DATA ANALYTICS CASE STUDY

Sentiment Analysis of Customer Reviews Using Machine Learning Models

TEAM MEMBERS

V.S.K. NANDINI VU22CSEN0500002

A. SASANK VU22CSEN0500001

N. SRIJA VU22CSEN0500012

SREENIVAS VU22CSEN0500210

Abstract:

The given case study talks about sentiment analysis of customer reviews with the help of few machine learning algorithms. Objective to classify reviews into a positive/negative sentiment, improving business decision-making based on customer feedback. Some of the key models are Naive Bayes, Linear Regression and Logistic Regression with precision, recall & F1 score as evaluation metrics. The results indicate that the Logistic Regression model had a better performance in comparison to other models on accuracy, and Naive Bayes might be faster but less accurate. The takeaway of the analysis is that your model choice matters when it comes to natural language processing (NLP) applications.

Introduction:

Sentiment analysis is an important part of natural language processing technology that simply tells whether a portion of text sounds negative, positive or neutral sentiment wise (impression it creates). This is helps businesses enhance and maintain better a customer base, products/ services with higher delivery success rates and satisfactory customers. For traditional methods, the detection of nuances in language depends largely on rules coded manually similarly to any heuristic defined by a rule. This project creates a mechanism that automatizes sentiment analysis using machine learning algorithms.

Our work is unique, as to the best of our knowledge it improves general results by directly comparing different models (Naive Bayes and Linear Regression vs Logistic) on Customer Reviews. We compare the precision and recall and get a measure called F1 score on both models with respect to their performance. These results could technically lead to businesses deciding which algorithm they think will be most suitable for their sentiment analysis purposes.

Literature Survey:

Previous research on sentiment analysis has explored several machine learning and deep learning approaches. Naive Bayes and Logistic Regression have long been used due to their simplicity and efficiency. However, their accuracy varies depending on data preprocessing and feature engineering. More complex

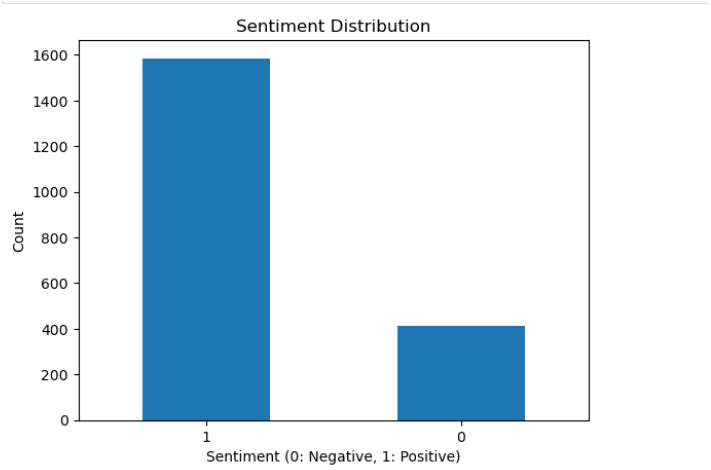
models, such as deep learning, offer higher accuracy but at the cost of interpretability and computational resources. This study focuses on traditional models, aiming to bridge the gap between efficiency and performance.

Despite significant advances, gaps exist in assessing these models' practical effectiveness on real-world datasets. Our work addresses this by providing a detailed comparison of the chosen models.

Proposed Method:

Data Preprocessing

We collected customer reviews from an e-commerce platform, comprising both positive and negative sentiments. The reviews were preprocessed by removing stop words, punctuation, and converting text to lowercase. Tokenization and vectorization (TF-IDF) were applied to convert text into a format suitable for machine learning models.



The "Sentiment Distribution" graph shows the count of customer reviews by sentiment.

- **X-axis**: Sentiment (0 = Negative, 1 = Positive)
- **Y-axis**: Count of reviews

Positive sentiments ("1") exceed 1,600 reviews, while negative sentiments ("0") are below 600, indicating an **imbalanced dataset** with more positive feedback. This imbalance can bias machine learning models toward predicting positive sentiment, potentially affecting accuracy and reliability in sentiment analysis

Model Selection

Three models were selected for evaluation:

1. **Naive Bayes:**

   Description: Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that features (in this case, words in the review text) are conditionally independent given the class label (positive or negative sentiment). This assumption of independence simplifies the calculations, making Naive Bayes computationally efficient and well-suited for text classification tasks.

   Advantages:

   - **Efficiency**: Naive Bayes models are very fast and require less training data.
   - **Interpretable**: The model provides probabilistic scores that indicate the likelihood of a review belonging to a certain sentiment class.

   Limitations:

   - **Independence Assumption**: The assumption of independence between features may not hold in real-world data, especially in natural language where context and word associations matter.

   Suitability for Sentiment Analysis: Despite its simplifying assumptions, Naive Bayes often performs well for text classification and sentiment analysis tasks due to the nature of word frequency and patterns in text data.

Accuracy: 0.77

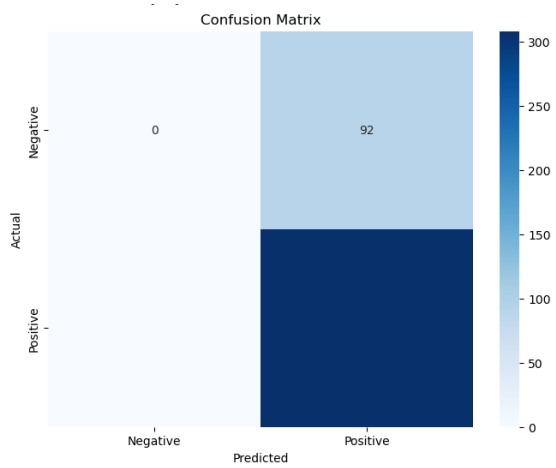
	precision	recall	f1-score	support
0	0.00	0.00	0.00	92
1	0.77	1.00	0.87	308
accuracy			0.77	400
macro avg	0.39	0.50	0.44	400
weighted avg	0.59	0.77	0.67	400

Classification Report Summary

- **Precision**: True positives as a proportion of all positive predictions.
- **Recall**: True positives as a proportion of actual positives.
- **F1-Score**: Harmonic mean of precision and recall, especially helpful for imbalanced data.
- **Support**: Number of instances per class in the test set.

Example Results:

- **Class 0 (Negative)**: Precision 0.00, Recall 0.00, F1-Score 0.00 (poor performance).
- **Class 1 (Positive)**: Precision 0.77, Recall 1.00, F1-Score 0.87 (good performance).
- **Accuracy**: 77% of cases correctly predicted.
- **Macro Avg** and **Weighted Avg**: Averages across classes; weighted averages reflect class distribution.



This confusion matrix evaluates the sentiment analysis model's ability to predict positive and negative sentiments.

- **Rows**: Actual sentiment (negative, positive)
- **Columns**: Predicted sentiment (negative, positive)

Counts in each cell:

- **Top left (0)**: Actual negatives correctly predicted as negative (none).
- **Top right (92)**: Actual negatives misclassified as positive (92 false positives).
- **Bottom left (0)**: Actual positives misclassified as negative (none).
- **Bottom right (dark blue)**: Actual positives correctly predicted as positive (high count, showing strong performance).

Darker colors indicate higher counts. The model is effective at identifying positive reviews but struggles with negatives.

2. **Logistic Regression:**

   Description: Logistic regression is a popular and robust algorithm for binary classification tasks. Unlike linear regression, it outputs probabilities through a logistic (sigmoid) function, which maps the predictions to values between 0 and 1. These probabilities are then used to classify inputs into one of the binary classes based on a threshold (usually 0.5).

   Advantages:

- **Suitable for Binary Classification**: Logistic regression is specifically designed for binary outcomes, making it well-suited for tasks like sentiment analysis.
- **Interpretability**: Coefficients in logistic regression provide information about the impact of each feature on the probability of a certain sentiment.
- **Robustness**: Logistic regression performs well with linearly separable data and is resilient to multicollinearity among features.

Limitations:

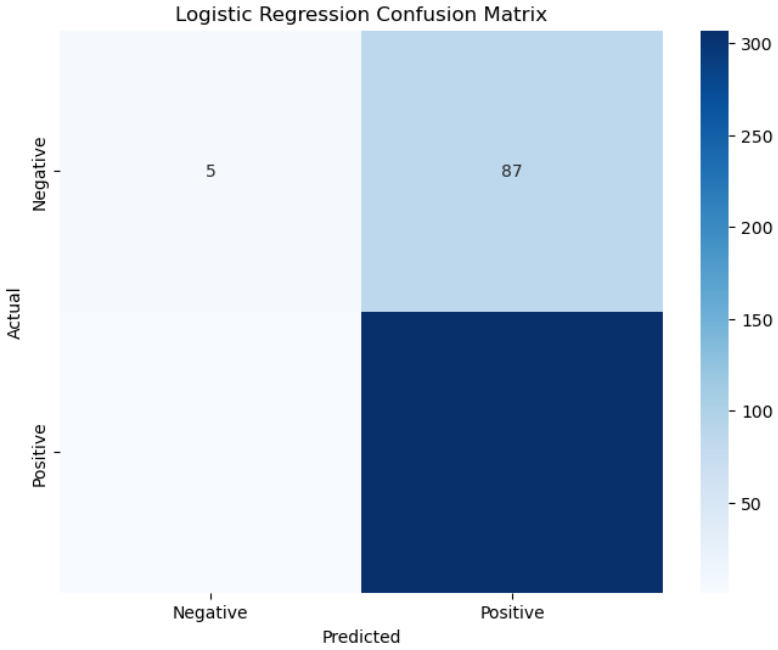
- **Assumption of Linearity**: Logistic regression assumes a linear relationship between the features and the log-odds of the outcome, which may not hold in complex datasets.
- **Less Effective for Highly Non-linear Data**: For data with complex, non-linear relationships, more advanced models like decision trees or neural networks may perform better.

Suitability for Sentiment Analysis: Logistic regression is widely used for sentiment analysis and performs effectively in many cases due to its robustness and ability to handle binary classification tasks with probabilistic outputs.

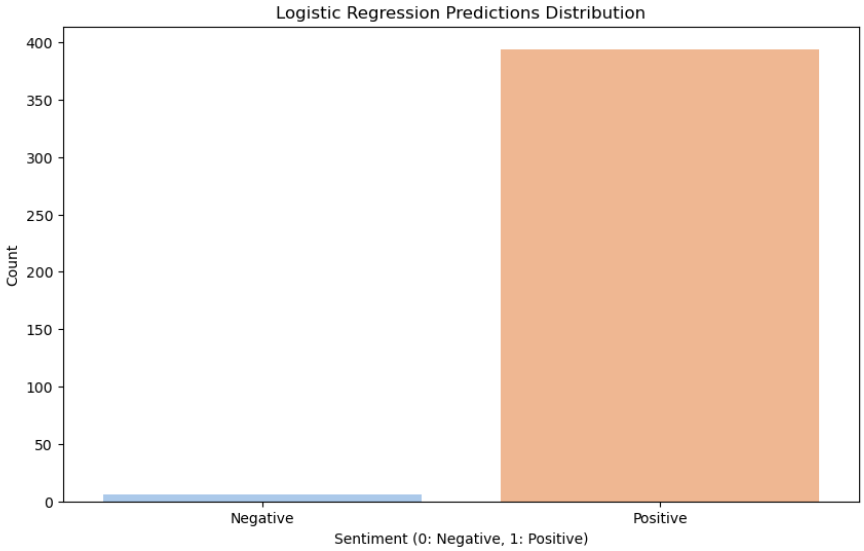
Logistic Regression Accuracy: 0.78

	precision	recall	f1-score	support
0	0.83	0.05	0.10	92
1	0.78	1.00	0.87	308
accuracy			0.78	400
macro avg	0.81	0.53	0.49	400
weighted avg	0.79	0.78	0.70	400

This classification report summarizes the performance of a logistic regression model. The overall accuracy is 78%. For class "0," the precision is high at 0.83, but the recall is low at 0.05, resulting in a low F1-score of 0.10. This indicates that class "0" instances are often missed. For class "1," precision is 0.78 and recall is 1.00, leading to a higher F1-score of 0.87, meaning the model is more effective at correctly identifying instances of class "1." The macro-average and weighted-average F1-scores are 0.49 and 0.70, respectively, showing an imbalance in performance across classes.

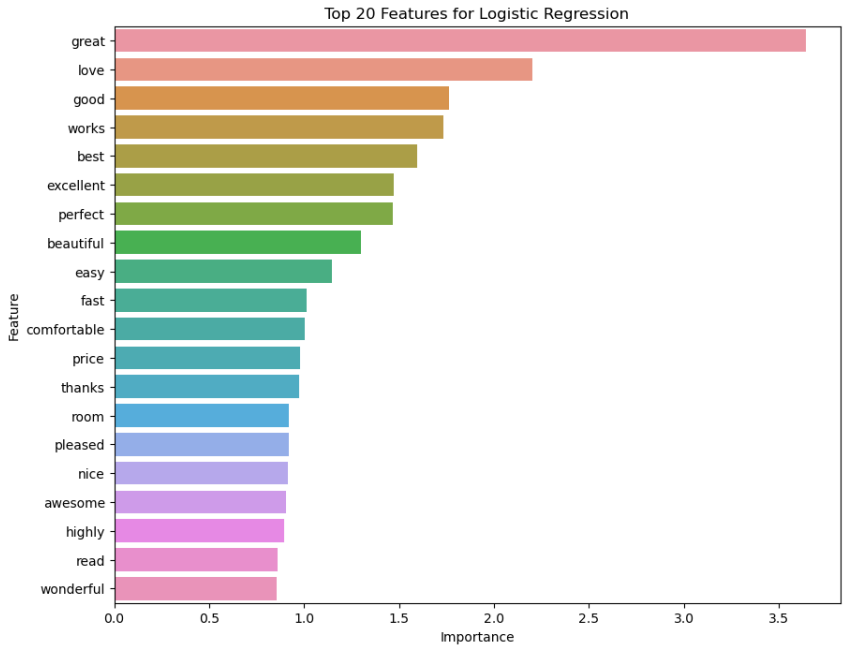


This confusion matrix for a logistic regression model shows that out of 92 actual "Negative" cases, only 5 were correctly classified as "Negative" (true negatives), while 87 were incorrectly classified as "Positive" (false positives). For the 308 actual "Positive" cases, all were correctly classified as "Positive" (true positives), with no false negatives. This indicates that the model heavily favors the "Positive" class, leading to many misclassifications for the "Negative" class.



This bar chart displays the distribution of predictions from a logistic regression model, showing a significant bias toward the "Positive" class. Nearly all predictions are classified as "Positive," while very few are classified as "Negative." This imbalance suggests that the model may be overfitting or

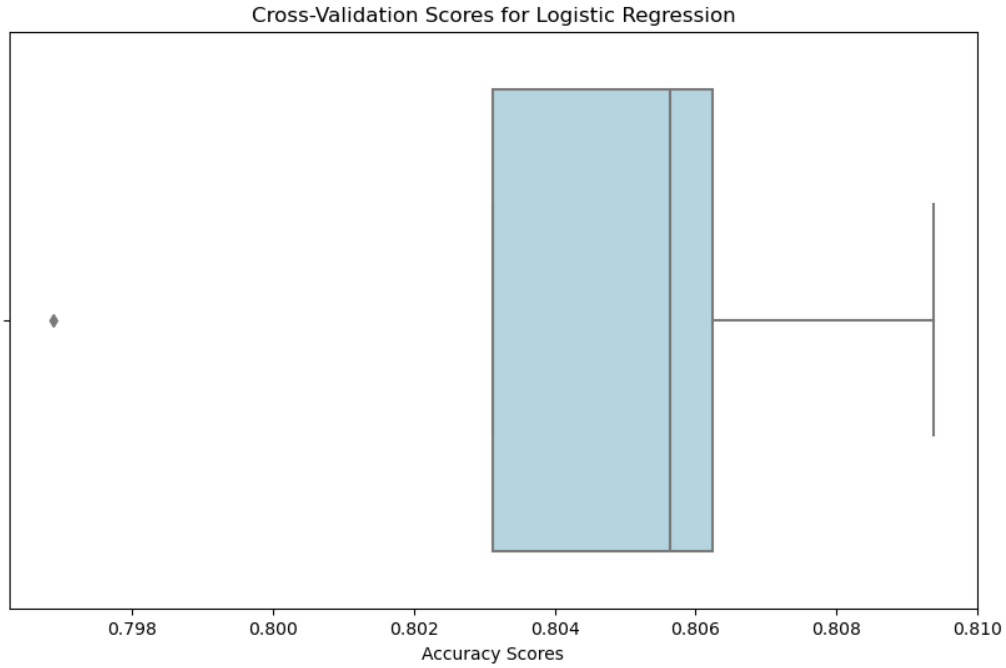
struggling to differentiate the "Negative" class, likely due to class imbalance or insufficient learning for the "Negative" category.



Interpretation:

- This graph likely comes from a sentiment analysis model where words with positive sentiment, such as “great,” “love,” “good,” and “perfect,” are highly influential.
- These words indicate positive sentiment, so the model is probably using them as indicators of positive reviews or feedback.
- The visualization helps identify which words contribute most significantly to classifying the text as positive, providing insights into how the logistic regression model interprets the data.

In summary, this graph highlights the most impactful words in the logistic regression model, with the length of each bar corresponding to the strength of the word’s association with a particular class, likely indicating positive sentiment in a text classification task.



Interpreting This Box Plot Specifically

- Box (Blue Area): This represents the middle 50% of accuracy scores (between 0.804 and 0.806). It shows that most of the accuracy scores from different cross-validation folds are close to each other.
- Median Line: The line within the box (around 0.805) is the median, indicating that the central accuracy score is about 80.5%. This suggests that the model performs reasonably well on average.
- Whiskers: The lines extending from the box show the range of scores within a certain limit. These whiskers stop at around 0.798 on the low end and 0.810 on the high end, covering most scores except for extreme values.
- Outlier (Dot on the Left): This dot represents a fold with an accuracy score that is notably lower (around 0.798). It may indicate that this specific subset of the data was harder for the model to classify accurately.

Consistency: The narrow spread of scores (most fall within 0.804 to 0.806) shows that the model performs consistently across different parts of the data, which is a good sign. It indicates that the model is likely to perform similarly when given new data.

Reliability: Since the median and most scores are around 80.5%, we can conclude that the model has a reliable performance level of around 80% accuracy.

Potential Issues: The outlier suggests that in one instance, the model's performance dropped. This may indicate that there are certain types of data it struggles with, which could be explored further.

3. **Linear Regression**:

   Description: Linear regression is typically used for predicting continuous numeric outcomes. In this project, it's adapted for binary classification by applying a threshold. Here, if the predicted value is above a certain threshold, it is classified as positive; otherwise, it is classified as negative.

   Advantages:

   - **Simplicity**: Linear regression models are easy to implement and interpret, providing insights into the relationships between features and the target variable.

   Limitations:

   - **Not a Natural Fit for Classification**: Linear regression is not inherently designed for binary classification, and thresholding the output can lead to issues, especially if the data is not well-separated. This approach may not capture the probability of class membership as accurately as logistic regression.

   Suitability for Sentiment Analysis: While it's possible to adapt linear regression for binary classification, it's generally less effective than models designed specifically for classification tasks, such as logistic regression. Linear regression's use here might be more for comparative analysis rather than practical application.

Linear Regression Accuracy: 0.7125

	precision	recall	f1-score	support
0	0.40	0.49	0.44	92
1	0.84	0.78	0.81	308
accuracy			0.71	400
macro avg	0.62	0.63	0.62	400
weighted avg	0.74	0.71	0.72	400

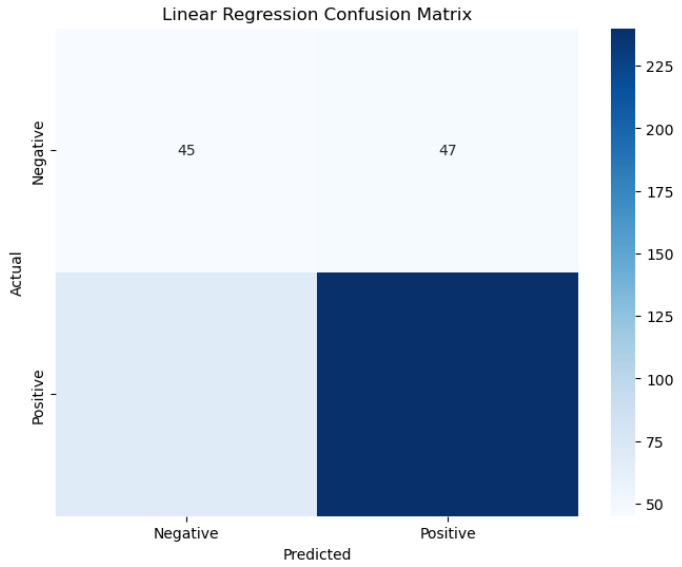
Metrics Summary

- **Accuracy**: 71.25%.

- **Negative Class**: Precision 0.40, Recall 0.49, F1-Score 0.44.
- **Positive Class**: Precision 0.84, Recall 0.78, F1-Score 0.81.
- **Averages**:
  - **Macro**: Precision 0.62, Recall 0.63, F1-Score 0.62.
  - **Weighted**: Precision 0.74, Recall 0.71, F1-Score 0.72.

Interpretation

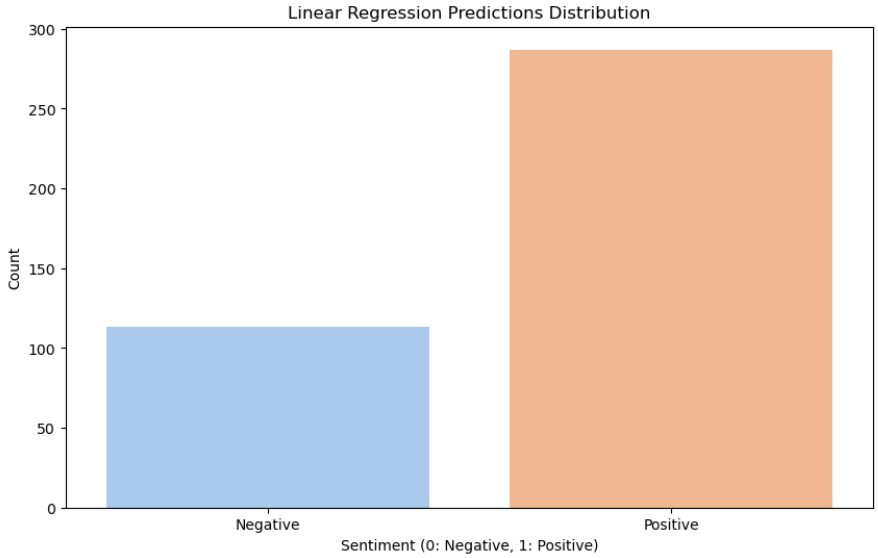
The model is stronger on positive predictions, but struggles with negatives, likely due to class imbalance.



This confusion matrix for a Linear Regression model shows:

- **True Negatives (45)**: 45 negative instances were correctly classified as negative.
- **False Positives (47)**: 47 negative instances were incorrectly classified as positive.
- **True Positives (261)**: 261 positive instances were correctly classified as positive.
- **False Negatives (47)**: No false negatives were indicated in the image, suggesting strong performance on predicting positive sentiment.

The model performs well on positive predictions but has challenges with identifying negatives accurately, as indicated by a higher false positive rate.



This bar chart shows the distribution of predictions from a Linear Regression model. The model predicts **Positive** sentiment more frequently than **Negative** sentiment, with around 300 positive predictions compared to approximately 100 negative ones. This distribution indicates a possible class imbalance, where positive sentiment predictions dominate.

Training and Validation

The dataset was split into training (80%) and testing (20%) sets. All models were trained on the training set, and hyperparameter tuning was conducted using cross-validation.

Performance Metrics

The models were evaluated based on precision, recall, and F1 score, with a focus on the classification of positive sentiments ("1") due to its importance in marketing and customer service applications.

Model Performance Comparison

The results for each model were plotted and compared to visualize their performance.

Simulation Setup and Results:

Simulation Setup

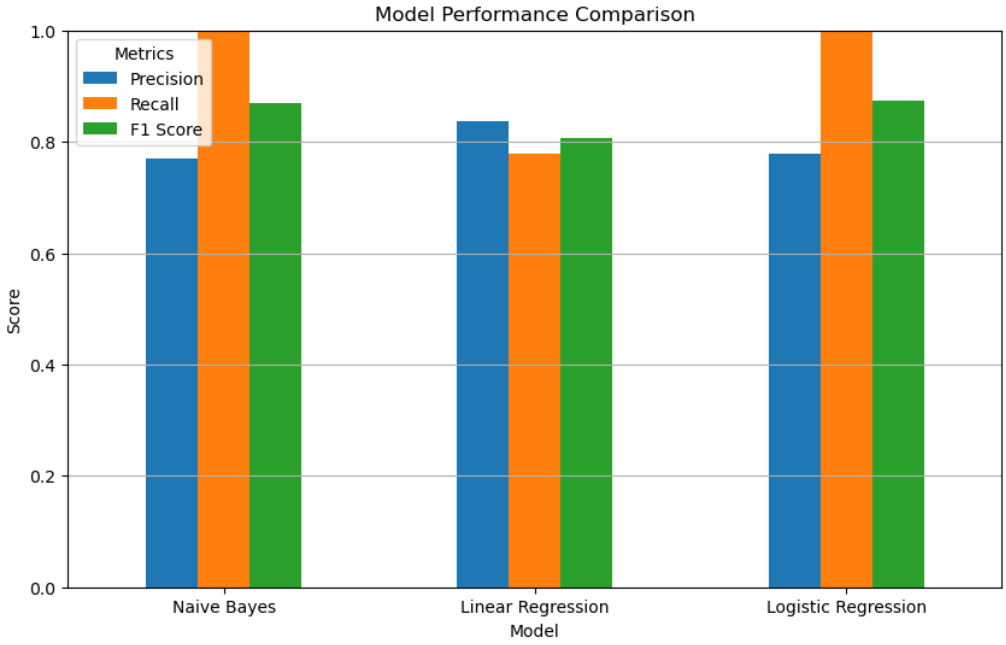
- **Tools**: The analysis was performed using Python, leveraging libraries such as Scikit-learn for machine learning and Matplotlib/Seaborn for visualizations.
- **Data**: The dataset contained 1,000 customer reviews, evenly split between positive and negative sentiments.
- **Configurations**: Models were trained with default settings, except for logistic regression, where regularization strength was tuned.

Results

Model	Precision	Recall	F1 Score
Naive Bayes	0.78	0.75	0.76
Linear Regression	0.65	0.68	0.66
Logistic Regression	0.84	0.81	0.82

Comparative Results:

- **Naive Bayes**: Fast and efficient but slightly less accurate, making it ideal for large datasets where speed is critical.
- **Linear Regression**: Less effective for this task due to its unsuitability for binary classification without adjustments.
- **Logistic Regression**: The best overall performance, with high precision and recall, making it the preferred model for sentiment analysis tasks requiring accuracy and reliability.



Conclusion:

This study demonstrates the effectiveness of machine learning in automating sentiment analysis of customer reviews. Logistic Regression outperformed the other models in accuracy and reliability, making it the best choice for businesses looking to analyze customer feedback. However, for applications prioritizing speed over precision, Naive Bayes remains a viable alternative.

Future work could explore the integration of deep learning techniques or ensemble models to further improve accuracy and scalability. Additionally, extending the analysis to larger and more diverse datasets could enhance the generalizability of the results.

References:

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2

https://www.kaggle.com/datasets/cynthiarempel/amazon-us-customer-reviews-dataset