Project NLP | Business Case: Automated Customer Reviews

Group: G4
Latif Calderón
Jaime Vélez
Larry W. Dávila
Juan Andujar

Index

Problem Statement

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 Generative AI 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.

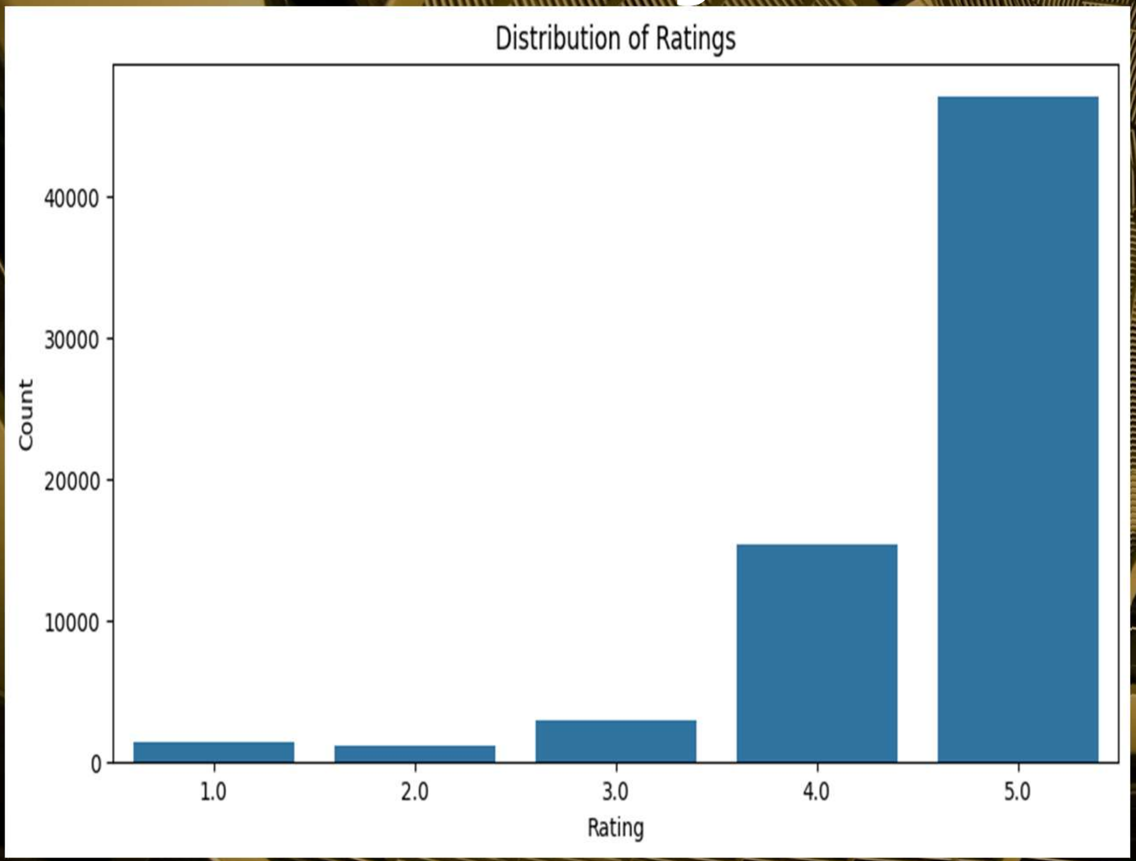
Project Goals

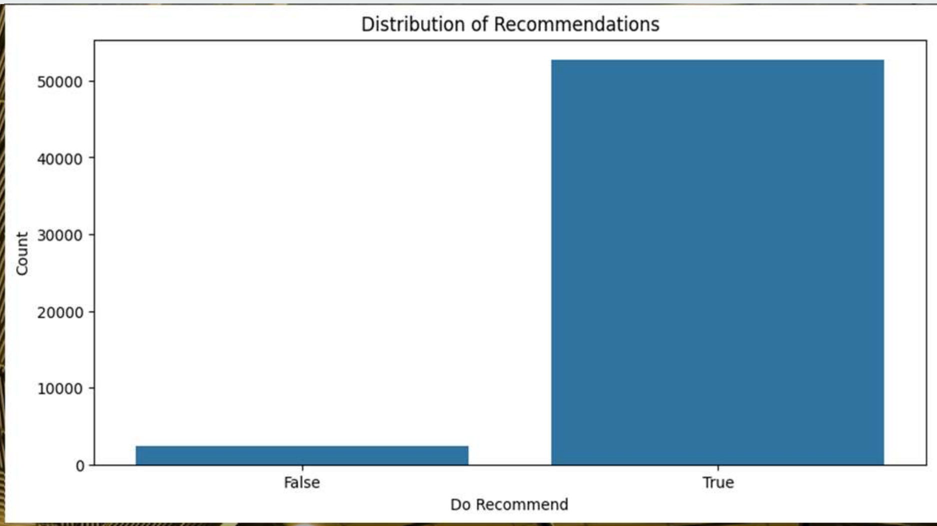
1. **Classify Customer Reviews**: Classify customer reviews (textual content) into positive, neutral, or negative.
2. **Summarize Reviews**: Summarize reviews for each product category broken down by star rating.
3. **Handle Multiple Product Categories**: Manage a feasible number of product categories,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

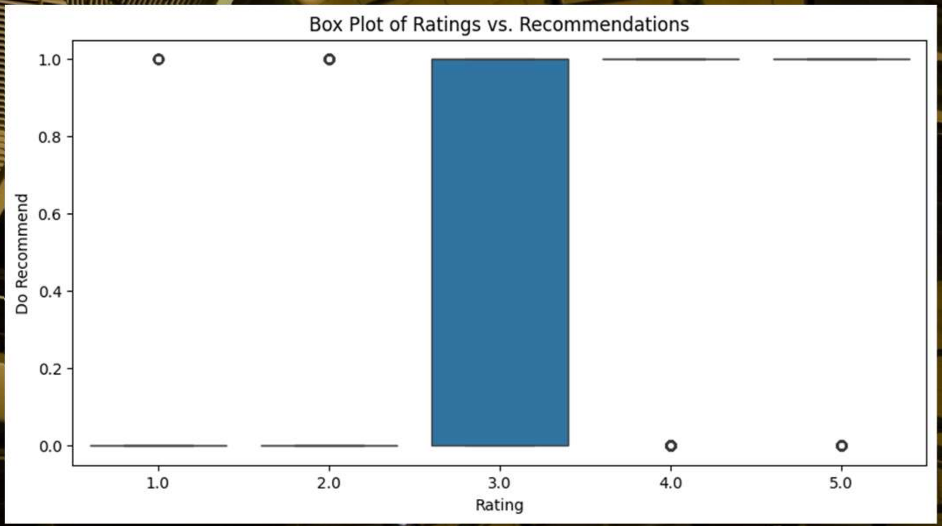
Data Collection

- Utilized publicly available datasets of Amazon customer reviews. (https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products/data?select=Datafiniti_Amazon_Consumer_Reviews_of_Amazon_Products_May19.csv)
- 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

Data Analysis







Steps for cleaning the data

- Drop unnecessary columns
- Check for missing values
- Drop rows missing text and ratings
- Calculate Reviews length
- Correlation between review length and rating
- Remove Lowercase , Stopwords, and Lemmatization
- Split the clean data into training and testing sets

- Encode ratings into positive (2), neutral (1), and negative (0)
  - `Original training set class distribution:`
  - `sentiment`
  - `2     49941`
  - `1      2361`
  - `0      1988`

  - `Resampled training set class distribution:`
  - `sentiment`
  - `2     49941`
  - `1     49941`
  - `0     49941`

Traditional NLP & ML Approaches

a. **Dataset Preparation:** Clean and preprocess the provided dataset, focusing on critical features.

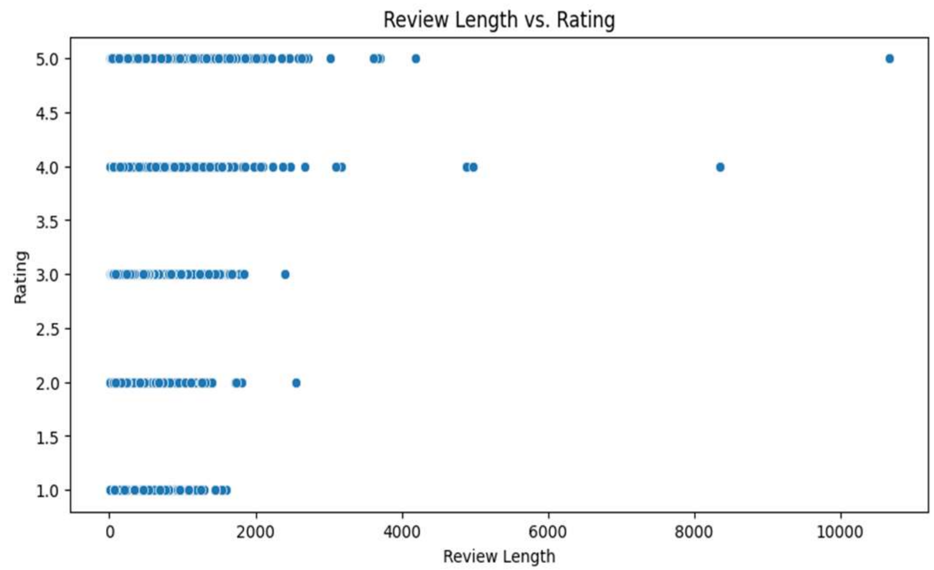
b. **Sentiment Mapping:** Map ratings to sentiment labels.

c. **Feature Engineering:** Balance classes suing SMOTE 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.



Training models & Findings

```
Training Random Forest...
Evaluating Random Forest...
Accuracy: 0.95
Precision: 0.95
Recall: 0.95
F1 Score: 0.95
Confusion Matrix:
[[  307    18   192]
 [   23   233   281]
 [   49    71 12399]]
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.59      0.69       517
           1       0.72      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
```

```
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 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 Logistic Regression...
Evaluating Logistic Regression...
Accuracy: 0.84
Precision: 0.93
Recall: 0.84
F1 Score: 0.88
Confusion Matrix:
[[  370    96    51]
 [   77   327   133]
 [  455  1317 10747]]
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 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 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]]
```

Model Selection

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**

Random forest Classifier Hyperparameter tuning

Class Weight Adjustment

```
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
```

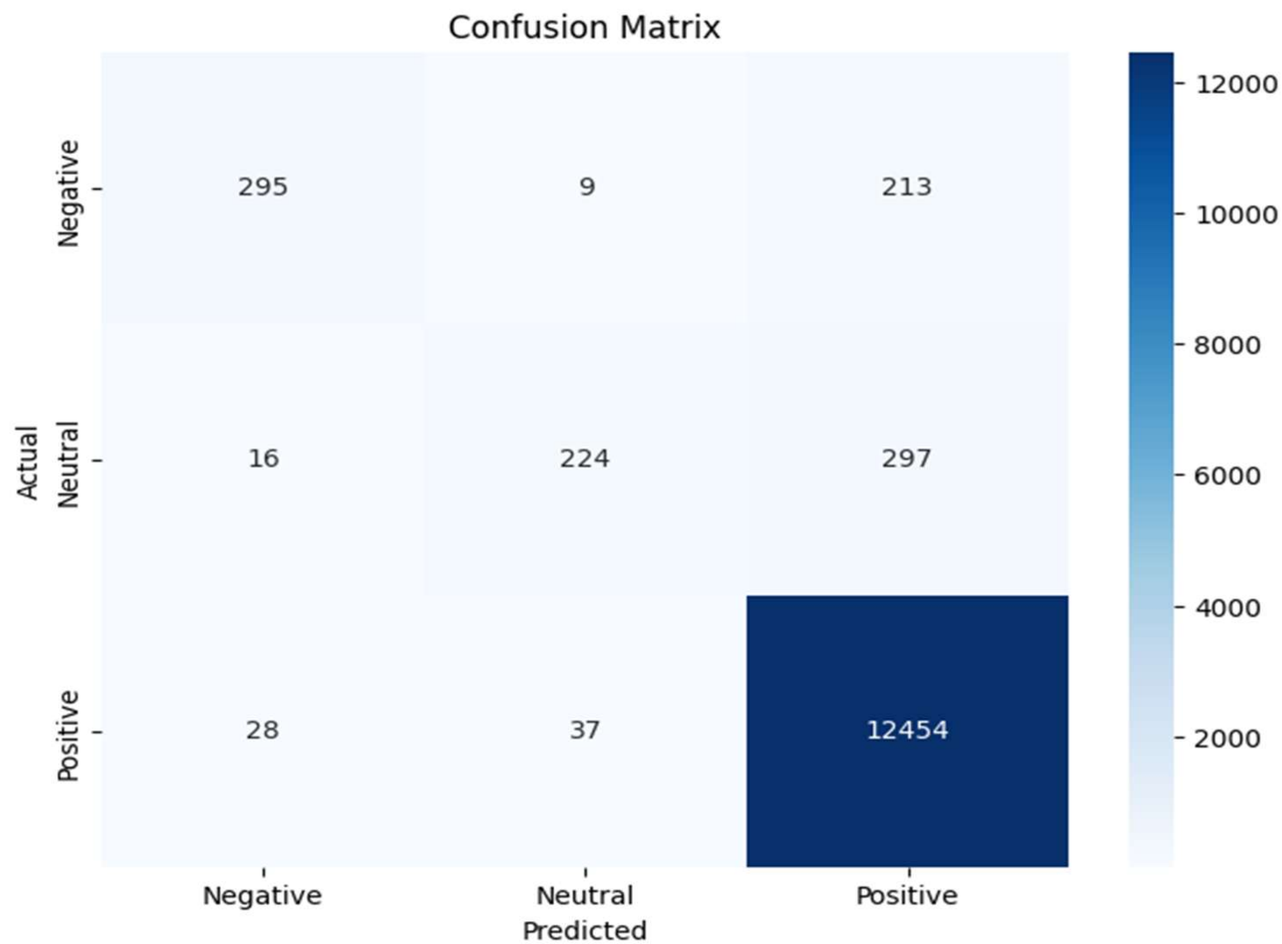
Hyperparameter Tuning with Grid Search

```
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
```

Confusion Matrix



Sequence-to-Sequence Modeling with LSTM

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

02 | **Preprocessing**: Applied the same preprocessing steps as for traditional models.

03 | **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

```
# 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}')
```

Transformer Approach (Hugging Face API)

Data Preprocessing

- Cleaned and tokenized the customer review data to remove special characters, punctuation, and unnecessary whitespace.
- Used HuggingFace Transformers tokenizer (distilbert-base-uncased, roberta-base)
- We reduced the dataset to 2,500 rows per sentiment. (7,500 in total from + 60,000)

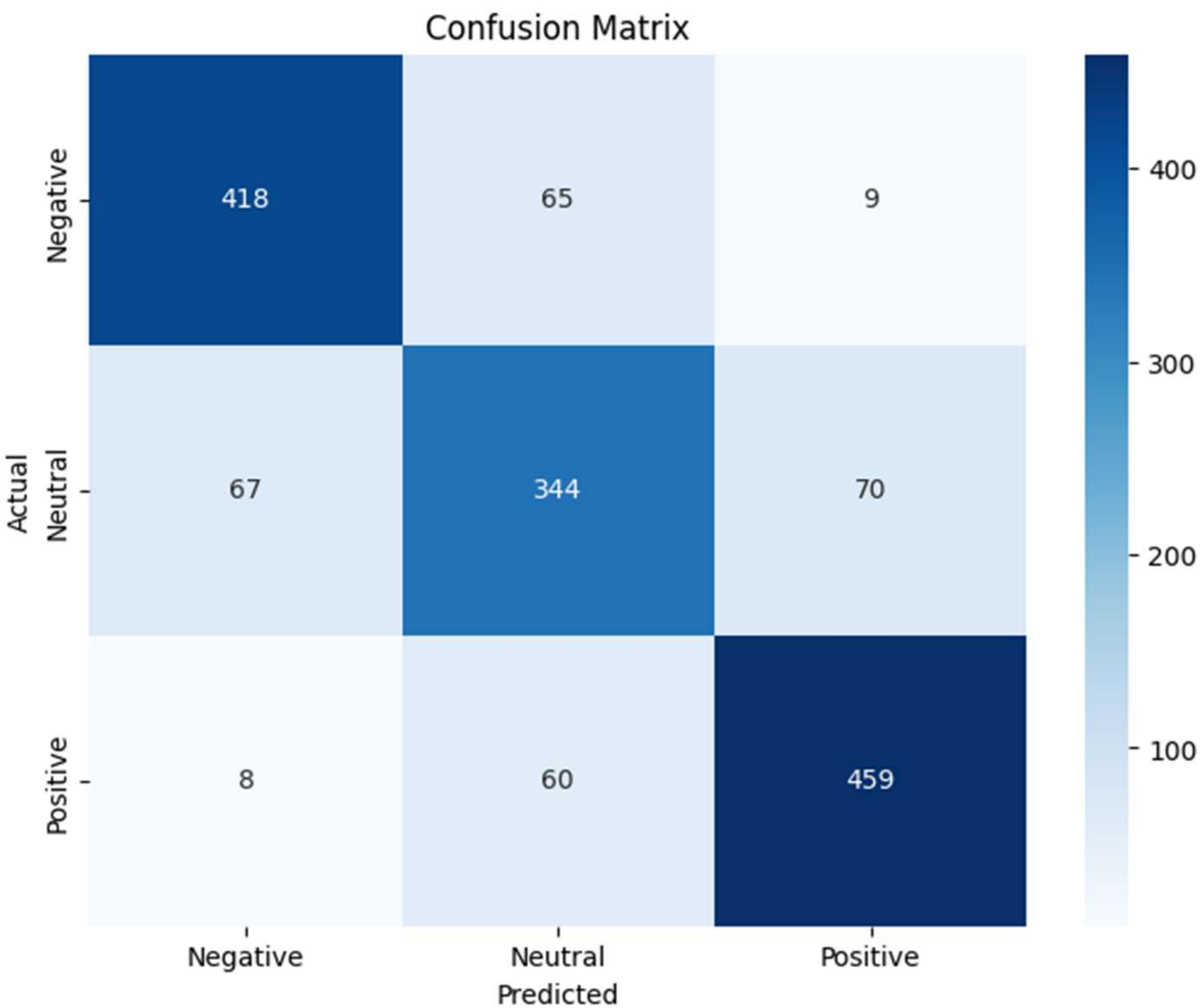
Model Selection

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

1. **Results: RoBERTa**

- F1 score(macro avg) 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

RoBERTa



Confusion Matrix
Negative
Neutral
Positive
Actual
418
65
9
67
344
70
8
60
459
Negative
Neutral
Positive
Predicted
400
300
200
100

Conclusion

- **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 memory capacity and a large dataset 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.

Reviews Summarization using Dynamic Visualization Dashboard

Implementation Process:

1. Data Preparation - Text cleaning & Tokenization
2. Summarization Using ***T5 Model***
3. Grouping by Product Category and by Rating Stars
4. Creation of an Interactive Dashboard Using Plotly
   a. Features: Dropdown menu, Dynamic Summaries, and Interactive Interface

Challenges Found

- Class Imbalance
- Text Preprocessing
- Conflicting Sentiment Labels
- Hyperparameter Tuning
- Model Performance

Gracias.