Online Purchasing Behaviour Prediction

David Wang

501208639

Tamer Abdou, Ph.D.

July 17, 2023

1. **Abstract**

Online commerce is an industry that has greatly benefited from the ever increasing development of today’s technologically connected world. Since the COVID-19 pandemic, businesses across Canada have reported trends of shoppers turning to online purchases over retail stores across different industries.

As such, developing an understanding of the patterns and mindsets of online shoppers would allow businesses to further increase sales. Being able to identify what separates a customer who purchases an item from those who leave the website without a purchase would allow businesses to develop their marketing and websites in a way to maximize purchases made on a website. The goal is then to predict which patterns of online shopping behavior lead to successful purchases. Such findings could help drive marketing decisions, website design and/or product design with the goal of increasing sales in businesses.

Ultimately, it was found that a Random Forest prediction model using standardized data oversampled by SMOTE, and filtered by the top quantile of features selected with Pearson’s correlation was found to be a model that could reliably predict the positive class with nearly 90% accuracy. Notably, this model was found to have a high recall value for the positive class, indicating strong ability in identifying transactions likely to result in purchase. While other models performed stronger in other facets, this model was chosen for overall performance, and lower time and computing resource costs.

1. **Literature**

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.

The data contains 12,330 records comprising 10,422 (84.5%) sessions that did not lead to a purchase and 1,905 (15.5%) sessions that led to a customer purchase. There are 10 numerical attributes and 8 categorical attributes that describe the browsing behavior of each potential customer, such as the types of pages they would visit and the duration of time spent on specific types of pages. External factors, such as the operating systems, browsers, and regions of the customer are also included. Additionally, time-based factors such as proximity to a holiday were recorded to determine if it had an effect on the shopping behavior of customers.

1. **Objective**

The goal is to build a classification model that will successfully be able to classify which customer behaviors led to a purchase and which did not, while using models that will be less costly in terms of time and computing resources than those used in previous literature. Data preprocessing will include data balancing methods as the disparity between records of outcomes will affect prediction performance. Additionally, data scaling techniques will be applied to test whether model performance would increase if numeric data is normalized or standardized. Afterwards, feature selection will proceed using Pearson’s correlation matrix, a Random Forest Classifier, and Recursive Feature Elimination in order to improve classification results. The data will then be split into training (70%) and test (30%) datasets. Decision Tree, Random Forest and Logistic Regression models will then be applied to classify whether specific customer behaviors led to purchases. K-fold and time-series cross-validation techniques will also be applied to compare with the control data to look for any improvements in performance, a step notably missing from prior literature.

One notable point of focus in the literature is noticing how many attributes/ features tend to get dropped when filtering for features. Principal component analysis will be looked at as a possible solution to reduce the dimensionality of features dropped, while being able to retain some information on which to further train and improve the performance of the model.

1. **Methodology**

**4.1 Data, Predictive Models, & Metrics**

Data was obtained from the *Online Shoppers Purchasing Intention Dataset* from the UCI Machine Learning Repository. It contained 10 numerical attributes and 8 categorical attributes that were one hot encoded to address potential issues with model performance and categorical variables. This left the dataset with a total of 75 attributes. Exploratory data analysis was performed using the PyPI library in Python.

Decision Tree, Random Forest and Logistic Regression models were used to predict the target class, “Revenue”, where 1 indicated a successful transaction and 0 indicated behavior where the user did not make a purchase. 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 model was built using libraries from sklearn and was used to determine a data sampling technique due to its low resource training cost. It had a default parameter of criterion = “entropy”.
* The random forest model was built using libraries from sklearn. It had default parameters of 100 estimators.
* The logistic regression model was built using libraries from sklearn. It had default parameters of 1000 max iterations.

sklearn’s classification\_report and confusion\_matrix libraries were used to report model performance metrics. The F1 score measures the harmonic mean of precision and recall, which are metrics that assess the model's ability to correctly identify positive instances (true positives) and the extent to which it avoids misclassifying negative instances (false positives), while accuracy calculates the ratio of correct predictions (true positives and true negatives) to the total number of 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. Additionally, the memory\_profiler was imported to measure the amount of mebibytes (MiB) of storage used by the model when the function containing the model was called.

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. MCC ranges from -1 to +1, where +1 indicates a perfect prediction, 0 indicates random prediction, and -1 indicates total disagreement between predictions and observations. MCC is particularly useful when dealing with imbalanced datasets.

Variance of metrics were also measured to calculate variation across 5 runs of each model and to assess stabilities of the predictive models.

**4.2 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.

**4.3 Feature Selection**

The literature mentions having used correlation, mutual information (MI) and Maximum Relevance — Minimum Redundancy(mRMR) as filters for feature selection. Here, Pearson’s correlation, Random Forest, and Recursive Feature Elimination methods were applied, using libraries from sklearn. Feature selection methods were applied onto the one hot encoded data, and the top quantile (25%) of features were kept. To test the performance of each set of feature selection filters, decision tree, random forest and logistic regression models were used.

**4.4 Data Scaling**

Previous literature mentions data standardization as a preprocessing technique. To apply similar techniques, the one-hot encoded dataset oversampled with SMOTE was split into categorical and numeric attributes, upon which the numeric attributes were scaled. Data was normalized to have a range between 0 and 1 using sklearn’s MinMaxScaler library. Data was also standardized to have a mean of 0 and a standard deviation of 1 using sklearn’s StandardScaler library. To test the performance of data scaling techniques, a decision tree model was applied to predict the target variable with normalization and standardization alongside a control group.

**4.5 Cross-Validation**

There was no mention of any cross-validation techniques having been applied to the dataset in previous literature. As such, it was a point of emphasis to see whether it could be used to improve the predictive model performance. K-fold and Time-Series cross-validation libraries were imported from sklearn to be applied onto SMOTE oversampled, standardized data and to judge the performance of the Random Forest model on unfiltered data, data filtered for Pearson’s correlation features, and data filtered for Random Forest features. The default number of folds to be split was set as five (5). Leave-one-out cross-validation (LOOCV) was also considered for this context, however, due to the size of the dataset, was ultimately abandoned due to insufficient 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.

**4.6 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.

**4.7 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.

1. **Results**

**5.1 Data Balancing**

| **Data** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| --- | --- | --- | --- | --- | --- |
| Imbalanced, raw | 0.92 | 0.55 | 0.86 | 0.046875 | 209.79 |
| Undersampling | 0.86 | 0.54 | 0.79 | 0.015625 | 209.77 |
| Oversampling | 0.91 | 0.53 | 0.86 | 0.078125 | 209.77 |
| SMOTE | 0.92 | 0.55 | 0.86 | 0.109375 | 210.83 |

Figure 1. Results of different data balancing techniques on the data.

Figure 1 displays the results of the data balancing/ sampling methods used. We see that the only significant result is that undersampling using imblearn’s RandomUnderSampler library appears to perform more poorly with lower abilities to predict the target class in terms of negative target class and overall accuracy. Another concern was the efficiency impacts of using a more intensive technique such as SMOTE, however the increased time and memory costs of SMOTE are minimal. 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.

**5.2 Feature Selection**

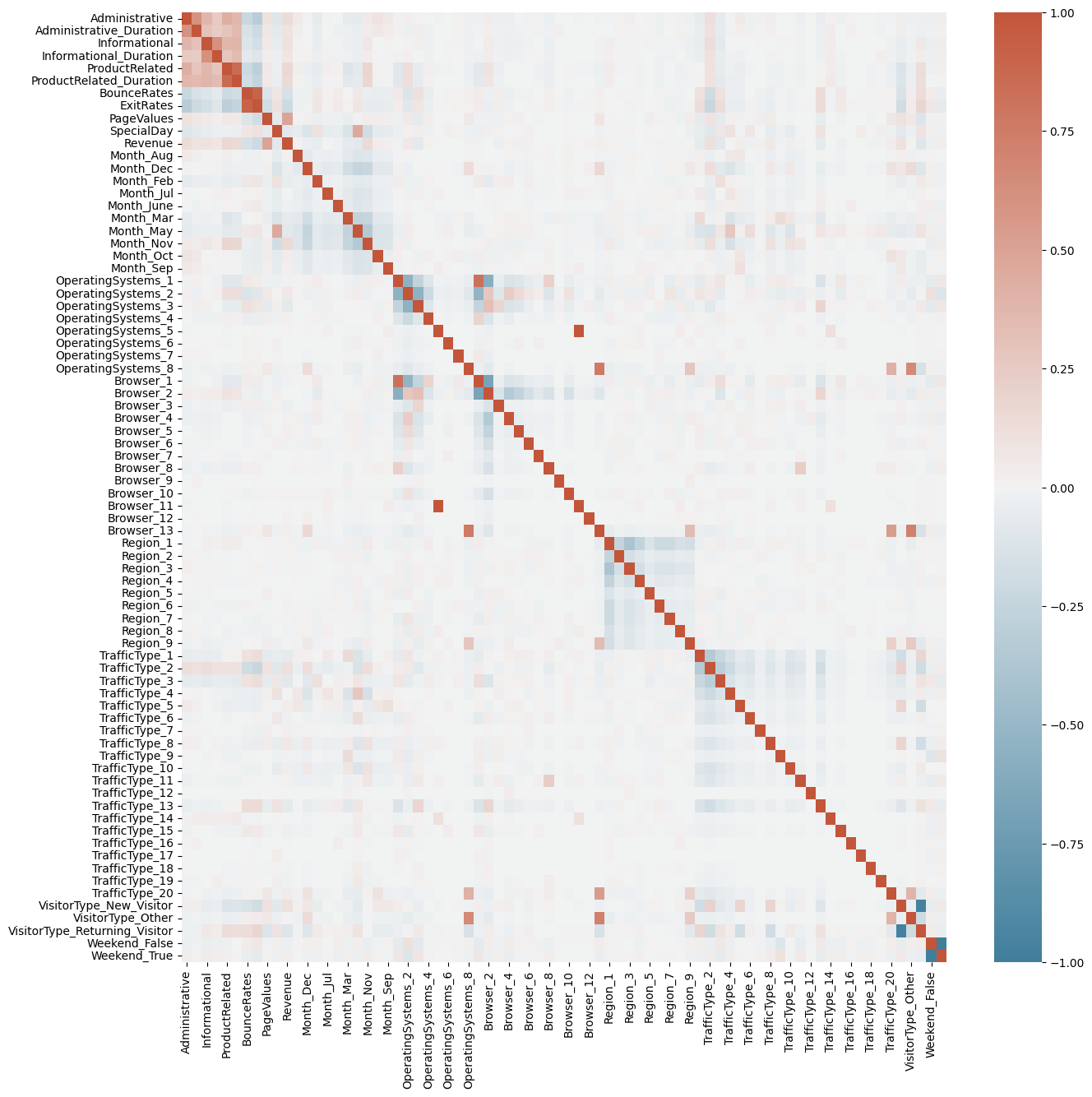


Figure 2. Pearson’s correlation matrix on the one-hot encoded dataset

Pearson’s correlation, Random Forest, and Recursive Feature Elimination (RFE) techniques were applied to filter for features. The results of the Pearson’s correlation are shown below in the correlation matrix (Fig 2.) Additionally, the rankings and results for the top 24 features filtered using each method are shown in Fig 3. Keeping the top quantile of features by their score from their respective feature selection methods (correlation, importance or rank) led to keeping the top 18 features using Pearson’s correlation, the top 19 features using Random Forest, and keeping the top 20 features using Recursive Feature Elimination. Upon examining the features kept, there are similarities between those kept by Pearson’s Correlation and Random Forest, with attributes PageValues, ExitRates, and ProductRelated showing strong contribution to correctly predicting the Revenue attribute. Notably RFE fails to keep many of the numeric attributes from the original dataset, with categorical attributes that have been one-hot encoded comprising most of the high-ranking attributes.

Model performance results show that on average, Random Forest model appears to perform the best in terms of accuracy and F1 score for both positive and negative target classes, albeit with greater memory and time efficiency costs. When examining the different feature selection filters, the only notable finding is that random feature elimination appears to perform much more poorly than other selection methods and even the control group. However, there is no distinction between the control data, Pearson’s correlation’s features and Random Forest’s selected features that would indicate a particular choice for feature selection method. Going forward, the Random Forest model will be used to test on control data, as well as data filtered by Pearson’s correlation and Random Forest features.

| **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 |

Figure 3. Ranking of features after feature selection filters. Retained features have been highlighted.

| **Decision Tree Model** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Feature Selection Method** | **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** | | | | | |
| **Feature Selection Method** | **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** | | | | | |
| **Feature Selection Method** | **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 |

Figure 4. Model performance results of different feature selection filters on SMOTE-balanced dataset.

**5.3 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 (no scaling)** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.64 | 0.90 | 1.28125 | 242.79 |
| Pearson Correlation | 0.93 | 0.69 | 0.89 | 1.59375 | 235.62 |
| Random Forest | 0.94 | 0.69 | 0.90 | 1.453125 | 239.36 |
| **Normalized Data** | | | | | |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.66 | 0.90 | 1.3125 | 271.36 |
| Pearson Correlation | 0.93 | 0.69 | 0.89 | 1.571825 | 266.69 |
| Random Forest | 0.94 | 0.68 | 0.90 | 1.484375 | 265.59 |
| **Standardized Data** | | | | | |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.66 | 0.90 | 1.328125 | 294.35 |
| Pearson Correlation | 0.93 | 0.69 | 0.89 | 1.59375 | 289.21 |
| Random Forest | 0.94 | 0.69 | 0.90 | 1.484375 | 288.36 |

Figure 5. Model performance results of different data scaling methods on SMOTE-balanced dataset using the Random Forest prediction model.

**5.4 Cross Validation**

Applying cross-validation techniques mainly improved model performance minorly in aspects of Negative Class F1-score. Notably, the recall of models increased when applying both K-fold and Time-Series cross-validation. However, the data without cross-validation applied still managed to perform well with considerable advantages in the Positive Class F1-score. Additionally, the time it took for the models to process increased significantly with cross-validation techniques applied.

To tune the models moving forward, focus will be placed on the control data with Pearson’s correlation features, the K-Fold cross-validated data with Pearson’s correlation features, and the Time-Series cross-validated data with Pearson’s correlation features.

| **Control Data (no cross-validation)** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.66 | 0.90 | 2.265625 | 290.05 |
| Pearson Correlation | 0.93 | 0.67 | 0.90 | 2.390625 | 273.88 |
| Random Forest | 0.93 | 0.65 | 0.89 | 2.40625 | 273.62 |
| **K-Fold Cross Validation** | | | | | |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.63 | 0.90 | 9.40625 | 280.73 |
| Pearson Correlation | 0.94 | 0.64 | 0.90 | 6.015625 | 256.45 |
| Random Forest | 0.94 | 0.61 | 0.90 | 9.84375 | 265.67 |
| **Time-Series Cross Validation** | | | | | |
| **Features**  **(Filter Method)** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| Control | 0.94 | 0.57 | 0.90 | 6.25 | 283.68 |
| Pearson Correlation | 0.94 | 0.65 | 0.90 | 6.5 | 263.99 |
| Random Forest | 0.94 | 0.62 | 0.90 | 6.625 | 263.36 |

Figure 6. Model performance results of different cross-validation methods on SMOTE-balanced and standardized dataset using the Random Forest prediction model.

**5.5 Model Tuning and Metrics**

Model Tuning

Adjusting the number of estimators for the Random Forest Model showed no gains in the model’s performance, however it led to greatly increased time and memory usage costs (Fig. 7).

Adjusting the number of folds involved in cross-validation showed no gains in the model’s performance, however it led to greatly increased time and memory usage costs (Fig. 8).

| **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 | 0.67 | 0.89 | 2.453125 | 235.53 |
| 500 | 0.93 | 0.68 | 0.89 | 10.265625 | 1494.98 |
| 1000 | 0.93 | 0.67 | 0.89 | 16.625 | 396.10 |
| **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 | 0.61 | 0.90 | 10.203125 | 1291.79 |
| 500 | 0.94 | 0.62 | 0.90 | 36.625 | 1332.65 |
| 1000 | 0.94 | 0.62 | 0.90 | 69.03125 | 1380.50 |
| **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 | 0.60 | 0.90 | 7.359375 | 1302.00 |
| 500 | 0.94 | 0.60 | 0.90 | 22.8125 | 1367.62 |
| 1000 | 0.94 | 0.60 | 0.90 | 44.0625 | 1414.58 |

Figure 7. Model performance results of number of estimator settings on SMOTE-balanced and standardized dataset using the Random Forest prediction model.

| **Pearson Correlation Filter/ K-Fold Cross Validation** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Number of Folds** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| 5 | 0.94 | 0.61 | 0.90 | 10.203125 | 1291.79 |
| 10 | 0.94 | 0.61 | 0.90 | 23.953125 | 1328.65 |
| 50 | 0.94 | 0.61 | 0.90 | 169.09375 | 1333.60 |
| **Pearson Correlation Filter/ Time-Series Cross Validation** | | | | | |
| **Number of Folds** | **Negative Class F1-Score** | **Positive Class F1-Score** | **Accuracy** | **Time**  **(s)** | **Memory**  **(MiB)** |
| 5 | 0.94 | 0.60 | 0.90 | 7.359375 | 1302.00 |
| 10 | 0.94 | 0.60 | 0.90 | 13.703125 | 1325.43 |
| 50 | 0.94 | 0.60 | 0.90 | 69.0625 | 1323.23 |

Figure 8. Model performance results of number of folds settings on SMOTE-balanced and standardized dataset using the Random Forest prediction model.

Complete Metrics

Brier’s Score and Matthew’s correlation were added as metrics alongside the previous metrics (Fig. 9). It is seen that the model without cross-validation techniques has a higher recall value for the positive class, as well as a higher Matthew’s correlation score. The higher recall indicates that it is more likely for the model to correctly identify transactions that are likely to lead to a purchase, minimizing false negatives, an important consideration to have when looking to predict a successful transaction. Additionally, the model without cross-validation techniques has considerably lower costs in time and computing resources.

Conversely, the models using cross-validation have a higher precision score for the positive class, and higher Brier Score values. The higher precision indicates that the model is likely to minimize false positive classifications and has a lower chance of recommending transactions that do not lead to actual purchases.

| **Metric** | **Pearson corr. features**  **No cross-validation** | **Pearson corr. features**  **K-fold CV** | **Pearson corr. features**  **Time-series CV** |
| --- | --- | --- | --- |
| **Negative Precision** | 0.95 | 0.92 | 0.92 |
| **Negative Recall** | 0.92 | 0.96 | 0.96 |
| **Negative F1-Score** | 0.94 | 0.94 | 0.94 |
| **Positive Precision** | 0.62 | 0.73 | 0.73 |
| **Positive Recall** | 0.72 | 0.56 | 0.56 |
| **Process F1-Score** | 0.67 | 0.63 | 0.64 |
| **Accuracy** | 0.89 | 0.90 | 0.90 |
| **Matthew’s Correlation** | 0.6054 | 0.0586 | 0.5877 |
| **Brier’s Score** | 0.1084 | 0.0978 | 0.0979 |
| **Time (S)** | 2.515625 | 9.6875 | 6.40625 |
| **Memory (MiB)** | 235.11 | 244.31 | 236.40 |

Figure 9. Metrics of each model

Model Variance

Variance of the models were calculated over 5 runs of each model (Fig. 10). The models demonstrated low variance among all metrics, with the highest variance metric being the time cost of each model. This demonstrates a relative high amount of stability among all models.

| **Variance of Metric** | **Pearson corr. features**  **No cross-validation** | **Pearson corr. features**  **K-fold CV** | **Pearson corr. features**  **Time-series CV** |
| --- | --- | --- | --- |
| **Negative Precision** | 0 | 1.23 e-32 | 1.60 e-05 |
| **Negative Recall** | 1.23 e-32 | 2.40 e-05 | 2.40 e-05 |
| **Negative F1-Score** | 1.59 e-05 | 1.23 e-32 | 1.23 e-32 |
| **Positive Precision** | 1.60 e-05 | 0 | 2.40 e-05 |
| **Positive Recall** | 2.40 e-05 | 2.40 e-05 | 2.40 e-05 |
| **Process F1-Score** | 1.60 e-05 | 2.40 e-05 | 2.40 e-05 |
| **Accuracy** | 0 | 0 | 0 |
| **Matthew’s Correlation** | 4.44 e-06 | 7.62 e-06 | 1.86 e-05 |
| **Brier’s Score** | 5.38 e-07 | 2.76 e-07 | 9.39 e-07 |
| **Time (S)** | 0.008 | 0.046 | 0.011 |

Figure 10. Variance of metrics of each model collected over 5 runs of each model.

**5.6 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.

When the datasets transformed by PCA were entered into the Random Forest model, the metrics showed poorer performance than the original dataset filtered by Pearson’s correlation feature selection. It was also shown to cost more in terms of time and memory resources. It was then determined that this dataset was not ideal for PCA.

A hybrid approach was also attempted, where PCA was applied on to the 57 features excluded after Pearson’s correlation feature selection. The goal was to retain some information kept from the features that would otherwise be excluded. The PCA was run with 57 principal components, which led to results comparable to the original Pearson’s correlation filtered data. A notable difference is an increase in precision of detecting the positive class, while there was a decrease in recall of the positive class and overall positive F1-score. While the hybrid approach demonstrated an ability to reliably predict the positive target class and successful transactions, it did not present any improvement over the Pearson’s correlated data while having increased computing costs.

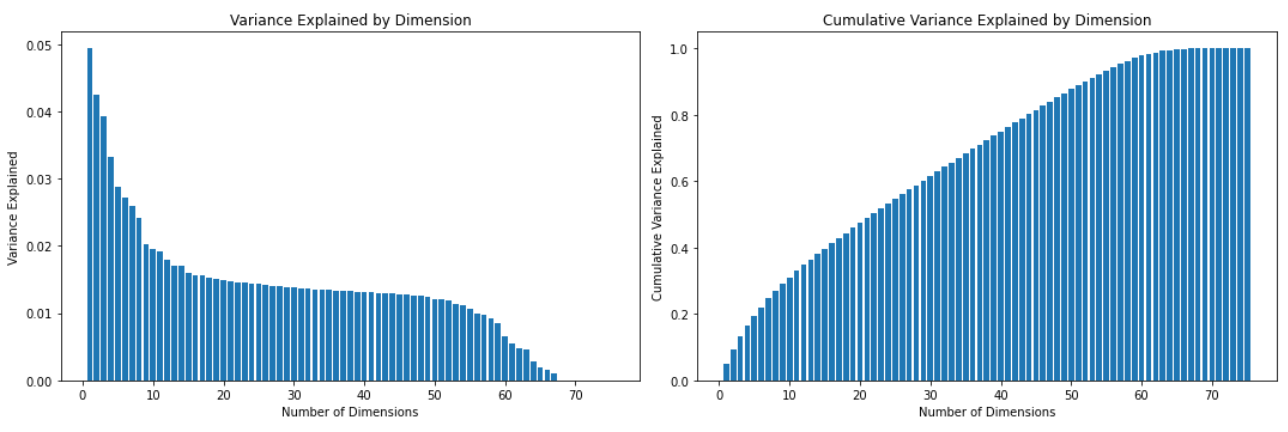


Figure 11. The variance explained by each principal component when applying PCA (left). The cumulative variance of added principal components (right).

| **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 |

Figure 9. Metrics of each PCA iteration with Pearson’s correlation data with Random Forest model as a baseline

1. **Conclusion**

In the end, a Random Forest prediction model was selected based on standardized data that was oversampled using SMOTE and filtered by top quantile features selected with Pearson's correlation. This model demonstrated a reliable prediction of the positive class with an accuracy close to 90%. Moreover, it exhibited a high recall value for the positive class, signifying its strong ability to identify transactions likely to lead to a purchase. Although some other models showed better performance in specific aspects, the chosen model was preferred for its overall effectiveness and reduced time and computing resource requirements.

Ultimately, it was shown that a prediction model low in computing cost and training time was able to be trained and predict user purchasing behavior with high accuracy. Focusing on the data filtered by feature selection can provide insight to which pages a user would interact with most, and the data assigned to users likely to commit to transactions. As such, this can lead to many applications in website design and for ecommerce marketing with the goal of increasing online sales as a better understanding of online shopping consumers has been obtained.

1. **References**

* Baati, K., & Mohsil, M. (2020, May 6). *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. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256375/
* 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