Online Purchasing Behavior Prediction

Student Name: Ishita Patel

Student ID: 501326439

Date of Submission: November 18, 2024

CIND820: Big Data Analytics Project

Tamer Abdou, Ph.D.

Table of Contents

Introduction

The fast expansion of internet shopping has completely changed how we buy things today – all thanks to the progress of tech and the popularity of stores with diverse products easily accessible to everyone through the internet connection boost. These days, many people choose to shop for their purchases. It is crucial for businesses to understand what influences consumer decisions and why they might walk out empty-handed. This understanding is essential for companies aiming to enhance customer satisfaction levels and boost their sales through smarter marketing tactics. This research delves into uncovering the elements influencing shopping habits and creating models to project buying behavior in advance. By utilizing the Online Shoppers Purchasing Intention Dataset sourced from the UC Irvine Machine Learning Repository with over 12k user session records as a basis for investigation, this study delves into inquiries like "What behaviors signal a purchase?" and "How can companies construct predictive frameworks for understanding customer buying choices?" This study seeks to discover patterns that can guide business strategies by utilizing classification and regression methods, like Logistic Regression and Naive Bayes along with Support Vector Machines combined with clustering and association rule mining techniques. The research aims to offer actionable insights for businesses looking to enhance their platforms and boost conversions by identifying key factors in online purchasing behavior, tackling class imbalances, and utilizing a Random Forest Classifier while meticulously assessing model performance to inform targeted marketing strategies and optimize customer engagement.

Literature Review

There has been previous literature demonstrating the goal of modeling online shopping behavior to predict successful transactions. Other research has focused on specific types of data such as clickstream data to model user transactions (Baati K. & Mohsil, M. 2020) or had used more complicated models such as Long Short Term Memory Recurrent Neural Network (LSTM-RNN) (Diamantas et al., 2021). Previous literature had indicated success in being able to accurately predict successful user transactions.

The data used in this project was first used in the paper Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural network by Sakar, C. O., Polat, S. O., Katircioglu, M., & Kastro, Y. The authors sought to create a model that would predict a customer's purchasing intention and the likelihood that they would abandon the website.

This was done through creation of two modules - the first of which would predict visitor purchasing intention from the dataset. The second module would focus on using sequential clickstream data of customers to train a neural network that would predict a customer who is likely to leave the site. The main research goal was to look into the feasibility of predicting purchasing intention gained from clickstream data as well as session information data.

This paper sought to use decision tree (C4.5 and random forest) classifiers, multilayer perceptron and support vector machines to predict user shopping behavior. These methods were chosen based on their use in past research. Upon running the models, the researchers found that they achieved poor model performance due to the large imbalance between negative and positive class samples. The researchers addressed the imbalance of class samples by dividing the dataset into training and test groups and then oversampling the dataset in order to look for stronger model performance. Afterwards, the data was standardized to prepare for feature selection. Next, the features were ranked using different methods, including correlation, mutual information, and mRMR methods. These were chosen as they sought to apply filter-based feature selection over wrapper algorithms that would require learning algorithms. Afterwards, feature selection was conducted using the MLP algorithm, which selected the top features to keep, with the model performing the best when using the top 6 features with the mRMR method.

Descriptive Statistics of Selected Dataset

The dataset consists of 12,330 observations with 18 features. Out of these 18 characteristics, 10 are considered numerical while the remaining 8 are labeled as categorical. The data indicates that 84. 50 % (10 422) Of the instances correspond to situations where a customer chose not to buy anything while 15. 50 % (1,908) Of the instances led to a purchase. The dataset does not have any data points, and the presence of entries is simply random occurrences that happened while gathering data from various users.

List of Attributes

1. Administrative

- **Type**: Continuous (numeric)
- **Description**: The number of pages visited by the user related to account management (e.g., logging in, signing up, etc.) during the session.

2. Administrative Duration

- **Type**: Continuous (numeric)
- **Description**: Total amount of time (in seconds) the user spent on administrative pages during the session.

3. Informational

- **Type**: Continuous (numeric)
- **Description**: The number of pages related to informational content (e.g., FAQs, company info) visited during the session.

4. Informational Duration

- **Type**: Continuous (numeric)
- **Description**: Total time (in seconds) the user spent on informational pages during the session.

5. Product-Related

- **Type**: Continuous (numeric)
- **Description**: The number of pages related to specific products (e.g., product details, reviews) visited during the session.

6. Product-Related Duration

- **Type**: Continuous (numeric)
- **Description**: Total time (in seconds) the user spent on product-related pages during the session.

7. Bounce Rates

- **Type**: Continuous (numeric)
- **Description**: The percentage of visitors who leave the website after viewing only one page (i.e., without interacting further).

8. Exit Rates

- **Type**: Continuous (numeric)
- **Description**: The percentage of page exits calculated for each page in the session (i.e., the rate at which users leave the site from that page).

9. Page Value

- **Type**: Continuous (numeric)
- **Description**: A calculated metric representing the average value of a page visited by users before completing an e-commerce transaction (if applicable).

10. Special Day

- **Type**: Continuous (numeric)
- **Description**: A score representing the closeness of the session to a special day (e.g., holiday). Values closer to 1 indicate proximity to a special shopping event or holiday.

11. Month

- **Type**: Categorical
- **Description**: The month in which the session occurred, represented as categorical values (e.g., "Jan," "Feb," etc.).

12. Operating System

- **Type**: Categorical
- **Description**: The type of operating system used by the user (e.g., Windows, MacOS, Linux, etc.).

13. Browser

- **Type**: Categorical
- **Description**: The browser used by the user during the session (e.g., Chrome, Firefox, Safari, etc.).

14. Region

- **Type**: Categorical
- **Description**: The geographical region or location from which the session was initiated.

15. Traffic Type

- **Type**: Categorical
- **Description**: The source of the traffic that brought the user to the website (e.g., direct, referral, organic search, etc.).

16. Visitor Type

- **Type**: Categorical
- **Description**: Whether the visitor is a **Returning Visitor** or a **New Visitor**.

17. Weekend

- **Type**: Boolean (Binary)
- **Description**: A binary variable indicating whether the session took place during the weekend (1 for Yes, 0 for No).

18. Revenue

- **Type**: Boolean (Binary)
- **Description**: The target label indicating whether the session resulted in a transaction (purchase). **1** indicates a purchase was made, while **0** indicates no purchase.

Numerical features

Feature name	Feature description	Min. val	Max. val	SD
Admin.	#pages visited by the visitor about account management	0	27	3.32
Ad. duration	#seconds spent by the visitor on account management related pages	0	3398	176.70
Info.	#informational pages visited by the visitor	0	24	1.26
Info. durat.	#seconds spent by the visitor on informational pages	0	2549	140.64
Prod.	#pages visited by visitor about product related pages	0	705	44.45
Prod.durat.	#seconds spent by the visitor on product related pages	0	63,973	1912.3
Bounce rate	Average bounce rate value of the pages visited by the visitor	0	0.2	0.04
Exit rate	Average exit rate value of the pages visited by the visitor	0	0.2	0.05
Page value	Average page value of the pages visited by the visitor	0	361	18.55
Special day	Closeness of the site visiting time to a special day	0	1.0	0.19

Categorical features

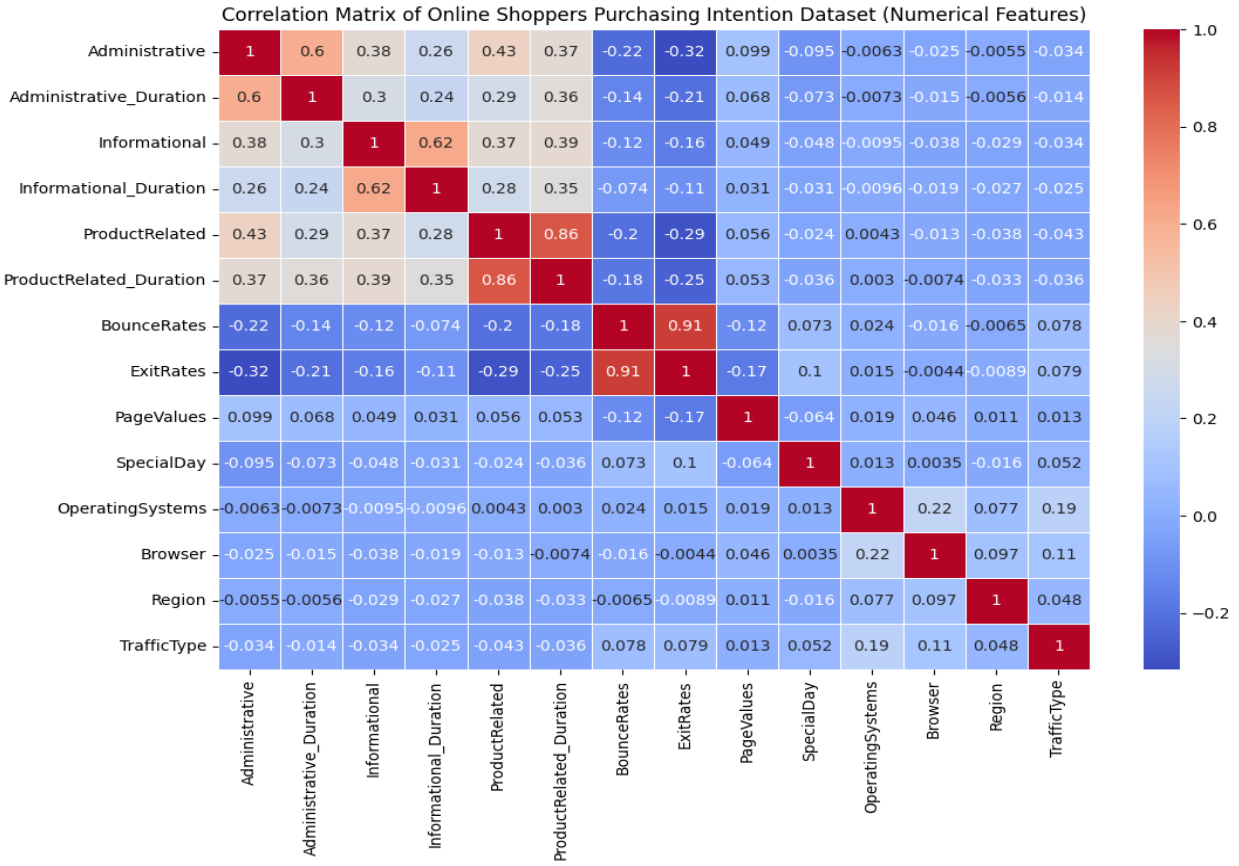
Feature name	Feature description	Number of Values
OperatingSystems	Operating system of the visitor	8
Browser	Browser of the visitor	13
Region	Geographic region from which the session has been started by the visitor	9
TrafficType	Traffic source (e.g., banner, SMS, direct)	20
VisitorType	Visitor type as “New Visitor,” “Returning Visitor,” and “Other”	3
Weekend	Boolean value indicating whether the date of the visit is weekend	2
Month	Month value of the visit date	12
Revenue	Class label: whether the visit has been finalized with a transaction	2

Objective

The goal is to create a classification model that can correctly identify which consumer behaviors resulted in a purchaseand which did not, while employing models that are less time-consuming and computationally expensive than those employed in earlier research. Data preprocessing will include data balancing algorithms, as discrepancies between records of outcomes will impair prediction accuracy. Additionally, data scaling approaches will be used to see if model performance improves when numeric data is normalized or standardized. Following that, feature selection will be performed using Pearson's correlation matrix, a Random Forest Classifier, and Recursive Feature Elimination to increase classification accuracy. The data will then be divided into two datasets: training (70%) and testing (30%). Decision Tree, Random Forest, and Logistic Regression models will then be used to determine whether various customer behaviours resulted in purchases. K-fold and time-series cross-validation techniques will also be used to compare the results to the control data to see whether there are any performance gains, which is noticeably absent from previous studies. One major point of emphasis in the literature is observing how many attributes/features are abandoned when filtering for features. Principal component analysis will be

considered as a feasible technique to minimize the dimensionality of lost features while retaining some information for further training and improving the model's performance.

Data Correlation



The correlation matrix shows that there are connections, between the numerical factors, in the Online Shoppers Purchasing Intention Dataset. Specifically, ProductRelated and ProductRelated_Duration have a correlation of 0 86 suggesting that how often users visit product related pages significantly affects how long they stay on the site. Moreover, BounceRates and ExitRates have a positive correlation of 0 91 indicating that users who leave a page are more likely to exit the session completely. On the other hand Indicators, like ExitRates show some negative connections with Administrative tasks, Administrative_Duration and ProductRelated activities indicating that spending more time on these aspects might reduce the likelihood of leaving. PageValues demonstrate associations with characteristics indicating minimal impact from other factors.

Methodology

The study's approach is centered on organizing the data through preprocessing steps, like handling missing values and selecting features before training and evaluating the model using Random Forest Classifier to enhance efficiency by identifying predictive features in a balanced dataset after applying SMOTE for class imbalance adjustments and encoding categorical variables for improved modeling suitability. The information was. Processed using SMOTE to address imbalanced classes in order to enhance the data, for analysis purposes optimally. Utilize the Random Forest Classifier to pinpoint and keep the features that boost the effectiveness of the model. Develop classification models (Logistic Regression, Naive Bayes, SVM), clustering (K-Means), and association rule mining (Apriori) to understand and predict purchasing behavior. Evaluated models based on their accuracy, precision, recall, F1-score, and AUC-ROC guarantee results and gather practical insights for e-commerce businesses online. Analyze consumer buying habits to understand how website design can be modified to enhance transactions.

Data, Predictive Models, & Metrics

The data for this project was sourced from the Online Shoppers Purchasing Intention Dataset available in the UCI Machine Learning Repository. The dataset originally included 10 numerical features and 8 categorical features. To enhance model performance and effectively handle categorical variables, one-hot encoding was applied, resulting in a total of 75 features. Exploratory Data Analysis (EDA) was conducted using Python's PyPI library. Predictive modeling was performed using Decision Tree, Random Forest, and Logistic Regression algorithms to classify the target variable, "Revenue." These models were chosen for their ease of interpretability and low time and computing resources when it came to training and evaluating performance.

The Decision Tree, Random Forest, and Logistic Regression models were developed using libraries from Scikit-learn. The Random Forest model employed its default configuration of 100 estimators, while the Logistic Regression model was configured with a default maximum of 1000 iterations.

To evaluate the performance of the models, used Scikit-learn's classification_report and confusion_matrix libraries. The F1 score was a key metric, as it balances precision and recall—assessing how well the model identifies true positives while avoiding false positives. Additionally, accuracy was calculated to measure the overall proportion of correct predictions, including both

true positives and true negatives, out of all predictions made.Time and memory efficiency were also metrics of concern when evaluating model performance. Python's process_time library was used to track how much time had passed between fitting the data to the model and predicting the test set.

After selecting certain models to focus on, Brier's score and Matthew's correlation metrics were imported from sklearn. Brier's score is a proper scoring rule that measures the accuracy of probabilistic predictions. It quantifies the average squared difference between predicted probabilities and the observed outcomes, where lower scores indicate better calibration and prediction accuracy. Matthews correlation coefficient (MCC) is a measure of the quality of binary classification predictions. It takes into account true positives, true negatives, false positives, and false negatives to provide a balanced evaluation of the classifier's performance. Variance of metrics were also measured to calculate variation across 5 runs of each model and to assess stabilities of the predictive models.

Data Balancing

The dataset contains 84.5% (10,422) samples where a shopper did not go on to purchase an item, and 15.5% (1,908) samples, indicating a large imbalance between negative and positive variables. Previous literature emphasized SMOTE (Synthetic Minority Oversampling Technique) as the chosen technique to address target variables. Other techniques such as undersampling and oversampling were chosen to compare with SMOTE. The RandomUnderSampler, RandomOverSampler and SMOTE libraries were imported from imblearn for this application. To test the performance of data balancing techniques, a decision tree model was applied to predict the target variable with undersampling, oversampling and SMOTE techniques alongside a control group. No other data preprocessing techniques were applied at this point.

Dataset	**Negative Class F1-Score**	**Positive Class F1-Score**	**Accuracy**	**Time (s)**
Imbalanced, Raw	0.92	0.55	0.86	0.1131
Under sampling on Raw Data	0.88	0.55	0.81	0.0449

Dataset	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)
Oversampling on Raw Data	0.93	0.60	0.88	0.1728
SMOTE	0.92	0.57	0.86	0.2287

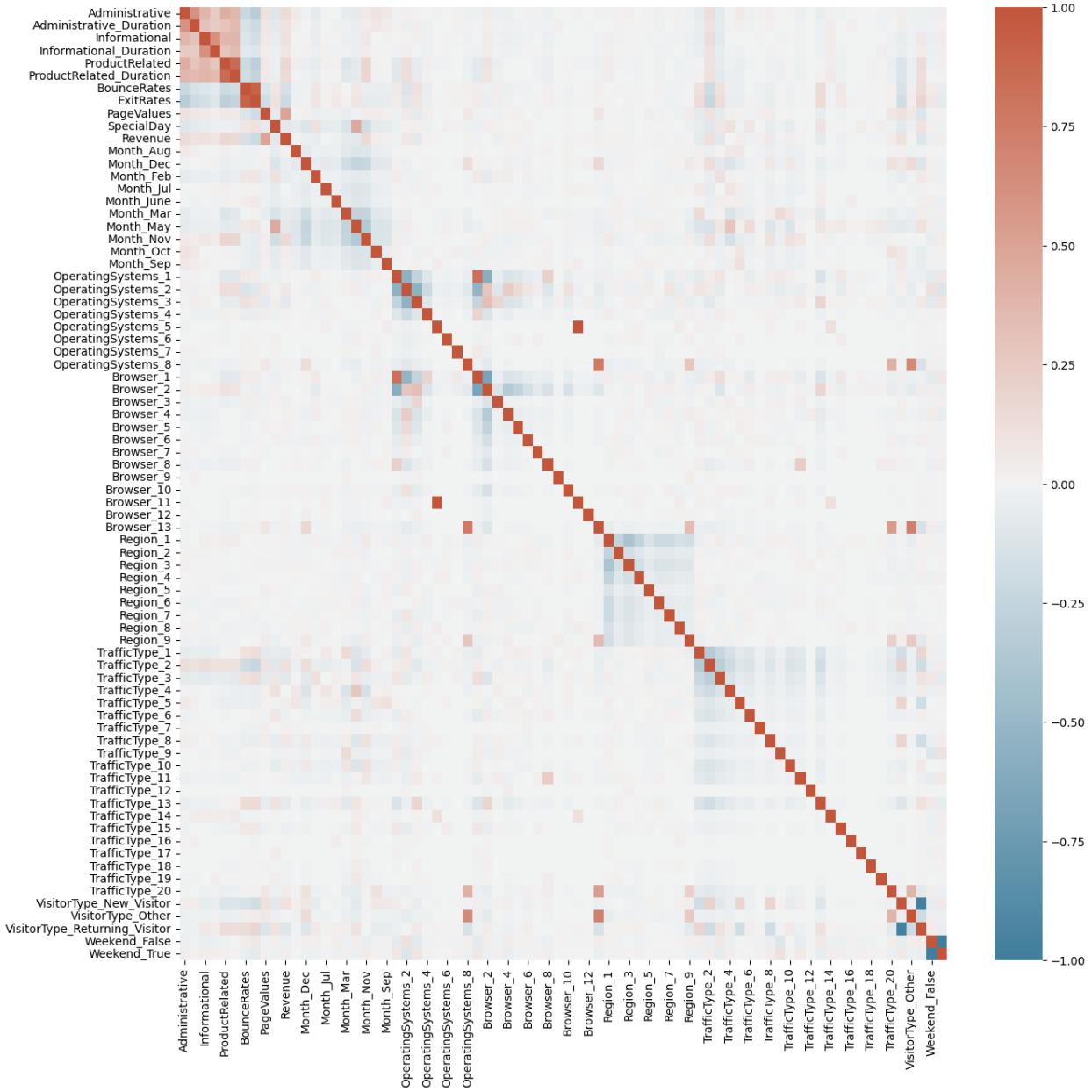
Results of different data balancing techniques on the data.

The results show that oversampling achieved the best overall performance with the highest accuracy (88%) and a positive class F1-score of 0.60 while maintaining a strong negative class F1-score of 0.93. SMOTE provided a balanced trade-off with an accuracy of 86% and F1-scores of 0.92 (negative class) and 0.57 (positive class), though it required slightly more time and memory. The raw dataset and undersampling methods performed comparably in terms of positive class F1-scores (0.55) but differed in resource usage and overall accuracy. Therefore, due to its accuracy and F1 scores in predicting class, as well as its usage in previous literature, SMOTE was chosen as the data balancing method in this project.

Feature Selection

Pearson's correlation, Random Forest, and Recursive Feature Elimination (RFE) techniques were applied for feature selection, utilizing Scikit-learn libraries. These methods were implemented on the one-hot encoded dataset, retaining the top 25% of features. To evaluate the effectiveness of each feature selection approach, Decision Tree, Random Forest, and Logistic Regression models were employed for testing.

Feature selection was performed using Pearson’s correlation, Random Forest, and Recursive Feature Elimination (RFE) techniques. The outcomes of Pearson’s correlation are displayed in the correlation matrix



Pearson's correlation matrix on the one-hot encoded dataset

The rankings and results for the top 24 features selected by each method. By selecting the top quantile of features based on their scores from each feature selection technique (correlation, importance, or rank), 18 features were retained using Pearson's correlation, 19 features using Random Forest, and 20 features using Recursive Feature Elimination (RFE). Notably, there is significant overlap in features retained by Pearson's correlation and Random Forest, with attributes like PageValues, ExitRates, and ProductRelated showing strong contributions to predicting the

Revenue attribute. However, RFE primarily retained one-hot encoded categorical attributes and excluded many original numeric features.

In terms of model performance, the Random Forest model consistently demonstrated superior accuracy and F1-scores for both positive and negative target classes, though it came with higher memory and time costs. Among the feature selection techniques, Random Feature Elimination underperformed significantly compared to the others and even the control group. Interestingly, there was no substantial difference in performance between models trained on features selected by Pearson's correlation, Random Forest, or the control dataset, indicating no clear preference for a specific feature selection method. Moving forward, the Random Forest model will be tested on the control dataset as well as datasets filtered using Pearson's correlation and Random Forest feature selections.

Rank	Pearson Correlation	Random Forest	Recursive Feature Elimination
1	PageValues	PageValues	Weekend_True
2	ExitRates	ExitRates	Weekend_False
3	ProductRelated	ProductRelated_Duration	Month_Sep
4	Month_Nov	Administrative_Duration	Month_Oct
5	ProductRelated_Duration	ProductRelated	Month_Nov
6	BounceRates	BounceRates	Browser_12
7	Administrative	Month_May	Browser_3
8	TrafficType_2	Administrative	TrafficType_13
9	VisitorType_New_Visitor	OperatingSystems_3	Month_Jul
10	VisitorType_Returning_Visitor	VisitorType_Returning_Visitor	Month_Feb

11	Informational	Month_Dec	TrafficType_15
12	Administrative_Duration	TrafficType_1	SpecialDay
13	TrafficType_3	Informational_Duration	TrafficType_3
14	SpecialDay	Month_Mar	ExitRates
15	Month_May	Browser_2	BounceRates
16	OperatingSystems_3	OperatingSystems_1	TrafficType_7
17	Informational_Duration	TrafficType_3	TrafficType_1
18	TrafficType_13	Weekend_True	TrafficType_18
19	TrafficType_1	Operating_systems_2	Month_Aug
20	Month_Mar	Weekend_False	OperatingSystems_3
21	OperatingSystems_2	Browser_1	TrafficType_19
22	TrafficType_8	Region_1	TrafficType_20
23	Month_Feb	Region_3	TrafficType_6
24	TrafficType_20	TrafficType_2	TrafficType_16

Feature Selection (w/ SMOTE)

Decision Tree Model

Data	**Negative Class F1-Score**	**Positive Class F1-Score**	**Accuracy**	**Time (s)**	**Memory (MiB)**
Control	0.91	0.57	0.85	0.9375	245.04

Pearson Correlation	0.91	0.59	0.86	0.078125	209.42
Random Forest	0.92	0.55	0.86	0.9375	239.09
Recursive Feature Elimination	0.81	0.34	0.71	0.46875	244.23

Random Forest Model

Data	**Negative Class F1-Score**	**Positive Class F1-Score**	**Accuracy**	**Time (s)**	**Memory (MiB)**
Control	0.93	0.64	0.89	1.26525	265.24
Pearson Correlation	0.93	0.66	0.88	1.53125	227.65
Random Forest	0.94	0.65	0.89	1.5	254.89
Recursive Feature Elimination	0.83	0.34	0.73	1.0625	285.75

Logistic Regression Model

Data	**Negative Class F1-Score**	**Positive Class F1-Score**	**Accuracy**	**Time (s)**	**Memory (MiB)**
Control	0.92	0.52	0.87	0.921875	253.05
Pearson Correlation	0.92	0.69	0.87	0.140625	210.72
Random Forest	0.92	0.53	0.86	0.578125	239.93
Recursive Feature Elimination	0.74	0.38	0.63	0.046875	244.79

Data Scaling

The results of data scaling are shown in Figure 5, where the Random Forest model was used on one-hot encoded, SMOTE oversampled data. Pearson's correlation and Random Forest filter techniques were also applied to the data to compare results to control data. We see that there is not a significant increase in model performance upon scaling the data, however there are minor improvements, mostly in the F1-Score (average of precision and recall) in the positive class upon standardization of the data to have a mean of 0 and a standard deviation of 1. Notably, there are increased time and memory costs with data scaling, however the increases in costs are still minimal. As such, due to slight improvements in performance and prior usage in the literature, data was chosen to be standardized moving forwards.

Control Data

Feature Selection Method	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
Control	0.94	0.67	0.90	3.2941	322.93
Pearson Correlation	0.95	0.62	0.89	3.2490	323.01
Random Forest	0.94	0.65	0.89	2.4509	323.04

Normalized Data

Feature Selection Method	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
Control	0.94	0.68	0.90	2.6782	351.00
Pearson Correlation	0.95	0.62	0.89	3.1820	342.43
Random Forest	0.94	0.65	0.89	2.5389	342.65

Standardized Data

Feature Selection Method	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)

Control	0.94	0.67	0.90	3.0611	376.74
Pearson Correlation	0.94	0.62	0.88	2.5837	376.74
Random Forest	0.94	0.67	0.90	2.5905	367.74

Model performance results of different data scaling methods on SMOTE-balanced dataset using the Random Forest prediction model.

Cross-Validation

It was important to explore whether cross-validation could enhance the predictive model's performance. To assess this, K-fold and Time-Series cross-validation techniques from sklearn were applied to the SMOTE-oversampled, standardized data. This allowed us to evaluate the performance of the Random Forest model on different datasets: the unfiltered data, data filtered based on Pearson's correlation, and data filtered using Random Forest feature selection. The default setting of five (5) folds was used for splitting the data. Although Leave-One-Out Cross-Validation (LOOCV) was also considered, it was ultimately not feasible due to the large size of the dataset and limited computing resources.

The functions used to call the Random Forest model and apply cross-validation techniques were modified from previous steps to include SMOTE oversampling and data standardization within the function as it was necessary for cross-validation to be performed. The classification reports generated by the function are an average of all 5 folds performed.

Applying cross-validation techniques led to minor improvements in model performance, primarily in the Negative Class F1-score. Notably, both K-fold and Time-Series cross-validation resulted in increased recall. However, the model without cross-validation still performed well, showing significant advantages in the Positive Class F1-score. It's also worth noting that the application of cross-validation significantly increased the time required for model processing. Moving forward, the focus will be on tuning models using the control data with Pearson's correlation features, as well as the K-Fold and Time-Series cross-validated data with Pearson's correlation features.

Control Data

Feature Selection Method	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
Control	0.94	0.67	0.90	2.4582	320.88
Pearson Correlation	0.94	0.67	0.89	3.3424	320.89
Random Forest	0.94	0.70	0.91	2.5755	320.91

K-Fold Cross Validation

Feature Selection Method	Negative ClassF1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
Control	0.94	0.61	0.90	11.8440	346.26
Pearson Correlation	0.94	0.63	0.90	11.8789	328.16
Random Forest	0.94	0.63	0.90	11.8573	327.20

Time-Series Cross Validation

Feature Selection Method	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
Control	0.94	0.61	0.90	7.7769	342.71
Pearson Correlation	0.94	0.63	0.90	7.5171	342.76
Random Forest	0.94	0.61	0.89	7.2702	342.78

Model Tuning

The number of estimators in each model and number of folds in K-fold and Time-series cross-validation techniques were also modified to look for increases in performance.

Pearson Correlation Filter/ No Cross-Validation

Number of Estimators	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
100	0.93	0.68	0.89	3.3532	315.92
500	0.93	0.68	0.89	16.8454	390.70
1000	0.93	0.68	0.89	30.8606	486.32

Pearson Correlation Filter/ K-Fold Cross Validation

Number of Estimators	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
100	0.94	0.63	0.90	13.262	426.45
500	0.94	0.65	0.90	57.424	426.61
1000	0.94	0.64	0.90	124.635	426.93

Pearson Correlation Filter/ Time-Series Cross Validation

Number of Estimators	Negative Class F1-Score	Positive Class F1-Score	Accuracy	Time (s)	Memory (MiB)
100	0.94	0.63	0.90	7.919	427.64
500	0.94	0.64	0.90	37.576	427.64
1000	0.94	0.64	0.90	74.096	427.64

Model performance results of number of folds settings on SMOTE-balanced and standardized dataset using the Random Forest prediction model.

The model without cross-validation techniques achieves a higher recall for the positive class and a superior Matthew's correlation coefficient. This indicates that the model is better at correctly identifying transactions likely to result in a purchase, thereby reducing false negatives—a crucial factor when aiming to predict successful transactions. Furthermore, this model demonstrates significantly lower time and computational resource requirements.

On the other hand, models employing cross-validation exhibit higher precision for the positive class and better Brier Score performance. This suggests these models are more effective at

minimizing false positives, reducing the likelihood of recommending transactions that fail to result in actual purchases.

PCA

Principal Component Analysis (PCA) is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving the most significant patterns and minimizing information loss. As many of the feature selection methods used in prior literature had led excluding much of the dataset, PCA was explored as a possible solution to maintain information while reducing dimensionality. PCA was done using the sklearn library, and applied on standardized data that had been scaled. Afterwards, SMOTE sampling was applied and the Random Forest prediction model predicted the target class. A hybrid approach was also utilized, where PCA was applied to data excluded from Pearson's correlation feature selection, and the datasets were combined afterwards to include both attributes chosen by the top quartile of Pearson's correlation features as well as principal components generated from PCA.

Principal component analysis was examined as a possible way to reduce attribute dimensionality, while maintaining some of the features from the data that would normally be filtered out through feature selection methods. When applying PCA on the dataset, it was found that each principal component explained very little of the variance (less than 0.05). It was also found that 18 principal components (equal to the number of features retained by Pearson's correlation) explained 0.44 of total variance, and it took over 60 principal components to explain all of the total variance.

Metric	**Pearson corr. features (No CV)**	**18 Principal Components**	**75 Principal Components**	**Hybrid Approach**
Negative Precision	0.95	0.89	0.91	0.93
Negative Recall	0.92	0.87	0.94	0.94
Negative F1-Score	0.94	0.88	0.92	0.94
Positive Precision	0.62	0.40	0.59	0.66
Positive Recall	0.72	0.44	0.49	0.60

Process F1-Score	0.67	0.42	0.53	0.63
Accuracy	0.89	0.80	0.87	0.89
Matthew's Correlation	0.6054	0.2961	0.4596	0.5645
Brier's Score	0.1084	0.1998	0.1346	0.1084
Time (S)	2.515625	6.234375	10.109375	7.75
Memory (MiB)	235.11	244.49	260.63	299.48

Metrics of each PCA iteration with Pearson's correlation data with Random Forest model as a baseline

Conclusion

Ultimately, the results highlighted that a computationally efficient model could be trained to predict user purchasing behavior with high accuracy. Leveraging data filtered through feature selection offers valuable insights into the pages users are most likely to engage with and the attributes linked to users likely to complete transactions. These findings have practical applications in website design and e-commerce marketing, enabling businesses to optimize their strategies for boosting online sales and gaining a deeper understanding of consumer behavior in online shopping.

In the end the findings showed that a model optimized for efficiency could accurately forecast how users make purchases. By utilizing data that has been refined through feature selection methods we gain information, about which pages users are more likely to interact with and the characteristics associated with users who tend to complete transactions. These findings have practical applications in website design and e-commerce marketing, enabling businesses to optimize their strategies for boosting online sales and gaining a deeper understanding of consumer behavior in online shopping.

References

- Sakar, C. O., Polat, S. O., Katircioglu, M., & Kastro, Y. (2019). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Computing & Applications, 31(10), 6893–6908. https://doi.org/10.1007/s00521-018-3523 0

- Diamantaras, K., Salampasis, M., Katsalis, A., & Christantonis, K. (2021, January). *Predicting shopping intent of e-commerce users using LSTM recurrent ...* Predicting Shopping Intent of e-Commerce Users using LSTM Recurrent Neural Networks. https://www.researchgate.net/publication/353397684_Predicting_Shopping_Intent_of_e-Commerce_Users_using_LSTM_Recurrent_Neural_Networks
- Sakar, C. O., & Kastro, Y. (n.d.). *Online Shoppers Purchasing Intention Dataset Data Set*. UCI Machine Learning Repository: Online Shoppers Purchasing Intention Dataset Data set. https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset
- Sakar, C. O., Polat, S. O., Katircioglu, M., & Kastro, Y. (2018, May 9). *Real-time prediction of online shoppers' purchasing intention using Multilayer Perceptron and LSTM recurrent neural networks - neural computing and applications*. SpringerLink. https://link.springer.com/article/10.1007/s00521-018-3523-0
- Zanzana, S., & Martin, J. (2023, February 21). Retail e-commerce and COVID-19: How online sales evolved as in-person shopping resumed. https://www150.statcan.gc.ca/n1/pub/11-621-m/11-621-m2023002-eng.htm