

Sentiment Analysis of Amazon Food Reviews:

Enhancing Product Insights with Machine Learning

Group 23: Shashwat Kumar(sk5520), Saatvik
Saradhi Inampudi(si2464), Tanner
Hillison(tkh2119), Mengxi Liu(mi5189),
Gaifan Zhang(gz2360)



COLUMBIA ENGINEERING

The Fu Foundation School
of Engineering and Applied Science



Problem Statement & Dataset

Objective: To develop a robust machine learning model that accurately predicts the sentiment (Positive, Neutral, Negative) of Amazon Fine Food reviews, leveraging natural language processing and various machine learning techniques to provide actionable insights for business strategies.

Motivation: Sentiment analysis on large-scale customer reviews can:

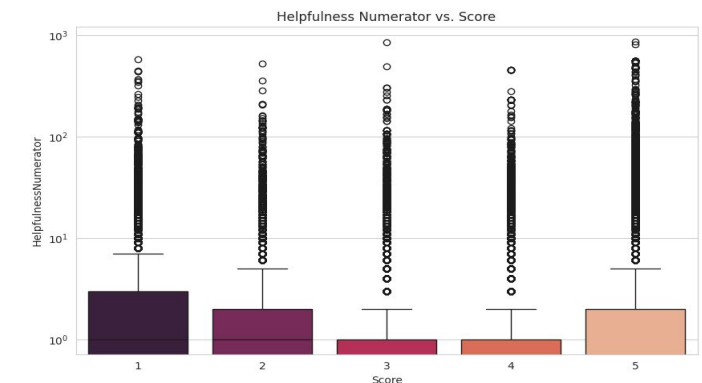
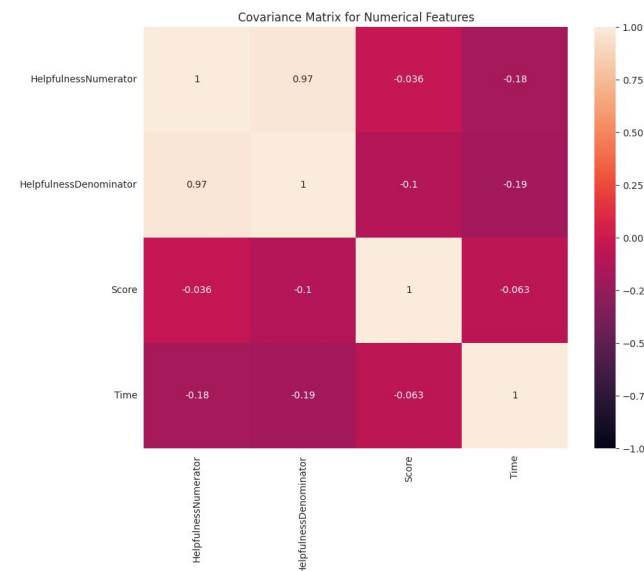
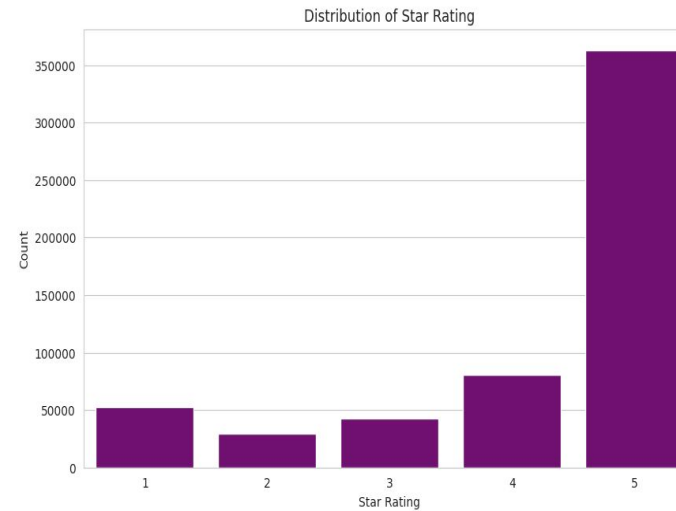
- Improve product recommendations.
- Support targeted marketing.
- Enhance customer satisfaction by addressing common complaints and preferences.
- **Dataset:** [Amazon Amazon Fine Food Reviews](#)
- **Description**
 - Size:** 568,454 reviews over 10 years.
 - Key Features:** ProductId, UserId, Score, Summary, Text, and Helpfulness

	ProductId	UserId	Score	Summary	Text
0	B001E4KFG0	A3SGXH7AUHU8GW	5	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	B00813GRG4	A1D87F6ZCVE5NK	1	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	B000LQOCH0	ABXLMWJIXXAIN	4	"Delight" says it all	This is a confection that has been around a fe...
3	B000UA0QIQ	A395BORC6FGVXV	2	Cough Medicine	If you are looking for the secret ingredient i...
4	B006K2ZZ7K	A1UQRSCLF8GW1T	5	Great taffy	Great taffy at a great price. There was a wid...
5	B006K2ZZ7K	ADT0SRK1MGOEU	4	Nice Taffy	I got a wild hair for taffy and ordered this f...
6	B006K2ZZ7K	A1SP2KVKFXXRU1	5	Great! Just as good as the expensive brands!	This saltwater taffy had great flavors and was...
7	B006K2ZZ7K	A3JRGQVEQN31IQ	5	Wonderful, tasty taffy	This taffy is so good. It is very soft and ch...
8	B000E7L2R4	A1MZY09TZK0BBI	5	Yay Barley	Right now I'm mostly just sprouting this so my...
9	B00171APVA	A21BT40VZCCYT4	5	Healthy Dog Food	This is a very healthy dog food. Good for thei...
10	B0001PB9FE	A3HDKO7OW0QNK4	5	The Best Hot Sauce in the World	I don't know if it's the cactus or the tequila...
11	B0009XLVG0	A2725IB4YY9JEB	5	My cats LOVE this "diet" food better than thei...	One of my boys needed to lose some weight and ...
12	B0009XLVG0	A327PCT23YH90	1	My Cats Are Not Fans of the New Food	My cats have been happily eating Felidae Plati...
13	B001GVISJM	A18ECVX2RJ7HUE	4	fresh and greasy!	good flavor! these came securely packed... the...
14	B001GVISJM	A2MUGFV2TDQ47K	5	Strawberry Twizzlers - Yummy	The Strawberry Twizzlers are my guilty pleasur...
15	B001GVISJM	A1CZX3CP8IKQIJ	5	Lots of twizzlers, just what you expect.	My daughter loves twizzlers and this shipment ...
16	B001GVISJM	A3KLWF6WQ5BNYO	2	poor taste	I love eating them and they are good for watch...
17	B001GVISJM	AFKW14U97Z6QO	5	Love it!	I am very satisfied with my Twizzler purchase....
18	B001GVISJM	A2A9X58G2GTBLP	5	GREAT SWEET CANDY!	Twizzlers, Strawberry my childhood favorite ca...
19	B001GVISJM	A3IV7CL2C13K2U	5	Home delivered twizlers	Candy was delivered very fast and was purchase...

Exploratory Data Analysis (EDA)

Rating Distribution, Correlations, and Term Frequency:

- 1. Distribution of Star Rating:** Illustrates the skewed distribution of ratings, with a notable majority of reviews being rated 5 stars.
- 2. Covariance Matrix for Numerical Features:** Visualizes correlations between numerical features, highlighting a strong correlation between Helpfulness Numerator and Helpfulness Denominator.
- 3. Helpfulness Numerator vs. Score:** Shows the distribution of helpfulness votes across different scores, indicating higher helpfulness for extreme ratings.
- 4. Common Words in Positive vs. Negative Reviews:** Word clouds displaying the most frequent terms in positive and negative reviews, revealing distinct vocabulary patterns based on sentiment.



Data Cleaning

Missing Values Check and Removal:

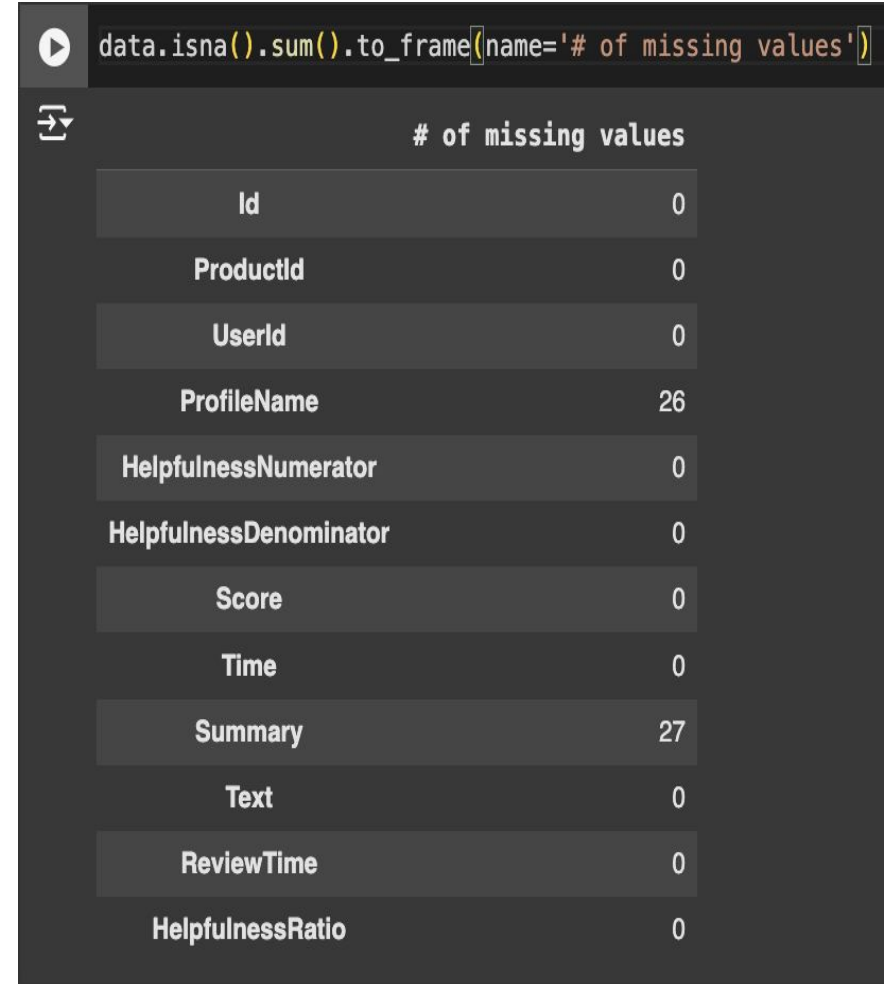
- **Method:** `data.isna().sum().to_frame(name='# of missing values')`
- **Justification:** Dropped rows with missing values in critical columns, removing only a small portion of data (0.01%), ensuring minimal data loss while retaining data quality.

Removing Duplicates:

- **Method:** Removed duplicates based on Score and Text columns.
- **Impact:** This reduced the dataset by 30.74%, removing redundant reviews and avoiding repeated influence from identical reviews.
- **Justification:** Removing missing values and duplicates ensures data integrity and minimizes redundancy, creating a cleaner dataset for accurate model training.

Overview of Features:

- **Numerical Features:** HelpfulnessNumerator, HelpfulnessDenominator
- **Categorical Features:** UserId, ProductId, ProfileName, Score
- **Text Features:** Summary, Text
- **Data or Time Features:** Time



```
data.isna().sum().to_frame(name='# of missing values')
```

	# of missing values
Id	0
ProductId	0
UserId	0
ProfileName	26
HelpfulnessNumerator	0
HelpfulnessDenominator	0
Score	0
Time	0
Summary	27
Text	0
ReviewTime	0
HelpfulnessRatio	0

Text Preprocessing and Exploration

Text Cleaning Pipeline: The preprocessor function applies several steps to prepare reviews for analysis:

- **Remove HTML Tags:** Eliminates any HTML elements to focus on review content.
- **Remove Punctuation:** Strips punctuation marks, making the text uniform.
- **Remove Digits:** Removes all numeric characters to focus solely on text content.
- **Lowercasing:** Converts all text to lowercase, ensuring uniformity across words.
- **Replace Multiple Whitespaces:** Replaces extra whitespaces with a single space to avoid unnecessary gaps in text.

```
▶ print("Before preprocessing : ")
data.Text.iloc[6]

↗ Before preprocessing :
'I am a big time tea drinker, and I absolutely enjoyed this particular tea, which has an amazing taste. Enjoy!'

[ ] data.Text = data.Text.apply(preprocessor)
print("After preprocessing : ")
data.Text.iloc[6]

↗ After preprocessing :
'big time tea drinker absolut enjoy particular tea amaz tast enjoy'
```

Stop Words Removal:

- **Definition:** Removed high-frequency, low-info words (e.g., “and,” “the”) that contribute minimally to sentiment detection.
- **Method:** Employed NLTK’s stop word list to streamline text, reducing dimensionality and focusing on terms that enhance sentiment analysis.

Stemming:

- **Definition:** Reduces words to their root forms (e.g., “running” becomes “run”).
- **Method:** Applied NLTK’s Porter stemmer to standardize word forms, reducing vocabulary size.
- **Justification:** Stemming enhances model efficiency by unifying variations of words into a single root, helping the model generalize better.

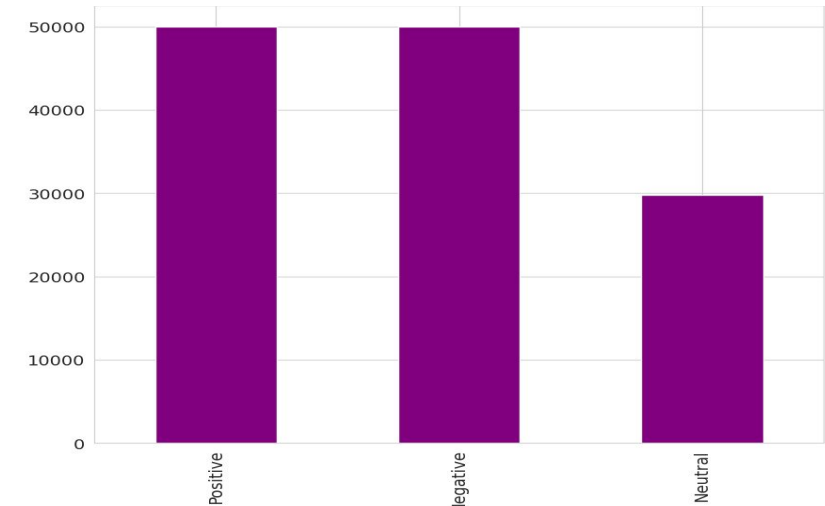
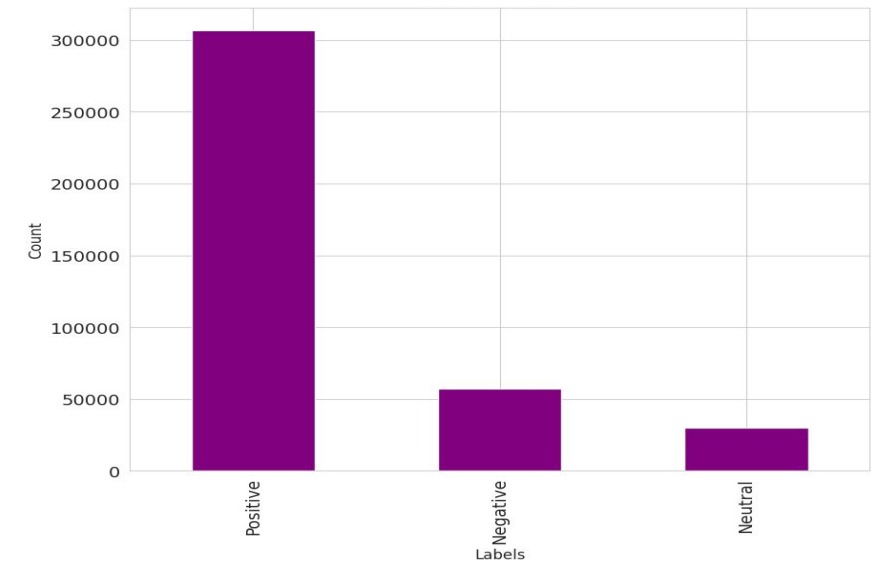
Advanced Data Processing and Feature Extraction

Feature Extraction methods:

- **Bag of Words (BoW):** Created word frequency vectors, capturing the presence of words without considering importance. Applied in models like Logistic Regression and Naive Bayes.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighed terms by balancing frequency with rarity, enhancing relevance of unique words. Used in Naive Bayes, LightGBM, Logistic Regression, and Random Forest models.
- **Tokenization with Embedding Layer:** Converted text to sequences of word indices, then mapped each word to a dense vector space through a learned Embedding layer. Applied in CNN and BiLSTM for semantic understanding.

Handling Class Imbalance:

- **Imbalance in Classes:** The dataset was skewed toward positive reviews, with fewer neutral and negative reviews.
- **Solution:** Used two techniques: (1) Downsampling for both the positive and negative classes to achieve a balanced dataset; (2) Reweighting each sample to balance the training procedure for CNN.
- **Justification:** Balancing the dataset prevents the model from being biased toward the majority class, ensuring fair representation of each sentiment class.



Implemented Machine Learning Models

Baseline Models:

- **Logistic Regression:** A linear classifier with **BoW** and **TF-IDF**, efficient for establishing baselines in text classification and used for deployment due to low computational demands.
- **Naive Bayes:** A probabilistic model based on Bayes' theorem, leveraging independence assumptions for fast, scalable classification, well-suited to sparse, high-dimensional data.

Advanced Models:

- **Random Forest:** An ensemble of decision trees trained on bootstrapped samples, capturing complex feature interactions and being robust against overfitting, especially in **TF-IDF** spaces.
- **XGBoost:** Gradient boosting framework optimizing with decision trees, using regularization and pruning to reduce overfitting, effective in imbalanced datasets.
- **LightGBM:** Gradient boosting optimized for large datasets, using histogram-based training for efficiency; handles categorical features and scales well with high-dimensional data.

Deep Learning Models:

- **BiLSTM (Bidirectional LSTM):** RNN variant processing data in forward and reverse, capturing long-term dependencies and context, ideal for sequential data like text.
- **CNN (Convolutional Neural Network):** Uses convolutional layers to detect local patterns in text; combined with word embeddings like **Word2Vec**, it effectively extracts semantic features for classification.

Model Training and Deployment

Training Process:

- **Data Splitting:** 80-20 train-test split
- **Hyperparameter Tuning:**
 - **Logistic Regression:** Tuned C and max_iter
 - **Naive Bayes:** Tuned alpha
 - **Random Forest:** Adjusted n_estimators and max_depth
 - **XGBoost:** Tuned n_estimators and learning_rate

Deployment Using Logistic Regression:

- **Model Choice:** Logistic Regression selected for efficiency and real-time compatibility.
- **Preprocessing:** Text cleaned and vectorized with TF-IDF for model compatibility.
- **Sentiment Function:** Custom function provides real-time sentiment prediction.
- **Testing:** Sample reviews validate model's sentiment classification accuracy.

Evaluation Metrics:

- **Accuracy:** Measures overall correctness.
- **Precision, Recall, F1-Score:** For each sentiment class.
- **Confusion Matrix:** Visual insights on misclassifications.
- **ROC-AUC:** Measures class distinction capability.
- **Classification Report**

Sentiment analysis of reviews

```
# positive review
review = "This chips packet is very tasty. I highly recommend this!"
print(f"This is a {get_sentiment(review)} review!")
```

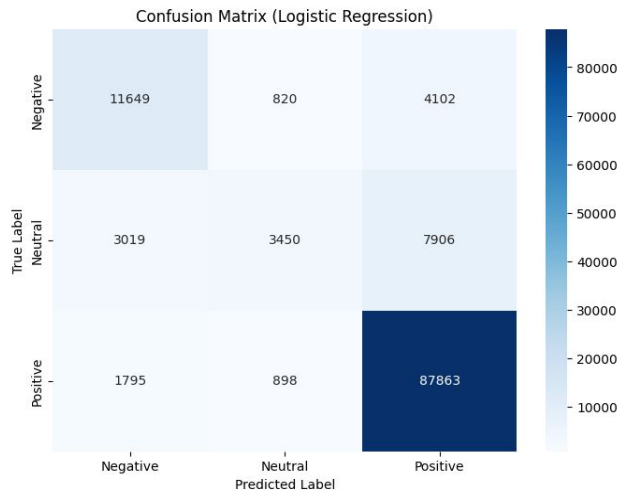
This is a Positive review!

```
# negative review
review = "This product is a waste of money. Don't buy this!!"
print(f"This is a {get_sentiment(review)} review!")
```

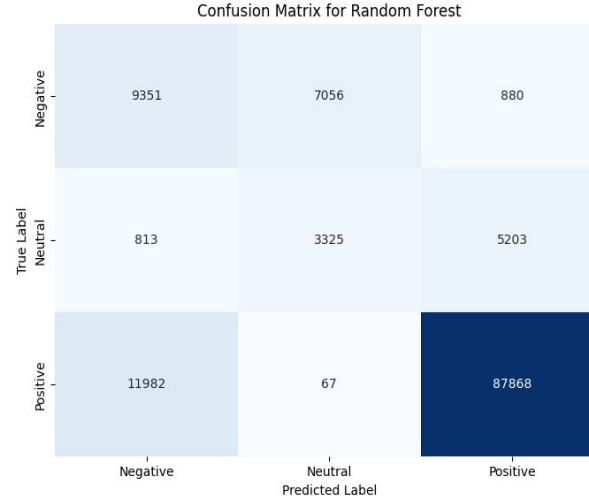
This is a Negative review!

Results and Model Comparison

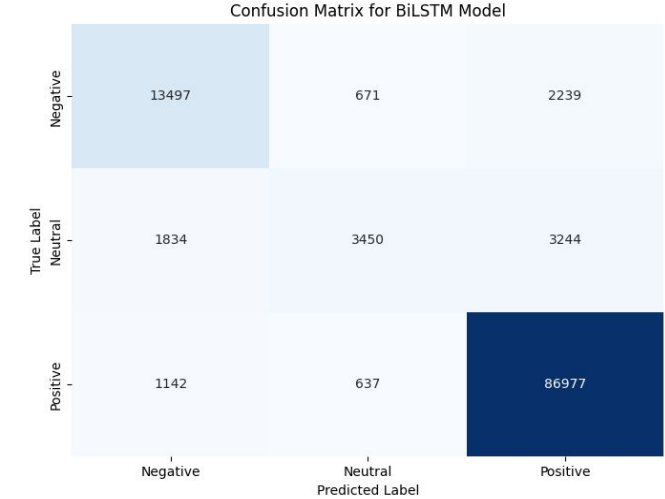
Logistic Regression (88.78% acc)



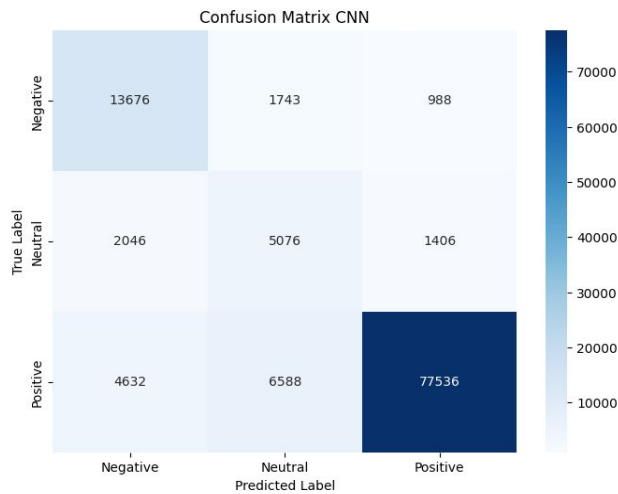
Random Forest (87.80% acc)



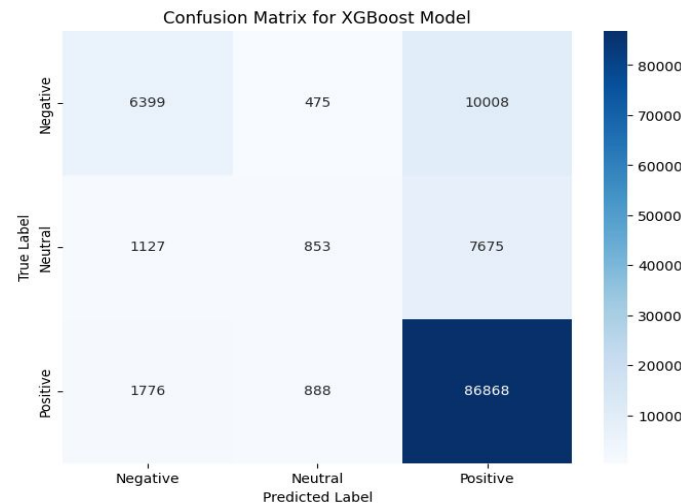
BiLSTM (91.41% acc)



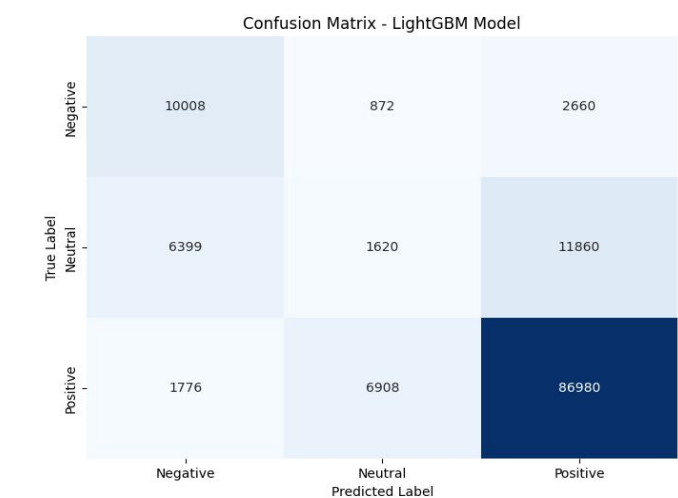
CNN Model (84.69% acc)



XGBoost (83.94% acc)



LightGBM (86.85% acc)



Conclusion and Future Scope

Conclusion:

- **Model Performance:** BiLSTM achieved the highest accuracy (91.41%), but Logistic Regression with TF-IDF was chosen for deployment due to its speed, efficiency, and lower computational cost, making it more suitable for real-time applications.
- **Deployment Suitability:** Logistic Regression's performance and lower resource requirements align well with the need for fast, scalable sentiment analysis in high-traffic environments like e-commerce.
- **Balance Between Accuracy and Efficiency:** The chosen model strikes an optimal balance between performance and practicality, ensuring scalability and responsiveness in real-time feedback systems.

Future Scope:

- **Model Interpretability:** Improve understanding of complex models (BiLSTM, CNN) through techniques like SHAP and LIME for better transparency.
- **Advanced Embeddings:** Implement state-of-the-art embeddings (GloVe, BERT) to capture richer semantic relationships and improve sentiment classification accuracy.
- **Address Class Imbalance:** Enhance classification of underrepresented classes, particularly Neutral sentiment.
- **Real-time Deployment:** Focus on deploying models in customer feedback systems with minimal latency for quick decision-making and insights.

[Github Link to the python notebook](#)

