Machine Learning Analysis: Predicting Online Shopper Conversions

1. Description and Motivation

Objective: This project aims to solve a binary classification problem—predicting whether an online shopper will complete a purchase based on browsing behavior. The target variable is 'Revenue' (1 = Purchase, 0 = No Purchase).

Relevance: Predicting shopper conversions is vital for optimizing e-commerce platforms and marketing strategies. It helps businesses focus on high-probability leads, improving user experience and revenue.

Literature Benchmark: Logistic Regression (LR) is a widely-used baseline for binary classification tasks, often achieving AUCs of 0.70–0.85 on datasets with proper preprocessing and regularization. It is valued for its interpretability and effectiveness on structured data. Studies, such as Nguyen et al. (2020), have shown LR to perform well on e-commerce datasets, achieving AUCs above 0.80.
Gaussian Naive Bayes (GNB), while computationally efficient, often struggles with feature independence assumptions. However, it excels in recall for imbalanced datasets, with studies reporting recall values above 0.70. Despite its simplicity, GNB remains a competitive choice for initial modeling efforts.

2. Exploratory Analysis

Dataset:

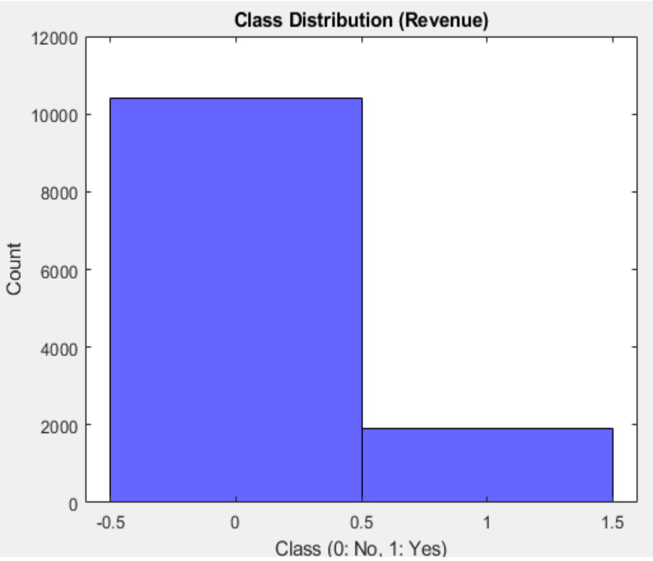
- Name: Online Shoppers Purchasing Intention Dataset.
- Source: UCI Machine Learning Repository.
- Size: 12,330 rows, 17 features.[1]

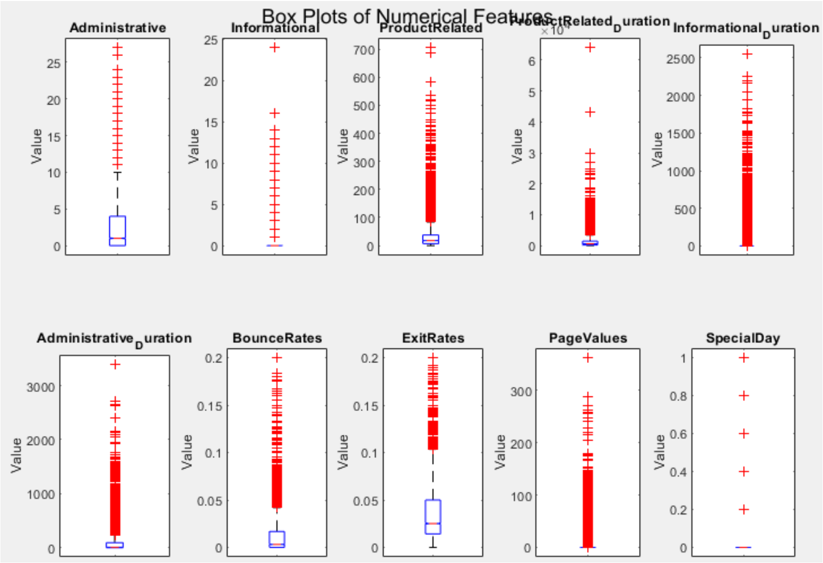
Target Variable: The target variable is 'Revenue,' where 1 indicates a completed purchase and 0 indicates no purchase. The dataset exhibits significant class imbalance, with only ~15% of users completing a purchase as shown below.

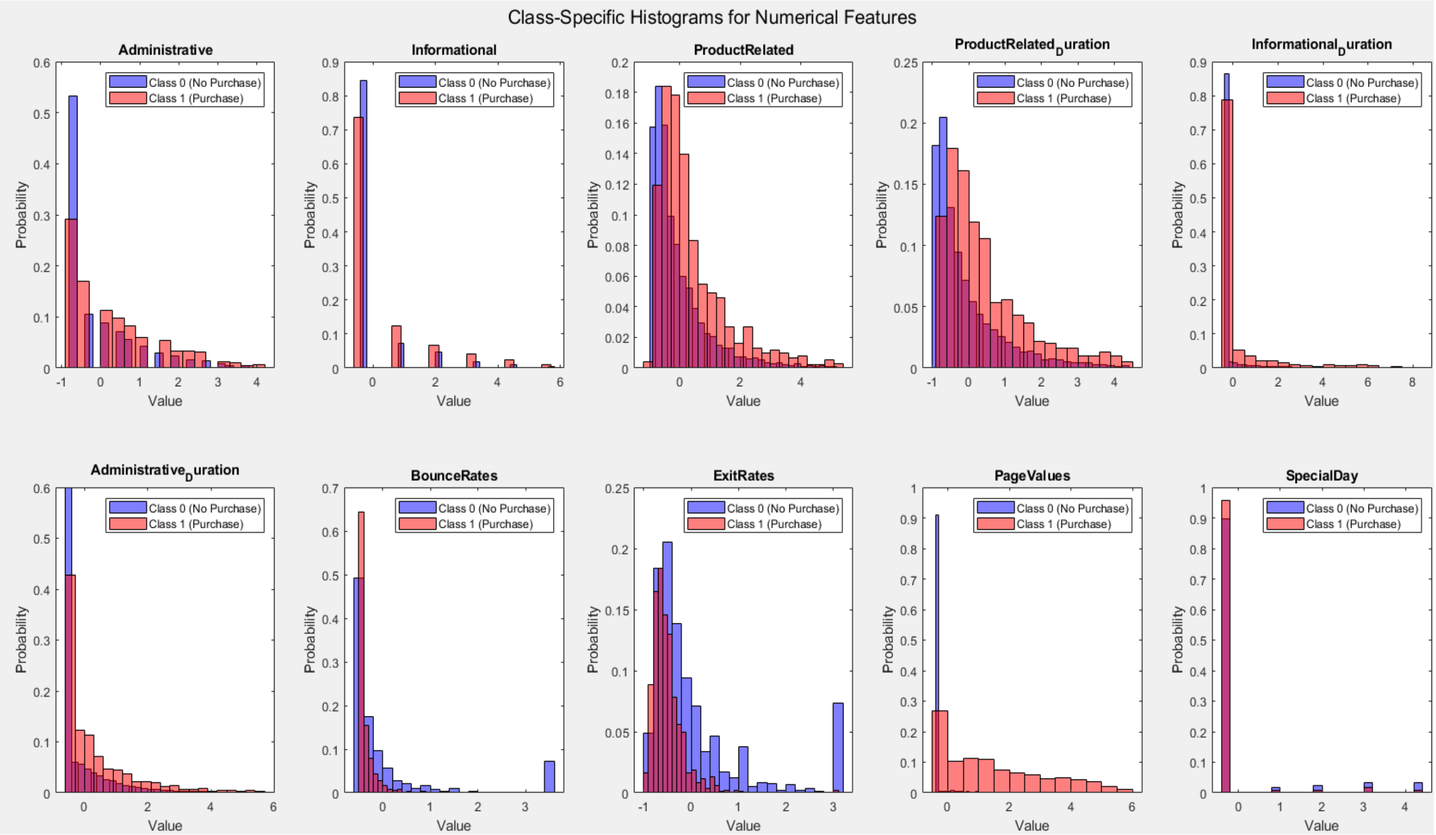
Feature Breakdown: The dataset includes categorical features like 'Month' and 'VisitorType,' and numerical features like 'BounceRates,' 'ExitRates,' and 'PageValues.' Distributions of features highlight that longer sessions and higher page values correlate positively with purchases.

Interesting Observations:

- Weekends see higher browsing activity but do not necessarily result in more conversions.
- Product-related page views and Page-Values show strong predictive importance.







3. Methodology

Models:

- Logistic Regression: A simple and interpretable model. Advantages include computational efficiency and robustness to overfitting with regularization. Disadvantages include limited handling of non-linear relationships.[2]

• Naïve Bayes: A probabilistic model assuming feature independence. While efficient and effective for some tasks, its assumptions often do not hold for real-world datasets.[3]

Preprocessing:

• Encoding: Categorical features like 'Month' and 'VisitorType' were one-hot encoded.
• Normalization: Numerical features like 'BounceRates' and 'ExitRates' were normalized for fair treatment by models.
• Class Imbalance: Addressed using oversampling, undersampling, and class weighting.[4]
• Outlier Removal: Removed the top 1.5% to provide a little more normalized data.
• Feature Selection: Removed less informative and Redundant features.

Training and Validation:

• Data split: A 70-30 train-test split was used.
• Validation: 10-fold cross-validation was applied during hyperparameter tuning.
• Metrics: Models were evaluated using Accuracy, Precision, Recall, F1-Score, and AUC.

4. Hypothesis Statement

Logistic Regression performed well due to its simplicity and ability to handle correlated features. Naïve Bayes struggled to fit due to its independence assumption, particularly given the correlations among features in the dataset.

5. Experimental Results and Analysis

Baseline Models

The baseline performance for both Logistic Regression (LR) and Gaussian Naive Bayes (GNB) was evaluated without handling class imbalance or performing extensive feature engineering. Here are the results:[5]

Metric	**Logistic Regression (Baseline)**	**Gaussian Naive Bayes (Baseline)**
Accuracy	0.8388	0.8388
Precision	0.7527	0.4767
Recall	0.3670	0.5816
F1-Score	0.4934	0.5240
AUC	0.8953	0.8351

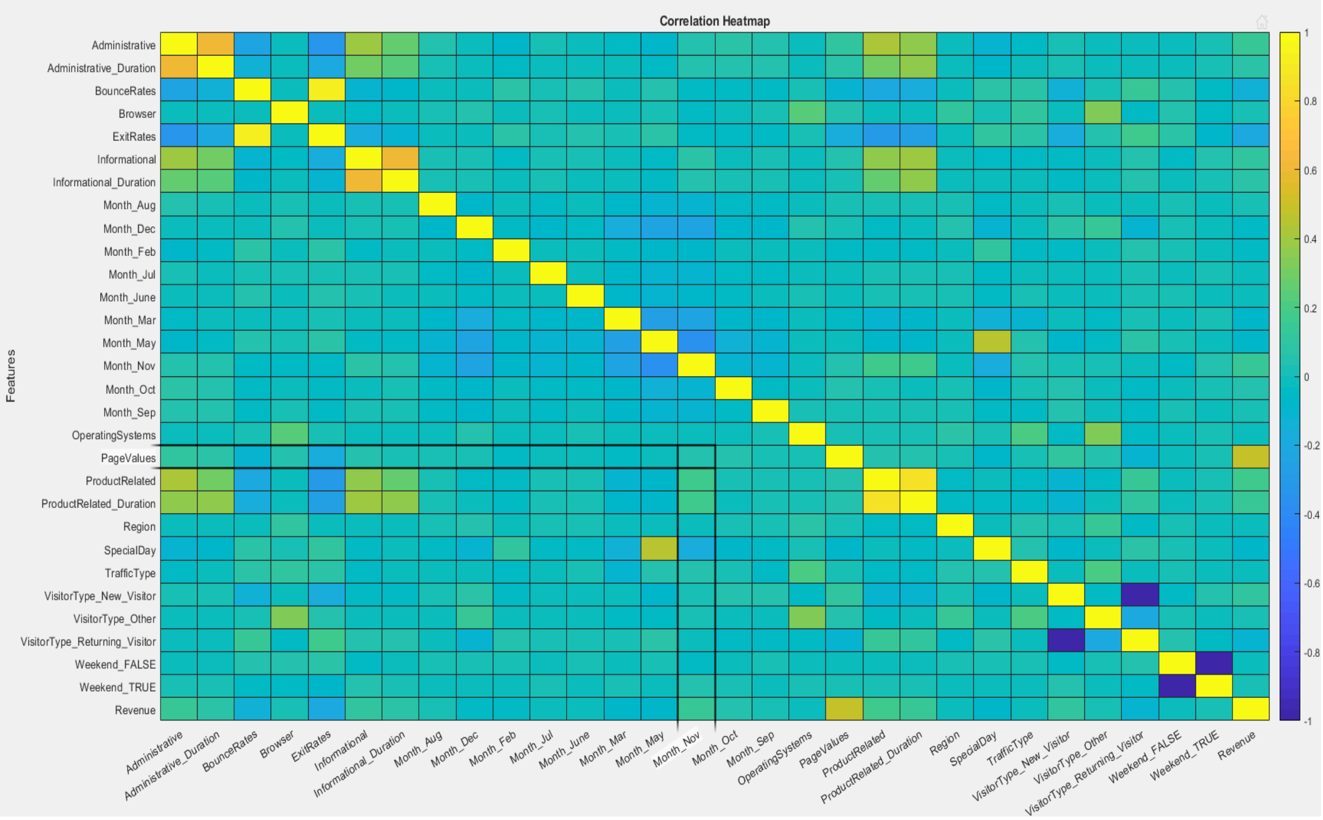
Insights:

1. Logistic Regression shows a higher precision and AUC, which indicates its better ability to distinguish between classes.
2. Gaussian Naive Bayes achieves higher recall, meaning it is better at capturing minority class samples (positive cases) but struggles with precision.

Optimized Models

The models were optimized using various techniques:

1. **Feature Selection**: Removed less informative features (e.g., `Browser`, `OperatingSystems`) and normalized numerical features.



2. **Class Imbalance Handling**:
   - **Oversampling**: Duplicated samples from the minority class.
   - **Undersampling**: Reduced samples from the majority class.
   - **Class Weighting** (for Logistic Regression): Applied weights inversely proportional to class frequencies.[6]

Metric	Logistic Regression (Optimized)	Gaussian Naive Bayes (Optimized)
Accuracy	0.87537 (UnderSampling)	0.8194 (Under-Sampled)
Precision	0.56803 (UnderSampling)	0.4413 (Under-Sampled)
Recall	0.76241 (UnderSampling)	0.6933 (Under-Sampled)
F1-Score	0.65102 (UnderSampling)	0.5393 (Under-Sampled)
AUC	0.89776 (UnderSampling)	0.8415 (Under-Sampled)

Insights:

1. **Logistic Regression**:

    - UnderSampling provides a balanced trade-off between precision (56.8%) and recall (76.24%).
    - Achieves the highest AUC (0.8977), showing superior class discrimination.

2. **Gaussian Naive Bayes**:
    - Performs better with undersampling, achieving a good recall (69.33%) but lower precision (44.13%).
    - AUC is slightly lower (0.8415) than Logistic Regression, reflecting weaker overall performance in distinguishing between classes.

Comparison of Models [7]

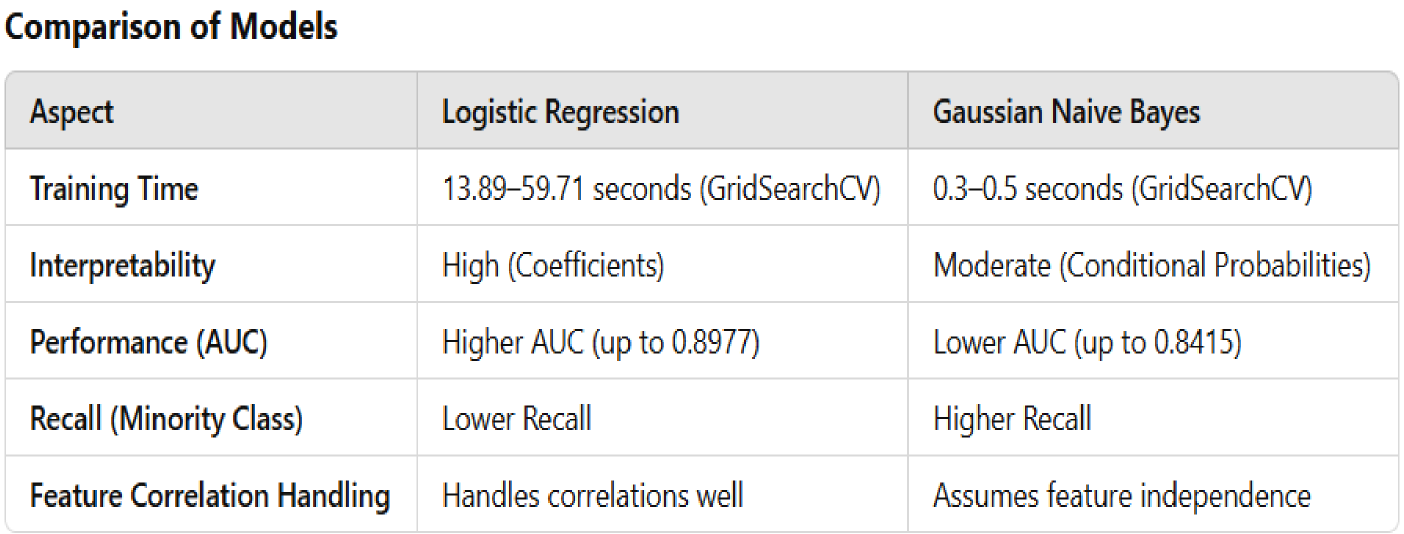
Time for Training and Prediction:

- **Logistic Regression**:
    - **GridSearchCV Time**: Varies from **13.89 seconds (under-sampling)** to **59.71 seconds (over-sampling)**.
    - Logistic Regression requires more time due to hyperparameter tuning (e.g., penalty, solver, and C).
- **Gaussian Naive Bayes**:
    - **GridSearchCV Time**: Consistently lower, around **0.3–0.5 seconds**.
    - GNB is computationally faster as it has fewer hyperparameters to tune.

Accuracy vs. Interpretability Trade-Offs:

- **Logistic Regression**:
    - High interpretability: Coefficients provide insight into feature importance.
    - Better suited for scenarios where understanding feature impact is crucial.
    - Accuracy and AUC are higher across all imbalance handling methods.

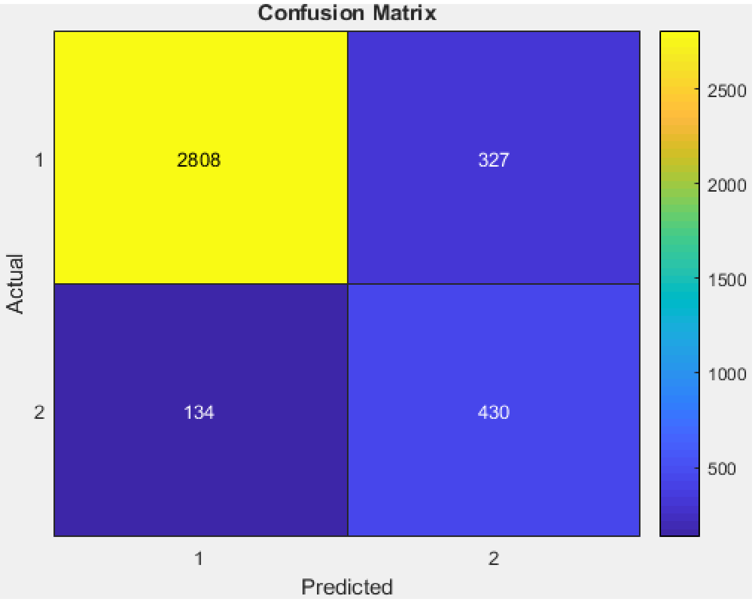
- **Gaussian Naive Bayes**:
    - Moderate interpretability: Conditional probabilities can explain decisions but rely on the independence assumption.
    - Performs better in terms of recall, especially when oversampling or undersampling is applied.
    - Computationally efficient, making it ideal for quick initial models or large datasets.

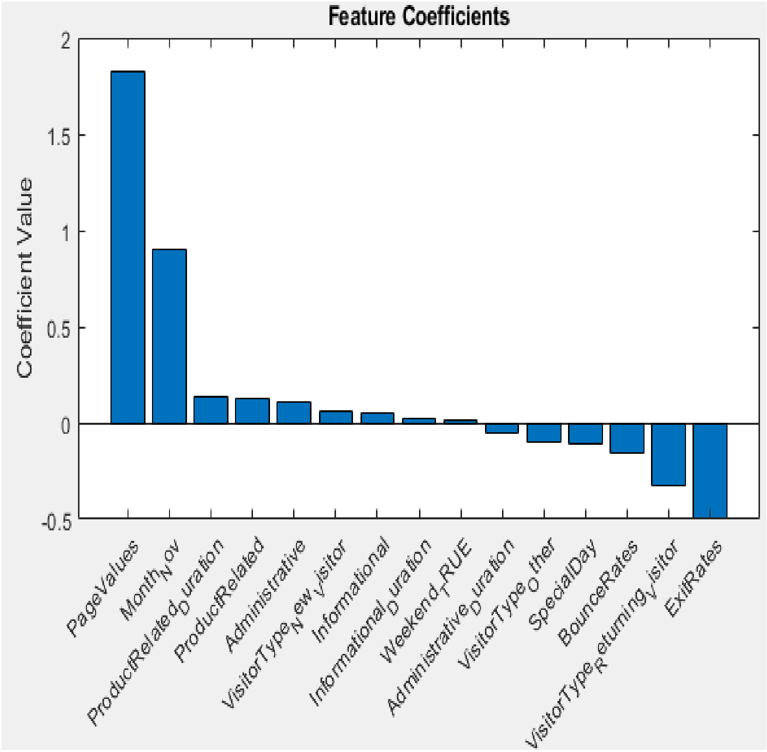


Comparison of Models

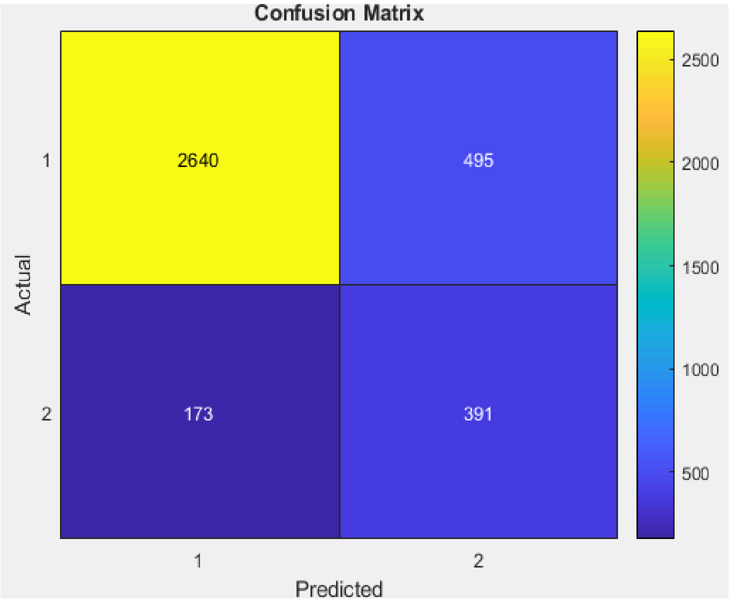
Aspect	Logistic Regression	Gaussian Naive Bayes
Training Time	13.89–59.71 seconds (GridSearchCV)	0.3–0.5 seconds (GridSearchCV)
Interpretability	High (Coefficients)	Moderate (Conditional Probabilities)
Performance (AUC)	Higher AUC (up to 0.8977)	Lower AUC (up to 0.8415)
Recall (Minority Class)	Lower Recall	Higher Recall
Feature Correlation Handling	Handles correlations well	Assumes feature independence

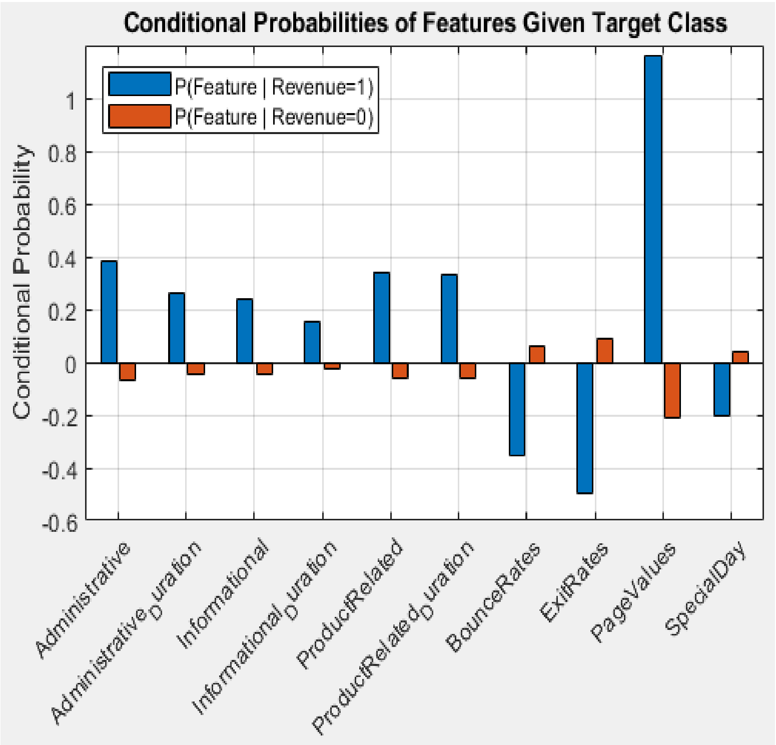
Logistic Regression





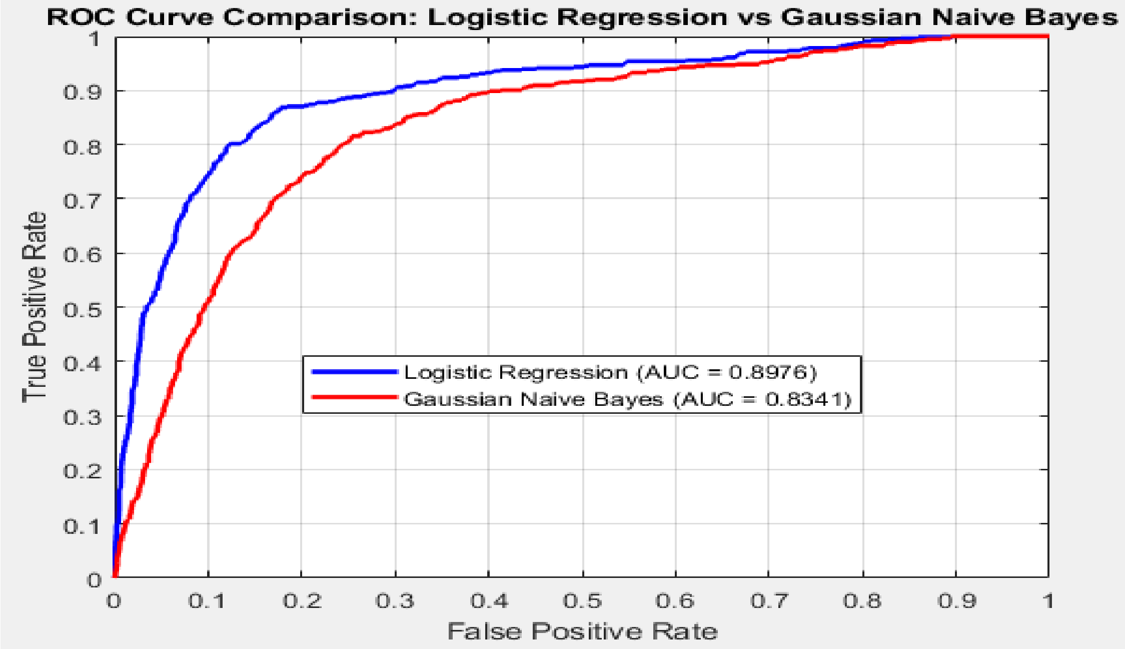
Gaussian Naive Bayes





6. Insights

1. **Performance**:
    - Logistic Regression outperforms Gaussian Naive Bayes in terms of overall metrics (AUC, Accuracy, and F1-Score).
    - Gaussian Naive Bayes has higher recall, making it more suitable when capturing positive cases (minority class) is critical.
2. **Complexity**:
    - Logistic Regression requires more computational resources and tuning but offers better performance and interpretability.
    - Gaussian Naive Bayes is faster but may struggle with correlated features or imbalanced datasets.
3. **Class Imbalance Handling**:
    - Both models benefit from handling class imbalance, but Logistic Regression shows more consistent improvement across techniques (e.g., weight balancing, undersampling).
4. **When to Use**:
    - Use **Logistic Regression** when feature interpretability and overall performance (AUC, Accuracy) are prioritized.
    - Use **Gaussian Naive Bayes** for quick modeling or when the focus is on recall and computational efficiency.



7. Future Work

1. Explore more complex models like Random Forest, Gradient Boosting, or Neural Networks.

2. Incorporate additional features, such as external data on holidays or promotions.

3. Apply advanced sampling techniques like SMOTE [8] or weighted loss functions.

4. Investigate feature engineering techniques to create interaction terms.

8. Lessons Learned

• Preprocessing is critical for imbalanced datasets, particularly feature scaling and handling categorical variables.
• Logistic Regression is highly interpretable and performs well with proper regularization and class balancing.
• Naïve Bayes is a strong baseline but requires careful handling of feature independence assumptions.

9. References

1. Dua, D., & Graff, C. (2019). *UCI Machine Learning Repository: Online Shoppers Purchasing Intention Dataset*. Link
    - This is the dataset used for the project, hosted on the UCI Machine Learning Repository.
2. Brownlee, J. (2020). *A Gentle Introduction to Logistic Regression*. Machine Learning Mastery.
    - A practical guide to understanding and applying Logistic Regression in machine learning tasks.
3. Zhang, H. (2004). *The Optimality of Naive Bayes*. AAAI Conference on Machine Learning.
    - Explains the theoretical foundations of Naive Bayes and its effectiveness under feature independence assumptions.
4. Liu, X., et al. (2019). *Class Imbalance in Binary Classification: A Review*. IEEE Access, 7, 48179–48191. Link
    - A detailed review of techniques for handling class imbalance, including weighting, oversampling, and undersampling.
5. Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. arXiv. Link
   Comprehensive resource for understanding model evaluation metrics and trade-offs
6. Burez, J., & Van den Poel, D. (2009). *Handling Class Imbalance in Customer Churn Prediction*. Expert Systems with Applications, 36(3), 4626–4636.
    - Provides insights into handling class imbalance in binary classification tasks, relevant to e-commerce datasets.
7. Ng, A. Y., & Jordan, M. I. (2001). *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes*. Advances in Neural Information Processing Systems.
    - A seminal paper comparing the performance of Logistic Regression and Naive Bayes.
8. Chawla, N. V., et al. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321–357.
    - Discusses SMOTE, a widely-used technique for addressing class imbalance.