# Predicting Purchasing Intention from Online Shopper Behavior: A Machine Learning Approach

Franck Sirguey – MFG Data Science Challenge

April 21, 2025

## Contents

# 1 Task and Dataset Description

The objective of this project is to build a predictive model capable of determining whether a visitor on an e-commerce website will make a purchase during their session. This problem is framed as a binary classification task.

The dataset used is the **Online Shoppers Purchasing Intention Dataset**, available on the UCI Machine Learning Repository. It contains records of 12,330 user sessions, each described by 18 features related to user behavior and technical characteristics. The target variable, `Revenue`, is a boolean indicating whether the session led to a purchase (`True`) or not (`False`).

## 1.1 Features Overview

The dataset includes the following types of features:

- **Session behavior:** such as `Administrative`, `Informational`, and `ProductRelated`, indicating the number of pages visited per type.
- **Duration variables:** e.g., `Administrative_Duration`, `Informational_Duration`, etc.
- **Engagement metrics:** including `BounceRates`, `ExitRates`, and `PageValues`.
- **Temporal and special occasion indicators:** such as `Month` and `SpecialDay`.
- **Technical information:** like `OperatingSystems`, `Browser`, `Region`, and `TrafficType`.
- **User information:** such as `VisitorType` (new, returning, or other) and `Weekend` (whether the session occurred on a weekend).

## 1.2 Imbalance Issue

The dataset is **significantly imbalanced**, with approximately **84.5%** of sessions labeled as non-conversions (`Revenue = False`) and only **15.5%** as conversions (`Revenue = True`). This imbalance poses a challenge for most machine learning algorithms, which tend to be biased toward the majority class. Several techniques were therefore considered to mitigate this issue during modeling, such as class weighting and SMOTE oversampling.

# 2 Exploratory Data Analysis

The purpose of this exploratory analysis is to better understand the structure and statistical properties of the dataset prior to modeling. This includes identifying potential imbalances, correlations, and behavioral patterns that could influence purchase decisions.

## 2.1 Target Variable Distribution

The target variable `Revenue` is highly imbalanced: approximately **84.5%** of the sessions did not result in a purchase, while only **15.5%** ended with a transaction. This imbalance was considered throughout the modeling phase and addressed through techniques such as class weighting, oversampling (SMOTE), and adjusted scoring metrics.

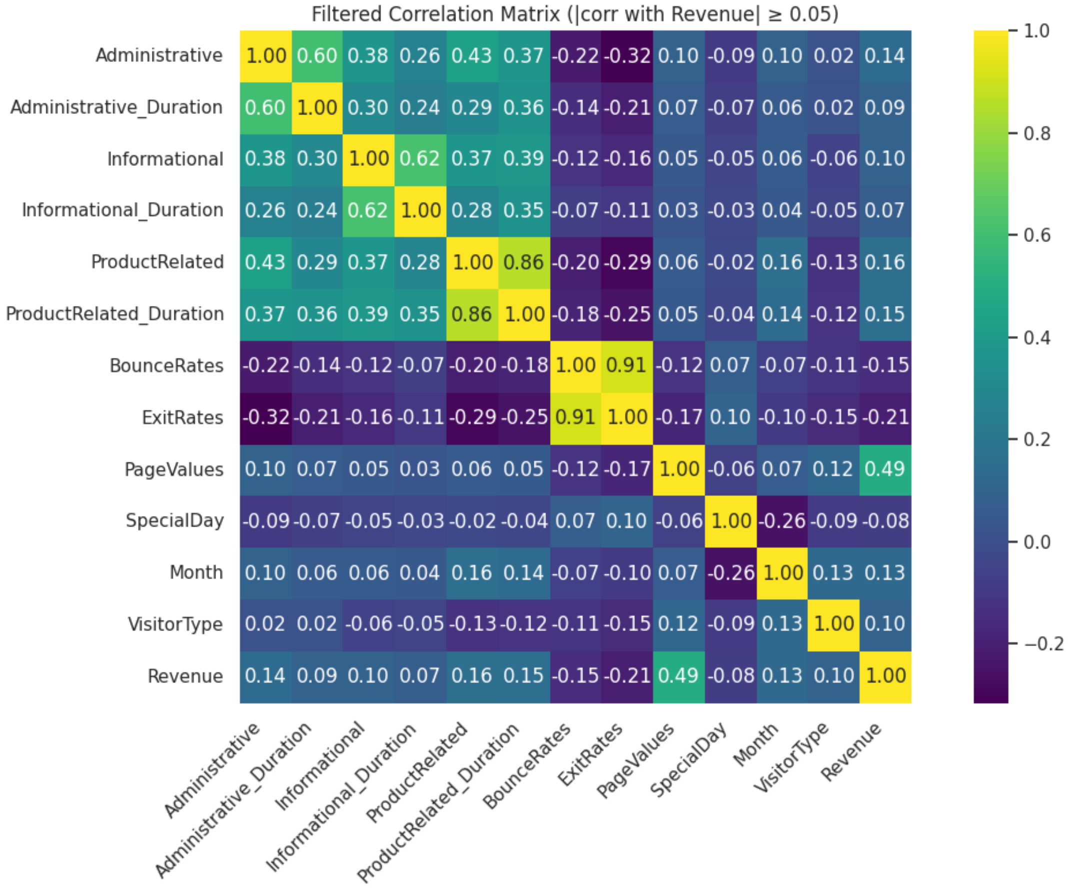
## 2.2 Visitor Type Distribution

Over 75% of the sessions were initiated by returning visitors, with new and other visitor types representing a minority. While this feature is itself imbalanced, it remains highly relevant to the classification task, as returning visitors may exhibit stronger purchase intent.

Unlike some low-impact behavioral features (e.g., number of informational pages viewed), `VisitorType` encodes prior familiarity with the site — a factor that can significantly influence conversion behavior. For this reason, the feature was retained and numerically encoded for use in all models.

## 2.3 Correlation Analysis

We computed a Pearson correlation matrix on the numerically encoded dataset. Figure 1 shows a filtered heatmap, retaining only features with a correlation above 0.05 (absolute value) with the target variable. This visual representation helped identify features that are most predictive of conversion behavior.



| | Administrative | Administrative_Duration | Informational | Informational_Duration | ProductRelated | ProductRelated_Duration | BounceRates | ExitRates | PageValues | SpecialDay | Month | VisitorType | Revenue |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Administrative | 1.00 | 0.60 | 0.38 | 0.26 | 0.43 | 0.37 | -0.22 | -0.32 | 0.10 | -0.09 | 0.10 | 0.02 | 0.14 |
| Administrative_Duration | 0.60 | 1.00 | 0.30 | 0.24 | 0.29 | 0.36 | -0.14 | -0.21 | 0.07 | -0.07 | 0.06 | 0.02 | 0.09 |
| Informational | 0.38 | 0.30 | 1.00 | 0.62 | 0.37 | 0.39 | -0.12 | -0.16 | 0.05 | -0.05 | 0.06 | -0.06 | 0.10 |
| Informational_Duration | 0.26 | 0.24 | 0.62 | 1.00 | 0.28 | 0.35 | -0.07 | -0.11 | 0.03 | -0.03 | 0.04 | -0.05 | 0.07 |
| ProductRelated | 0.43 | 0.29 | 0.37 | 0.28 | 1.00 | 0.86 | -0.20 | -0.29 | 0.06 | -0.02 | 0.16 | -0.13 | 0.16 |
| ProductRelated_Duration | 0.37 | 0.36 | 0.39 | 0.35 | 0.86 | 1.00 | -0.18 | -0.25 | 0.05 | -0.04 | 0.14 | -0.12 | 0.15 |
| BounceRates | -0.22 | -0.14 | -0.12 | -0.07 | -0.20 | -0.18 | 1.00 | 0.91 | -0.12 | 0.07 | -0.07 | -0.11 | -0.15 |
| ExitRates | -0.32 | -0.21 | -0.16 | -0.11 | -0.29 | -0.25 | 0.91 | 1.00 | -0.17 | 0.10 | -0.10 | -0.15 | -0.21 |
| PageValues | 0.10 | 0.07 | 0.05 | 0.03 | 0.06 | 0.05 | -0.12 | -0.17 | 1.00 | -0.06 | 0.07 | 0.12 | 0.49 |
| SpecialDay | -0.09 | -0.07 | -0.05 | -0.03 | -0.02 | -0.04 | 0.07 | 0.10 | -0.06 | 1.00 | -0.26 | -0.09 | -0.08 |
| Month | 0.10 | 0.06 | 0.06 | 0.04 | 0.16 | 0.14 | -0.07 | -0.10 | 0.07 | -0.26 | 1.00 | 0.13 | 0.13 |
| VisitorType | 0.02 | 0.02 | -0.06 | -0.05 | -0.13 | -0.12 | -0.11 | -0.15 | 0.12 | -0.09 | 0.13 | 1.00 | 0.10 |
| Revenue | 0.14 | 0.09 | 0.10 | 0.07 | 0.16 | 0.15 | -0.15 | -0.21 | 0.49 | -0.08 | 0.13 | 0.10 | 1.00 |

Figure 1: Correlation matrix filtered on features with |corr| ≥ 0.05 with `Revenue`.

- PageValues was positively correlated with Revenue, indicating that sessions with higher page value are more likely to lead to a transaction.
- BounceRates and ExitRates were negatively correlated with Revenue, suggesting that high exit/bounce behavior corresponds to lower purchase likelihood.
- ProductRelated_Duration also showed positive correlation, highlighting that spending time on product-related pages increases conversion probability.

Features with near-zero correlation were not removed, as modern models such as Random Forests and XGBoost are robust to irrelevant features.

### 2.4 Revenue vs. Key Behavioral Metrics

Boxplots and group-wise comparisons confirmed that converting sessions had significantly higher values for PageValues, and lower values for BounceRates. These visual trends further supported the predictive value of several features.

## 3 Preprocessing and Data Splitting

Before training any model, a consistent and clean preprocessing of the dataset was applied. Most machine learning algorithms expect numerical input features, so all categorical and boolean variables were transformed accordingly.

### 3.1 Variable Encoding and Transformation

The following preprocessing steps were performed:

- **Categorical mapping:**
  - Month was converted from month abbreviations (e.g., 'Jan', 'Feb', ...) to numerical values (1–12).
  - VisitorType was mapped to integers: Returning_Visitor = 0, New_Visitor = 1, and Other = 2.
- **Boolean encoding:** Revenue and Weekend were converted from True/False to 1/0.

This transformation ensured all features were numerical and compatible with both linear and tree-based models.

### 3.2 Train/Test Split

The data was then divided into training and test sets using an 80/20 split, with stratification on the Revenue variable to preserve class distribution across the sets. The training set was used for cross-validation and hyperparameter tuning, while the test set was held out for final evaluation.

### 3.3 Feature Selection Strategy

Initially, all features were retained to allow the models to capture as much information as possible. This decision was motivated by the relatively small number of features (17) and moderate dataset size (12,330 sessions), which did not justify aggressive dimensionality reduction at this stage. Moreover, tree-based models like Random Forest and XGBoost are known to be robust to irrelevant or weakly informative features.

Nevertheless, a feature selection experiment was conducted in a second phase, with the goal of evaluating model performance when trained on a more compact representation of the data. In this setup, only features with a Pearson correlation coefficient of at least 0.05 (in absolute value) with the target variable `Revenue` were retained.

This allowed for a fair comparison between models trained on the full dataset and those trained on a reduced but informative subset of variables, providing insights into the trade-off between model complexity, training time, and predictive performance.

## 4 Model Evaluation and Comparison

To assess the effectiveness of different modeling strategies, we trained and evaluated several classifiers using both the full feature set and a reduced set of variables selected through correlation filtering and multicollinearity removal.

### 4.1 Performance Comparison

Table 1 summarizes the performance of each model on the held-out test set. The key metric used for model selection is the **recall for class 1** (i.e., purchase), as false negatives are more costly in this context.

Table 1: Model Performance on Test Set

| **Model** | **Features** | **Recall (Class 1)** | **F1 (Class 1)** | **Accuracy** |
|---|---|---|---|---|
| Logistic Regression | All | 0.70 | 0.61 | 0.86 |
| Logistic Regression | Selected | 0.70 | 0.61 | 0.86 |
| Logistic Regression + SMOTE | All | 0.70 | 0.59 | 0.85 |
| Random Forest | All | 0.76 | 0.65 | 0.88 |
| Random Forest | Selected | 0.76 | 0.66 | 0.88 |
| XGBoost | All | **0.84** | 0.64 | 0.86 |
| XGBoost | Selected | **0.84** | 0.64 | 0.86 |
| TabPFN | All | 0.72 | **0.67** | 0.90 |

### 4.2 Observations

- The performance of all models remained remarkably stable after feature selection, confirming that the reduced subset retained the most informative variables.

- XGBoost achieved the highest recall (84%), making it the best option for minimizing false negatives (i.e., missing potential buyers).
- Logistic Regression and Random Forest also performed competitively, especially considering their interpretability and lower complexity.
- Feature selection helped reduce training time (30–40% faster) with no drop in performance, highlighting the benefit of compact models in production settings.

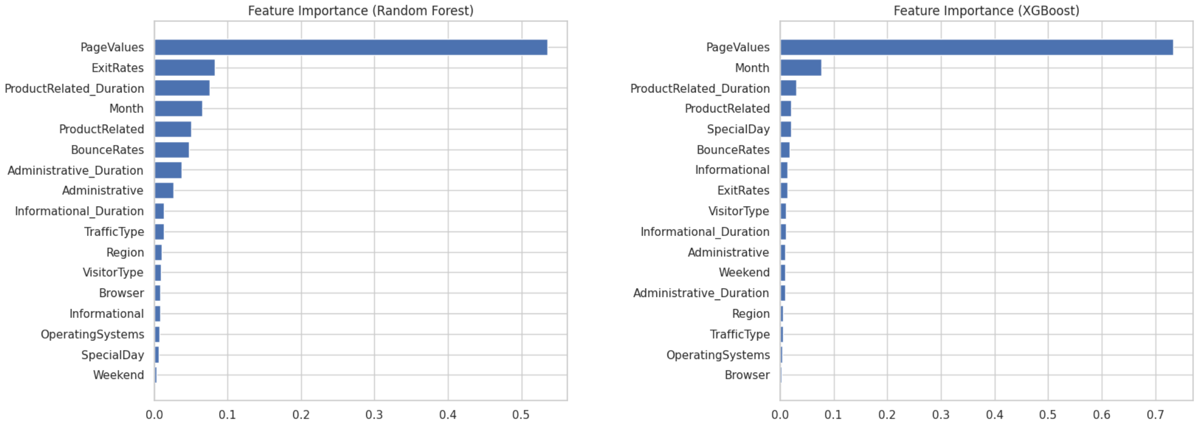
### 4.3 Additional Note on SMOTE Oversampling

SMOTE (Synthetic Minority Oversampling Technique) was tested alongside logistic regression to artificially balance the training set by generating synthetic samples of the minority class. While the recall remained similar (around 70%), the F1-score slightly decreased, and no significant improvement was observed compared to using standard class weighting.

Given this limited added value, and considering the potential risk of introducing synthetic noise, SMOTE was deemed unnecessary — especially when working with models that natively handle class imbalance through weighting strategies, such as logistic regression, Random Forest, or XGBoost.

## 5 Feature Importance Analysis

To better understand the decision-making process of the models, we extracted feature importance scores from both Random Forest and XGBoost — the two most powerful models tested in this study.



(a) Random Forest Feature Importance

(b) XGBoost Feature Importance

Figure 2: Side-by-side comparison of feature importances from Random Forest and XGBoost models.

### 5.1 Insights

In both models, `PageValues` clearly emerged as the most influential predictor, confirming its strong relationship with purchase behavior. However, Random Forest appeared to leverage a more diverse set of features, assigning meaningful importance to variables such as ExitRates, ProductRelated_Duration, Month, and BounceRates, suggesting that it integrates more contextual signals into its predictions. In contrast, XGBoost concentrated most of its weight on PageValues, which likely explains its high predictive power, though at the expense of reduced feature diversity. Overall, despite their architectural differences, both models exhibit a strong dependence on PageValues, making it a central driver of purchasing predictions in this dataset.

## 5.2 Model Robustness Consideration

The strong reliance on PageValues — especially in XGBoost — can be a double-edged sword. While it boosts predictive power, it also increases sensitivity to changes in this variable (e.g., tracking logic, website structure).

## 5.3 Bonus: TabPFN – A Foundation Model for Tabular Data

TabPFN (Tabular Prior-data Fitted Network) is a novel transformer-based model introduced at NeurIPS 2022. Unlike traditional machine learning models, TabPFN is trained on millions of synthetic datasets generated from causal models, allowing it to learn general principles of tabular data learning.

At inference time, TabPFN receives the full dataset (including the test set) and produces predictions in a single forward pass, without any gradient-based optimization. Its architecture applies attention across both features and samples, ensuring invariance to row and column order.

Despite being pretrained on synthetic data, TabPFN achieved the highest F1-score in this project, surpassing both Random Forest and XGBoost, with virtually no hyperparameter tuning. While it remains a black-box and computationally heavier at inference time, it demonstrates the growing potential of foundation models in tabular machine learning.

### 5.3.1 Effectiveness on Imbalanced Data

One of the most remarkable features of TabPFN is its robustness to imbalanced datasets. Although the model does not rely on traditional balancing techniques (such as class weighting or SMOTE), it handles imbalance effectively thanks to its pretraining strategy. During training, TabPFN was exposed to millions of synthetic classification tasks, many of which included strong class imbalance. By learning to approximate the Bayes-optimal predictor in these settings, the model implicitly learns to adapt to skewed label distributions.

This generalization ability likely explains why TabPFN achieved the highest F1-score on our test set, even outperforming tuned Random Forest and XGBoost models — without any class weighting or oversampling.

# 6 Business Applications and Final Thoughts

## 6.1 Business Applications

The predictive model developed in this project has clear practical applications for e-commerce platforms seeking to increase conversion rates and optimize marketing efforts. By identifying sessions with a high probability of purchase in real time, the model can support several business use cases:

- **Personalized Engagement:** High-probability visitors can be targeted with tailored offers, real-time chat support, or limited-time discounts to encourage conversion.
- **Resource Allocation:** Customer support and marketing resources can be dynamically focused on sessions most likely to convert, improving cost-efficiency.
- **Segmentation and Reporting:** The output of the model can be used to segment traffic into "hot leads" vs. "cold traffic", supporting analytics and high-level strategic decisions.
- **A/B Testing Guidance:** By understanding the behavioral signals most correlated with conversion, UX teams can design and prioritize more impactful website experiments.

## 6.2 Final Thoughts

This project demonstrates the effectiveness of combining behavioral data with machine learning techniques to forecast purchase intent. Despite a significant class imbalance, the use of robust models (Random Forest, XGBoost), proper class weighting, and feature selection led to accurate and interpretable predictions.

Interestingly, reducing the input features to a compact subset (based on correlation and multicollinearity filtering) did not degrade performance, highlighting the importance of exploratory analysis and model simplification.

The best performing model — XGBoost with selected features — achieved a recall of 84% for the positive class, making it a strong candidate for deployment in production environments where identifying potential buyers is critical.

Future work may include deploying the model in a real-time inference pipeline to enable instant decision-making, integrating advanced interpretability techniques such as SHAP to better understand feature contributions, and continuously retraining the model on updated session data to ensure adaptability to evolving user behavior.