Online Shoppers' Prediction

Kartik Kumar Singh and Aman Divya

February 9, 2025

Contents

1 Introduction

This report provides a detailed analysis and visualization of data for Assignment 1. The report is divided into two sections: Non-Competitive and Competitive portions to differentiate the aspects of the analysis. The problem statement focuses on predicting shopper conversion, which involves identifying whether an online shopper will complete a purchase based on their browsing behavior and other relevant features.

2 Non-Competitive Portion

2.1 Understanding the Dataset

The dataset was loaded using appropriate libraries and examined for missing values and data types. An overview of the dataset, including the number of observations, key variables, and their characteristics, was noted.

2.2 Feature Distribution Visualization

Histograms and box plots were used to examine feature distributions. Box plots identified outliers, revealing that some columns contained significant numbers of zeros, making non-zero values appear as outliers.

2.3 Data Preprocessing

Categorical variables were encoded using one-hot encoding, and missing values were appropriately handled. Standardization was performed using `StandardScaler`, and Z-score normalization was applied to manage data distributions and mitigate errors.

2.3.1 One-Hot Encoding

Converted categorical variables to binary vectors to handle non-numeric data effectively.

2.3.2 Normalization

Standardized features using Z-score normalization to ensure consistency in data scaling.

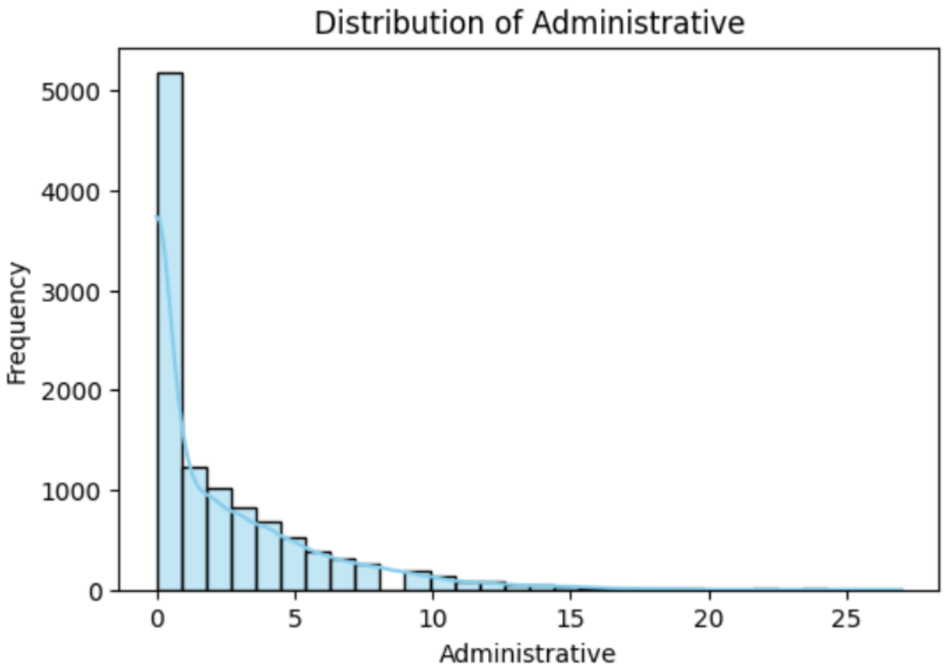

Figure 1: Feature Distribution Visualization 1

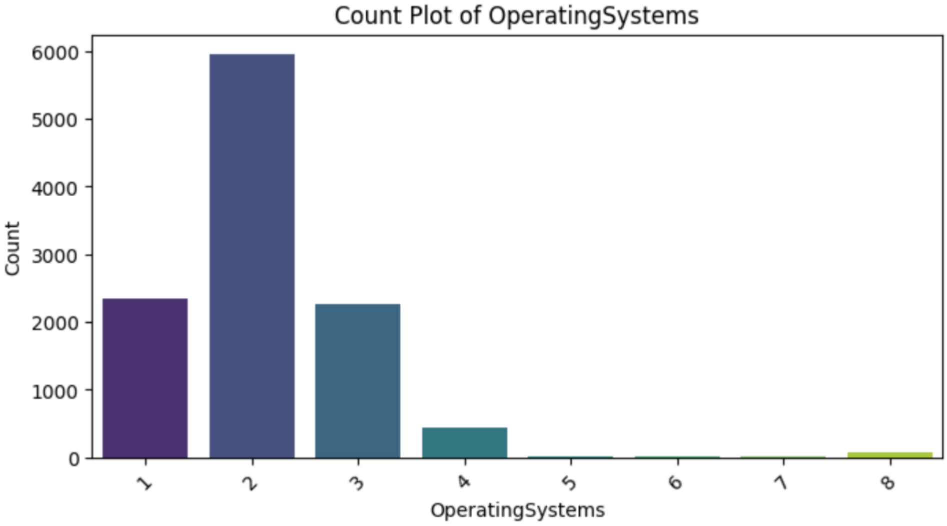

Figure 2: Feature Distribution Visualization 2

3 Model Training and Evaluation

Three models were trained to classify the target variable:

3.1 Logistic Regression

A linear model estimating probabilities using the logistic function.

- **max_iter**: Maximum iterations for solver convergence.
- **penalty**: Set to `None` for no regularization.

- **class_weight='balanced'**: Adjusts weights for class imbalance.

3.2 Support Vector Classifier (SVC)

Effective in high-dimensional spaces, SVM finds the optimal hyperplane for class separation.

- **class_weight='balanced'**: Addresses class imbalance.
- **Kernel Functions**: Handles non-linear data using kernels like `rbf`.

3.3 Decision Tree Classifier

Non-linear model splitting data based on feature values.

- **Non-linear Modeling**: Captures complex feature relationships.
- **Feature Importance**: Identifies key features influencing decisions.

3.4 Model Evaluation

Models were evaluated using:

- **Accuracy**: Correct predictions ratio.
- **Precision**: Correct positive predictions ratio.
- **Recall**: Correct positive observations out of all actual positives.
- **F1-score**: Harmonic mean of Precision and Recall, critical for imbalanced datasets.

3.5 Custom Logistic Regression Implementation

Explored using gradient descent with the following components:

- **Sigmoid Function**: Converts linear output to probabilities.
- **Cost Function**: Measures error using cross-entropy loss.
- **Gradient Descent**: Optimizes by iteratively updating weights.

3.6 Confusion Matrix Visualization

Confusion matrices were generated to visually assess model performance, highlighting true positives, false positives, true negatives, and false negatives.

3.7 Observations and Insights

- Missing values were identified and handled.
- Outliers were managed using binning techniques.
- Z-score normalization was applied to standardize data.
- Categorical variables were encoded using one-hot encoding.

4 Competitive Portion

4.1 Data Exploration

Initial dataset exploration identified patterns, value ranges, and key metrics. Basic plots visualized distributions and detected anomalies.

4.2 Key Metrics and Descriptive Statistics

Statistical summaries (mean, median, standard deviation) identified outliers and missing values.

4.3 Preprocessing Techniques

4.3.1 One-Hot Encoding

Converted categorical variables to binary vectors using `handle_unknown='ignore'` to prevent errors with unseen categories.

4.3.2 Standardization

Ensured features had a mean of zero and a standard deviation of one using `StandardScaler`, crucial for models like SVM.

4.3.3 Handling Class Imbalance

Used `class_weight='balanced'` in models to address class imbalance by assigning weights inversely proportional to class frequencies.

4.4 Feature Selection

To improve model performance, various feature selection techniques were applied:

4.4.1 Outlier Removal Using Binning

Features with extreme outliers were handled by categorizing values into bins, effectively reducing the impact of outliers on model performance.

4.4.2 Correlation Matrix

A correlation matrix was used to identify relationships between features. Highly correlated features were considered for removal to reduce multicollinearity and improve model efficiency.

5 Modeling Approaches

5.0.1 Model 1: Logistic Regression

Binary classification model with linear decision boundaries.

- `max_iter=10000`: Ensured convergence.

- `penalty=None`: No regularization.
- `class_weight='balanced'`: Addressed class imbalance.
- F1-score: Primary evaluation metric.

5.0.2 Model 2: Decision Tree

Captured non-linear patterns with key hyperparameters:

- Maximum Depth
- Minimum Samples Split
- `class_weight='balanced'`

5.0.3 Model 3: Support Vector Machine (SVM)

Maximized class separation margin.

- F1-score, Precision, Recall
- `class_weight='balanced'`

5.0.4 Model 4: Bagging Classifier

Reduced variance using multiple Decision Trees on different data subsets.

- `n_estimators=50`
- `class_weight='balanced'` in base estimators

5.0.5 Model 5: Random Forest Classifier

Enhanced generalization by introducing feature randomness.

- `n_estimators=100`
- `class_weight='balanced'`

5.0.6 Model 6: Gradient Boosting Classifier

Sequentially corrected model errors.

- `n_estimators=100`

6 Evaluation Metrics

- **Accuracy**: Measures overall correctness but can be misleading in imbalanced datasets.
- **Precision**: Proportion of true positives among all predicted positives.
- **Recall**: Proportion of true positives identified out of all actual positives.
- **F1-score**: Harmonic mean of precision and recall, crucial for imbalanced datasets.

7 Visualization Analysis

Visualizations provided insights into feature distributions and model performance. Key plots included confusion matrices and feature importance visualizations.

8 Conclusions

- EDA revealed key relationships among features, informing model selection.
- Data preprocessing, including handling missing values, encoding categorical features, and addressing class imbalance, significantly improved model performance.
- Logistic Regression provided baseline performance but struggled with non-linear relationships.
- Decision Tree captured complex patterns but required careful hyperparameter tuning to avoid overfitting.
- SVM demonstrated robust performance with appropriate kernel selection.
- Ensemble methods like Bagging and Random Forest enhanced stability and generalization.
- Gradient Boosting delivered strong performance by sequentially correcting errors.
- Visualizations highlighted feature importance and performance trends, guiding further analysis.

In conclusion, the combined insights from the non-competitive and competitive portions of the analysis provided a comprehensive understanding of the dataset and guided the development of accurate predictive models.