DSCI 310 Group 8: Online Shopper

Calvin Choi, Nour Abdelfattah, Sai Pusuluri, Sana Shams

Table of contents

Summary

E-commerce pages host several customers at any given moment, yet its metric of success lies in the visitors who ultimately make purchase. This project uses several machine learning models to learn from webpage data and customer browsing behaviour in order to predict whether or not a given customer will finalize their purchase.

Introduction

It has been no surprise that retail giants like Walmart and Ikea have aggressively invested and developed their e-commerce experiences transitioning away from big box store fronts and converting those assets to hubs for location-based fulfillment Monteros (2023). The post-pandemic affects on consumer behaviour have accelerated our dependency on digital platforms and have pushed the e-commerce industry to grow a whopping 25% to an industry worth over $4 trillion USD Shaw, Eschenbrenner, and Baier (2022). Consequently, online storefronts get a lot of site traffic but what ultimately matters is their decision to purchase and the volume of revenue. Marketing and User Experience teams are tasked with optimizing a site's interface and content in order to improve customer retention and the site's revenue. Given this, understanding customer browsing behaviour and web page features is crucial for not only improving the user's experience, but also maximizing the retailer's revenue. Traditionally marketing and user experience studies are conducted through surveys, interviews and ethnographic studies, taking weeks up to months to process. However, machine learning-based marketing research has exponentially reduced the rate at which web metrics and purchase conversion strategies can be processed, while significantly increasing purchase prediction accuracy Gkikas and Theodoridis (2022). A common method to evaluate user retention for online web browsing is through clickstream data of the user's navigation path, however Saka et al. found that combining this information with session information significantly improves the purchase success rate Sakar et al. (2018).

This project aims to analyze various features of online shopper's sessions on a site to predict whether the customer makes a purchase. We will use the dataset, Online Shoppers Purchasing Intention dataset from the UCI Machine Learning Repository. This dataset was chosen specifically due to its coverage of both user navigation data and session information, allowing for a well-rounded analysis of both the user and e-commerce page's profile.

Question

Using all variables provided in the dataset, which group of explanatory variables form the best prediction for the user's purchase intent?

Exploratory Data Analysis

Before starting our analysis, we will perform exploratory data analysis in order to have a better understanding of the distributions of the features in our dataset, as well as their contribution to our target feature, Revenue.

The dataset provides the following features:

Summary of features

- **Administrative**: the number of pages visited by user of this administrative type
- **Administrative_Duration**: the amount of time spent on pages of this administrative type
- **Informational**: the number of pages visited by user of this informational type
- **Informational_Duration**: the amount of time spent on this informational category of pages
- **ProductRelated**: the number of pages of this type of product the user visited
- **ProductRelated_Duration**: the amount of time spent on pages featuring related products
- **BounceRates**: percentage of visitors who enter the web page then leave ("bounce") without triggering any other requests to the analytics server during the session
- **ExitRates**: the percentage of pageviews where the given page is the last page before exiting website
- **PageValues**: the average value for a web page that a user visited before completing an e-commerce transaction
- **SpecialDay**: the temporal proximity between the day the user is visiting the page and a special day (eg. Valentines Day, Christmas, Mother's Day, etc.).
- **Month**: the month the page was viewed
- **OperatingSystems**: an integer value representing the operating system of the user when viewing the page
- **Browser**: an integer value representing the user's browser when viewing the page

- **Region**: an integer value representing the user's traffic type. Learn more about user traffic types here
- **VisitorType**: categorizes the user into 'New Visitor', 'Returning Visitor', and 'Other'.
- **Weekend**: a boolean value, indicating whether the user's session took place during a weekend or not
- **Revenue**: a boolean value, indicating whether the user made the purchase or not
  - **This will be our target feature**

Exploratory Visualization

To inform the chosen method of visualization, let us first document if the features are continuous values, or if they are discrete categorical values. Some features are categorical but represented as integers so this step will allow for clarification.

Table 1: Data Types Summary

Feature	Type
Administrative	numerical, continuous
Administrative_Duration	numerical, continuous
Informational	numerical, continuous
Informational_Duration	numerical, continuous
ProductRelated	numerical, continuous
ProductRelated_Duration	numerical, continuous
BounceRates	numerical, continuous
ExitRates	numerical, continuous
PageValues	numerical, continuous
SpecialDay	numerical, continuous
Month	categorical, discrete
Browser	categorical, discrete
Region	categorical, discrete
TrafficType	categorical, discrete
VisitorType	categorical, discrete
Weekend	categorical, discrete, boolean
Revenue	categorical, discrete, boolean

The summary of each datatype is provided in Table **??**

Given that Revenue is our target feature, let us examine its distribution in Figure **??**.

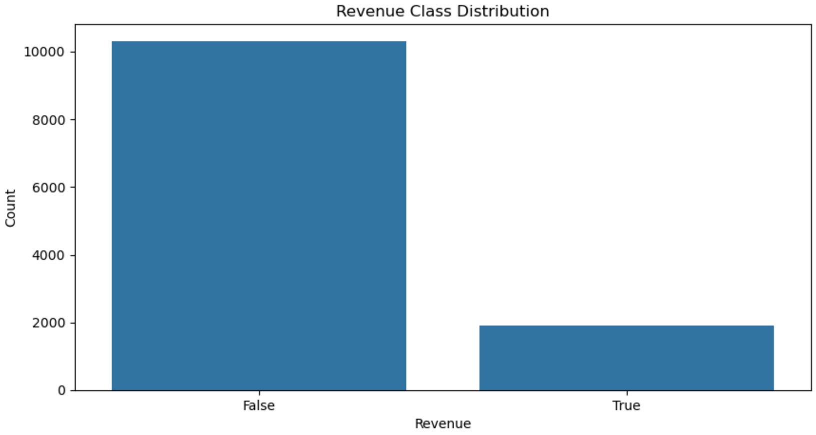

Figure 1: Revenue Class Distribution

Figure 2 shows that there does seem to be some class imbalance in the Revenue feature. This might create bias in our models that may perform poorly on the 'True' Revenue cases as there were less examples to fit on.

To compare Revenue with other features, we will perform some feature engineering by creating the feature, Total Revenue, for each of the categorical revenues explored below.

Categorical Features: Examining the Distributions

Let us examine the distribution of certain categorical feature to better understand the user demographic.The distributions are:

- distribution of classes within the categorical feature
- distribution of Revenue=True across the different classes for a given feature

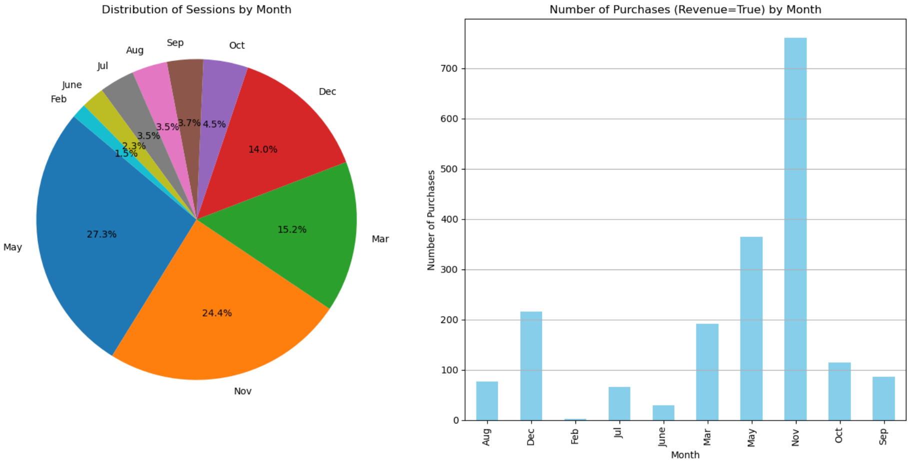

Figure 2: Distribution of Sessions by Month

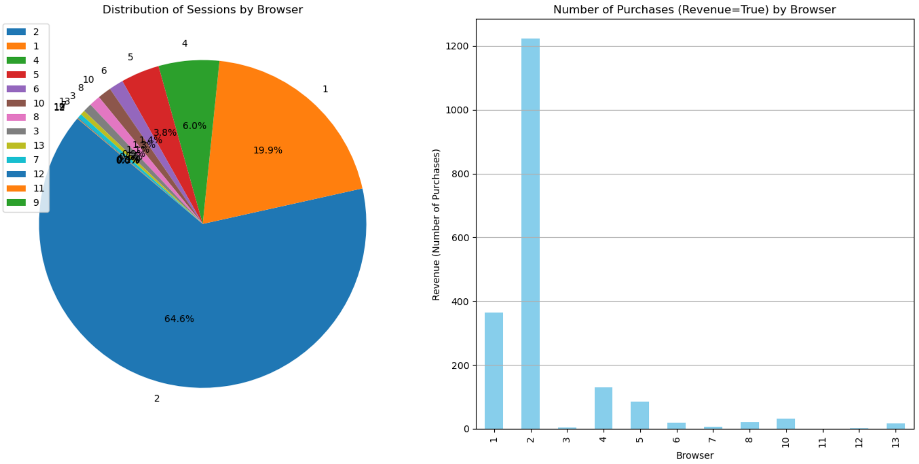

Figure 3: Distribution of Sessions by Browser

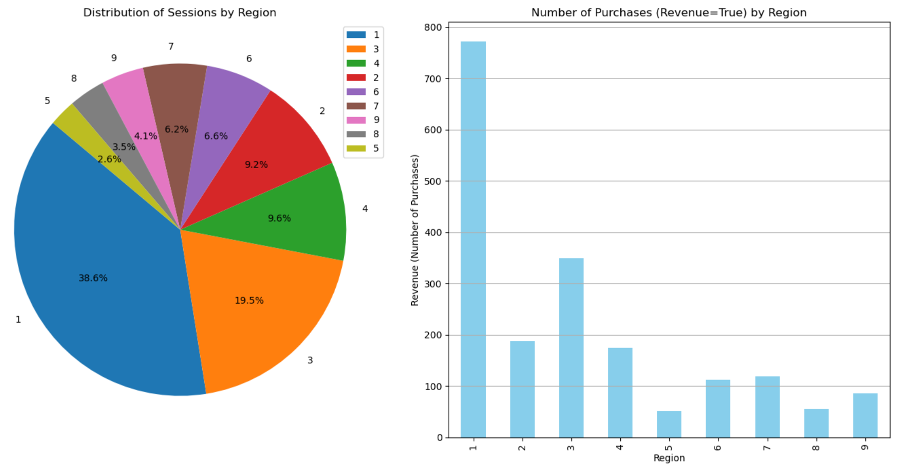

Figure 4: Distribution of Sessions by Region

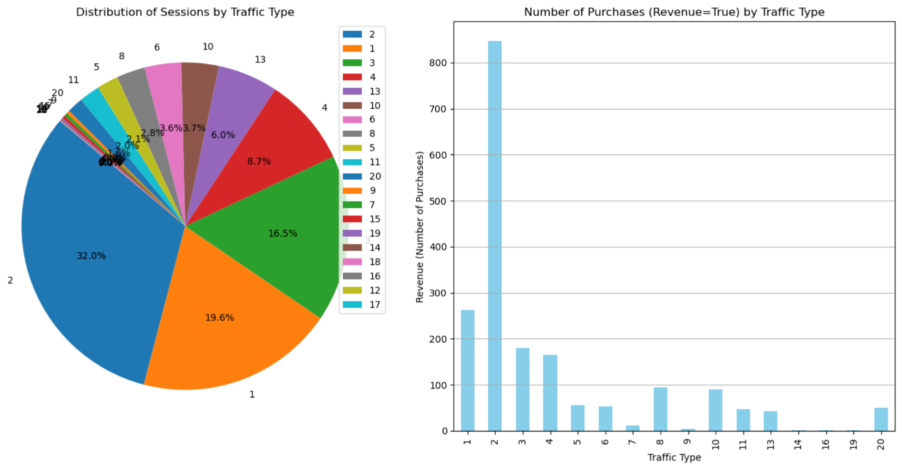

Figure 5: Distribution of Sessions by Traffic Type

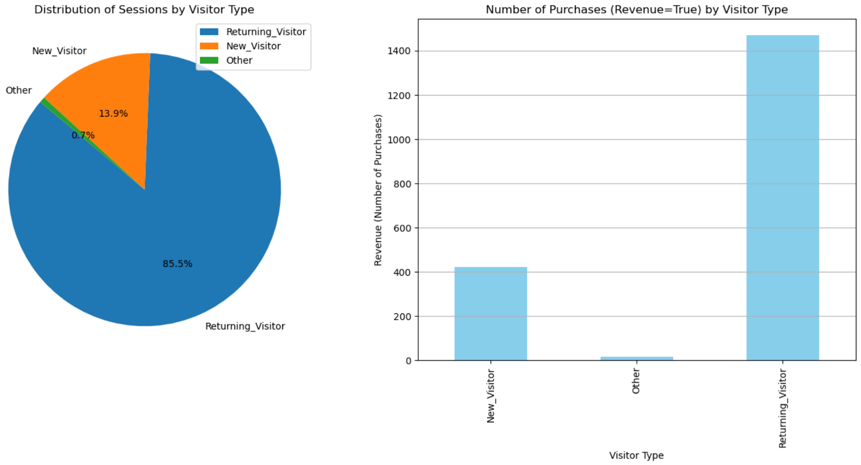

Figure 6: Distribution of Sessions by Visitor Type

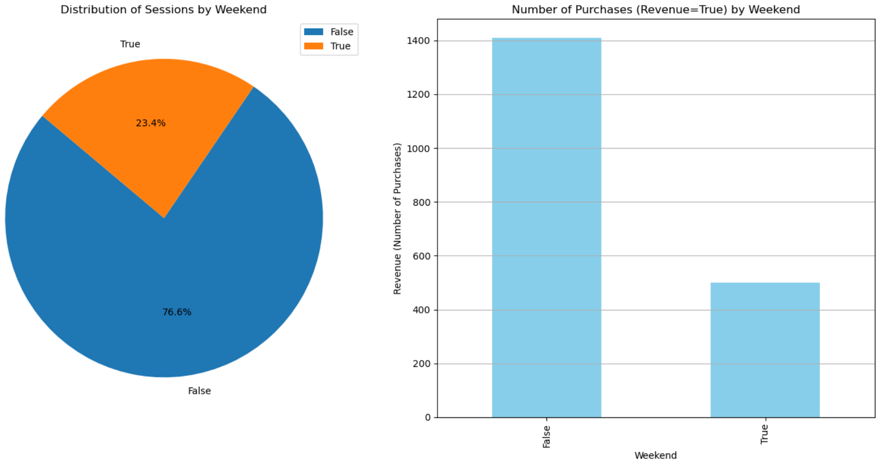

Figure 7: Distribution of Sessions by Weekend

Continuous numerical Features: Correlation with Revenue

The remaining features are continuous numerical features. In order to understand their significance to the target feature, Revenue, we will make a correlation plot. To do this, we will create a copy of the original data, and modify it so that:

- Only the numerical features are kept
- Revenue is represented as a numerical feature so that it can be compared with the other numerical features

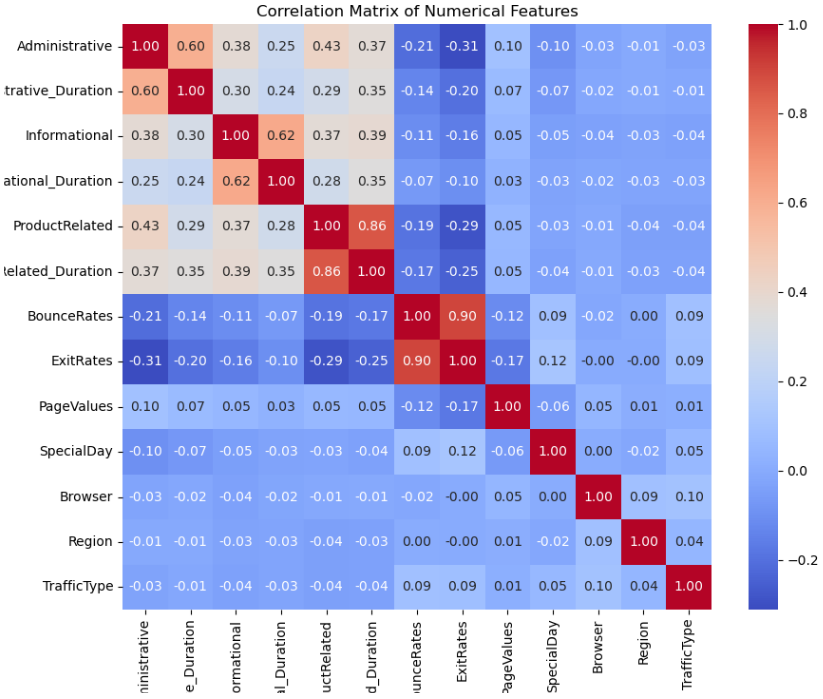

Figure 8: Correlation Matrix of Numerical Features

From Figure ??, we can see that the features most strongly correlated with Revenue are: PageValues, ProductRelated, and ProductRelated_Duration. It is important to note that the correlation matrix only represents linear correlations, between *pairs* of features. Some correlations may be confounded with other features, so while this matrix is a good starting point, it may not capture all relevant relationships.

Methods

Analysis Plan

1. Train-Test Split

Before applying any transformations or conducting any analysis, we will first create a 70-30 split of our data:

- 70% split for the training subset
- 30% split for the testing subset

All training of the models will be strictly conducted on the training set. The testing set will only be used once the model is finalized, and will only be deployed for scoring on the testing set once. This is to ensure that the model is not exposed to the testing set so that it does not 'learn' off of what it is trying to predict.

2. Preprocessing and Transformations

Given that we have different data types, we will apply some transformations to each feature type depending on if the feature is numerical and continuous, discrete and categorical, binary, etc. The transformations will be detailed in Figure[INSERT LATER].

3. Training Models

Our target feature, Revenue, is binary so the models chosen will be trained to perform binary classification:

3.1 Dummy Classifier Model: The Dummy Classifier Model makes predictions that ignore the input features, in other words, it does not attempt to 'learn' anything from the data. This classifier serves as a baseline to compare to the following models.

3.2 kNN:

kNN is a simple cluster-based model. Given k, the number of nearest data points, the kNN classifier takes a data point and classifies it according to the the class of its k-nearest neighbours.

3.3 SVM RBF:

Support Vector Machines with RBF Kernels act as weighted KNN's. Unlike KNN's, this model bases its decision boundary only on key examples, known as support vectors. The model transforms the input features into a higher dimensional space, generating a decision boundary based on a set of positive and negative examples and their weights along with their similarity measure. This model uses a kernel called RBFs as the similarity metric.

3.4 Random Forest Classifier:

A Random Forest Classifier fits a series of decision tree classifiers on subsets of the given data. Each tree 'overfits' on a select feature, however the model uses averaging of individual trees to improve the predictive accuracy and therefore prevent overfitting. Given that there are many features in our dataset, this model is a strong candidate for our classification problem.

Evaluating the models

Each model will be evaluated on the following:

- fit time
- score time
- test score (this is the validation score, not the score from the actual test subset)
- train score

The model with the best (validation) test score will then be deployed **once** on the test data to obtain a final test score.

Analysis

1. Train Test Split

Note:

Before splitting the data, the feature Revenue will be transformed at this step as it needs to be transformed to a numerical format using one-hot encoding. This is done at this step because it will be removed from the X_train and X_test sets.

2. Preprocessing and Transformations

Table 2: Data Preprocessing Summary

Feature	**Transformation**	**Explanation**
Administrative	scaling	standardize scale with other numerical features
Administrative_Duration	scaling	standardize scale with other numerical features
Informational	scaling	standardize scale with other numerical features
Informational_Duration	scaling	standardize scale with other numerical features
ProductRelated	scaling	standardize scale with other numerical features
ProductRelated_Duration	scaling	standardize scale with other numerical features
BounceRates	scaling	standardize scale with other numerical features

Feature	**Transformation**	**Explanation**
ExitRates	scaling	standardize scale with other numerical features
PageValues	scaling	standardize scale with other numerical features
SpecialDay	scaling	standardize scale with other numerical features
Month	one-hot encoding	categorical feature, need a numerical representation to pass through models
OperatingSystems	drop	justified in EDA, not relevant
Browser	n/a	would apply one-hot encoding but already represented in numerical form
Region	n/a	would apply one-hot encoding but already represented in numerical form
TrafficType	n/a	would apply one-hot encoding but already represented in numerical form
VisitorType	one-hot encoding	categorical feature, need a numerical representation to pass through models
Weekend	one-hot encoding with 'binary=True'	categorical feature, need a numerical representation to pass through models

The summary of preprocessing transformations and the explanation behind each is provided in Table **??**.

Training Models

For this analysis, we defined a custom function that returns the mean and standard deviation cross validation scores. For each model, a pipeline was defined to first preprocess the training split of the data, then pass it through the model to be fit. The documentation of the function is as follows:

Parameters * model :scikit-learn model * X_train : numpy array or pandas DataFrame * X in the training data * y_train : * y in the training data

Returns * pandas Series with mean scores from cross_validation

Note: this function definition is taken from CPSC330 2023W1 Course Notes

Moreover, the results of each of the models described in the Analysis Plan section is stored in a table to facilitate comparison of their fit time, score time, test score, and train scores – including the cross validation scores for each metric.

Model Results

	Unnamed: 0	fit_time	score_time	test_score	train_score
0	Dummy	0.006 (+/- 0.002)	0.004 (+/- 0.001)	0.847 (+/- 0.000)	0.847 (+/- 0.000)
1	KNN_best_k	0.016 (+/- 0.007)	1.324 (+/- 0.328)	0.877 (+/- 0.005)	0.880 (+/- 0.001)
2	SVM_RBF	8.417 (+/- 1.140)	2.303 (+/- 0.344)	0.887 (+/- 0.002)	0.889 (+/- 0.000)
3	Random_Forest	3.847 (+/- 0.409)	0.090 (+/- 0.029)	0.902 (+/- 0.004)	1.000 (+/- 0.000)

Model Selection

Among the many factors that data scientists consider for model selection, the two main factors we will consider are accuracy and efficiency.

Efficiency

Time efficiency and computational complexity are key factors to model selection. We want models that strike the best balance between getting the results we want and doing it in an efficient manner. KNN is known for its computational intensity, although not reflected in the fit times above, we can be concerned how it might performed on a larger scale. Similarly, we know that the SVM model struggles with computational efficiency and that is what our results reflected in the the fit time. The random forests model came out on top of the efficiency tables with the quickest fit time and scoring time while outperforming all other models in terms of score as well.

Performance

In assessing the accuracy of machine learning models, particularly in the context of comparing Random Forests, kNN, and SVM, it's essential to evaluate various metrics and considerations that provide an in-depth evaluation truly reflective of performance. Each of these models offers distinct approaches to classification and regression tasks, leading to variations in performance across different datasets and problem domains. In terms of our results, the clear winner is the random forest model.

Evaluating True Positive Rate, Precision Score

The accuracy score at face value seems to be good, however if we break down further our results by looking at our true positive rate (precision score), we can see that our model actually struggled and because of the class imbalance the effect of this on our score was not reflected in our accuracy. This analysis helps us tame our optimism and helps us insights into how well our model correctly identifies positive instances within the dataset, which is particularly crucial in scenarios with imbalanced classes. Understanding the nuances of class imbalances allows us to make informed decisions about model adjustments, such as implementing techniques like oversampling, undersampling, or adjusting class weights, to improve the model's predictive capability for minority classes. Moreover, it underscores the importance of considering multiple evaluation metrics beyond just accuracy, enabling a more comprehensive assessment of model effectiveness and guiding future iterations or refinements to enhance predictive performance across all classes.

Figure **??** illustrates the distribution of the true positives, false positives, true negatives, false negatives for the Random Forests Model. This will be used to calculate the precision score of the model.

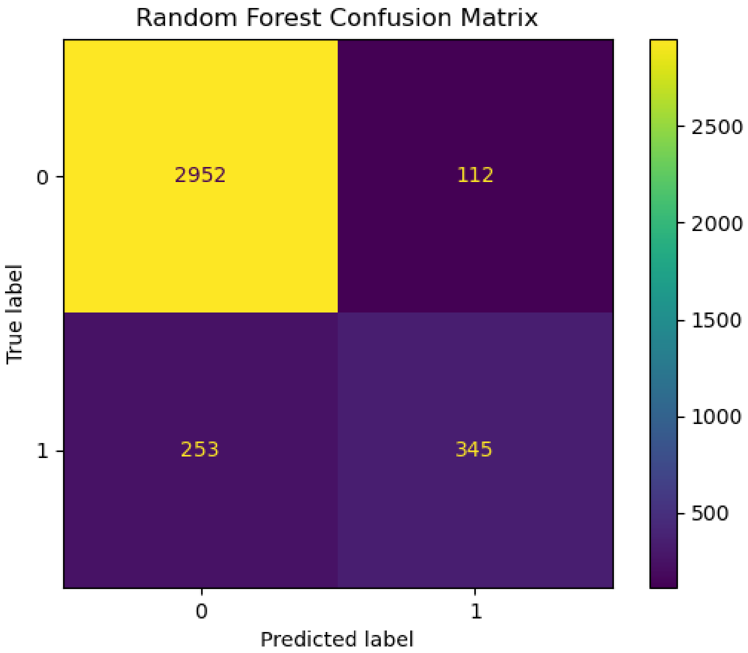

Figure 9: Random Forests Confusion Matrix

To obtain the precision score of our model, the following formula was used:

$$s = TP/(TP + FP)$$

Given this formula, the precision score is: Precision = 0.5585.

Conclusion

Our findings can be summed up into 2 main points:

When it comes to the application of ML to predict online consumer behaviour, the random forests model performs the best and produced the best results.

Random forests was both efficient and accurate. The results provided a very simple solution without much to be further analyzed.

Margin For Error

Our results are purely based on the models that we chose to test, and therefore, may not be the absolute best off the shelf models for the given project but in favour of simplicity and the essence of time, among the 3 tested models (KNN, SVM and RF), we've concluded that the random forests model performed the best.

While it was the best-performing model, the final model's precision score of 56% indicates that the model only predicts the positive class correctly roughly half the time. This could be indicative that the model was quite overfitted on the training model, or that despite the randomization of our train-test split, the training sample was not a good representation of the test split and perhaps the true population of online shoppers sampled in our dataset. Likely, the low precision score is highly influenced by the fact that there was class imbalance in our dataset, as demonstrated in Figure **??**, which shows that there are over 4 times more false cases than true for the feature Revenue. As a consequence, the models, no matter how well they perform on the training dataset, are bound to better learn the predictive features for a false Revenue case than a true revenue case simply because of the class distribution in the original dataset.

For future analysis, perhaps the class imbalance may be addressed by selectively sampling to have fair distribution of both Revenue classes, rather than pure random sampling as was done in our analysis.

References

Gkikas, Dimitris C., and Prokopis K. Theodoridis. 2022. "AI in Consumer Behavior." In *Advances in Artificial Intelligence-Based Technologies: Selected Papers in Honour of Professor Nikolaos g. Bourbakis—Vol. 1*, edited by Maria Virvou, George A. Tsihrintzis, Lefteri H. Tsoukalas, and Lakhmi C. Jain, 147–76. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-80571-5_10.

Monteros, Maria. 2023. "Big-Box Retailers Continue to Ramp up Investments in Store-Based Fulfillment." *Modern Retail*. https://www.modernretail.co/operations/big-box-retailers-continue-to-ramp-up-investments-in-store-based-fulfillment/.

Sakar, C. Okan, S. Olcay Polat, Mete Katircioglu, and Yomi Kastro. 2018. "Real-Time Prediction of Online Shoppers' Purchasing Intention Using Multilayer Perceptron and LSTM Recurrent Neural Networks." *Neural Computing and Applications* 31 (10): 6893–6908. https://doi.org/10.1007/s00521-018-3523-0.

Shaw, Norman, Brenda Eschenbrenner, and Daniel Baier. 2022. "Online Shopping Continuance After COVID-19: A Comparison of Canada, Germany and the United States." *Journal of Retailing and Consumer Services* 69 (November): 103100. https://doi.org/10.1016/j.jretconser.2022.103100.