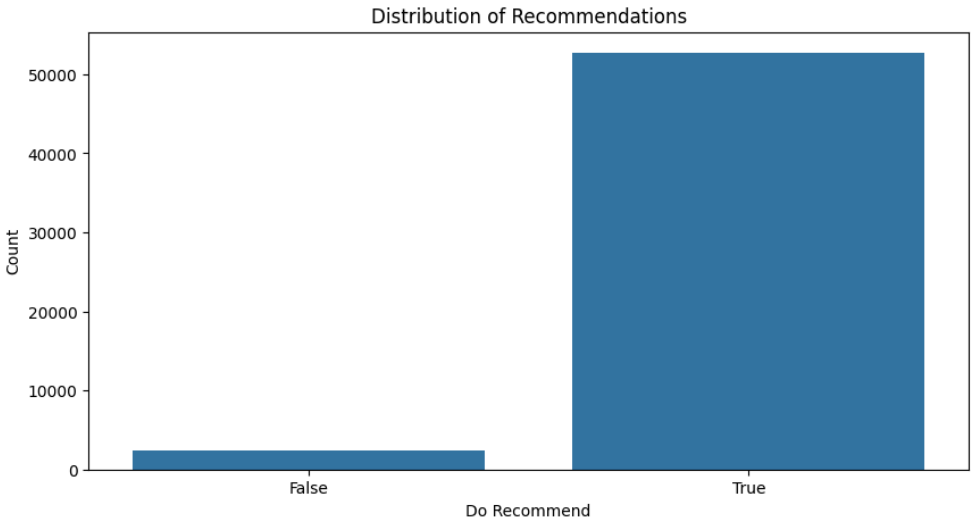
- Utilized publicly available datasets of Amazon customer reviews.
- Ensured computing resources could handle the dataset size and machine learning processes.
- Combined all 3 datasets available to get as much data as possible to train the model

1. Data Preprocessing

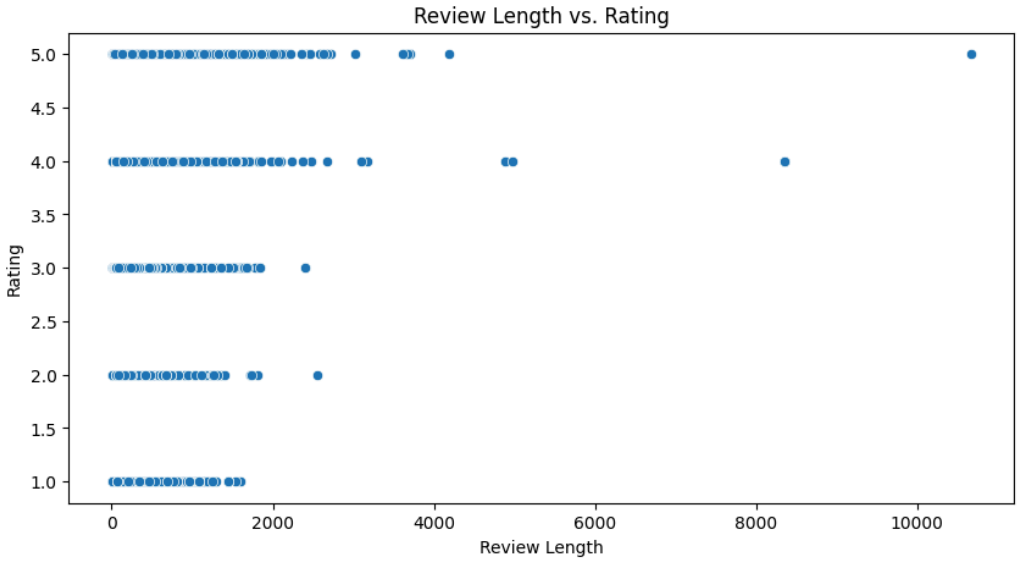
1.1. Data Analysis

- **Relevant Columns:** Explored the data to identify useful columns for the model: doRecommend, numHelpful, rating, text, title.
- **Data Balance:** Plotted the distribution of ratings and doRecommend to assess data balance. Conclusion: The data is very unbalanced, necessitating resampling.



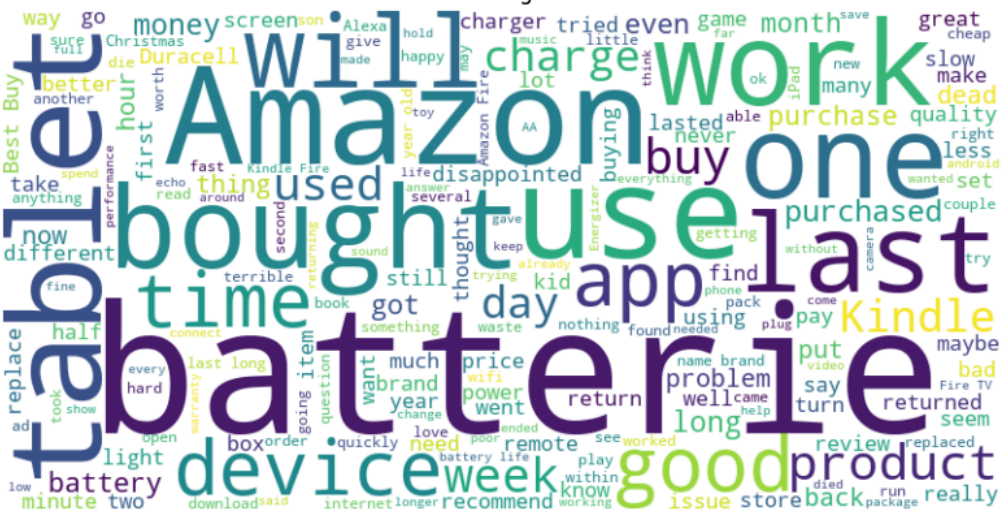


- **Length and Rating Correlation**: Explored the correlation between review length and rating, finding no significant correlation.



- Correlation between Review Length and Rating: -0.11
- Correlation between Review Length and Do Recommend: -0.08
- **Word Frequency:** Created word clouds for each sentiment to visually analyze the most used words. Added common words lacking sentimental context (tablet, amazon, kindle, bought, one, use) to the stopword list.

Word Cloud for Negative Reviews



Word Cloud for Neutral Reviews

Phase #1: Traditional NLP & ML Approaches

Objective: The objective of this project is to implement a sentiment analysis pipeline on customer reviews using a combination of traditional Natural Language Processing (NLP) techniques and machine learning classifiers. The goal is to classify sentiments into three categories: positive, neutral, and negative, and evaluate the performance of various classifiers.

Original Action Plan

a. **Dataset Preparation:** Clean and preprocess the provided dataset, focusing on critical features.
b. **Sentiment Mapping:** Map ratings to sentiment labels.
c. **Feature Engineering:** Balance classes and tokenize/vectorize text data.
d. **Model Training:** Train machine learning models (Naive Bayes, Logistic Regression, SVM, Random Forest) and evaluate performance.
e. **Evaluation:** Use metrics like accuracy, precision, recall, F1-score, and confusion matrix.
f. **Comparison:** Analyze results across different models and approaches.

Data Cleaning

- **Label Encoding Ratings into Sentiments**: Encoded 1 & 2-star ratings as negative (0), 3-star as neutral (1), and 4-5 stars as positive (2).
- **Standard Data and Text Cleaning**: Removed special characters, punctuation, unnecessary whitespace, and rows with null values in ***reviews.text*** or ***reviews.rating***. Converted text to lowercase and removed stopwords.
- **Tokenization and Lemmatization**: Tokenized text data and applied lemmatization.
- **Vectorization:** Used TF-IDF Vectorizer to convert text data into numerical vectors, creating a document-term matrix.

1.2. Resampling for Classes Balancing

- Applied SMOTE and dataset reduction techniques to ensure balanced representation of sentiments.

```
Original training set class distribution:
sentiment
2    49941
1     2361
0     1988
Name: count, dtype: int64

Resampled training set class distribution:
sentiment
2    49941
1    49941
0    49941
Name: count, dtype: int64
```

2. Model Building

2.1. Model Selection

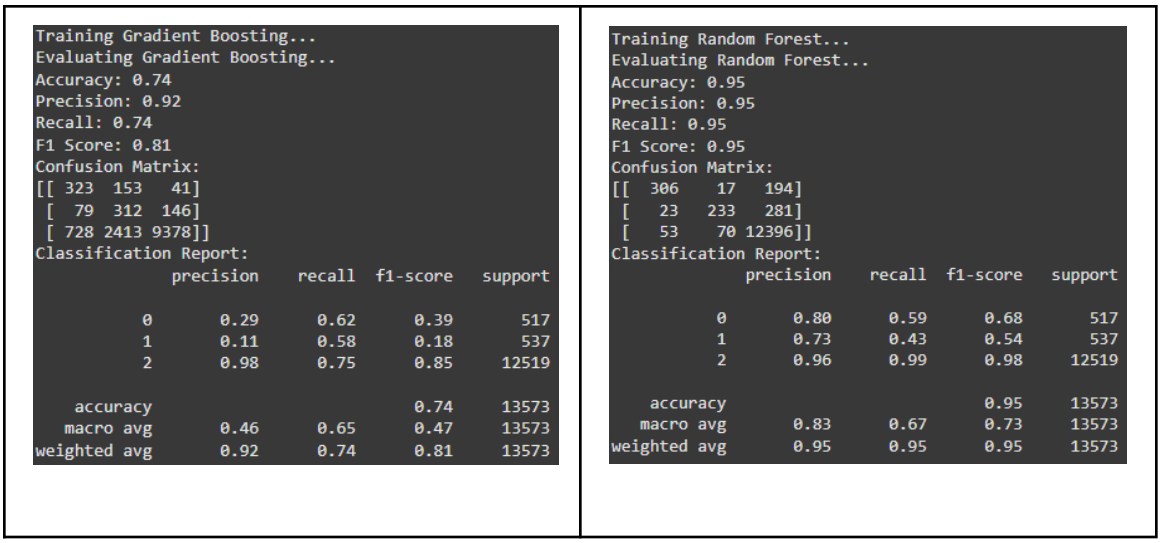
- Explored various algorithms: Naive Bayes, Logistic Regression, Support Vector Machines, XGBoost, and Random Forest. Selected the best based on performance metrics.

```
Training Naive Bayes...
Test Accuracy (Naive Bayes): 0.7238

Classification Report (Naive Bayes - Test Set):
              precision    recall  f1-score   support

    negative       0.70      0.82      0.75       390
     neutral       0.58      0.20      0.30       290
    positive       0.76      0.92      0.83       587

    accuracy                           0.72      1267
   macro avg       0.68      0.65      0.63      1267
weighted avg       0.70      0.72      0.69      1267


Confusion Matrix (Naive Bayes - Test Set):
[[539  18  30]
 [123  58 109]
 [ 46  24 320]]
```

```
Training Logistic Regression...
Evaluating Logistic Regression...
Accuracy: 0.84
Precision: 0.93
Recall: 0.84
F1 Score: 0.88
Confusion Matrix:
[[  370    95    52]
 [   76   329   132]
 [  455  1330 10734]]
Classification Report:
              precision    recall  f1-score   support

           0       0.41      0.72      0.52       517
           1       0.19      0.61      0.29       537
           2       0.98      0.86      0.92     12519

    accuracy                           0.84     13573
   macro avg       0.53      0.73      0.58     13573
weighted avg       0.93      0.84      0.88     13573
```

```
Training SVM...
Test Accuracy (SVM): 0.7395

Classification Report (SVM - Test Set):
              precision    recall  f1-score   support

    negative       0.74      0.76      0.75       390
     neutral       0.54      0.40      0.46       290
    positive       0.80      0.89      0.84       587

    accuracy                           0.74      1267
   macro avg       0.70      0.69      0.69      1267
weighted avg       0.72      0.74      0.73      1267


Confusion Matrix (SVM - Test Set):
[[523  44  20]
 [ 90 116  84]
 [ 38  54 298]]
```

```
Training XGBoost...
Evaluating XGBoost...
Accuracy: 0.88
Precision: 0.93
Recall: 0.88
F1 Score: 0.90
Confusion Matrix:
[[  348    86    83]
 [   59   299   179]
 [  320   958 11241]]
Classification Report:
              precision    recall  f1-score   support

           0       0.48      0.67      0.56       517
           1       0.22      0.56      0.32       537
           2       0.98      0.90      0.94     12519

    accuracy                           0.88     13573
   macro avg       0.56      0.71      0.60     13573
weighted avg       0.93      0.88      0.90     13573
```



```
Training Gradient Boosting...
Evaluating Gradient Boosting...
Accuracy: 0.74
Precision: 0.92
Recall: 0.74
F1 Score: 0.81
Confusion Matrix:
[[ 323  153   41]
 [  79  312  146]
 [ 728 2413 9378]]
Classification Report:
              precision    recall  f1-score   support

           0       0.29      0.62      0.39       517
           1       0.11      0.58      0.18       537
           2       0.98      0.75      0.85     12519

    accuracy                           0.74     13573
   macro avg       0.46      0.65      0.47     13573
weighted avg       0.92      0.74      0.81     13573
```

```
Training Random Forest...
Evaluating Random Forest...
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
F1 Score: 0.95
Confusion Matrix:
[[  306    17   194]
 [   23   233   281]
 [   53    70 12396]]
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.59      0.68       517
           1       0.73      0.43      0.54       537
           2       0.96      0.99      0.98     12519

    accuracy                           0.95     13573
   macro avg       0.83      0.67      0.73     13573
weighted avg       0.95      0.95      0.95     13573
```

Model	Accuracy (%)	Precision (%)			Recall (%)			F1-Score (%)		
		1	2	3	1	2	3	1	2	3
Naive Bayes	72	70	58	76	**82**	20	92	75	30	83
Logistic Regression	84	41	19	98	72	**61**	86	52	29	92
SVM	74	74	54	80	76	40	89	75	46	84
XGBoost	88	48	22	**98**	67	56	90	56	32	94
Gradient Boosting	74	29	11	98	62	58	75	39	18	85
Random Forest	**95**	**80**	**73**	96	59	43	**99**	**68**	**54**	**98**

2.2 **Model Training and fine tuning**

- Selected Random Forest Classifier based on accuracy, precision, recall, and F1-score.
- Employed class weight adjustment and hyperparameter tuning with Grid Search to improve results.
- Class Weight Adjusted Random Forest Classifier

```
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
F1 Score: 0.95
Confusion Matrix:
[[  315    12   190]
 [   22   235   280]
 [   48    68 12403]]
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.61      0.70       517
           1       0.75      0.44      0.55       537
           2       0.96      0.99      0.98     12519

    accuracy                           0.95     13573
   macro avg       0.84      0.68      0.74     13573
weighted avg       0.95      0.95      0.95     13573
```

- Hyperparameter Tuning with Grid Search

```
Fitting 5 folds for each of 216 candidates, totalling 1080 fits
Best Parameters: {'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300}
Best F1 Score: 0.9931117563745216
```

```
Accuracy: 0.96
Precision: 0.95
Recall: 0.96
F1 Score: 0.95
Confusion Matrix:
[[  295     9   213]
 [   16   224   297]
 [   28    37 12454]]
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.57      0.69       517
           1       0.83      0.42      0.56       537
           2       0.96      0.99      0.98     12519

    accuracy                           0.96     13573
   macro avg       0.89      0.66      0.74     13573
weighted avg       0.95      0.96      0.95     13573
```

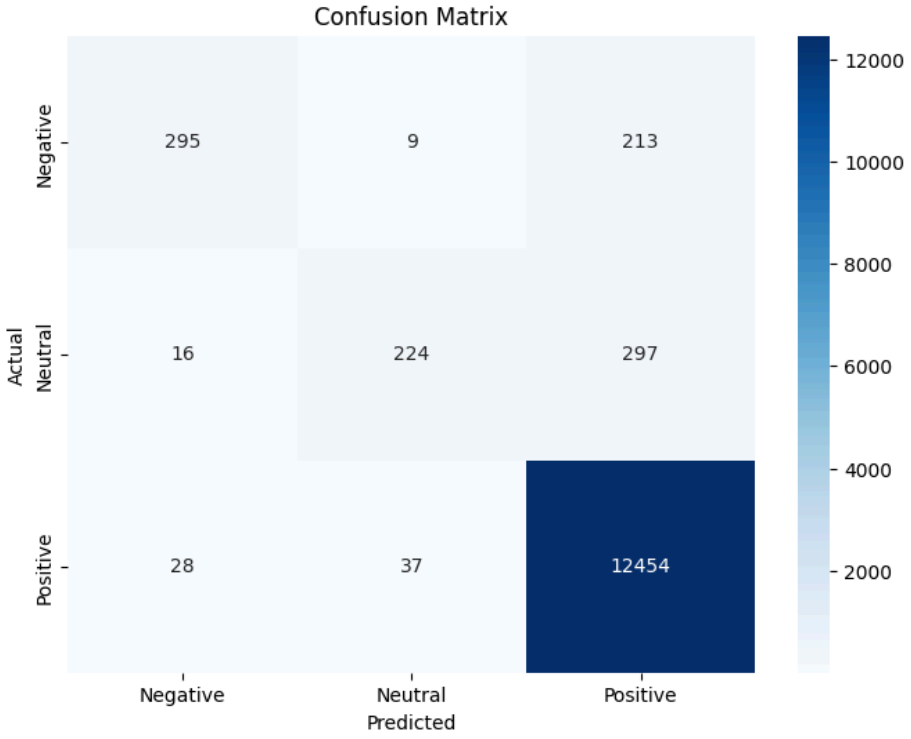
3. Model Evaluation

3.1. Evaluation Metrics

- Evaluated the model's performance on a separate test dataset using various metrics:
    - **Accuracy**: **0.96** (Percentage of correctly classified instances.)

- **Precision**: **0.95** (Proportion of true positive predictions among all positive predictions.)
- **Recall**: **0.96** (Proportion of true positive predictions among all actual positive instances.)
- **F1-score**: **0.95** (Harmonic mean of precision and recall.)

- Calculated a confusion matrix to analyze the model's performance across different classes.



Phase #2: Sequence-to-Sequence Modeling with LSTM

- **Model Architecture**: Built a Bidirectional LSTM model with four layers: embedding layer, two hidden layers, and an activation layer.

```
# Build the Bidirectional LSTM model
model = Sequential()
model.add(Embedding(input_dim=5000, output_dim=128, input_length=max_sequence_length))
model.add(Bidirectional(LSTM(128, return_sequences=True)))
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(64)))
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))

# Compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X_train_padded, y_train, epochs=10, batch_size=64, validation_data=(X_test_padded, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(X_test_padded, y_test)
print(f'Test Accuracy: {accuracy:.2f}')
```

- **Preprocessing**: Applied the same preprocessing steps as for traditional models.
- **Evaluation**: Achieved good results but lagged behind the Random Forest Classifier.

	precision	recall	f1-score	support
0	0.80	0.60	0.68	517
1	0.52	0.41	0.46	537
2	0.97	0.98	0.98	12519
accuracy			0.95	13573
macro avg	0.76	0.66	0.71	13573
weighted avg	0.94	0.95	0.94	13573

Phase #3: Transformer Approach (HuggingFace API)

1. **Data Preprocessing**

1.1. Data Cleaning and Tokenization

- Cleaned and tokenized the customer review data to remove special characters, punctuation, and unnecessary whitespace.
- Used HuggingFace Transformers tokenizer (distilbert-base-uncased, roberta-base).

1.2. Data Encoding and Padding

- Encoded the tokenized input sequences into numerical IDs using the tokenizer's vocabulary.
- Padded input sequences to a maximum length for uniform input size.

1.3. Dataset Reduction

- We implemented a function called reduce_data to effectively reduce the dataset size while maintaining diversity.
- This function ensures that the resulting dataset retains a balanced distribution across different categories, such as star ratings.
- We reduced the dataset to 2,500 rows per sentiment. (7,500 in total from + 60,000)

2. **Model Building**

2.1. Model Selection

- Explored transformer-based models: BERT, RoBERTa, DistilBERT. Selected RoBERTa and DistilBERT for their respective strengths.
- **RoBERTa:** Chosen for robustness, extensive pre-training, and state-of-the-art performance.
- **DistilBERT:** Chosen for efficiency, lightweight nature, and high performance...

2.2. Model Fine-Tuning

- Fine-tuned the selected pre-trained model on the customer review dataset using transfer learning.

- Configured the fine-tuning process by specifying the following parameters: batch size 32, learning rate 2×10−5, and number of training epochs of 10.

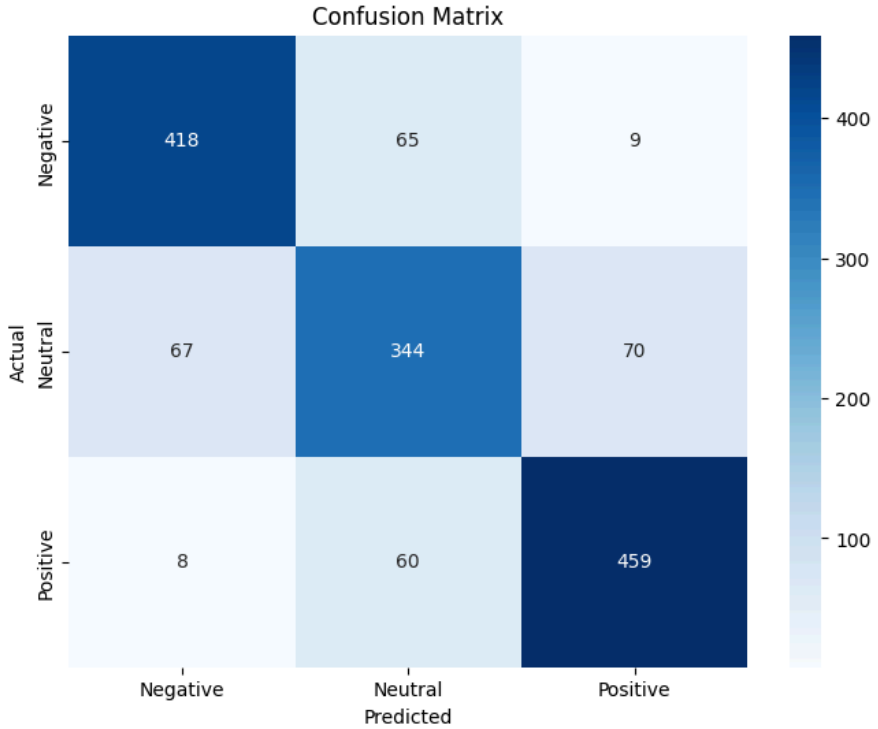
3. **Model Evaluation**

3.1. Evaluation Metrics for RoBERTa

- Accuracy, Precision, Recall, F1-score calculated for each class.

4. Results

- Accuracy: 81%
- Precision, Recall, F1-score:
    - Negative: Precision=0.85, Recall=0.85, F1-score=0.85
    - Neutral: Precision=0.73, Recall=0.72, F1-score=0.72
    - Positive: Precision=0.85, Recall=0.87, F1-score=0.81



5. Conclusion

- **Performance**: The Random Forest classifier achieved higher accuracy, precision, recall, and F1 score compared to RoBERTa. This can be attributed to the use of the entire dataset and effective resampling.

- **Resource Efficiency**: RoBERTa demonstrated the ability to perform reasonably well with a reduced dataset size (7500 rows) but required significant computational resources (preferably GPUs) for fine-tuning. Random Forests, while achieving higher performance, required substantial CPU and memory resources for handling the full dataset with resampling and grid search.
- **Use Case Considerations**: Depending on the available computational resources and the specific requirements of the task, either approach can be preferred:
    - **Random Forest**: Suitable for scenarios where high computational power and memory are available, and achieving the highest possible performance is critical.
    - **RoBERTa**: Suitable for scenarios where GPU resources are available, and a balanced performance with a reduced dataset is sufficient.

Phase #4: Reviews Summarization using Dynamic Visualization Dashboard

6. Category Rating Summary and Interactive Dashboard

6.1. Category Rating Summary Using T5

- To better understand customer feedback across various product categories, we implemented a Text-to-Text Transfer Transformer (T5) model to generate summaries of reviews based on their ratings. This approach ensures that the summarized reviews provide a clear and concise understanding of customer sentiment for each rating within a product category.

6.2. Implementation Steps:

6.2.1. Data Preparation:

- The review data was cleaned by removing special characters, punctuation, and unnecessary whitespace.
- Tokenized the text data to break it into individual words or tokens.
- Converted text to lowercase to ensure consistency.

6.3. Summarization Process:

- We used the T5 model to generate summaries of reviews for each product category and star rating.
- The summarization function included the rating before the review text to provide context.

6.4. Grouping and Summarizing:

- Grouped the reviews by product category and star rating.
- Applied the summarization function to each group, ensuring that the rating was included before the review text.

6.5. Results:

- The generated summaries provide an insightful overview of customer feedback for each product category and star rating. This approach highlights key sentiments expressed by customers, helping the company improve its products and services based on specific ratings.

6.6. Interactive Dashboard Using Plotly

- We developed an interactive Plotly dashboard to visualize the summarized reviews for each product category and star rating. The dashboard allows users to select a category from a dropdown menu and view the corresponding review summaries, including ratings.

6.7. Dashboard Features:

- **Dropdown Menu:**
    - The dashboard includes a dropdown menu with all the product categories, allowing users to select a specific category to view its review summaries.
- **Dynamic Summaries:**
    - Once a category is selected, the dashboard dynamically updates to display the summarized reviews, including ratings, for that category.
- **Interactive Interface:**
    - The dashboard provides an interactive and user-friendly interface for exploring customer feedback across different product categories and ratings.

Review Summaries Dashboard

Electronics

- Rating: 1.0, Summary: amazon's kindle line is mainly v v whats the difference and around amp which is watts w why the mystery guys also this deal is for us v market only i travel. i like that the previous simple fire charger was universal ie vv i like that the first kindle charger was also universal ie vv.
- Rating: 2.0, Summary: hama binders look as good as newoverally this binder has a lot of storage at a good price. the netting trim gives a nice appearance compared to plain look of most budget cd binders. each group of pages shares the same anchor point in the spine.
- Rating: 3.0, Summary: case logic els chromebooksurface sleeve fits the macair perfect the amazon sleeve is simply neoprene which snags and has so much friction it doesnt easily slide inout of my backpack the amazon quality is decent but looks like a beer can cozy koozie compared to the case logic the case logic is not just neoprene it is highly protective light low density foam encased in canvas its more durable and slide in and out of
- Rating: 4.0, Summary: amazon basics logo embossed on black zippers but there is no obvious branding on the front. main compartment is generous in size open and closed with two zippers. back area is generously padded on both sides and is meant to house the laptop.
- Rating: 5.0, Summary: really cool device instantly noticed the difference in quality when i switched from the regular fire stick to the fire tv with k love it works great one in each of the main rooms a lazy mans drean when it is combined with alexa if you get the harmony hub you can really impress with home automation.