University of Michigan - SIADS 696 - Milestone II
Analyzing E-Commerce Customer Behavior Using Website Analytics
June 2023

Greg Holden (gsholden@umich.edu), Emmanuel Sengendo (esenge@umich.edu)

Section 1: Introduction

The primary objective of our project is to analyze the behavior of visitors to the website of an online electronics retailer. Traditional offline retailers have long analyzed customer behavior in the hope of identifying purchasing patterns and developing customer personas. These patterns allow for more effective marketing strategies to be developed that are ultimately meant to improve sales and profitability. However, e-commerce has become an integral part of the retail landscape. The prevalence of online shopping has created an opportunity for merchants to leverage enhanced analytics data generated during the online shopping experience. Harnessing this data provides retailers with a more nuanced perspective on customer behavior that can be used to incrementally enhance the effectiveness of marketing campaigns. Our goal is to combine traditional and more modern web-based customer metrics and analyze which are most impactful as it relates to product recommendation and customer segmentation.

Our motivation for conducting this analysis is a fascination with the seismic changes in the retail landscape and a desire to analyze just how impactful modern, web-based marketing metrics might be. We intend to do this using both supervised and unsupervised models. Supervised learning models will enable us to analyze the relationship between website behavior and purchases of certain product categories or brands using classification models. Unsupervised learning models will be used to cluster customers into groups based on both traditional and web-based customer metrics. Key characteristics of these groups can used to develop more targeted marketing strategies, or make more effective product recommendations.

Key findings from our analysis include: (1) classification models (particularly XGBoost) can be effectively used to predict whether a specific product or category is likely to be purchased, and different combinations of traditional and web-based marketing metrics are more important depending on the category/product (2) using the XGBoost classification model, we can gain valuable insights into customer purchases by leveraging time-based patterns in user session behavior and identifying crucial factors influencing purchase decisions, and (3) despite our reliance on anonymized customer data with no demographic or location information**,** unsupervised learning methods can still be used to effectively segment customers into groups based on purely on feature engineered traditional and web-based marketing metrics. These findings are promising for e-commerce merchants seeking to devise more effective marketing strategies.

Section 2: Related Work

There are numerous studies discussing the potential of ML in e-commerce. A few examples include:

#1 ”Towards Accurate Predictions of Customer Purchasing Patterns”: This reference highlights a time-based purchase analysis conducted on online retail customers of a UK company. Various algorithms were utilized to classify customers based on specific marketing-focused behaviors. Our approach was different as we employed a time-based analysis to analyze customer behavior.

#2 “Predicting Customer Class using Customer Lifetime Value with Random Forest Algorithm”: This work focused on the importance of customer relationship management (CRM) in the e-commerce industry. By utilizing Customer Lifetime Value (CLV), a predictive model is developed to classify individual customers and determine their predicted business value. Our approach was different as we employed a time-based analysis to analyze customer behavior.

#3 “eCommerce behavior using XGBoost”: This article uses an earlier, narrower version of our dataset to specifically address whether a customer will purchase a given item once it is in the shopping cart. This precedent includes a single website-based metric (number of pages viewed) and only the Frequency component of the RFM marketing framework. Our approach is differentiated given that we include

additional web event metrics, provide more complete RFM metrics, and address a different and more in-depth set of business questions.

Section 3: Data Source

Our dataset is based on web event transactions for an online retailer of electronics goods. A few highlights of the dataset are as follows:

- 885,000 web events, including page views, items added to the cart, and purchase transactions
- Events occurred between October 2019 and February 2020
- Over 700 different product categories. The most common categories include video cards, phones, printers, and audio equipment.
- Over 53,000 unique products
- ~1,000 different brands, included Asus, Gigabyte, Samsung, AMD, Canon and Panasonic
- Unique, anonymized identifiers for each customer, browsing session, and product id
- Data available for download in a .csv file

The dataset can be found here:

https://www.kaggle.com/datasets/mkechinov/ecommerce-events-history-in-electronics-store

Fortunately, most of the variables had relatively complete data - with the exception of the "product category code" and "brand" columns. Many of these missing values were related to browsing sessions rather than purchases. After eliminating these missing rows, we still had over 23,000 completed purchase transactions.

However, one critical limitation of our dataset is that the user data is anonymous. We are given a numeric 'user_id' but no demographic or geographic information for each customer. This limits our ability to be as granular as we would like in terms of clustering or product recommendations.

4. Feature Engineering

Our primary feature engineering efforts involved calculating a series of metrics for each purchase transaction. This included both traditional marketing metrics widely used outside of the e-commerce sphere, along with statistics specifically related to web-based transactions. Our objective was to use combinations of these metrics to estimate the impact on purchase behavior. Traditional metrics included (1) how recently the user made a purchase (2) how frequently the user made a purchase and (3) how much each customer spent on the website. Web-based metrics included (4) the length of each browsing session (5) the number of web pages each user visited during a browsing session, and (6) a time delta feature for each unique user session, who both viewed and purchased within the same hour (7) time based features capturing user behavior on the site including the hour, time, day of the week, and month.

The data set contains several key columns which were used to calculate these metrics. These included the "event_type" column (which characterizes each row as a "view", a "cart" placement, or a "purchase"), the "event_time" column, which needed to be converted to a pd_datetime format, a unique "user_session" id for each browsing session, and a unique "user_id" value for each anonymized customer.

Recency was calculated for each purchase transaction by comparing the purchase date to the latest date of the dataset (11/6/2020). **Frequency** was calculated using groupby on each "user_id" and then counting the number of "purchases" in the "event_type" column. **Monetary Value** was calculated using groupby on each "user_id" and then summing the occurrences in the "price" column. **Browsing Duration** was calculated using a groupby on each "user_session", then using an aggregate function (.agg) plus a lambda function to calculate the difference between the first and last "event_time". **Pages Visited** was calculated using a groupby on each "user_session", then counting the number of events in the "event_type" column, where each row represented a single page view. **Unique User_Session** was calculated as each user session who both viewed and purchased an item within the same hour.

Our other main feature engineering goal was to reduce the 53,000+ unique product and 700+ unique category codes (both columns of lengthy numerical values) to a more manageable number and convert this information into an interpretable form. Fortunately, there was also a text-based 'category_code' column that we could split into a high-level "category" grouping and a more specific "product" type. This was done by using str.split and breaking the string apart. We then treated the first word as the "category" and the last word as the "product". For instance, a string of "computers.components.cooler" would result in a category of "computers" and a product type of "cooler". This process reduced the number of products to 103 and the number of categories to 14. These newly-created product and category columns were then converted from categorical to numerical variables using pd.get_dummies for use in later classification experiments.

Once these marketing metrics were created, the database for the first supervised learning question had 9 original columns (event_time, event_type, product_id, category_id, category_code, brand, price, user_id, user_session), plus the first 5 marketing metrics described above (recency, frequency, moentary value, browsing duration and page views) and 117 new product/category dummy columns, for a total of 132 initial columns. The database for the second supervised learning question ultimately used a subset of 10 of the above variables (more details in the Supervised Learning section).

Part A: Supervised Learning

Description of Methods – First Business Question

The first question we sought to answer is: **"How effective are traditional and web-based metrics at predicting whether a customer will purchase a specific product or category of product?"**. Are more traditional metrics such as Recency and Frequency more influential? Or can web-based metrics also teach retailers how to effectively target prospective customers?

We treated this question as a classification task. Given that we had previously created 117 new 1/0 columns for each of our narrowed-down product and category groupings, we used a specific product/category as the dependent variable and used four marketing metrics (Recency, Frequency, Pages Visited and Browsing Duration) as the independent variables. We excluded Monetary Value (the cumulative value of all purchases by a customer) given its undue influence at the product level (computers/stationery are relatively expensive/inexpensive products, so including Monetary Value dominated other variables).

We also performed two other key steps before running our models. First, given the skewed distribution of the marketing variables (i.e. there are many customers who visit the website infrequently but few who visit many times) we first normalized the data. Second, we quickly recognized the imbalanced nature of a dataset where any given product or category represents a small percentage of total purchases. In order to strike a balance between 1s (purchases) and 0s (non-purchases), we used the SMOTE algorithm.

We then ran our analysis on seven different classification models and gauged relative performance. This included a dummy classifier, which served as a baseline, plus Logistic Regression, Naïve-Bayes, SVM, k-Nearest Neighbors, Random Forest and XGBoost. This diverse choice of models was intended to see which general family best captured the characteristics of our marketing metric data. Logistic Regression and Naïve-Bayes both rely on probabilistic approaches, although given the non-normalized nature of the marketing metrics we were skeptical that these models would perform well. We included kNN and SVM to see how well instance-based models would perform on a dataset with few features but a decent number (over 23,000) of observations. Finally, we included Random Forest and XGBoost to see which type of tree-based algorithm (bagging vs boosting) would best capture patterns and handle outliers. Hyperparameter tuning began with the default parameters and involved experimenting to improve performance.

Modeling Results

The results of our comparative modeling analysis are shown in Figure 1 below. This example uses Telephones (a.k.a. Smartphones) as the chosen product category, although many other categories and products produce directionally similar results. This analysis shows metrics for the top performing model in

each category described above. Given the imbalanced nature of the data and the importance of predicting the positive class (a purchase of a specific category or product), we would normally have focused on F1 Score. However, given the use of SMOTE to balance the dataset, we have also included Accuracy scores. These figures represent average scores assuming 10-fold stratified cross-validation.

Figure 1: Summary of Supervised Learning Evaluation Metrics

Model	Accuracy	Std. Dev.	F1	Std. Dev.	Test Set Accuracy
Dummy	0.503	0.020	0.503	0.020	0.512
Logistic Regression	0.542	0.009	0.539	0.010	0.479
kNN	0.897	0.006	0.897	0.006	0.850
XGBoost	0.931	0.065	0.930	0.070	0.922

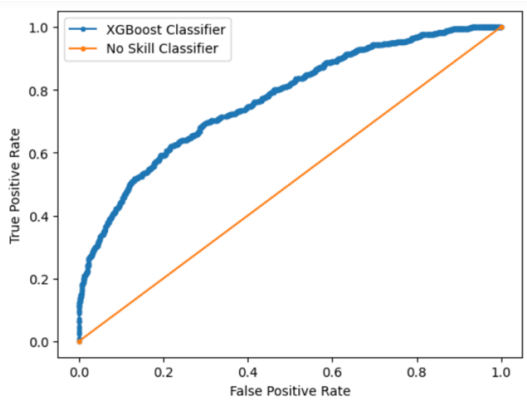
The results suggest that the XGBoost Classifier does the best job of using the marketing metrics to predict Smartphone purchases, but also offers insights on the other models. First, the performance of the dummy classifier at ~50% is in line with expectations based on the SMOTE-balanced dataset. Second, the relative performance of XGBoost is impressive – especially vs. Random Forest (our other tree-based candidate). Random Forest's Test Set Accuracy of 61% (see Appendix 1 for more detail) highlights the power of the boosting mechanism within XGBoost. Finally, it is worth noting that while XGBoost has the highest accuracy and F1 Scores, it also has the highest standard deviation by a wide margin.

Hyperparameter Tuning

Once we assessed relative performance, we conducted hyperparameter tuning on XGBoost (now without normalization) using RandomizedSearchCV. Final parameter values from this process included using 400 estimators, a learning rate of 0.2, max depth of 12, gamma of 0.4 and a minimum child weight of 5. See Appendix 2 for the more detail. Tuning increased both **Accuracy and F1 scores to ~0.938.**

The ROC-AUC plot in Figure 2 shows the effectiveness of our model (ROC-AUC score of 0.769)

Figure 2: XGBoost ROC-AUC Curve



XGBoost Classifier: ROC AUC=0.769
No Skill Classifier: ROC AUC=0.500

Feature Importance & Ablation Analysis

The SHAP summary plot (Figure 3) provides useful insights regarding feature importance. All four features carry meaningful weight, but the order changes depending on the chosen product or category. In the case of Smartphones, Browsing Duration and Page Views are the two most impactful features. Additional examples for different products/categories can be found in Appendix 3. In the case of Smartphones, high Page Views and Browsing Duration figures are intuitive, and suggest that shoppers spend significant time comparing phones given their high cost and importance in modern life. Interestingly, the SHAP summary plot also suggests that very high values of Browsing Duration (and Frequency) have a negative impact on the model. This is surprising but provides marketers with helpful guidance for developing a persona for a typical Smartphone shopper. It also suggests that the model benefits from the wider range of values/higher variance of Browsing Duration and Page View values relative to Recency and Frequency values.

We can also remove each feature from the XGBoost model and calculate the impact on the evaluation metrics (Figure 4). It is interesting that the relative impact differs slightly compared to the feature importance analysis, although individually removing the two "web-based" metrics still has a greater impact than removing the "traditional" metrics of Recency and Frequency.

Figure 3: SHAP Feature Importance

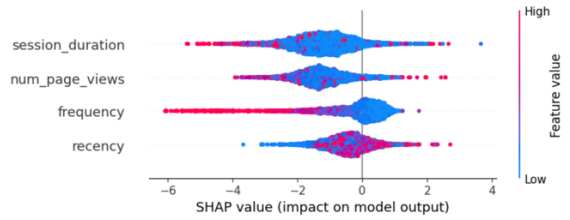


Figure 4: Ablation Analysis

Ablation Analysis	Accuracy	Change vs. All 4	F1 Score	Change vs. All 4
All 4 Variables	**0.94**		**0.94**	
Remove:				
Frequency	0.93	-0.01	0.93	-0.01
Recency	0.92	-0.02	0.92	-0.02
Session Duration	0.89	-0.04	0.89	-0.04
Pages Visited	0.86	-0.08	0.86	-0.08

Sensitivity Analysis

In order to assess the XGB model's sensitivity, we focused on three key parameters – learning rate, max depth and the number of estimators. The charts (Figures 5-7) below show that the accuracy is sensitive to very low levels of both learning rate and max depth, but less so as the parameter values increase. The number of estimators is only sensitive to values of 100 or lower, while max depth gradually starts to decline for values above 15. In general, number of estimators and max depth can lead to to overfitting above a certain point, although a comparison of the training and test accuracies does not suggest that this is an issue in this case. It is also interesting that accuracy begins to decrease fractionally for learning rates above 0.30, which suggests that the larger step sizes allow the model to run more quickly but slightly less accurately.

Figure 5: Learning Rate

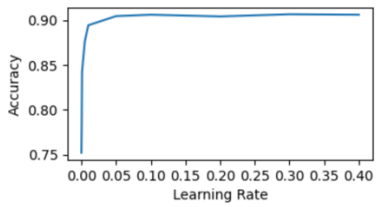


Figure 6: Max Depth

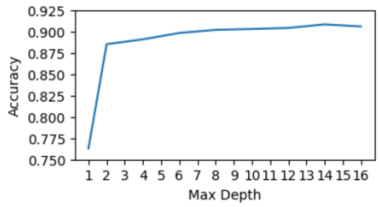
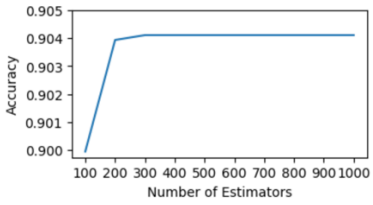


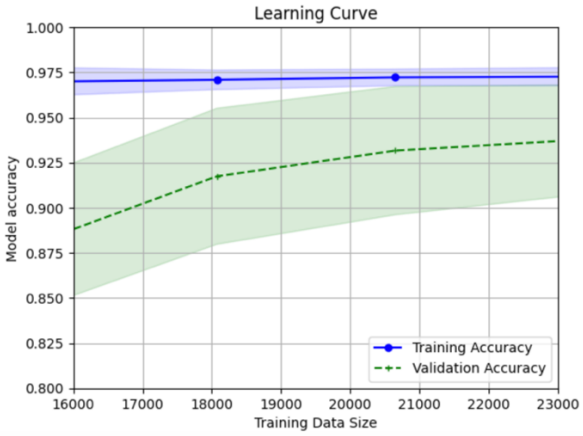
Figure 7: Number of Estimators



Tradeoffs

The use of SMOTE had a sizable, positive, impact on our analysis. The ability to balance the target variable for our classification tasks led to more balanced precision and recall scores than we might have otherwise expected when faced with an imbalanced dataset. However, the learning curve in Figure 8 still reveals an interesting relationship between the size of the data and validation accuracy. Based on the SMOTE analysis of Smartphone purchases, our model initially assumed ~16,000 observations in each class. Using this figure as the starting point, we see that validation accuracy increases as the assumed size of the dataset increases. It also narrows the gap between the accuracy of the training set.

Figure 8: Learning Curve



Failure Analysis

The confusion matrix (Figure 9) is a solid starting point for analyzing different types of classification errors. Our Smartphone example reveals slightly more False Positives (289) vs. False Negatives (265). Figure

10 compares mean and median statistics for cases where the model wrongly predicted a purchase (False Positive) or where actual purchases were missed by the model (False Negative). This data shows that (a) False Positive customers tend to view significantly fewer web pages than Correct Predictions in the test set (median of 3 vs. 7), and have very short Browsing Sessions, and (b) False Negative customers tend to view slightly fewer pages (5 vs 7), browse for ~40% less time and make fewer purchases.

Figure 9: Confusion Matrix

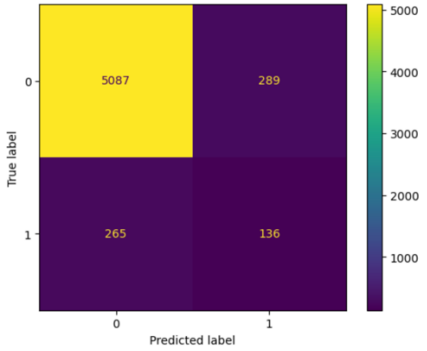


Figure 10: Failure Analysis – Summary Statistics

Category	Samples	Metric	Number Of Pages Viewed	Browsing Session Duration (seconds)	Recency of Last Purchase	Frequency of Purchases
All 4 Variables	5,777	Mean	10.4	8,918	72.2	3.5
		Median	6.0	949	68.0	2.0
Correct Predictions	136	Mean	12.6	12,460	81.9	3.6
		Median	7.0	1,049	81.5	3.0
False Positives	289	Mean	3.5	3,031	74.3	1.4
		Median	3.0	93	73.0	1.0
False Negatives	265	Mean	5.8	5,768	79.0	1.9
		Median	5.0	630	82.0	2.0

Figure 11: Sample False Positives

y_test	y_pred	num_page_views	session_duration	recency	frequency
0	1	3.0	38.0	129	1
0	1	8.0	417.0	53	1
0	1	3.0	261.0	41	1

Figure 12: Sample False Negatives

y_test	y_pred	num_page_views	session_duration	recency	frequency
1	0	1.0	0.0	41	2
1	0	1.0	0.0	45	3
1	0	1.0	0.0	150	1

The sample of 3 False Positives (Figure 11) shows that the model tends to get fooled either by (a) short web browsing sessions or (b) small numbers of page views. Given the relative importance of Browsing Duration and Page Views in the Feature Importance analysis, one way to mitigate this issue is to set thresholds for each of these values below which very low observations can be excluded.

The sample of 3 False Negatives (Figure 12) tells a similar story that the model can also be fooled in a different way by shorter web browsing sessions or fewer page views. This issue can be also addressed by setting threshold levels (perhaps at a slightly higher level than for the False Positives) that could reduce outlier observations and improve model performance.

Description of Methods - Second Business Question

In our study, we delved into the question of whether it was possible to predict customer purchase behavior using user session activity on an e-commerce site. Our objective was to utilize supervised machine learning techniques to forecast customer purchases by leveraging the patterns in user session behavior. The dataset we worked with consisted of user events, such as 'view' and 'purchase' events, along with their respective timestamps. Our goal was to develop a reliable supervised machine learning model that could accurately predict customer purchases using this target variable (Unique user Session) and other pertinent features derived from the dataset. By analyzing the patterns in user session behavior, we aimed to gain valuable insights and identify the crucial factors that influence purchase decisions. The ultimate outcome of our study was to create a predictive model that businesses can employ to optimize their marketing strategies, personalize recommendations, and enhance customer experiences, thereby boosting sales and customer satisfaction.

We treated this as a classification task. Firstly, the DataFrame is processed by categorizing its rows based on the hour of the event time. This categorization involves creating a new column called 'time_label' that assigns a specific label to each row. To achieve this, the hour component is extracted from the datetime

value using a lambda function. The lambda function checks the hour and assigns a label to each row: 1 for morning (before 12), 2 for afternoon (between 12 and 18), and 3 for evening (after 18).

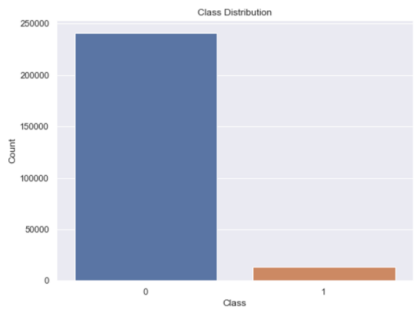
Next, a new column named 'purchase' is added to the DataFrame. This column serves the purpose of indicating whether a particular row corresponds to a purchase event. The values in the 'purchase' column are set to 1 if the corresponding value in the 'event_type' column is 'purchase', and 0 otherwise.

In addition to the 'purchase' column, another new column is introduced in the DataFrame. This column is created to identify purchases made during a specific "desirable time" during the daytime, as opposed to the evening and night.

Finally, the probability of a customer making a purchase is calculated based on the available data. This involves determining the total number of unique customers and the number of unique customers who actually made a purchase. By dividing the number of buyers by the total number of unique customers, the probability of a customer making a purchase is obtained. This probability metric provides valuable insights into the purchasing behavior of customers within the dataset.

In the feature engineering process, we first filter the DataFrame to include only the top 10 categories and top 10 brands based on their frequency count. Once we have this filtered DataFrame, we proceed with extracting additional time-related features from the 'event_time' column. The Purchase_label was created as a new feature based on two conditions involving the 'event_type' and 'time_label' columns. It assigns a value of 1 to rows where both conditions are satisfied and 0 otherwise. This contains values for when a user actually did make a purchase. Hour_of day was extracted from the 'event_time', column, which contains datetime values, and this new feature contains only the hour component of each datetime value. Category code features were derived from the 'category_code' column by splitting and assigning specific words. Category_construction, Category_auto, Category_appliances, Category_electronics, Category_furniture, Category_computers, Category_medicine, Category_audio; features were extracted, by splitting and assigning the third-to-last word (element) from the 'split_categories' Series to each row of the 'category' column. which corresponds to the third word before the last period in the original 'category_code' string. The target feature for our model, was Unique User_Session which was calculated as each user session who both viewed and purchased an item within the same hour. We then decided to select a subset of 10 features and all rows to start off our classification model selection process, with the "Unique_sessions" (initialized with 0, and its value is set to 1 for rows for each use session which viewed an item on the site and then made a purchase within that hour) as our target variable. As shown in Figure 13, our data though is highly imbalanced with 95% observations with 0 and 5% with observations with 1.

Figure 13: Distribution of Classes And Value Counts By Class




```
0    241227
1     13355
Name: unique_sessions,
```

To establish a baseline for comparison, we employed a dummy regressor. The dummy model was created using a constant strategy, in which unique sessions were set to 1. The model was then fitted on the scaled training data and utilized to generate predictions on the scaled test data. The results in Figure 14 provide an understanding of the baseline performance against which other models can be compared.

Figure 14: Summary of Dummy Classifier Evaluation Metrics (pre-SMOTE)

```
Dummy Model Score: 0.05161341005950861
Accuracy: 0.05161341005950861
F1 Score: 0.09816042581006629
Precision: 0.05161341005950861
```

We then compute and extract the top 10 features (Figure 15) and conduct an ablation analysis (Figure 16) based on the XGBoost Classifier.

Figure 15: XGBoost Feature Importance

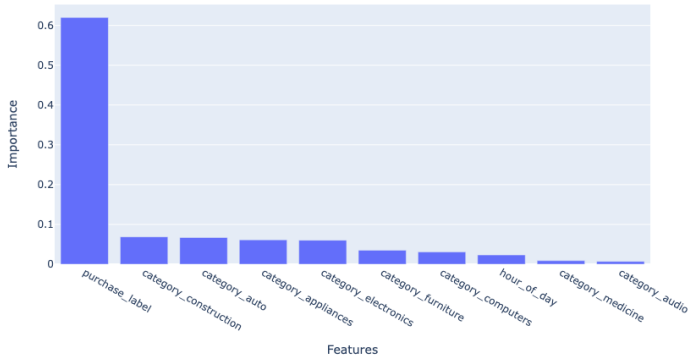


Figure 16: XGBoost Ablation Analysis

```
Feature Ablation Scores:
month: 0.9183376868236541
category_audio: 0.9179448907044798
time_label: 0.9167075829290806
category_medicine: 0.9158630712728558
season: 0.9154113557358053
category_furniture: 0.9120922285287821
day_of_week: 0.8954769526877074
hour_of_day: 0.8943771235540193
category_construction: 0.8800793448160732
category_appliances: 0.879882946756486
category_auto: 0.8763085020719995
category_electronics: 0.8584951980674431
purchase_label: 0.847948622267612
category_computers: 0.8220437182080641
price: 0.765402517823124
```

Following that, we conducted a model selection process to assess and compare various machine learning models, including Random Forest, Support Vector Machine (SVM), XGBoost, and Logistic Regression. This evaluation aimed to determine the model that exhibited the best performance based on the selected features and relevant performance metrics. By analyzing the performance metrics of each model (Figure 17), we identified and selected the model that demonstrated superior predictive capabilities in relation to our specific objectives and chosen set of features.

Figure 17: Comparison of Supervised Learning Evaluation Metrics (pre-SMOTE)

	Score	F1 Score	Accuracy	Precision
Random Forest	0.964020	0.585708	0.964020	0.721851
SVM	0.948387	0.000000	0.948387	0.000000
XGBoost	0.971208	0.613193	0.971208	1.000000
Logistic Regression	0.971149	0.612094	0.971149	1.000000

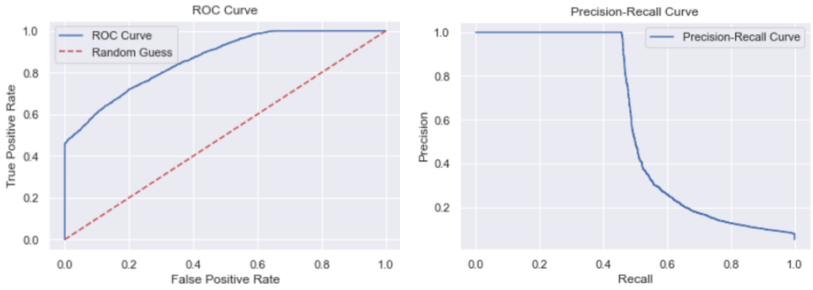
Modeling Results

After conducting model selection, it was determined that the imbalanced data required addressing. To tackle this issue, the SMOTE (Synthetic Minority Over-sampling Technique) algorithm was employed. Hyperparameter tuning was performed by defining a parameter grid with different values for the hyperparameters. RandomizedSearchCV was then utilized to search through the parameter grid, fitting the model with the best parameters hyperparameter combinations, and evaluating their performance through cross-validation (Figure 18).

Subsequently, a pipeline was set up to streamline the workflow. The pipeline consisted of two steps: data scaling using StandardScaler and classification using the XGBoost classifier.

Figure 18: XGBoost Evaluation Metrics (post-SMOTE)

```
Accuracy: 0.9709723667930161
Precision: 0.9700551615445232
F1 Score: 0.6248730964467004
AUC-ROC: 0.8649253398124206
Recall: 0.4608760763758892
```

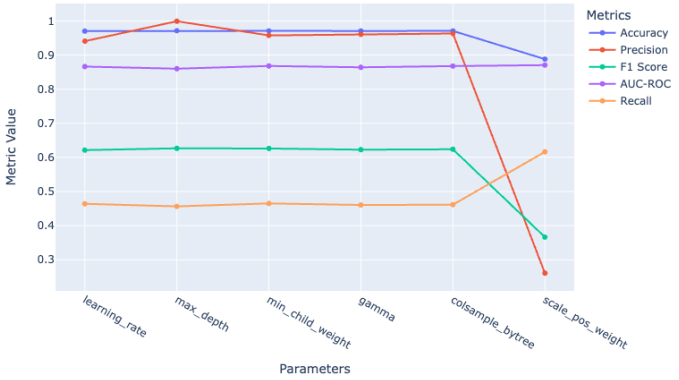


Ablation Analysis

In Figure 19 we evaluated mode performance by varying each parameter individually while keeping the other parameters constant. The **learning rate** parameter yielded fairly good results with high accuracy and precision. The F1 score indicated a moderate balance between precision and recall. The model had a good ability to distinguish between positive and negative instances, as reflected by the AUC-ROC score. However, the recall was relatively low. The **max_depth** parameter produced high accuracy and precision, indicating accurate predictions. The AUC-ROC score was slightly lower. The recall was relatively low. The **min_child_weight** parameter resulted in good performance with high accuracy and precision. The AUC-ROC score was good, but the recall was relatively low. The **gamma** parameter yielded good performance with high accuracy and precision. The AUC-ROC score was good. The recall was relatively low.

Figure 19: Ablation Analysis

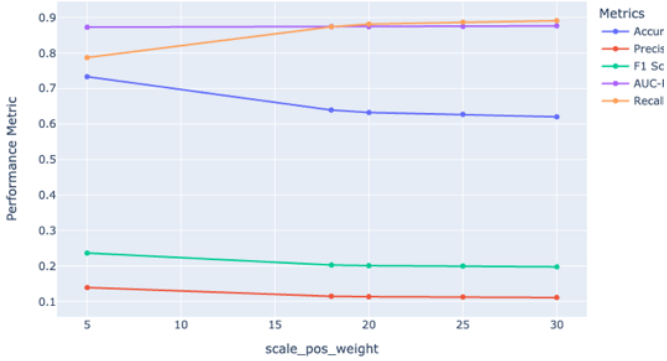
The **colsample_bytree** parameter produced good performance with high accuracy and precision. The AUC-ROC score was good. However, the recall was relatively low. Finally, the **scale_pos_weight** parameter resulted in relatively poor performance with low accuracy and precision. The F1 score and AUC-ROC score were also low. However, the recall was relatively high, indicating a better ability to capture positive instances but at the cost of lower precision



Sensitivity Analysis

Figure 20: Sensitivity Analysis

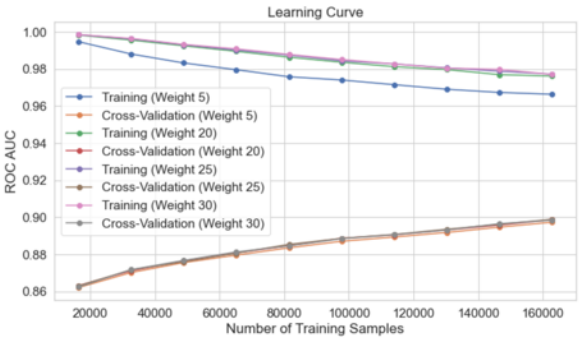
By conducting a sensitivity analysis (Figure 20), we investigated the impact of varying the **scale_pos_weight** parameter while keeping all other parameters fixed. We specifically focused on the range of scale_pos_weight values from 5 to 30 and examined their effects on the model's performance metrics, with a particular emphasis on recall, precision, and AUC-ROC. Notably, we observed that the best results were obtained with scale_pos_weight values of 25 and 30.



Tradeoffs

First, the learning rate (Figure 21) showed a tradeoff between precision and recall. While a higher learning rate led to better accuracy, it resulted in a lower recall, indicating a potential tradeoff between capturing more positive instances and maintaining precision. Second, increasing the max depth improved accuracy and precision but slightly decreased the AUC-ROC score and recall. The scale_pos_weight parameter demonstrated a tradeoff between precision and recall. Higher values improved recall at the expense of lower precision, underlining the need to find the right balance for correctly identifying positive instances while minimizing false positives. We focused on varying the scale_pos_weight parameter while keeping other parameters fixed. By increasing the ratio of scale_pos_weight from 5 to 30, we observed the impact on performance metrics, particularly recall, precision, and AUC-ROC.

Figure 21: Learning Curve



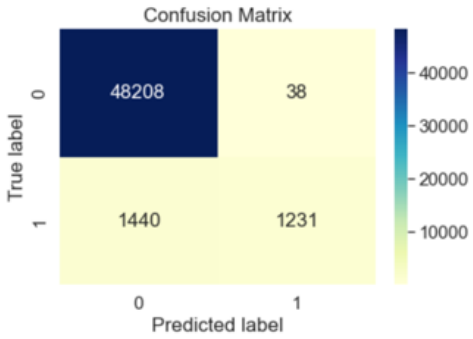
Failure Analysis

Based on the results of the XGBoost model, it demonstrates a high accuracy of 97.10% and precision of 97.01%, indicating its overall strong performance in correctly predicting both positive and negative instances, with a low false positive rate. The model's AUC-ROC score of 0.8649 further confirms its ability to distinguish between positive and negative instances.

However, the model exhibits a weakness in capturing all our positive customer unique sessions who made purchases, as indicated by the relatively low recall score of 0.4609. This suggests that the model struggles to identify all positive instances, resulting in a significant number of false negatives (Figure 22). Despite its good overall performance, there is room for improvement in enhancing the model's ability to identify more customer unique sessions.

To address this issue, further analysis and model enhancements should be explored to reduce false negatives and improve the model's recall rate for positive instances.

Figure 22: Confusion Matrix



Part B: Unsupervised Learning

Description of Methods

Our objective in this portion of the project was to use unsupervised models to form clusters of customers based on the combination of traditional and web-based marketing metrics. Creating customer clusters is a common practice in the retail industry, and incorporating web-based statistics provides e-commerce merchants with incremental information that can be used to customize product recommendations.

The features we used were five of our key marketing metrics – Recency, Frequency, Monetary Value, Browsing Duration, and Pages Visited. Our goal was to look for trends that could be translated into actionable marketing insights. For instance, do customers who spend more time browsing tend to spend more? Or do people who browse more web pages ultimately end up making more purchases (higher frequency)?

We chose kmeans and agglomerative clustering as our primary models. Our rationale for choosing these models was to see how the results would vary between centroid and tree-based algorithms.

Our first step in running these models was to use PCA to reduce dimensionality and the impact of potentially correlated variables (see correlation matrix in Appendix 4). It is worth noting that our preliminary clustering efforts did not include PCA analysis, which resulted in overlapping clusters heavily condensed along the x and y-axes. We used the explained variance curve (Figure 23) to decide how many components were required to capture a significant portion of the total variance. The analysis shows that the first two components explain ~85% of the total variance. The next major decision was to choose the number of clusters. For kmeans, we did this using the scores of the two PCA components as inputs to the elbow method graph below (Figure 24). We then tested a range of potential cluster numbers. We ultimately identified the point on the elbow graph where the slope of the line flattens significantly – in this case, at three clusters – and used that value in our kmeans model.

Figure 23: Explained Variance

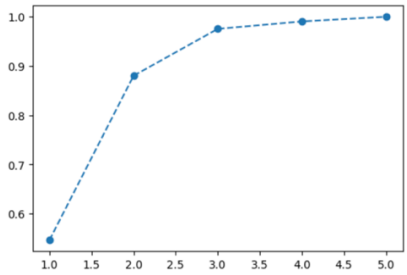
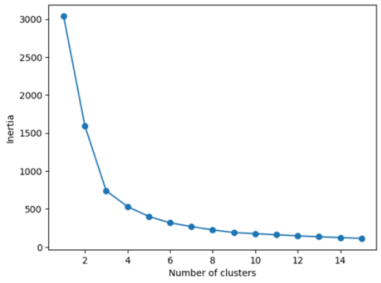
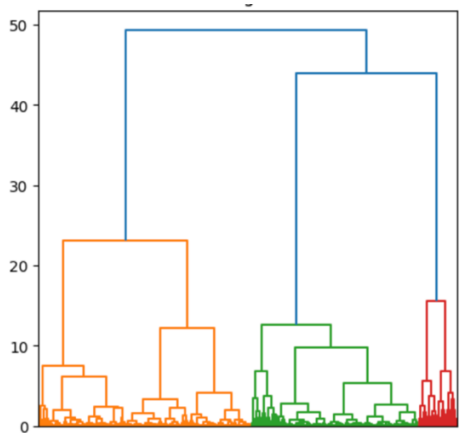


Figure 24: Elbow Method



For the agglomerative clustering model, our goal was to use the same number of PCA components (i.e. 2) that we used in our kmeans model as starting point. This was done to isolate the impact of the clustering mechanism within each underlying model. We chose the appropriate number of clusters for the agglomerative clustering model using a dendogram (Figure 25). First, we identified the greatest vertical distance between horizontal lines (between ~22 and ~44). We then drew an imaginary, horizontal line across this zone and counted the number of vertical lines we crossed. Once again, the analysis suggested that the optimal number of clusters is three.

Figure 25: Dendogram Analysis



Unsupervised Learning - Evaluation

A side-by-side comparison of the clustering results shows relatively similar shapes, although the cluster in the lower right corner of the agglomerative clustering chart (Figure 27) extends further into the red cluster than its counterpart in the kmeans chart (Figure 26).

Figure 26: Kmeans

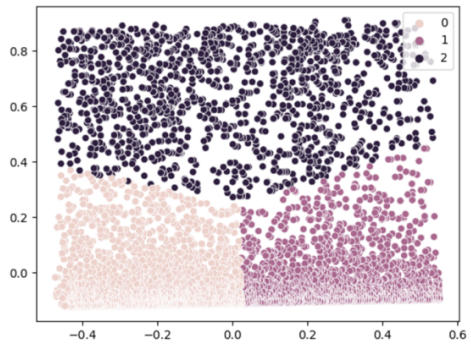
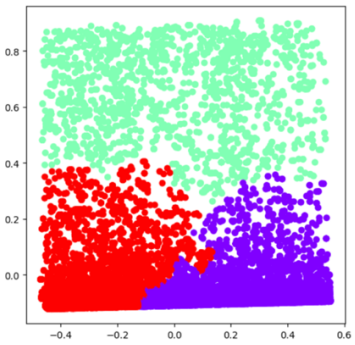


Figure 27: Agglomerative Clustering



But what do these plots suggest about average customer behavior in each cluster? The table below (Figure 28) shows median statistics for each marketing metric based on each clustering method. Although the models number the clusters differently, the distinct sizes (roughly 11,000, 9,000 and 2,000) allow for easy mapping between each model.

Figure 28: Median Cluster Statistics By Model

Unsupervised Model	Cluster	Cluster Size	Number Of Pages Viewed	Browsing Session Duration (seconds)	Recency of Last Purchase	Frequency of Purchases	Monetary Value
Kmeans	0	11,405	6.0	776	35.0	2.0	345.0
	1	9,541	5.0	734	113.0	2.0	165.9
	2	2,160	14.0	65,248	67.0	3.0	408.6
Agg. Clustering	0	11,780	5.0	665	106.0	2.0	172.9
	1	2,097	13.0	65,827	72.0	3.0	405.3
	2	9,229	6.0	976	28.0	2.0	399.5

This table provides useful insights related to feature selection. For instance, Frequency has very similar statistics across all clusters, which suggests a limited amount of valuable modeling variance. This finding is also consistent with the Feature Importance charts from our supervised analysis, where Frequency was generally the least important variable. Another interesting distinction involves Cluster 0 in each model (the largest cluster) and the difference in Monetary Value. For kmeans, the median is $345 vs. only $173 for the agglomerative model. This inconsistency creates a quandary for marketers attempting to target this cluster. Should this customer persona be treated as low or high spending? A final observation is how Browsing Duration is much higher for the smallest cluster in each model. This distinction could help marketers develop a distinct profile for this group (they spend a lot of time browsing before purchasing, but also tend to spend more than those in other clusters).

Evaluation Metrics

We used several metrics to compare the performance of the two models (Figure 29). These include (1) Silhouette scores, which measure separation between the clusters, (2) Calinski-Harabaz scores, which

measure the ratio of the sum of between-cluster to within-cluster dispersion, and (3) Davies-Bouldin scores, which measure cluster size vs. the average distance between clusters.

Figure 29: Summary of Unsupervised Learning Metrics

	Model	
Metric	**Kmeans**	**Agg. Clustering**
Silhouette Score	0.522	0.528
Calinski-Harabaz	23,102	29,511
Davies-Bouldin	0.676	0.611

The silhouette scores of ~0.52-0.53 suggest that both models do a reasonable job of clustering the data points. Interestingly, the other metrics also favor the tree-based agglomerative clustering approach. In some ways these findings are consistent with the results of our supervised learning analysis, where the boosted tree structure (XGBoost) performed better than instance-based models such as KNN and SVM. This difference could also be explained in part by (a) the relatively small number of independent variables (5), which makes it easier for agglomerative clustering to identify similar points, and (b) potentially more irregular shapes of the clusters, which has a greater impact on the kmeans model.

Silhouette Plots

The silhouette plots (Figures 30 & 31) provide similar results for the two models. The large Cluster 0s in each model have similar silhouette scores, while the other two clusters have slightly higher scores in the aggregate clustering chart. Interestingly, the weightings used in the silhouette chart show a slightly higher overall score for keams compared to the calculated silhouette_score metric for each model.

Figure 30: Kmeans

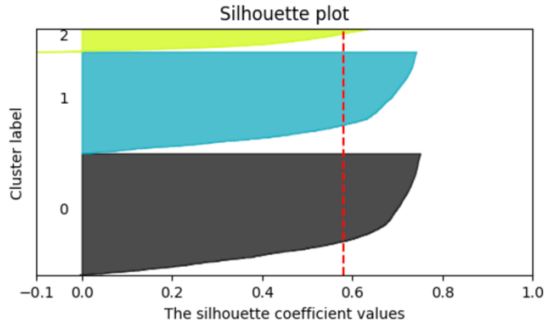
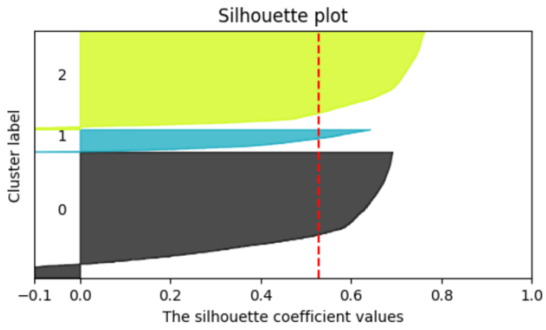


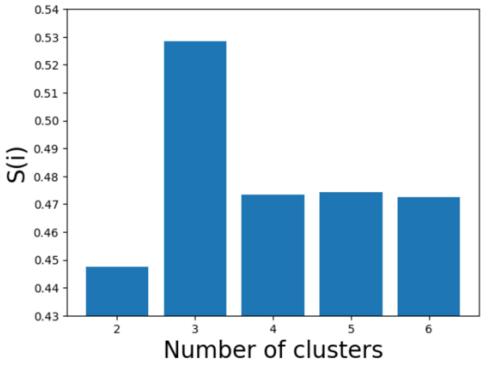
Figure 31: Agg. Clustering



Sensitivity Analysis

The most impactful parameter to sensitize in these unsupervised models is the number of clusters. This choice has a noticeable impact on the on the silhouette scores. Figure 32 shows different silhouette scores for different numbers of clusters using the aggregate clustering model. The silhouette score is highest for 3 clusters (0.528) and lowest for 1 cluster (0.448), while scores for 4-6 clusters all fall between 47.0 and 47.5.

Figure 32: Silhouette Score Sensitivity



Discussion – Supervised Learning

Business Question 1 – Product/Category Classification Using Marketing Metrics

This business question led to several interesting results. First, we were surprised at the significant difference in performance across supervised models. XGBoost was the clear winner in terms of Accuracy and F1 Scores (although ROC-AUC scores were a little lower than hoped), which is impressive given that the analysis was conducted using only four independent variables. Second, it was interesting that different products/categories had different orders of feature importance (other than Frequency, which was generally the least impactful variable). This is encouraging for e-commerce marketers. If the feature importance figures had been the same for every product/category, it would have made it more difficult to develop distinct customer personas. This is even more impressive given that we were working with anonymized data. The main challenges in this process involved (1) the need to engineer customer features given that we had anonymized data (2) deciding how to narrow down the vast number of products and categories, and (3) addressing the issue of imbalanced classes. We addressed the feature engineering issue by creating a series of diverse metrics with the hope that they would capture different information about the customers. We dealt with the large number of products through a bit of trial and error and ended up with a meaningful but not overwhelming number of choices. The critical imbalanced class issue was addressed using SMOTE, which made a big difference. Given more time and resources, the most obvious incremental improvement would be to get actual rather than anonymized user data. Another interesting analysis would be to run the XGBoost model for every product and category rather than selected subsamples. This could allow us to identify synergistic opportunities between product groups (i.e. people who buy Smartphones are also good candidates for headphones).

Business Question 2 – Predict purchase behavior using time-based user session behavior

In this business analysis, we aimed to predict customer purchase behavior using time-based user session behavior. We believe we were successful. The XGBoost model achieved a high accuracy and precision and AUC-ROC using the ten feature inputs we extracted mainly time_based, and the rest categorical. (1) The highly imbalanced dataset posed a challenge. To address this issue, we explored further analysis and model enhancements which we addressed by using SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes and improve the model's performance. (2) In addition to this, the model had a challenging parameter which need to be tuned precisely. We focused on adjusting the scale_pos_weight parameter of the model while keeping other parameters fixed. By increasing this parameter, we observed the impact on performance metrics, particularly recall, precision, and AUC-ROC. This tradeoff between precision and recall highlighted the importance of finding the right balance to correctly identify positive instances while minimizing false positives. (3) The extraction of time-based explanatory features from the main 'event_time' column posed a challenge due to gaps in the time data, resulting in the presence of NaT (Not a Time) values. Consequently, many rows with NaT had to be dropped, leading to the loss of potential data. Overall, while the model demonstrated good performance, although there is room for improvement in enhancing its ability to identify more customer unique sessions. Further analysis, model refinement, and experimentation with different parameters are recommended to reduce false negatives and improve the model's recall rate for positive instances. This analysis offers valuable insights that can benefit businesses by optimizing marketing strategies, personalizing recommendations, enhancing customer experiences, and ultimately driving increased sales and customer satisfaction.

Discussion – Unsupervised Learning

One surprising result in this portion of the project was that both models chose three as the optimal number of clusters and created two large clusters and one very small one. Given the number of samples (over 23,000), we would have expected the models to make more granular distinctions. Perhaps this suggests that one or more of the variables (likely Frequency) added limited new information, so the models effectively relied on a smaller subset of the five original variables. Another surprise was how the median marketing metrics (such as Monetary Value) differed between equivalent clusters in each model. This finding speaks to the importance of considering diverse modeling processes before characterizing customers. For example, a marketer could easily be led down a path of thinking that a customer in Cluster 0 is either a big

spender (average Monetary Value of $345 based on kmeans) or budget conscious ($173 based on agglomerative clustering). The biggest challenge in this exercise was that our initial modeling efforts led to some unusual looking, overlapping clusters bunched along the x and y-axes. Implementing a two-step process involving PCA prior to running the clustering algorithm helped greatly. Similar to the supervised analysis, having actual rather than anonymous data would have been helpful and likely made the results more impactful. Given more time, we might also have considered additional algorithms with potentially different clustering mechanisms (such as DBSCAN), which would have allowed us to get a third perspective on the most appropriate clustering mechanism for this dataset.

Ethical Considerations

One of the few benefits of using anonymized user data in this project is that customer privacy is maintained, and the lack of informed consent is less of an issue.

In the first supervised analysis we are relying on a black box classification model to make product recommendations. This raises a concern that we may not be able to explain these recommendations. We may also be recommending unsuitable products. If, for instance, someone has a gaming addiction and our black box model recommends promotional campaigns regarding gaming products (videocards, consoles, etc.), we could be subjecting that individual to future harm. Potential fixes are to (1) use multiple models to create consensus recommendations, and (2) provide enhanced disclosures in any potential marketing materials about the potential risks associated with certain products.

Ethical considerations arise in the second supervised analysis, particularly in the context of privacy, consent, and fairness. Collecting and analyzing customer data over time requires ensuring the protection of personal information and respecting individuals' privacy rights. Transparent communication and obtaining informed consent from customers for data collection and analysis is essential. It is crucial to handle customer data securely and responsibly, adhering to applicable data protection regulations. Fairness should be maintained throughout the analysis process to avoid biases and discrimination.

In the unsupervised portion of the project, one potential issue is assigning customers to clusters with which they do not identify. Again, this would be a bigger issue if the data contained personally identifiable information, but it could still result in customers receiving marketing materials for a lower-quality products (based on the average prices paid by customers in their cluster) than they might have otherwise chosen for themselves. This could lead to an economic cost if the low-quality product needs to be replaced sooner than expected. This issue could be addressed by developing additional marketing metrics and/or using personally identifiable information, both of which would allow for more granular recommendations.

Statement of Work

In general, we divided many of the preprocessing and feature engineering tasks, then combined our results before seeking to address different supervised learning business questions. We maintained regular contact throughout the process and shared code using both Slack and Deepnote.

	Primary Responsibilities
Greg Holden	EDA, Feature Engineering (Web-Based Analytics), Unsupervised Analysis, Supervised Analysis (First Business Question), Report Writing And Editing
Emmanuel Sengendo	EDA, Feature Engineering (Time-Based Analytics), Related Work Research, Supervised Analysis (Second Business Question), Report Writing And Editing

References

Sefara, T. (2020). *eCommerce Behavior using XGBoost*. Kaggle. https://www.kaggle.com/code/tshephisho/ecommerce-behaviour-using-xgboost

R. Valero-Fernandez, D. J. Collins, K. P. Lam, C. Rigby and J. Bailey, "Towards Accurate Predictions of Customer Purchasing Patterns," *2017 IEEE International Conference on Computer and Information Technology (CIT)*, Helsinki, Finland, 2017, pp. 157-161, doi: 10.1109/CIT.2017.58.

T. T. Win and K. S. Bo, "Predicting Customer Class using Customer Lifetime Value with Random Forest Algorithm," *2020 International Conference on Advanced Information Technologies (ICAIT)*, Yangon, Myanmar, 2020, pp. 236-241, doi: 10.1109/ICAIT51105.2020.9261792.

Appendix 1: Summary of Evaluation Metrics – First Supervised Learning Analysis

	Model	Accuracy	Precision	Recall	F1 Score	Acc Stdev	F1 Stdev	Accuracy on Test Set
0	Dummy Classifier	0.503	0.503	0.503	0.503	0.020	0.020	0.512
1	Logistic Regresion	0.542	0.542	0.542	0.539	0.009	0.010	0.479
2	NB	0.545	0.600	0.545	0.471	0.006	0.008	0.223
3	SVM	0.566	0.568	0.566	0.562	0.005	0.005	0.472
4	KNN	0.897	0.901	0.897	0.897	0.006	0.006	0.850
5	Random Forest	0.704	0.713	0.704	0.701	0.032	0.031	0.611
6	XGBoost	0.931	0.939	0.931	0.930	0.065	0.070	0.922

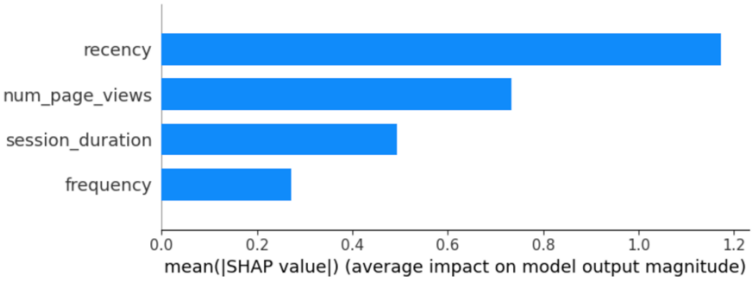
Appendix 2: XGBoost – Output from RandomizedSearchCV – First Supervised Learning Analysis

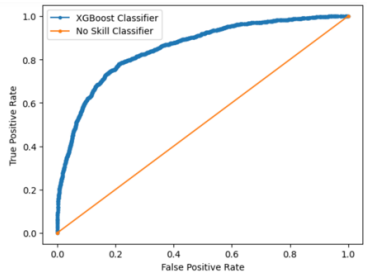
```
Fitting 10 folds for each of 5 candidates, totalling 50 fits

XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1.0,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0.4, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.2, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=12, max_leaves=0, min_child_weight=5,
              missing=nan, monotone_constraints='()', n_estimators=400,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=99,
              reg_alpha=0, reg_lambda=1, ...)
```

Appendix 3: Select Feature Importance Results For Different Products and Categories – First Supervised Learning Analysis

Product – Video cards

Model	Accuracy	Precision	Recall	F1 Score	F1 Stdev
XGBoost	0.843345	0.853274	0.843344	0.841451	0.071845

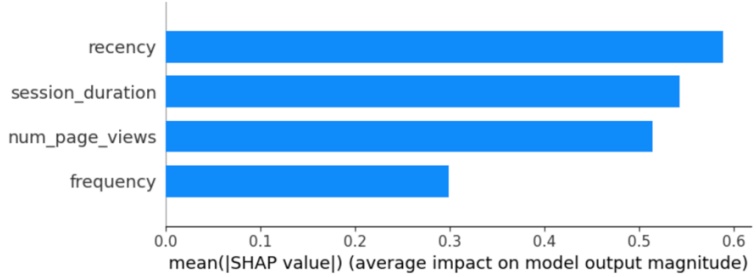


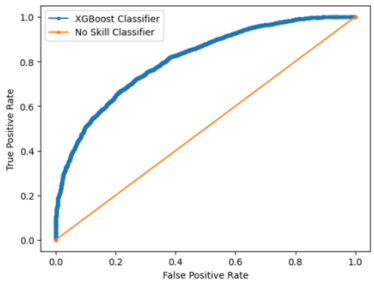


XGBoost Classifier: ROC AUC=0.853
No Skill Classifier: ROC AUC=0.500

Category - Computers

Model	Accuracy	Precision	Recall	F1 Score	F1 Stdev
XGBoost	0.761976	0.770935	0.761976	0.759886	0.058175

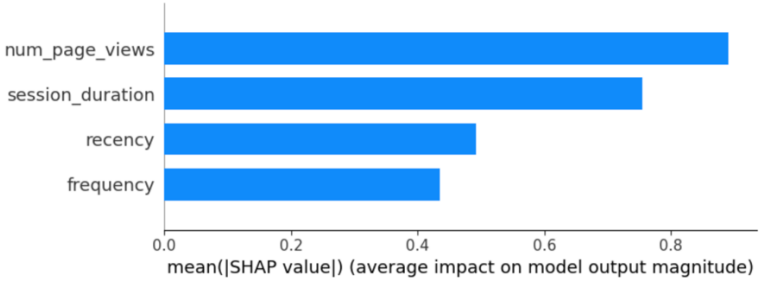


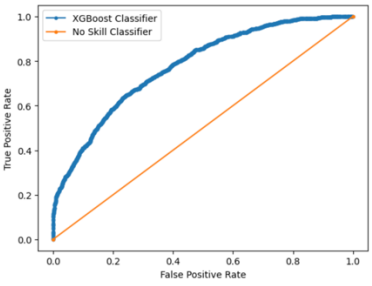


XGBoost Classifier: ROC AUC=0.809
No Skill Classifier: ROC AUC=0.500

Category – Electronics

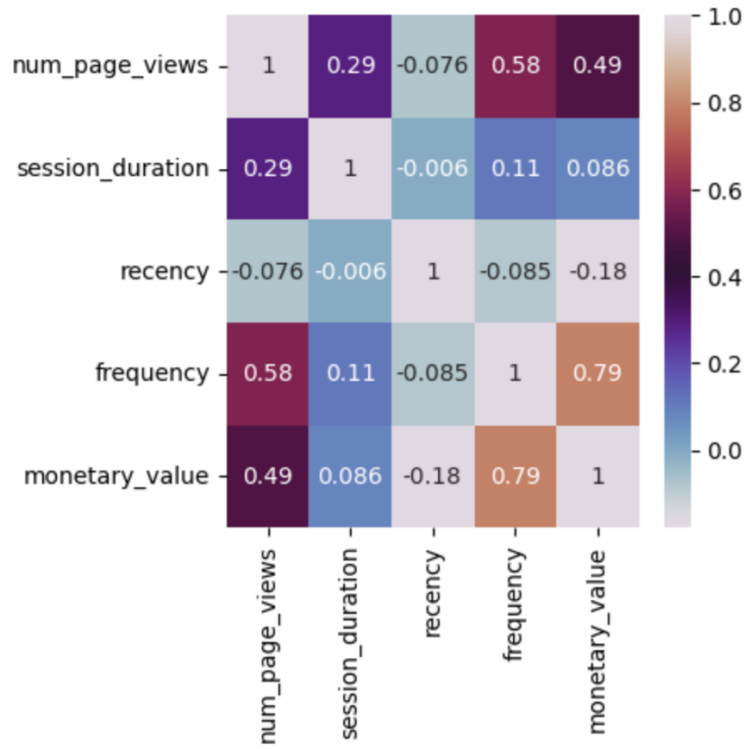
Model	Accuracy	Precision	Recall	F1 Score	F1 Stdev
XGBoost	0.876628	0.889256	0.876628	0.873254	0.08858





XGBoost Classifier: ROC AUC=0.778
No Skill Classifier: ROC AUC=0.500

Appendix 4: Correlation Matrix – Unsupervised Learning Variables



Appendix 5: Summary of Evaluation Metrics – Second Supervised Analysis

	Score	F1 Score	Accuracy	Precision
Random Forest	0.964020	0.585708	0.964020	0.721851
SVM	0.948387	0.000000	0.948387	0.000000
XGBoost	0.971208	0.613193	0.971208	1.000000
Logistic Regression	0.971149	0.612094	0.971149	1.000000

Appendix 6: XGBoost – Second Supervised Analysis

Best Parameters:

'scale_pos_weight': 25,
'min_child_weight': 7,
'max_depth': 12,
'learning_rate': 0.2,
'gamma': 0.1,
'colsample_bytree': 0.7

Appendix 7: Correlation Matrix – Second Supervised Analysis