


## Task 1: Sentiment Analysis on Amazon Reviews

 **Objective:** Develop a sentiment analysis model to classify Amazon reviews into positive or negative categories based on the review text.

### ★ Data Preprocessing

#### 1. Initial Data Inspection:

##### ○ Column Renaming:

- Renamed columns in the training and test datasets to "polarity", "title", and "text" for consistency.

##### ● Handling Missing Values:

- Filled missing values in the "title" column with the string "no title" to handle NA values.

#### 2. Sampling for Class Balance:

- Created a function `sample_equal_polarity` to sample a fixed number of samples (`num_samples_per_polarity = 50000`) for each polarity class from the dataset.

#### 3. Text Data Cleaning:

##### ○ Implemented `clean_data` function to preprocess text data:

- Removed special characters and extra spaces.
- Replaced escaped quotes and new lines with spaces.
- Stripped leading and trailing spaces.

#### 4. Text Preprocessing:

##### ● Tokenization and Normalization:

##### ○ Defined `preprocess_text` function for further text preprocessing:

- Tokenized text and removed stopwords.
- Applied either stemming or lemmatization based on the specified method ('`lemmatization`' for this task).

##### ● Preprocessing Combined Text:

- Created `preprocess_combined` function to combine and preprocess "title" and "text" columns.
- Added a "`processed_text`" column in both training and test datasets after applying the preprocessing function.

## 5.Vectorization:

- **TF-IDF Vectorization:**

- Used `TfidfVectorizer` to convert the "processed\_text" column into numerical features.

How TF-IDF works?

- **TF-IDF Vectorization** (Term Frequency-Inverse Document Frequency) is a numerical representation of text that reflects the importance of a word in a document relative to a collection of documents (or a corpus). It is commonly used in text mining and information retrieval to convert text data into features for machine learning models

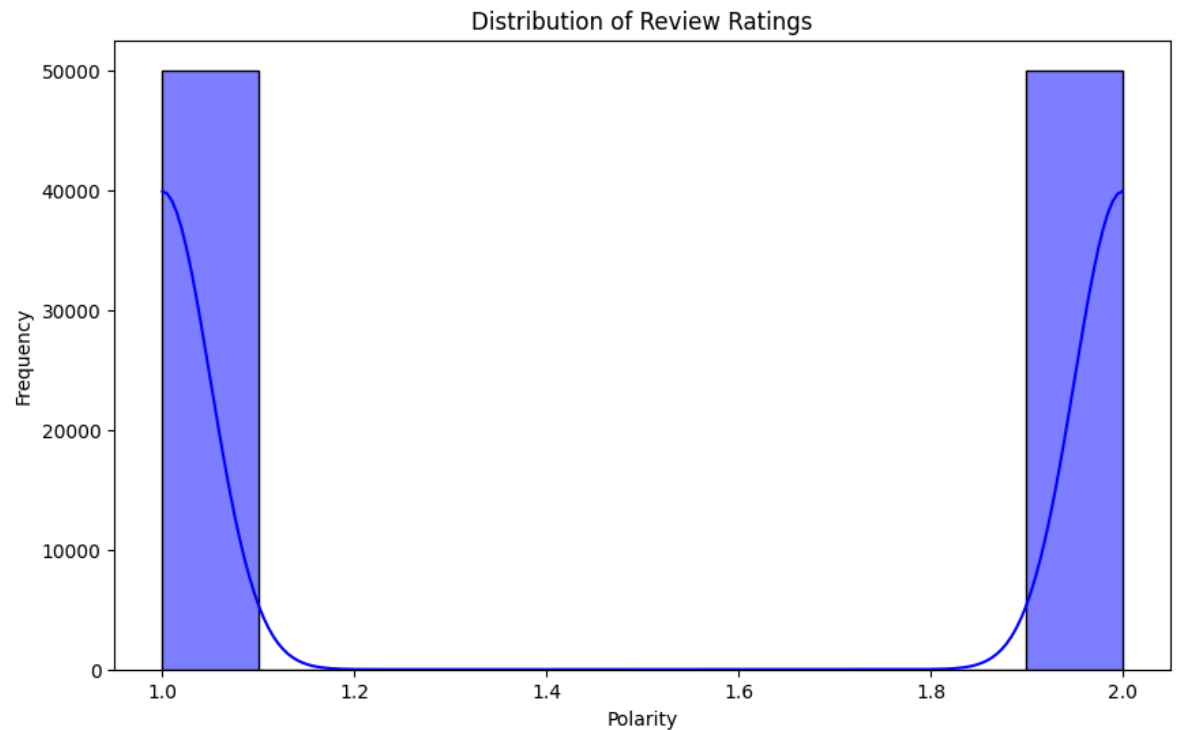
## 6.Preparing sentiment as target column for binary classification:

- The polarity values were converted into binary sentiment labels, with 1 mapped to 'negative' and 2 to 'positive', to prepare the sentiment column as the target variable for binary classification

## ★ Exploratory Data Analysis (EDA):

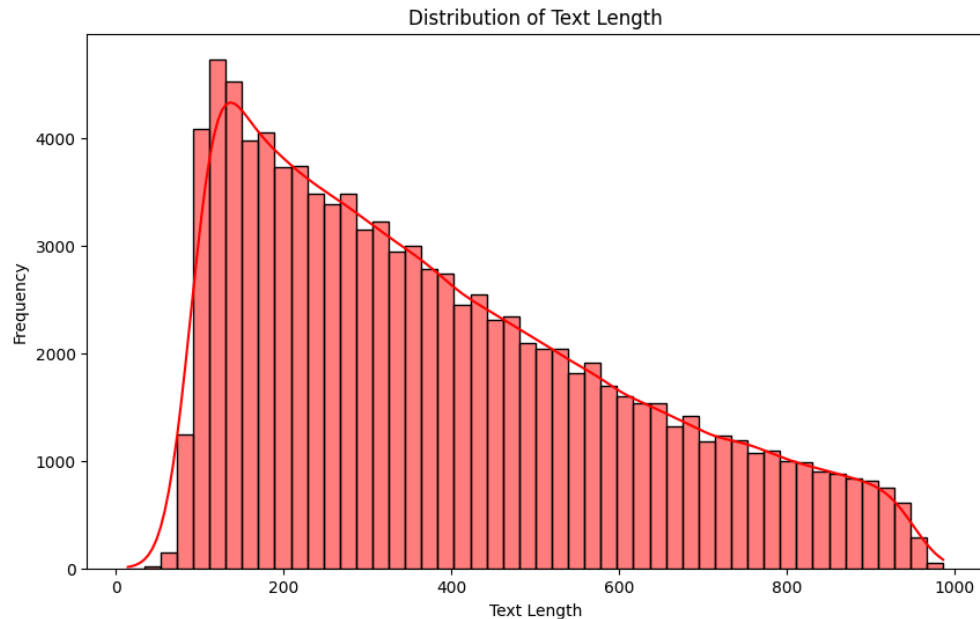
### 1. Distribution of Review Ratings:

- **Graph:** Histogram with KDE (Kernel Density Estimate) of `train['polarity']`.
  - **Purpose:** Visualize the distribution of polarity values in the training dataset.
  - **Insights:**
    - This graph shows the frequency of different polarity ratings.
    - Helps identify if the dataset is balanced with respect to polarity ratings or if there is a skew towards specific ratings.



## 2. Distribution of Text Length:

- **Graph:** Histogram with KDE of `train['text_length']`
  - **Purpose:** Examine the distribution of text lengths in the training dataset.
  - **Insights:**
    - This graph illustrates how text length varies across different reviews.
    - Helps understand if there are extreme values or if the text lengths are uniformly distributed.



## 3. Word Cloud of All Reviews:

- **Graph:** Word Cloud generated from `train['processed_text']`.
  - **Purpose:** Visualize the most frequent words in the processed text of the training dataset.
  - **Insights:**
    - This graph highlights the most common words and their relative frequencies.
    - Useful for understanding the prevalent terms and themes in the dataset, which can guide further feature engineering or model interpretation.



#### 4. Sentiment Distribution in Training Data:

- **Graph:** Count plot of `train['sentiment']`
- **Purpose:** Visualize the distribution of sentiment labels in the training dataset.
- **Insights:**
  - This graph shows the count of each sentiment class ('positive' and 'negative').
  - Helps assess the class balance and determine if there is a need for any balancing techniques during model training.



## ★ Model Training:

- Training Configuration:

- Logistic Regression, Naive Bayes Models:

- Trained using scikit-learn, with standard settings and default training parameters.

- Bi-LSTM Model:

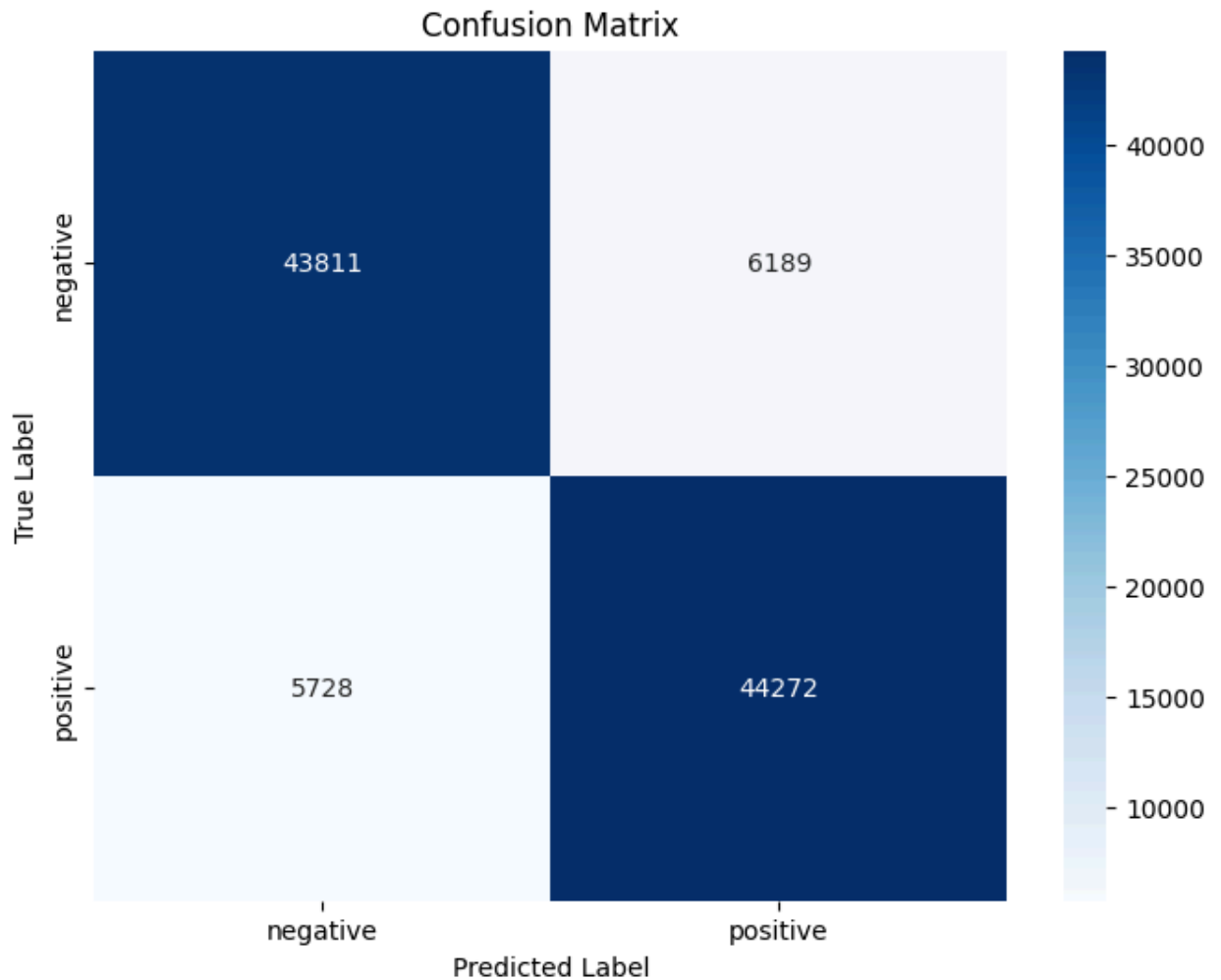
- This LSTM model for sentiment analysis uses an **embedding layer** to convert **words into dense vectors**, followed by a **bidirectional LSTM** layer to capture context from both past and future in the text.
    - Dropout layers are included to **prevent overfitting** by randomly deactivating neurons during training.
    - A second LSTM layer further processes the sequence, and a final dense layer with a **sigmoid activation** function outputs a probability for **binary classification**, indicating positive or negative sentiment.
    - Trained for 10 epochs with a batch size of **64** and **20%** of the training data used for validation.
    - Adam optimizer was used with a learning rate of **1e-4**.
    - The **binary\_crossentropy** loss function is used when you are dealing with a **binary classification problem**. In such tasks, the goal is to classify inputs into one of two classes, often labeled as 0 or 1.

- BERT Model:

- Fine-tuned for 5 epochs using the Hugging Face **Trainer**
    - Configured with warmup steps and weight decay to improve convergence.

## Confusion Matrix:

### Logistic Regression:

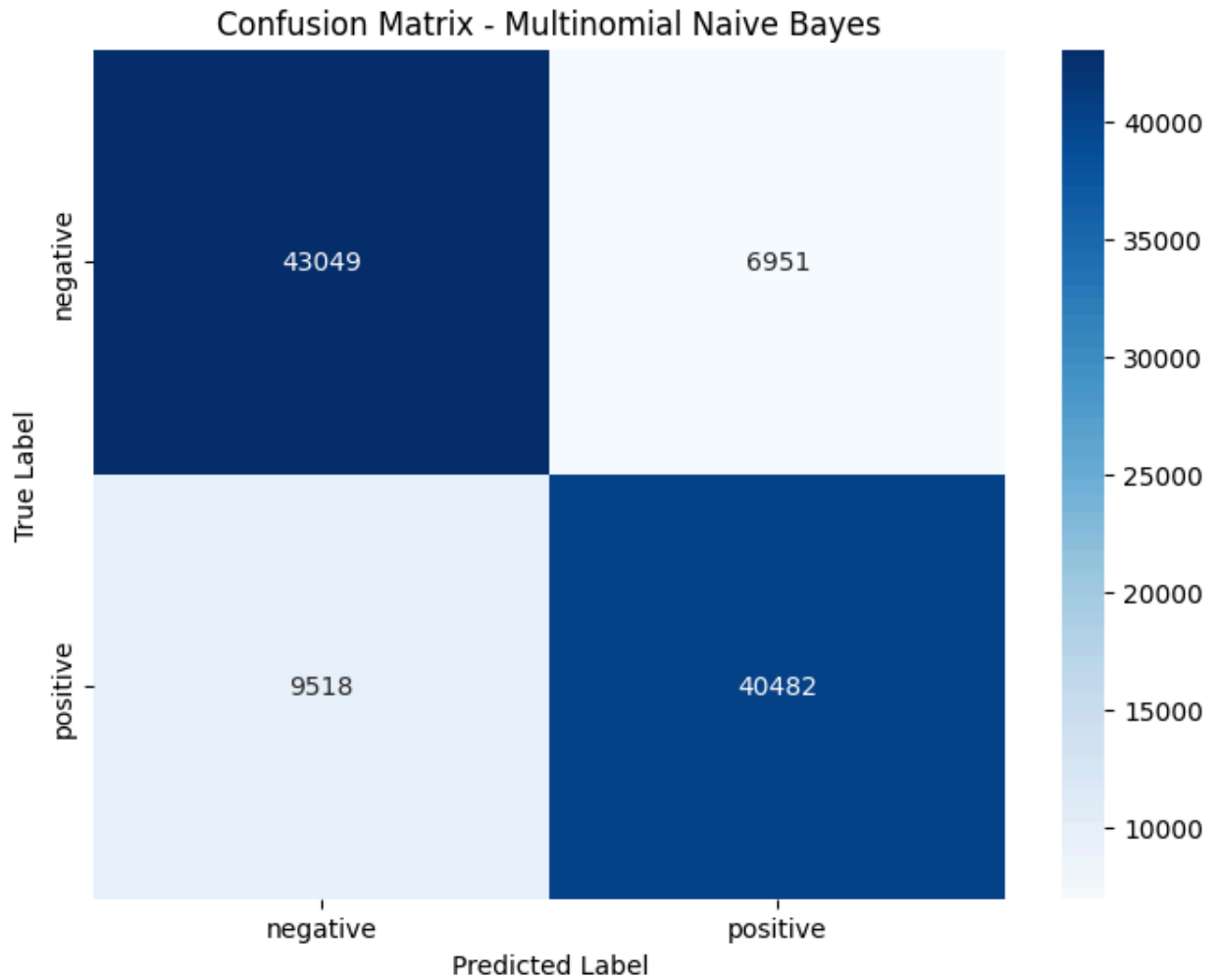


The model shows good overall accuracy but exhibits some **bias towards predicting positive cases**.

It has a higher false positive rate (6189) compared to false negatives (5728), indicating a tendency to overpredict the positive class.

The **variance appears relatively low**, as the model makes consistent predictions for both classes, with the majority of samples correctly classified.

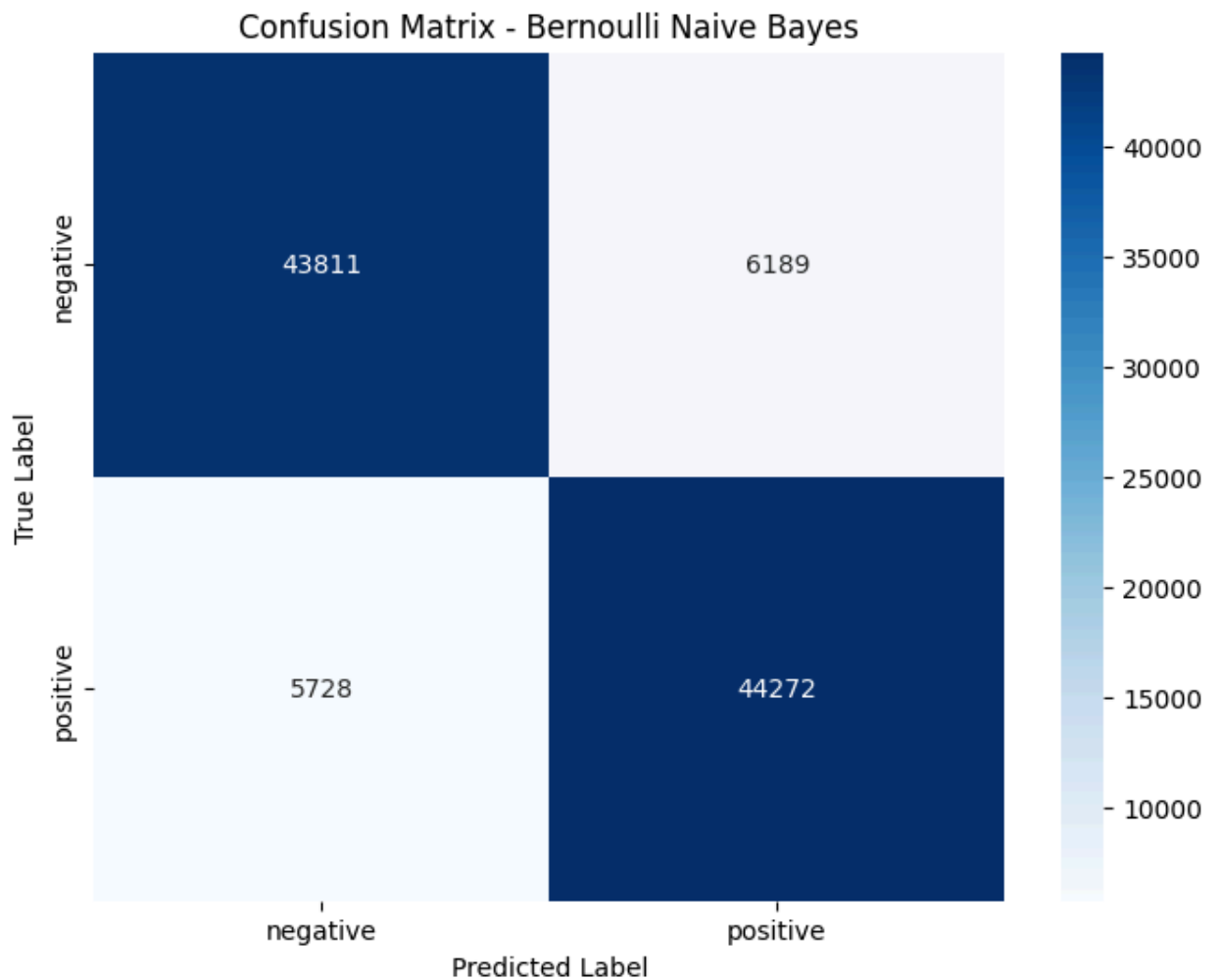
## Multinomial Naive Bayes:



The model is better at identifying true positive instances but has a **higher false negative rate**.



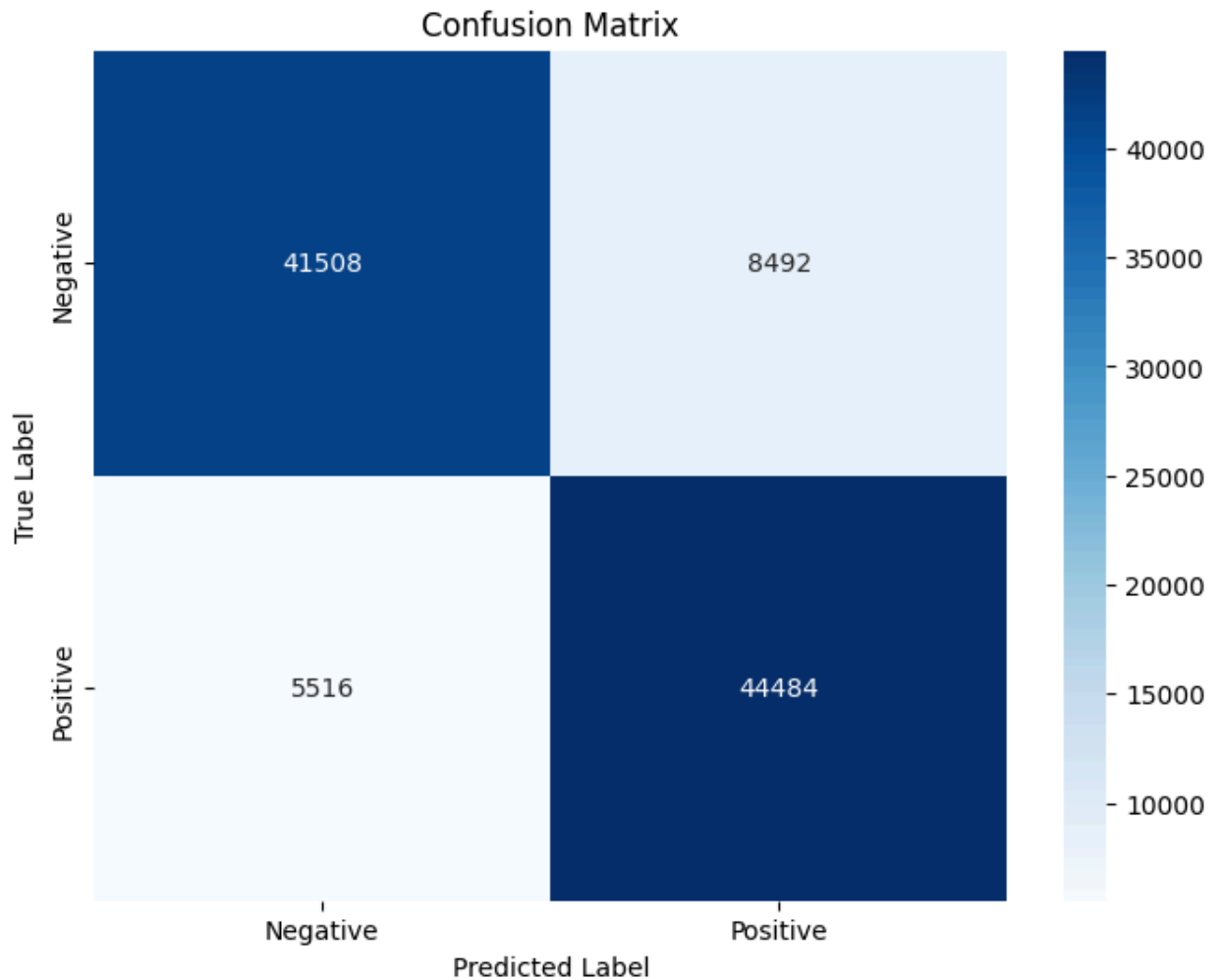
## Bernoulli Naive Bayes:



The model is performing **better** with the **prediction of positive** reviews compared to negative reviews.

However, the number of **misclassified negative reviews** is quite high.

## Bi-LSTM:

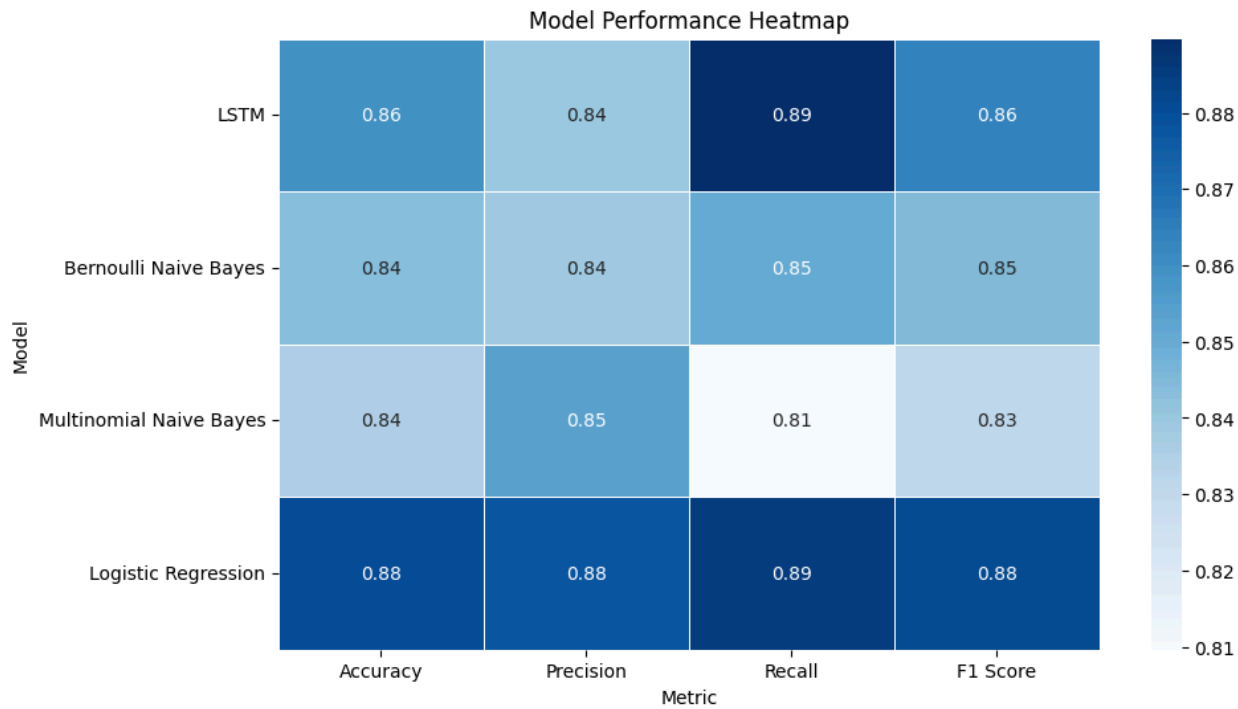


The confusion matrix shows that the Bi-LSTM model has a **high accuracy**, as it correctly classified a large number of samples.

However, it also has a **high number of false positives**, which means that it is **biased towards predicting positive cases**.

★ **Model Evaluation:**

- **Heatmap Analysis**



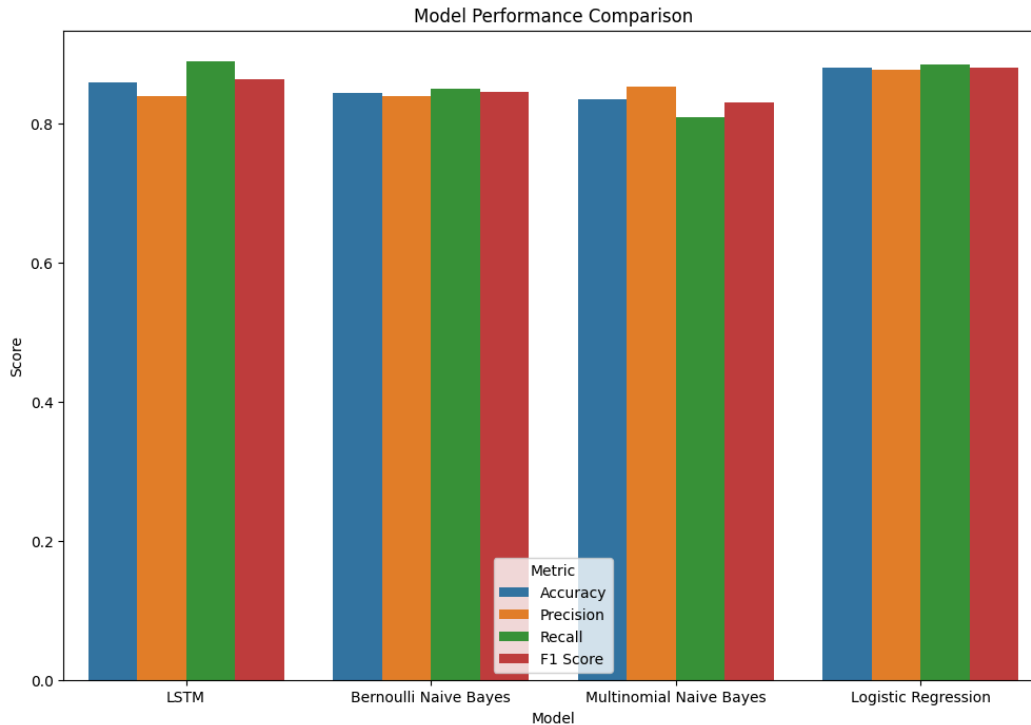
This heatmap shows the performance of four different models on four different metrics: accuracy, precision, recall, and F1 score.

Overall, **Logistic Regression** seems to be the best performing model, with consistently high scores across all metrics. It achieves the highest scores on accuracy (0.88), precision (0.88), recall (0.89), and F1 score (0.88).

**Bi-LSTM** also performs well, with particularly high scores on recall (0.89).

Bernoulli Naive Bayes and Multinomial Naive Bayes, on the other hand, achieve similar performance across all metrics but lag behind the other two models.

- **Model Performance Bar Analysis:**



- The bar chart likely compares accuracy, precision, recall, and F1-score across different models.
- **Accuracy** measures the overall correctness of the model's predictions.
- **Precision** measures how many of the positive predictions were actually correct.
- **Recall** measures how many actual positives were correctly predicted by the model.
- **F1-Score** is the harmonic mean of precision and recall, providing a single metric to evaluate the model's performance.

### ★ **Analysis:**

- Bi-LSTM has a high recall which implies that it is good at identifying positive cases even if it also predicts some negative cases as positive (false positives).

This means the model is able to capture most of the positive cases, which is valuable in situations where it's more important to minimize false negatives than false positives.

- The accuracy of the model can vary depending on the dataset used for training. Logistic regression has shown to be the best performing model on this particular dataset, but it is not guaranteed to perform the best on a different dataset. Bi-LSTM might perform better on a different dataset.
- As seen, Bernoulli Naive Bayes performs better than Multinomial Naive Bayes for binary classification tasks like this one .

### ★ **Recommendations:**

The BERT model can also be trained on a large sample like the other models and since it is a more complex model , it can provide better results.