Predicting Online Shoppers' Purchasing Intention

Anthony Stefanuto
Computer Science
Western University

Luke Jang
Computer Science
Western University

Aimee Tang
Statistics
Western University

Nicol Elias
Statistics
Western University

Abstract*—Predicting purchasing intention in e-commerce is crucial for optimizing marketing strategies and improving conversion rates. This study presents a comprehensive machine learning approach to classify online shopping sessions as purchase or non-purchase. Using the UCI Online Shoppers Purchasing Intention Dataset, we approached the challenge of predicting user behavior from metrics including page views, session duration, bounce rates, and visitor characteristics. Our methodology involves extensive exploratory data analysis to identify key behavioral patterns, followed by systematic feature engineering and preprocessing to handle the imbalanced dataset. We implement and compare multiple classification algorithms, including Logistic Regression, Decision Trees, Random Forest, and XGBoost. Our results show that the Random Forest model achieves the highest overall performance across accuracy, F1-score, and ROC-AUC. Feature analysis reveals that engagement attributes such as *ProductRelated_Duration*, *BounceRates*, and *Total_Time* are the strongest predictors of purchase behavior. These findings demonstrate the value of behavioral metrics and machine learning in understanding online purchasing decisions, providing a practical basis for identifying high-intent shoppers and improving conversion strategies.

I. Introduction

The growth of e-commerce platforms depends heavily on how businesses interact with consumers. These interactions create copious amounts of behavioural and transactional data. Comprehension of this information creates opportunities to improve market strategies, increasing conversion rates. A predictive analysis can help businesses allocate time and resources more efficiently. Targeting customers with a high purchase intention increases profits while reducing wasted advertising spend.

The task of predicting customer purchase intention must be navigated cautiously. User behaviour on websites is highly variable and influenced by numerous factors, including browsing duration, the types of pages visited, and bounce rate. Conventional rules and analysis fail to capture advanced interactions and data correlations. This is where the implementation of machine learning (ML) models can identify patterns in data, facilitating data-driven business decisions.

This study uses the Online Shoppers Purchasing Intention Dataset from the UCI Machine Learning Repository to classify user sessions as either purchase or non-purchase. Using this dataset, we investigate how user engagement metrics correlate with conversion rates.This study makes three main contributions:

1) *We conduct a comprehensive exploratory data analysis (EDA) to understand trends, correlations, and target variable relationships.*
2) *We implement and compare three machine learning models, including Logistic Regression, Decision Tree, and Random Forest, to evaluate their predictive performance.*
3) *We comprehend the key features driving purchase decisions, providing valuable insights for e-commerce platforms.*

The remainder of this report is organized as follows. Section II discusses related research on purchase intention and online behavioural analytics. Section III details the dataset, preprocessing, and model design. Section IV presents and discusses experimental results, and Section V concludes with implications and potential directions for future work.

II. Related Work

Related Work Research on predicting online shoppers' purchasing intentions has become increasingly important as e-commerce platforms strive to identify users most likely to make purchases. One of the main challenges in this field is developing models that are both accurate and interpretable while handling the class imbalance existing between purchasing and non-purchasing visits to websites.

Baati and Mohsil (2020) proposed a real-time prediction system in their paper, which classifies a visitor's purchasing intent immediately after they visit a website. Using the Online Shoppers Purchasing Intention Dataset from the UCI Machine Learning Repository, they compared three classification models: Naïve Bayes, C4.5 Decision Tree, and Random Forest. To address class imbalance, they applied the Synthetic Minority Oversampling Technique (SMOTE), which generated synthetic samples of the minority (purchasing) class to ensure an even sample distribution. Their results showed that Random Forest achieved the best performance, with an accuracy of 86.78% and an F1-score of 0.60, outperforming the other two models and demonstrating that Random Forest is appropriate for real-time classification of shopper intent. However, Baati and Mohsil's model is limited in several ways as it focuses on real-time prediction and relies mainly on categorical data available at the start of a visit, such as browser type, region, and visitor status. This excludes important behavioural variables that develop as the user interacts with the site, such as time

spent on pages or the bounce rate. In addition, their study emphasized predictive accuracy but provided little analysis of which features most influence purchase decisions, which makes the results more challenging to interpret and less influential in decision-making.

In contrast, our study focuses on offline prediction and analysis of shopper behaviour using the same UCI dataset. We begin with a detailed Exploratory Data Analysis (EDA) to identify trends and relationships between engagement metrics and purchasing outcomes, which is not an element emphasized in Baati and Mohsil's paper. We then compare Logistic Regression, Decision Tree, and Random Forest models to evaluate both predictive accuracy and interpretability. Our goal is not only to predict purchases but also to understand which features drive purchase intention, so that we can provide more actionable insights for e-commerce platforms.

In summary, while Baati and Mohsil (2020) demonstrated the effectiveness of Random Forest in real-time classification, our work extends this by emphasizing data-driven interpretation and feature understanding, combining prediction accuracy with practical insights.

III. Methodology / Proposed Method

This section presents the complete modeling pipeline used in our project, including system architecture, data preprocessing, feature engineering, algorithm design, and model training. All hyperparameters used to train our models are explicitly listed to ensure full reproducibility.

A. *System Architecture*

Our system follows a supervised learning workflow consisting of four major components:

1) **Data Ingestion and EDA**: Load the Online Shoppers Intention dataset (12,330 samples, 39 features). Perform exploratory analysis to understand distributions, correlations, and class imbalance.
2) **Feature Engineering and Preprocessing**: Encode categorical variables, scale numerical features, and remove redundancy using variance thresholds and correlation filtering.
3) **Modeling Layer**: Train linear and nonlinear models, including Logistic Regression (baseline, L1, L2), Random Forest, XGBoost, and a Decision Tree baseline.
4) **Evaluation and Selection**: Evaluate models using Accuracy, Precision, Recall, F1-score, ROC–AUC, and PR–AUC, selecting the model that best balances performance and interpretability.

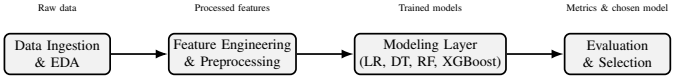

Fig. 1. Overall modeling pipeline for predicting online shoppers' purchasing intention.

B. *Data Preprocessing*

1) Exploratory Data Analysis: Initial EDA revealed substantial class imbalance, with only 15.5% of samples labeled as positive (`Revenue = 1`). Numerical features exhibited several high correlations (e.g., *ProductRelated* and *ProductRelated_Duration*, $\rho > 0.90$). Categorical variables such as *Month*, *VisitorType*, and *Region* required encoding.

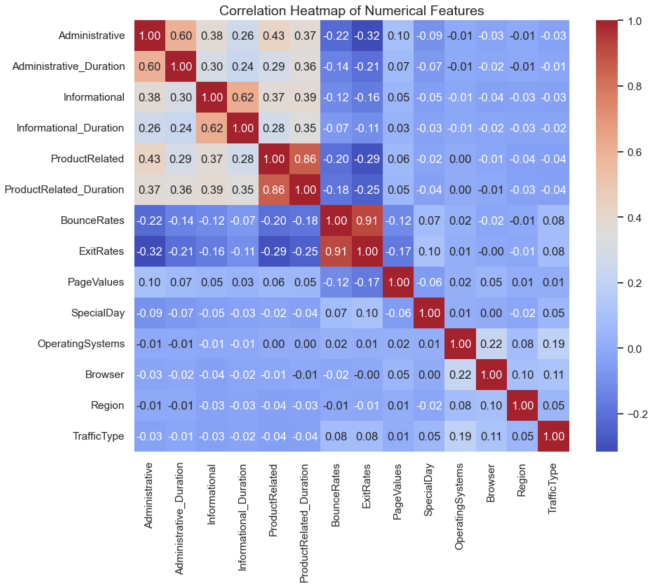

Fig. 2. Correlation heatmap of numerical features.

2) Handling Missing Data: The dataset contains no missing values; therefore, no imputation procedures were required.

3) Feature Generation: In addition to cleaning and encoding the raw attributes, we created two derived features to capture additional behavioural patterns not explicitly represented in the original dataset.

a) Total_Time: To aggregate a user's total engagement across different types of pages, we summed the duration-based variables:

$$\begin{aligned}\text{Total\_Time} = {} & \text{Administrative\_Duration} \\ & + \text{Informational\_Duration} \\ & + \text{ProductRelated\_Duration}\end{aligned} \tag{1}$$

This feature represents the overall time spent during a session and provides a single continuous measure of user engagement intensity.

b) Is_SpecialDay: The original `SpecialDay` variable encodes proximity to a holiday using a continuous value between 0 and 1. To simplify interpretability and model use, we converted it into a binary indicator:

$$\text{Is\_SpecialDay} = \begin{cases} 1, & \text{if SpecialDay} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

This transformation distinguishes sessions that occur near significant holiday periods, which often influence purchasing behaviour.

The inclusion of these two derived variables increased the dataset from 18 to 20 features and introduced higher-level representations that improved model interpretability while preserving the underlying behavioural information.

4) Feature Encoding: Categorical variables were transformed using one-hot encoding:

```
pd.get_dummies(df, columns=categorical_cols,
    drop_first=True)
```

Boolean variables (*Weekend*, *Revenue*) were encoded as 0/1.

5) Feature Scaling: Numerical variables were standardized using training-set statistics:

$$x_{\text{scaled}} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \tag{3}$$

```
scaler = StandardScaler()
X_train[num_cols] =
    scaler.fit_transform(X_train[num_cols])
X_test[num_cols]  =
    scaler.transform(X_test[num_cols])
```

Scaling ensures logistic regression behaves properly while maintaining consistent preprocessing across models.

6) Feature Selection: Two filter-based methods were applied:

a) Variance Threshold: A threshold of 0.0 was used:

```
VarianceThreshold(threshold=0.0)
```

And no features were removed.

b) Correlation-Based Filtering: A pair of highly correlated features ($|\rho| \geq 0.90$) was identified, and one feature from the pair was removed to avoid redundancy.

TABLE I
FEATURES REMOVED VIA CORRELATION FILTERING ($|r| \geq 0.90$)

Feature Removed	Correlated With
ExitRates	BounceRates ($r = 0.91$)

C. Algorithm Design

We evaluated linear and nonlinear classifiers to compare interpretability and predictive power.

1) Logistic Regression Models: Three variants were trained:

- **Baseline (No Penalty)**

```
LogisticRegression(penalty='none',
                   solver='lbfgs',
                   max_iter=2000)
```

- **Ridge (L2)**

```
LogisticRegression(penalty='l2',
                   C=0.01,
                   solver='lbfgs',
                   max_iter=2000)
```

- **LASSO (L1)**

```
LogisticRegression(penalty='l1',
                   C=0.1,
                   solver='liblinear',
                   max_iter=2000)
```

L1 regularization provided embedded feature selection, retaining 17 out of 38 features.

2) Decision Tree Classifier: A shallow decision tree (for interpretability) was tuned via GridSearchCV:

```
DecisionTreeClassifier(criterion='entropy',
                       max_depth=4,
                       min_samples_split=2)
```

3) Random Forest Classifier: A Random Forest was tuned using ROC–AUC scoring:

```
RandomForestClassifier(n_estimators=300,
                       max_depth=None,
                       min_samples_split=10,
                       class_weight='balanced',
                       random_state=42)
```

4) XGBoost Classifier: Boosted trees were tuned with:

```
XGBClassifier(n_estimators=100,
              max_depth=3,
              learning_rate=0.1,
              subsample=1.0,
              colsample_bytree=1.0,
              eval_metric='logloss',
              random_state=42)
```

D. Model Training

All models were trained using the same protocol:

- 70% training, 15% validation (via internal cross-validation), 15% testing.
- Training performed with:

```
model.fit(X_train, y_train)
```

- Predictions computed as:

```
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)[:, 1]
```

- Evaluation metrics included Accuracy, Precision, Recall, F1-score, ROC–AUC, and PR–AUC.

This methodology ensures reproducibility and allows for direct comparison of linear and ensemble-based approaches under consistent preprocessing and evaluation criteria.

REFERENCES

[1] Moro, S., Rita, P., and Cortez, P. (2016). "A Data-Driven Approach to Predict the Success of Bank Telemarketing." *Decision Support Systems*, 62, 22–31.

[2] UCI Machine Learning Repository (2018). "Online Shoppers Purchasing Intention Dataset."

[3] Pedregosa, F. et al. (2011). "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830.

[4] Powers, D. M. W. (2011). "Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation." *Journal of Machine Learning Technologies*, 2(1), 37–63.

[5] Baati, K., & Mohsil, M. (2020). Real-Time Prediction of Online Shoppers' Purchasing Intention Using Random Forest. Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part I, 583, 43–51. https://doi.org/10.1007/978-3-030-49161-1_4