Sentiment Analysis Of Product Reviews

Maneesh Kumar Chintakindi
Department of
Computer Science and Engineering
University Of North Texas
Denton, Texas, USA
maneeshkumarchintakindi@my.unt.edu

Hemanth Anuginti
Department of
Computer Science and Engineering
University Of North Texas
Denton, Texas, USA
hemanthanuginti@my.unt.edu

Revanth Putta
Department of
Computer Science and Engineering
University Of North Texas
Denton, Texas, USA
revanthputta@my.unt.edu

1. Introduction

1.1. Motivation:

In several domains, including corporate intelligence, social media monitoring, and customer experience management, sentiment analysis–the act of examining textual data to comprehend views and emotions has become essential. In order to ascertain if user evaluations are favorable or negative and to derive more profound insights, such the causes of certain feelings, this study focuses on analyzing user reviews from websites. We advocate for sentiment analysis in order to better understand customer behavior and improve services.

Utilitarianism is the ethical theory used here since it emphasizes using sentiment analysis findings to maximize benefits for the largest number of stakeholders.

1.2. Significance:

1.2.1. Financial Impact: *Identifies trends in customer opinions about products.*Predicts future sales trends and reduces product returns.

1.2.2. Customer Experience: *Empowers businesses to identify and address customer concerns proactively. *Ensures continuous improvement of products , services.

1.2.3. Operational Efficiency: *Provides useful insights for product lifecycle management.*Supports efficient inventory planning and reduces burden on support teams.

1.3. Objectives:

*Understanding and exploring the dataset.
Performing EDA and data- preprocessing. Identifying patterns in the data.*Identifying common themes in reviews. *Analyzing sentiment trends by product category. *Understanding the relationship between sentiment and star ratings. *Identifying trends in review activity overtime. *Exploring the impact of review length on sentiment. *Examining the relationship between product price and sentiment. *Analyzing review behaviour by reviewer profiles. *Building ML models for sentimental classification. *Comparing model performance.

1.4. Features:

*Textual Data: The primary input for sentiment analysis comes from the textual content of the product reviews.
*Preprocessed Features: Text processing includes tokenization , stopword removal, stemming and lemmatization, and TF-IDF.
*Sentiment Analysis: The target variable for the model is the sentiment label, which classifies the review as positive, negative or neutral.

1.5. Related Work:

*Data Preparation: Cleaning the reviews dataset by removing stop-words, special characters and performing tokenization.
*Model Development and Tuning:
Developing Naive Bayes, SVM and CNN models.
*Model Comparision:
Evaluating the models based on their F1 score.
*Evaluation Metric:
The F1 score will be used as the primary metric to evaluate models performance.

2. Background

*Computational linguistics and Natural Language Processing(NLP) are the foundation of sentiment analysis. Overtime, advances in Machine Learning and Deep Learning have altered its use. At first, sentiment analysis was based on lexicon-based methods, in which word lists were used to establish polarity. However, the advent of machine learning models like Naive Bayes and SVMs led to the creation of data-driven classifications.

*Deep Learning techniques, particularly CNN and RNN have revolutionized the accuracy of sentiment analysis, as they can capture contextual information in the text. Modern sentiment analysis is used by e-commerce platforms, new outlets and public opinion researchers. However, problems including bias in datasets, the ethical implications of utilizing data, and the intricacies of interpreting nuanced sentiments still pose substantial dilemmas.

3. Ethical Framework:

*This research uses utilitarianism as its frame of ethics, highlighting that the social community can benefit from sentiment analysis. User reviews also give insights into how to improve a product or service, which can ultimately help improve customer satisfaction. Also sentiment analysis provides insight into the public opinion which guides policies, marketing strategies and products.

*Utilitarianism is relevant in its concern for maximizing positive outcomes in the deployment of sentiment analysis and minimizing the negative, such as privacy infringements and misuse of user data. Leveraging this framework can help ensure that when valuable insights are derived from sentiment analysis, it will be performed responsibly to generate value for both organizations and users alike, while also respecting ethical considerations, such as data anonymity and fairness.

4. Personal Position:

Engagement is lovely but driving effective delivery means determining this utility. For example, it has been found that through effective analysis of reviews, businesses are able to improve customer experiences by directly tackling specific issues they have faced. Sentiment classification is done using Machine Learning algorithms of hypertuned Naive Bayes, SVM and CNN and sentiment reasoning with CNN.

Here is our evidence supporting our side:

*Enhanced Accuracy:In contrast to conventional ML models, CNNs use word embeddings, allowing them to comprehend each character's contextual significance for sentiment classification.

*Practical Applications:By enabling companies like Amazon and Netflix to personalize recommendations, sentiment analysis increases user engagement.

*Scalability:Organizations can efficiently process large volumes of text data using automated sentiment analysis.

*Better Customer Insights:Sentiment analysis provides useful information about customer preferences that goes beyond polarity, assisting companies in customizing their product and service offerings.

*Quicker Choice-Making: Business may improve customer happiness and loyalty by promptly responding to consumer feedback thanks to real-time analytic technologies.

*Global Adaptability: Organizations may expand their reach by using mutilingual sentiment analysis to comprehend feelings across many languages and geographical areas.

*Applications Across Industries: Sentiment analysis has shown promise in politics, healthcare and entertainment for gaining insight into public trends and opinion, in addition to e-commerce.

Our team for this research included **Revanth, Hemanth and Maneesh** each of whom contributed distinct responsibilities and viewpoints to the study. **Maneesh** was tasked with running the code and showcasing the usefulness of machine learning models such as hypertuned Naive Bayes, SVM and CNN while **Revanth** and **Hemanth** served as critics, contesting the implementation and raising doubts about the underlying assumptions of sentiment analysis.

*Execution vs Critics: The validity of sentiment analysis results was frequently at the heart of the discussion. According to **Revanth** and **Hemanth**.

Complex human emotions may be oversimplified by sentiment analysis. Despite their strength, models like CNN may still overlook colloquial language or sarcasm, among other contextual nuances. They also warned against relying too much on computer models for decision making and underlined the moral dangers of unintentional biases in training data. **Maneesh** refuted these accusations by presenting the actual outcomes of his applications:
1. CNN beat previous models by using word embeddings to capture contextual meanings, which led to improved accuracy.
2. Business relevant results, such identifying common pain areas and actionable insights from user evaluations, were inline with sentiment patterns produced from study.

Ultimately, this relationship between critics and implementers produced a balanced strategy, guaranteeing that the initiative was both motivated by real world results and ethical concerns.
Revanth and **Hemanth** kept an eye on things to make sure they were transparent and accountable, but **Maneesh's** technical know-how made the idea a reality.

4. ANALYSIS:

Data Collection

*Dataset selection:
Gather a comprehensive dataset of product reviews.
*Data Size Considerations:
Ideally contain thousands of product reviews for improved model accuracy.

Preprocessing of Data:

*Cleaning like handling missing values and removing dupicates and whitespaces.
*Tokenization means splitting review data into chunks.

Model Construction:

*Dataset divide into training and testing sets.
*Several Machine Learning algorithms are trained on the trained dataset.

5. IMPLEMENTATION:

***Data preparation:**
Includes data cleaning, preprocessing, handling imbalances, feature engineering and additional feature construction.

***Model Validtion and Training:**
In this, model selection, training the models and cross validaton and hold out set are performed.

***Performance Evaluation:**
This includes focusing on evaluation metrics like accuracy,precision,recall,F1 score and model comparision.

***Model Refinement:**
Refinement has been done through hyperparameter tuning and analysis has been done.

6. Preliminary Results:

6.1. Initially loading the libraries:

```
Step 1: Importing Required Libraries and Data Collection

[ ] # Importing necessary libraries
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
    from sklearn.linear_model import LogisticRegression
    from sklearn.feature_extraction.text import TfidfVectorizer
    import joblib
    import warnings
    from collections import Counter
    from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report, confusion_matrix

[ ] # Text preprocessing
    import nltk
    nltk.download('stopwords')
    nltk.download('punkt')
    from nltk.corpus import stopwords
    from nltk.tokenize import word_tokenize
```

Figure 1. loading the libraries

6.2. Reading the dataset

```
[85] # Loading the dataset with the correct delimiter
     file_path = r"/content/Amazon Product Review.txt"
     data = pd.read_csv(file_path, delimiter=",", encoding="utf-8")

     # Display the first few rows of the dataset to verify its structure
     data.head()
```

...	verified_purchase	review_headline	review_body	review_date	sentiment	cleaned_headline	cleaned_body
...	Y	Five Stars	Great love it	2015-08-31	1	five stars	great love
...	N	Lots of ads Slow processing speed Occasionally...	Lots of ads Slow processing speed Oc...	2015-08-31	0	lots ads slow processing speed occasionally sh...	lots adsslow processing speedoccasionally shut...
...	Y	Well thought out device	Excellent unit. The versatility of this table...	2015-08-31	1	well thought device	excellent unit versatility tablet besides comp...

Figure 2. By using the read csv,loaded the dataset

6.3. Understanding the Dataframe

```
data.info()

<class 'pandas.core.frame.DataFrame'>
Index: 30840 entries, 0 to 30845
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   marketplace          30840 non-null  object
 1   customer_id          30840 non-null  int64
 2   review_id            30840 non-null  object
 3   product_id           30840 non-null  object
 4   product_parent       30840 non-null  int64
 5   product_title        30840 non-null  object
 6   product_category     30840 non-null  object
 7   star_rating          30840 non-null  int64
 8   helpful_votes        30840 non-null  int64
 9   total_votes          30840 non-null  int64
 10  vine                 30840 non-null  object
 11  verified_purchase    30840 non-null  object
 12  review_headline      30840 non-null  object
 13  review_body          30840 non-null  object
 14  review_date          30840 non-null  object
 15  sentiment            30840 non-null  int64
 16  cleaned_headline     30840 non-null  object
 17  cleaned_body         30840 non-null  object
 18  sentiment_label      30840 non-null  object
 19  cleaned_review_body  30840 non-null  object
 20  review_length        30840 non-null  int64
dtypes: int64(7), object(14)
memory usage: 5.2+ MB
```

Figure 3. Finding out the number of columns and rows: We have 30,840 rows and 21 columns.

6.4. Statiscal understanding of numerical columns:

```
data.describe()
```

	customer_id	product_parent	star_rating	helpful_votes	total_votes	sentiment	review_length
count	3.084000e+04	30840.0	30840.000000	30840.000000	30840.000000	30840.000000	30840.000000
mean	2.470843e+07	2693241.0	4.336316	0.686543	0.894942	0.835376	16.286770
std	1.611152e+07	0.0	1.120569	30.746359	32.359474	0.370847	30.454573
min	1.134600e+04	2693241.0	1.000000	0.000000	0.000000	0.000000	0.000000
25%	1.150586e+07	2693241.0	4.000000	0.000000	0.000000	1.000000	3.000000
50%	2.294012e+07	2693241.0	5.000000	0.000000	0.000000	1.000000	8.000000
75%	4.008417e+07	2693241.0	5.000000	0.000000	0.000000	1.000000	18.000000
max	5.309351e+07	2693241.0	5.000000	3720.000000	3875.000000	1.000000	1174.000000

Figure 4. Describe function is used for this statistical understanding of columns.

6.5.Sentiment labels mapped to star ratings

Sentiment labels based on star ratings:

	star_rating	sentiment_label
0	5	positive
1	3	neutral
2	5	positive
3	4	positive
4	5	positive

Figure 5. In this the labels are mapped to star ratings.

7. Exploratory Data Analysis:

7.1 Sentiment distribution across start ratings

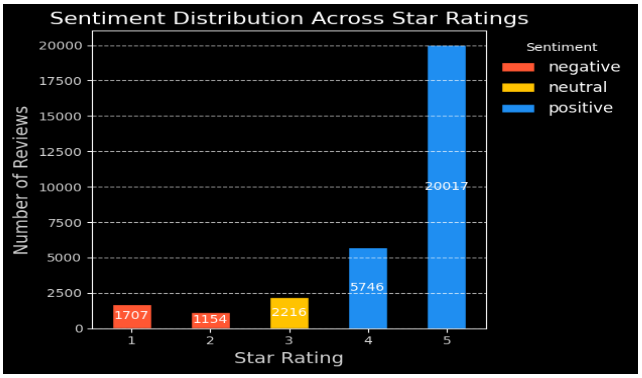


Figure 6. The graph shows variation in sentimental reviews with respect to number of reviews.

7.2 Sentiment distribution across product categories

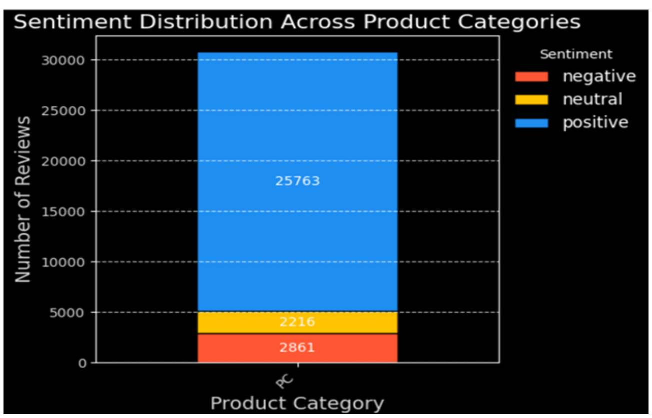


Figure 7. This graph shows product category sentiment distribution analysis.

7.3 Length of reviews by sentiment

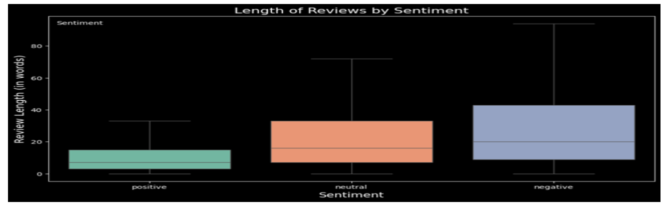


Figure 8. The review length analysis reveals that positives reviews are short, with a median of 15-20 words, while neutral reviews are slightly longer.Negative reviews are longest and positive reviews are short.

7.4 Most common words in reviews

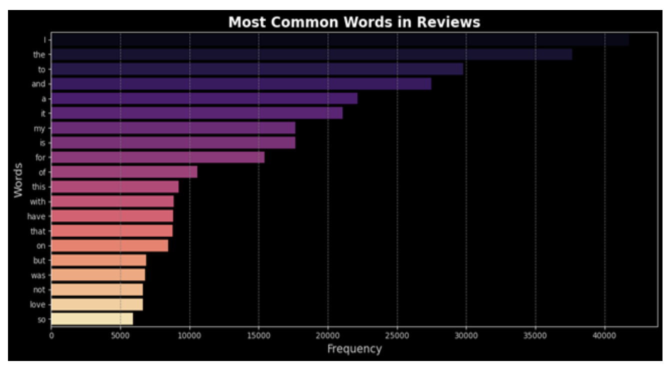
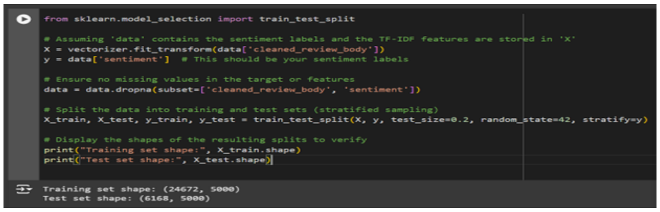


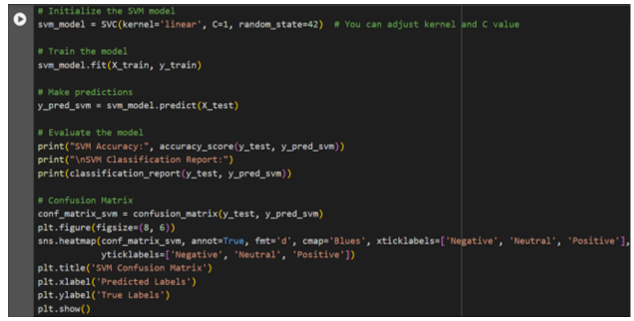
Figure 9. The plot shows the most frequently occuring words in product reviews, with word like "the","to","and" and "my" dominating the dataset. These words are mostly stopwords.

8. Model Selection and Model Training:



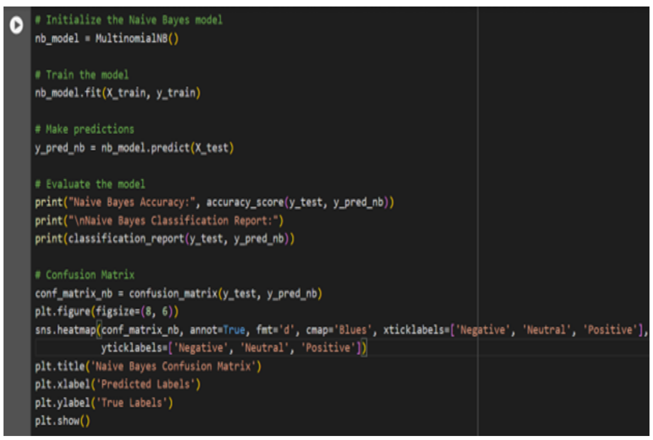
```
from sklearn.model_selection import train_test_split

# Assuming 'data' contains the sentiment labels and the TF-IDF features are stored in 'X'
X = vectorizer.fit_transform(data['cleaned_review_body'])
y = data['sentiment']  # This should be your sentiment labels

# Ensure no missing values in the target or features
data = data.dropna(subset=['cleaned_review_body', 'sentiment'])

# Split the data into training and test sets (stratified sampling)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Display the shapes of the resulting splits to verify
print("Training set shape:", X_train.shape)
print("Test set shape:", X_test.shape)

Training set shape: (24672, 5000)
Test set shape: (6168, 5000)
```

Figure 10. The data is divide into training and testing set.



```
# Initialize the SVM model
svm_model = SVC(kernel='linear', C=1, random_state=42)  # You can adjust kernel and C value

# Train the model
svm_model.fit(X_train, y_train)

# Make predictions
y_pred_svm = svm_model.predict(X_test)

# Evaluate the model
print("SVM Accuracy:", accuracy_score(y_test, y_pred_svm))
print("\nSVM Classification Report:")
print(classification_report(y_test, y_pred_svm))

# Confusion Matrix
conf_matrix_svm = confusion_matrix(y_test, y_pred_svm)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_svm, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Neutral', 'Positive'],
            yticklabels=['Negative', 'Neutral', 'Positive'])
plt.title('SVM Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

Figure 12.Training the model through SVM.



```
# Initialize the Naive Bayes model
nb_model = MultinomialNB()

# Train the model
nb_model.fit(X_train, y_train)

# Make predictions
y_pred_nb = nb_model.predict(X_test)

# Evaluate the model
print("Naive Bayes Accuracy:", accuracy_score(y_test, y_pred_nb))
print("\nNaive Bayes Classification Report:")
print(classification_report(y_test, y_pred_nb))

# Confusion Matrix
conf_matrix_nb = confusion_matrix(y_test, y_pred_nb)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_nb, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative', 'Neutral', 'Positive'],
            yticklabels=['Negative', 'Neutral', 'Positive'])
plt.title('Naive Bayes Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

Figure 11.Training the model through Naive Bayes.

```
# Initialize the CNN model
cnn_model = Sequential()

# Add an embedding layer to transform words into dense vectors (you could also use pre-trained embeddings)
cnn_model.add(Embedding(input_dim=len(vectorizer.get_feature_names_out()), output_dim=100, input_length=X.shape[1]))

# Add convolutional layer (1D CNN)
cnn_model.add(Conv1D(filters=64, kernel_size=5, activation='relu'))

# Add a max-pooling layer
cnn_model.add(MaxPooling1D(pool_size=4))

# Add a global max pooling layer
cnn_model.add(GlobalMaxPooling1D())

# Add a fully connected layer
cnn_model.add(Dense(64, activation='relu'))

# Add the output layer with softmax activation for multi-class classification
cnn_model.add(Dense(3, activation='softmax'))  # Assuming 3 classes: Negative, Neutral, Positive

# Compile the model
cnn_model.compile(loss='sparse_categorical_crossentropy', optimizer=Adam(), metrics=['accuracy'])
```

Figure 13.Training the model through CNN.

9. Model Evaluation(Results of each model):

```
SVM Accuracy: 0.9045071335927367

SVM Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.62      0.68      1015
           1       0.93      0.96      0.94      5153

    accuracy                           0.90      6168
   macro avg       0.84      0.79      0.81      6168
weighted avg       0.90      0.90      0.90      6168
```

Figure 14.
This is the result of SVM model, it achieved the highest accuracy at 90.45 with a precision of 0.75 and 0.93 for 0,1. However, it struggled with recall for class 0, only correctly identifying 62 percent of the actual negative instances.
For Positive reviews, the SVM performed well with a 96 per recall and a high 94per F1-score, indicating strong performance in distinguishing positive reviews but potential challenges with negative reviews.

```
Naive Bayes Accuracy: 0.8934824902723736

Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.51      0.61      1015
           1       0.91      0.97      0.94      5153

    accuracy                           0.89      6168
   macro avg       0.84      0.74      0.78      6168
weighted avg       0.89      0.89      0.88      6168
```

Figure 15.
The Naive Bayes model had a slightly lower accuracy with 76 and 91 precision for negatives, positives. However, its recall for negative sentiment dropped significantl to 51 per, indicating it failed to capture more than half of the actual negative reviews. Despite this, the recall for positive sentiment remained high at 97per, making it highly effective at identifying positive reviews.

```
Epoch 1/5
771/771 ━━━━━━━━━━━━━━━━━━━━ 89s 113ms/step - accuracy: 0.8377 - loss: 0.4767 - val_accuracy: 0.8354 - val_loss: 0.4482
Epoch 2/5
771/771 ━━━━━━━━━━━━━━━━━━━━ 141s 112ms/step - accuracy: 0.8349 - loss: 0.4521 - val_accuracy: 0.8354 - val_loss: 0.4530
Epoch 3/5
771/771 ━━━━━━━━━━━━━━━━━━━━ 90s 116ms/step - accuracy: 0.8334 - loss: 0.4537 - val_accuracy: 0.8354 - val_loss: 0.4478
Epoch 4/5
771/771 ━━━━━━━━━━━━━━━━━━━━ 140s 114ms/step - accuracy: 0.8365 - loss: 0.4476 - val_accuracy: 0.8354 - val_loss: 0.4487
Epoch 5/5
771/771 ━━━━━━━━━━━━━━━━━━━━ 89s 116ms/step - accuracy: 0.8374 - loss: 0.4453 - val_accuracy: 0.8354 - val_loss: 0.4474
193/193 ━━━━━━━━━━━━━━━━━━━━ 5s 26ms/step - accuracy: 0.8453 - loss: 0.4319
CNN Accuracy: 0.835440993309021
193/193 ━━━━━━━━━━━━━━━━━━━━ 5s 27ms/step
```

Figure 16.
The CNN model with accuracy of 83.54per lags behind both the SVM and Naive Bayes models, suggesting a limited ability to generalize or learn from the dataset. CNN may require further tuning or more extensive data to improve results.

10. Analyzing Performance through Confusion Matrix:

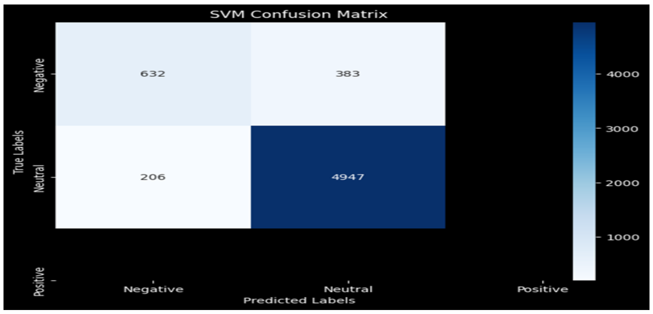


Figure 17.
The SVM model is effective in predicting neutral sentiments with 4947 correct classifications. However there is a confusion between negative and neutral classes, with 383 negative instances misclassified as neutral and 20 as negative. This suggests the model may struggle to distinguish between reviews with midly negative or neutral sentiments. The absence of positive sentiment in the confusion matrix suggests either no positive reviews were present or the model failed to identify any positive sentiments. This could indicate a class imbalance or a need for further tuning to capture positive sentiments.

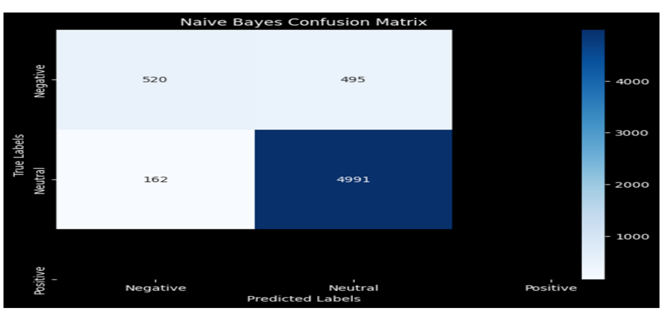


Figure 18.
The Naive Bayes confusion matrix outperformed the CNN confusion matrix in predicting negative and neutral sentiments. However, it still misclassified negative reviews as neutral and neutral reviews as negative, indicating confusion due to overlapping text data features. The study suggests that while Naive Bayes performs better in distinguishing between sentiments, further optimization is needed. Techniques like feature engineering, hyperparameter tuning and incorporate additional features like sentiment scores or part of speech tags could be considered. Analyzing misclassified examples can provide insights into the models struggles and guide adjustments to improve classification accuracy.

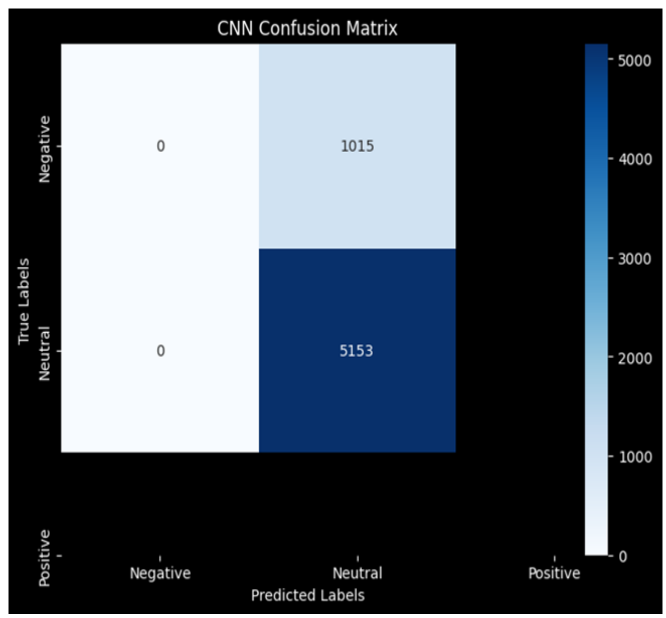


Figure 19.
The CNN Confusion Matrix reveals significant issues with the sentiment analysis model, indicating that it predicted neutral sentiment for all instances, while true negative suggets a bias towards predicting a single dominant class, possibly due to dataset imbalance, inadequate training, or a poorly tuned model. The lack of diversity in predictions suggests the need for further refinement, including rebalancing the dataset, using techniques like oversampling or undersampling, and enhancing feature extraction methods to capture sentiment related nuances.

11. Rebuttal:

Despite its advantages, sentimental analysis is criticized for a number of reasons. We discuss the main counter arguments below:

*Privacy Issues
Critics contend that it violates privacy to analyze user data without authorization.
Rebuttal: Our work protects privacy by anonymizing data and according to ethical standards.

*Sentiment Classification Bias:
Potential biases in training data that might provide distorted findings are brought up by opponents.
Rebuttal: In order to reduce bias, we use a variety of datasets and validate models in a number of domains.

Altough legitimate, these counterarguments are lessened by ethical behaviour and technological developments, supporting our position on the value of sentiment analysis.

12. Conclusion:

In summary, sentiment analysis is a game changing technique for better decision making and comprehending user view points. This study illustrates the practical and societal benefits of using ethical frameworks, strong methodology and sophisticated visualization approaches. Sentiment analysis can be applied responsibly to have significant effects by addressing issues like privacy concerns and data biases.

13. Implementation Status Report:

**Dataset Understanding - All 3*
**Dataset Preprocessing - Hemanth,Maneesh*
**EDA Analysis - Revanth,Hemanth*
**Model Creation:*
**Naive Bayes - Maneesh*
**SVM - Hemanth*
**CNN - Revanth*
**Model Comparision and Evaluation - All 3*
**Report - All teammates contributed.*

14. References:

1. Tanushree Dholpuria, Y.K Rana, Chetan Agrawal, “A Sentiment analysis approach through deep learning for a movie review” IEEE-2018
2. Upma Kumari, Dr. Arvind K sharma, Dinesh Soni, “Sentiment analysis of Smart Phone Product Review Using SVM Classification Technique” 2017 IEEE
3. Seetha Letchumy M. Belaidan, Lee Hua Fung, “Sentiment Analysis in Online Products Reviews Using Machine Learning” 2021 https://www.webology.org/data-cms/articles/20211102112758amWEB18271.pdf
4. Xing Fang and Justin Zhan, “Sentiment analysis using product review data “ https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2
5. Mrs. J. Samatha, Vaibhav Lahoti, Anusha.V, Deekshitha, “SENTIMENT ANALYSIS ON PRODUCT REVIEWS USING BIG DATA “ 2017
6. Alfredo Daza, Néstor Daniel González Rueda, Mirelly Sonia Aguilar Sánchez , Wilmer Filomeno Robles Espíritu , María Elena Chauca Quiñones, “Sentiment Analysis on E-Commerce Product Reviews Using Machine Learning and Deep Learning Algorithms: A Bibliometric Analysis, Systematic Literature Review, Challenges and Future Works”