PREDICTING PURCHASING INTENT FROM USER BEHAVIOUR ON E-COMMERCE PLATFORM

Institute of Data | Capstone Project

Zaki Hamzah Lee

1. Abstract

This report presents the results of an experimental analysis of predicting purchasing intent from user behaviour on an e-commerce platform. The preliminary raw dataset of user behaviour on the website of an electronics e-commerce store was obtained from Kaggle. The October 2020 portion of the data was selected from the raw data for analysis in this study. The study was conducted using Python programming on a Jupyter Notebook with extensive utilisation of functions and objects from Python packages such as pandas, numpy, and scikit-learn.

The selected October 2020 data was explored and processed, and features were engineered from the data as inputs for the classification models. The data was split into train-test and validation datasets according to a stratified 80:20 ratio. Several algorithms were compared for their effectiveness in classifying daily user activity data into visits with purchasing intent and visits without purchasing intent. The key performance indicators of the models consisted of recall scores and model training time. 5-fold cross validation was performed for each model to obtain the mean recall scores and mean model training time across folds.

Mean model training time on the train-test data was found to be short for all models, and was therefore not a key factor in the selection of the most effective model. The XGBoost algorithm produced the highest mean recall score of 89.2% on the train-test data. Hyperparameter tuning of the XGBoost model ('learning_rate' = 0.01, 'max_depth' = 3, 'n_estimators’ = 50) was successful in improving the recall score to 92.9% on the train-test data. The XGBoost model with these hyperparameters was selected for validation on the validation dataset.

The recall scored produced by the selected model was 93.0%, and was therefore considerably successful in predicting purchasing intent. Extraction of feature importances from the model revealed that the feature corresponding to the addition of products to users’ carts was highly important in predicting purchasing intent (0.895 out of 1). A marketing strategy

focused on cart abandonment is proposed as a possible application of the model to convert browsers with high purchasing intent into customers.

2. Acknowledgement

The author would like to express his gratitude to his trainers Ms. Gurkiran Kaur, Mr.Lim Zheng Wei, and Mr. Matt Burnham for their guidance, time, and inputs during the entire course of this project and the course as a whole.

Table of Contents

3. Introduction

Over the past few years, e-commerce has grown to be a key mode of consumer retail transactions. According to a Forbes report, worldwide e-commerce sales are expected to reach $4.2 trillion USD in 2021 [1]. From a separate Forbes report, Euromonitor International estimates that approximately 50% of the value growth of the global retail market from 2020 to 2025 will be digital [2]. It is also estimated that the number of e-commerce website visits globally in the month of June 2020 alone was 22 billion [3].

However, the size of the market is not as important to e-commerce businesses as the key performance indicator known as conversion rate. Conversion rate is the ratio of purchases to visits, represented as a percentage. It is a better indicator of whether e-commerce businesses are generating sales on their e-commerce platforms compared to the total number of visits on their platforms. Globally, the average conversion rate of e-commerce platforms in the first three quarters of 2020 was 2.19%. This low value of conversion rate is the essence of the problem to be studied, which is that visits themselves do not indicate purchasing intent, and that purchasing intent is not always acted upon to result in a successful transaction for businesses. Users with low or no purchasing intent may simply be browsing or ‘window shopping’, while users with high purchasing intent may choose not to follow through with their purchases due to high fees and taxes, or due to the cheaper price offered by a competitor.

Recognising the potential sales value in the high percentage of visits which do not result in purchases, e-commerce platforms go to great lengths to increase their conversion rates. For example, e-commerce giants such as Shopee and Lazada have set up a seller education hub and an online seller center respectively to educate their sellers on strategies to increase their conversion rates [4] [5]. In order to compete for survival with such giants, smaller businesses must explore all possible means to increase purchases on their own e-commerce platforms, thereby increasing their conversion rates. Failure to maintain competitiveness could cost smaller businesses their entire businesses, making everyone invested in the business a stakeholder, from shareholders to the most junior employees.

Traditional means of increasing purchases, such as increasing advertising, offering discounts, and tailoring customer rewards schemes, are well known business strategies. However, researchers have acknowledged difficulty in generalising the findings of available studies on the prediction of purchasing activity from e-commerce user activity data, also known as ‘clickstream prediction’ [6]. There is active research to understand ‘clickstream prediction’ and it’s various specificities in greater detail, underscoring the amount of work that can yet be done in the field [7] [8]. Therefore, even smaller businesses may be able to tap into a certain degree of ‘first mover’ advantages by assessing the potential of machine learning models to predict purchasing intent from user behaviour data.

This project seeks to develop and compare machine learning models to answer the data question of what features and model best predict purchasing intent from user behaviour data. Subsequently, a business marketing strategy is presented as a follow-up to answer the business question of how marketing strategies can be applied to users with purchasing intent.

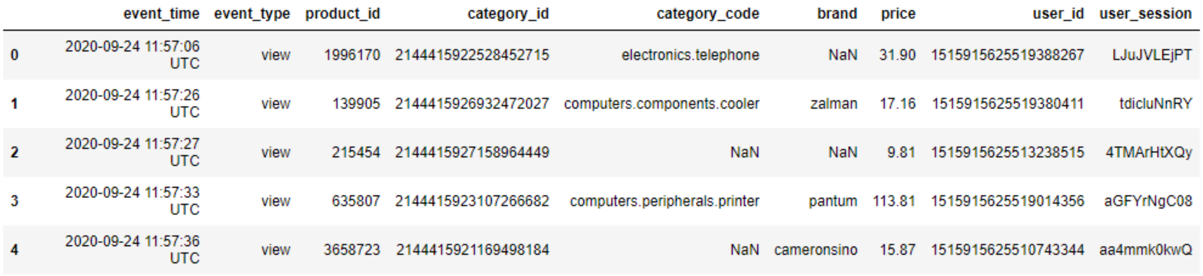
4. Methodology

4.1 Data Background and Setup

The preliminary raw dataset of user behaviour on the website of an electronics e-commerce store was obtained from Kaggle and provided by REES46 Marketing Platform [9]. The study was conducted using Python programming on a Jupyter Notebook with extensive utilisation of functions and objects from Python packages such as pandas, numpy, and scikit-learn. The version of Python used was 3.7.4.

4.2 Exploratory Data Analysis (EDA)

The basic characteristics of the data were explored to understand the nature and context of the data as well as its limitations and deficiencies. The first 5 rows were printed to obtain a snapshot of the columns of the raw dataset and the kinds of inputs that were to be expected.

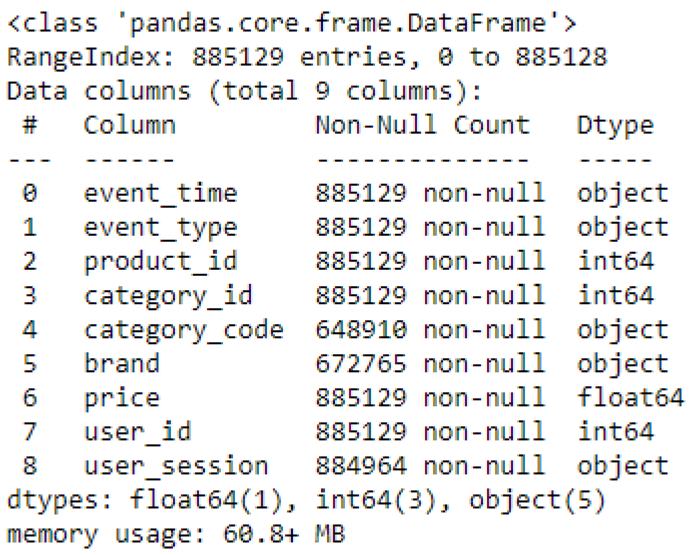


	event_time	event_type	product_id	category_id	category_code	brand	price	user_id	user_session
0	2020-09-24 11:57:06 UTC	view	1996170	2144415922528452715	electronics.telephone	NaN	31.90	1515915625519388267	LJuJVLEjPT
1	2020-09-24 11:57:26 UTC	view	139905	2144415926932472027	computers.components.cooler	zalman	17.16	1515915625519380411	tdicluNnRY
2	2020-09-24 11:57:27 UTC	view	215454	2144415927158964449	NaN	NaN	9.81	1515915625513238515	4TMArHtXQy
3	2020-09-24 11:57:33 UTC	view	635807	2144415923107266682	computers.peripherals.printer	pantum	113.81	1515915625519014356	aGFYrNgC08
4	2020-09-24 11:57:36 UTC	view	3658723	2144415921169498184	NaN	cameronsino	15.87	1515915625510743344	aa4mmk0kwQ

Figure 1: First 5 Rows of Raw Dataset

From Figure 1, 9 columns were observed, containing event time, event type (either 'view', 'cart', or 'purchase'), product ID corresponding to the specific product, category ID, category code corresponding to the category ID, brand, price, unique user ID, and user session. It was also observed that null values existed in the category code and brand columns.

The properties of the dataset, structured as a dataframe, and the individual columns were investigated using the pandas DataFrame.info method:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 885129 entries, 0 to 885128
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   event_time     885129 non-null  object 
 1   event_type     885129 non-null  object 
 2   product_id     885129 non-null  int64  
 3   category_id    885129 non-null  int64  
 4   category_code  648910 non-null  object 
 5   brand          672765 non-null  object 
 6   price          885129 non-null  float64
 7   user_id        885129 non-null  int64  
 8   user_session   884964 non-null  object 
dtypes: float64(1), int64(3), object(5)
memory usage: 60.8+ MB
```

Figure 2: Information of the Dataset DataFrame and Columns

From Figure 2, it was observed that the data consisted of 885,129 rows and 9 columns. There were also null cells in the category code, brand, and user session columns. As the dataset spanning October 2020 to February 2021 was too large to allow for rapid prototyping of prediction models, the October 2020 portion of the data (consisting of 161,544 rows) was selected from the raw data for analysis in this study. Any references to ‘the data’ from this point onwards shall refer to this October 2020 portion of the original data.

Several key observations were also made regarding the data:

Observation #1

As this study focuses on user behaviour on the e-commerce platform, the event type column formed a crucial part of the feature engineering process. The distribution of event types between ‘view’, ‘cart’, and ‘purchase’ was heavily skewed towards ‘views’ (146,539 total) as compared to ‘cart’ actions referring to the addition of items to the users’ shopping carts (8,729 total) and ‘purchase’ actions (6,276 total).

Observation #2

The user sessions in the user session column were initially thought to represent distinct periods of active user engagement with the e-commerce platform, and were therefore a potential candidate to select as the individual datapoints for analysis. However, the user sessions were found to span across several days. This corresponds to a valid user behaviour pattern where a user opens a browser session to conduct activities within a certain period of time, but stops interacting with the browser and returns to it at a later date. Considering such user sessions as distinct visits goes against the implicit premise of the study that user sessions spanning different days should not be considered as a single visit because user activity and purchasing intent may vary across different periods of platform engagement. A decision was therefore made to disregard the user session column, and to instead engineer daily user activity as distinct datapoints which could be analysed by the models.

4.3 Feature Engineering and Data Transformation

The raw data consisted of rows corresponding to individual events or actions (‘view’, ‘cart’, or ‘purchase’) with the distinct user ID and time when these events occurred. To construct meaningful collections of actions as features, the events had to be grouped by user and by the date on which the events occurred. The final transformed data would therefore contain daily user visits as rows or datapoints.

Three broad categories of event grouping were considered: Current Visit Activity, Most Recent/Previous Visit Activity, and Historical Visit Activity. Examples of the features which could be engineered under these categories are duration of current visit, number of actions performed in the previous visit, and historical user expenditure. However, it was recognised that features related to the Most Recent/Previous Visit Activity and Historical Visit Activity would introduce bias against new users of the platform. As a result, only Current Visit Activity was adopted to explore possible features to engineer from the data.

Engineered features were identified by considering the metrics from which it could be inferred that users had purchasing intent. Based on the 9 columns available from the data, the following were the features engineered and the reasons for selecting them as features:

Feature #1: Duration of Visit in Seconds

This feature was selected because it was inferred that users with purchasing intent would naturally be willing to put in a larger time investment into their visits compared to users who were just browsing or had no purchasing intent. This feature was engineered by grouping all actions performed by individual users on the same days together, followed by obtaining the time difference between the last action and the first action performed by the users on those days.

Feature #2: Count of 'Views' and 'Carts' for Most Frequently Browsed Category ID

This feature was selected because it was inferred that users with purchasing intent would repeatedly engage with products in the same category as their desired purchase, while users with little or no purchasing intent would not do so since they would not have a desired purchase in mind. This feature was engineered by grouping all actions performed by individual users on the same days together, followed by obtaining the count of the 'view' and 'cart' actions performed for the most frequently browsed product category on those days.

Feature #3: Cart Quantity

This feature was selected because it was inferred that users with purchasing intent would add desired items to their shopping carts, which is a pre-purchase action that is common on e-commerce platforms. Users with little or no purchasing intent would have no use for adding items to their shopping carts, and were therefore expected not to add anything to their carts. This feature was engineered by grouping all actions performed by individual users on the same days together, followed by obtaining the count of the 'cart' actions on those days regardless of product or category ID.

A python function was defined to perform all the necessary transformations on the raw data for the feature engineering. The Python code is included in the Appendix of this report. A target variable was also created to label the daily user visits with '1' if there was at least 1 purchase by the end of the visit or '0' if there were no purchases during the visit.

The transformed data contained a total of 92,277 daily user visits as rows, out of which 3,983 (4.3% of total) resulted in at least 1 purchase while 88,294 (95.7% of total) did not result in any purchases. The class imbalance is considered and discussed in section 4.5 of this report on model training and validation.

4.4 Classification Algorithms and Key Performance Indicators

Several classification algorithms were selected for comparison, namely Logistic Regression, Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, AdaBoost, and XGBoost. Neural networks were considered for comparison, but the follow-up step to the modelling would require the extraction of feature importances for the application of an appropriate marketing strategy. Since such extraction methods are not typically available for neural network classifiers, neural network classifiers were not explored.

The Key Performance Indicators (KPI) upon which the models were compared were recall scores (indicating high correctness in predicting purchasing intent) and model training time. A high recall score and short model training time were desirable characteristics of a successful model. However, the recall score was prioritized as a primary KPI while the model training time was designated as a secondary KPI due to the business objective of the problem.

While False Positives are sometimes used as part of the performance measures of classification models, for this problem they do not necessarily reflect weakness in the models nor do they have negative impact for business employing the models. For this problem, False Positives refer to instances of predicted purchasing intent, but no resultant purchase. However, the reason for the lack of purchase actions may not simply be because there was a misclassification of purchasing intent, but also that users had other reasons for not purchasing any products. Examples of such reasons could be the availability of a cheaper price offered by a competitor business, or that the fees and taxes upon checkout of the cart were too high. A study by the Baymard Institute found that the second most quoted reason for users abandoning their cart items was that the extra costs such as shipping, taxes, and fees were too high [10]. Essentially, the False Positives predicted by a successful model are not a failure of the model to predict purchasing intent accurately, but instead represent an essential target group of ‘missed opportunities’ for businesses to apply marketing strategies to.

4.5 Model Training, Testing, and Validation

The transformed data was split into train-test and validation datasets according to an 80:20 ratio respectively. The split was performed in a stratified manner to ensure that the percentage of purchases in the validation dataset was the same as the percentage of purchases in the train-test dataset. This helped to prevent the possibility that most of the positive class (purchases) was assigned to the train-test dataset.

The severe class imbalance in the train-test dataset was recognised, and methods such as undersampling and oversampling were considered to remedy this imbalance in order to train the classification models with enough instances of the positive class to make accurate predictions. However, no treatments were applied to the data because several implicit simplifying assumptions were made during the feature engineering stage, and performing undersampling or oversampling techniques would introduce further assumptions and uncertainty into the models.

The first assumption was that daily user activity was assumed to be a continuous and distinct period of user engagement regardless of the time of the first and last action performed. However, users may have interacted with the e-commerce platform at multiple times during a single day, each time with the intention to purchase different products. The second assumption was that the most frequently browsed category of product was relevant to the eventual product purchased, but it is possible that the two are unrelated. This assumption could have easily been removed by introducing a variable to verify if the purchased product belong to the most frequently browsed category of product, but the value of this variable would be undefined for the majority of visits which did not result in a purchase, and therefore this variable could not be used. In consideration of the assumptions above, the data was not modified further in order not to distort model effectiveness when classifying on the validation dataset.

5-fold cross validation was performed for each algorithm to obtain the mean recall scores and mean model training time across folds.

The model with the best mean recall score and an acceptable/reasonable model training time was selected for further hyperparameter tuning to improve the KPI measures, and the resultant tuned model was validated on the validation dataset. Feature importances were extracted from the final model for the development of an appropriate marketing strategy.

5. Results

Algorithm	Mean Train-Test Recall	Mean Model Train Time (seconds)
Logistic Regression	28.2%	0.332
Support Vector Machine	33.5%	47.8
Naïve Bayes	60.2%	0.0195
Decision Tree	60.9%	0.0419
Random Forest	65.5%	1.91
AdaBoost	80.7%	0.965
XGBoost	89.2%	1.60

Figure 3: Performances of Models on Train-Test Dataset

Referring to Figure 3, the mean model training time on the train data was found to be less that one minute for all models. As the train data consisted of 64% of the October 2020 data, the model training time on one month's worth of data is expected to scale accordingly and amount to less than 1.5 minutes for all models. Since all models offered practical times for re-training and re-configuration on a month's worth of data, the mean model training time was therefore not a key differentiating factor in the selection of the most effective model.

The mean train-test recall scores varied widely across different algorithms. The lowest mean recall score was observed for Logistic Regression at 28.2% while the XGBoost algorithm produced the highest mean recall score of 89.2%. The XGBoost model was therefore selected for hyperparameter tuning.

The hyperparameters of the XGBoost model were tuned using the GridSearchCV method of the model_selection module of the scikit-learn library. The hyperparameters and their tuned values are as follows:

Hyperparameter	Tuned Values
‘learning_rate’ – The learning rate for the XGBoost model	0.01, 0.05, 0.1, 0.2
'max_depth' – The maximum depth of each estimator/tree in the XGBoost model	3, 6, 10
‘n_estimators’ – The number of estimators/trees used in the XGBoost model	50, 100, 200, 300

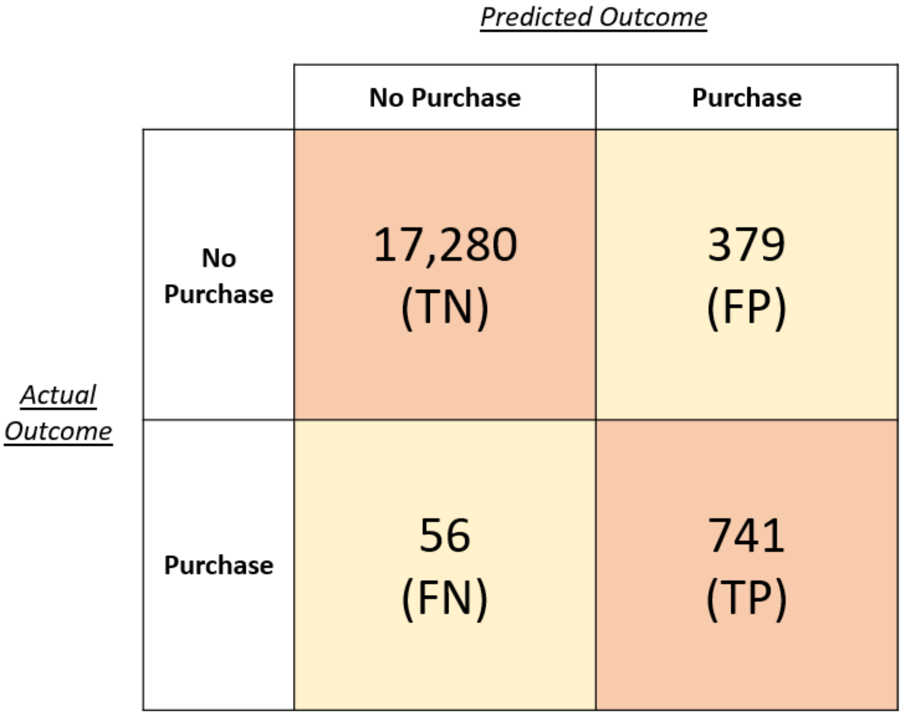
Figure 4: Hyperparamter Tuning Values

The hyperparameter values which produced the highest recall score of 92.9% on the train-test dataset were 'learning_rate' = 0.01, 'max_depth' = 3, 'n_estimators’ = 50. The XGBoost model with these hyperparameters was selected for validation on the validation dataset. The recall scored produced by the tuned XGBoost model was 93.0%, and was therefore considerably successful in predicting purchasing intent.

Extraction of feature importances from the model revealed that the feature corresponding to cart quantity was the most important in predicting purchasing intent (0.895 out of 1) while the count of ‘views’ and ‘carts’ for the most frequently browsed product category was far less important (0.104 out of 1) and the duration of visits was not important at all (0.001 out of 1).

6. Discussion

The final tuned XGBoost model was highly accurate in predicting purchasing intent with a 93.0% recall score. The confusion matrix of the model's classifications of the validation dataset is as follows:



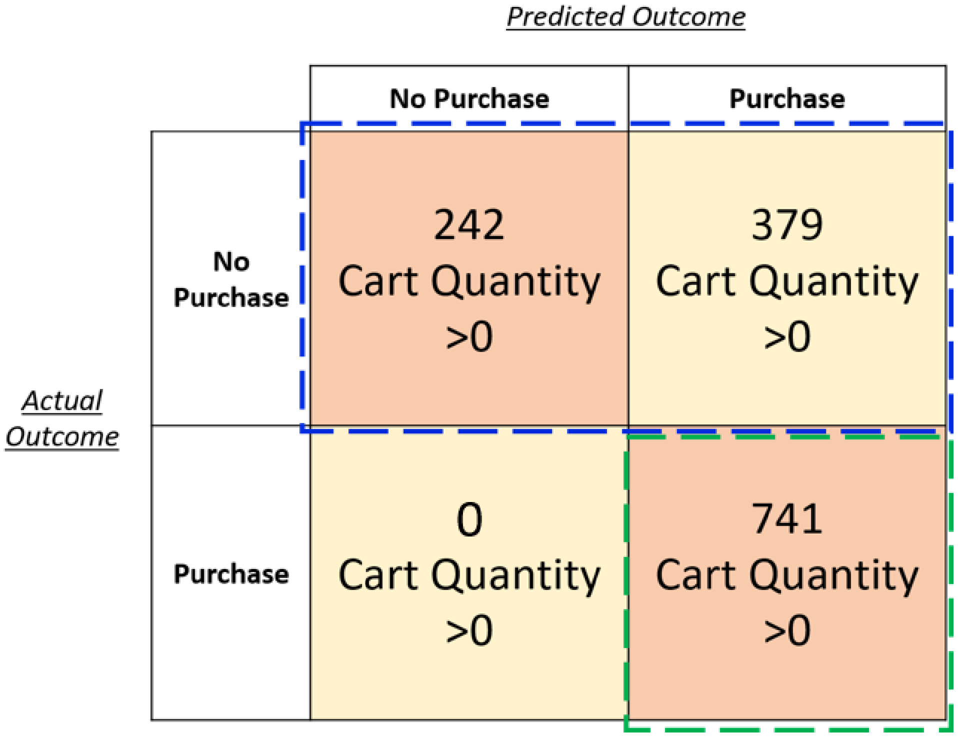
Predicted Outcome

		No Purchase	Purchase
Actual Outcome	No Purchase	17,280 (TN)	379 (FP)
	Purchase	56 (FN)	741 (TP)

Figure 5: Comparative Matrix of tuned XGBoost Model on Validation Dataset

The 56 False Negatives (FN) corresponding to the daily user visits which the model did not predict purchasing intent for were investigated in further detail. It was observed that all 56 False Negatives had a cart quantity of 0. The purchases occurred when users did not put the purchased products in their carts before purchasing the products, but instead clicked on the available option to purchase the products directly from the 'view' pages. Considering that the cart quantity was the most important feature for predicting purchasing intent (0.895 out of 1), and that such purchases with any 'cart' actions formed only 0.00003% of all visits (56 out of 18,456), such user activity data without any 'cart' actions was difficult for the model to classify as indicating purchasing intent. This further implies that the tuned XGBoost model had obtained the highest possible reasonable recall score, and that the model was highly accurate in predicting purchasing intent.

Examining the user visits which did not result in purchases, it was observed that all 379 False Positives (FP) were user visits where users had added items to their carts. Recognising the possibility that the model may have used positive values of cart quantity as the sole predictor of purchasing intent, the number of visits with positive cart quantities in the confusion matrix was examined and is represented below:



		Predicted Outcome	
		No Purchase	**Purchase**
Actual Outcome	**No Purchase**	242 Cart Quantity >0	379 Cart Quantity >0
	Purchase	0 Cart Quantity >0	741 Cart Quantity >0

Figure 6: Number of Visits with Positive Cart Quantities

It was observed that the model had also predicted that some visits with positive values of cart quantity were still classified as having no purchasing intent, and that these visits did not result in any purchases (242 True Negatives). Deeper analysis revealed that the profile of the features (cart quantity, the count of ‘views’ and ‘carts’ for the most frequently browsed product category, and the duration of visits) were different for these True Negatives than for the 379 False Positives. The cart quantity for False Negatives did not exceed 2, while the cart quantity for False Positives ranged from 1 to 17. While the count of ‘views’ and ‘carts’ for the most frequently browsed product category for the True Negatives did not exceed 2, the same count for False Positives ranged from 3 to 63. The mean duration of visits for True Negatives was 102 seconds while the mean duration of visits for False Positives was 2534 seconds. The differences in the feature profiles for the True Negatives and False Positives aided the model in differentiating between visits with no purchasing intent and visits with purchasing intent. Positive values of cart quantity were therefore not the sole determining factor of purchasing intent.

The results of the model applied on the validation dataset may be subsequently used to demonstrate how a marketing strategy can be targeted at users with purchasing intent. Since the most important feature for purchasing intent prediction was found to be the cart quantity, the marketing strategy which should be prioritised should target users who added items to their shopping carts, but later abandoned them and did not make purchases. According to a study by the Baymard Institute on cart abandonment, the most frequently cited reason for cart abandonment was that users were just browsing or not ready to make purchases (58.6% of respondents), while the second most frequently cited reason was that the extra fees, such as shipping and taxes, were too high (20.3% of respondents) [10]. The latter reason is more within a business’s control than the former reason, and the latter can therefore be targeted for the application of a marketing strategy. An example calculation of a discount strategy to mitigate extra fees is elaborated as follows:

Monthly Marketing Budget for Strategy

Assuming 5% of monthly revenue is allocated for the monthly marketing budget, the total marketing budget for October 2020 amounts to 5% of the total $562,590 (obtained from the data), or $28,129.

Cost of Discount Strategy

The proposed discount strategy is the offer of an average $5 discount per purchase for all user visits which the model predicted as having purchasing intent.

Assuming that 20.3% of users in the validation dataset who abandoned their carts are converted into paying customers through this strategy, the number of additional purchases gained is 20.3% of 621 visits (the sum of the numbers in the blue rectangle in Figure 6), or 126 additional purchases.

However, the $5 discount cannot be withheld from the 741 user visits which the model classified as having purchasing intent and which resulted in actual purchases (the green square in Figure 6).

The total cost of this discount strategy for the 20% validation dataset is therefore $5 multiplied by 867 visits, or $4,335. The total cost for the entire dataset is therefore $21,675.

Additional Revenue Gained

The additional revenue gained through the discount strategy would come from the 126 additional purchases which were gained. With an average cart value of $141 among abandoned carts, the potential revenue gained from the additional purchases is therefore 126 multiplied by $141, or $17,766 in total for the 20% validation dataset. For the entire dataset, the additional revenue gained may be approximated as $88,830.

Final Evaluation

The total cost of $21,675 for the discount strategy is within the monthly marketing budget of $28,129. The strategy presents a potential revenue increase of more than

400% of the total cost of the strategy. On this basis, it is justifiable to claim that the prediction model will bring positive monetary impact to the business.

7. Conclusion

This study analysed the possibility of predicting purchasing intent from e-commerce user behaviour data for the business purpose of applying marketing strategies to increase purchases. Data of an electronics e-commerce store was obtained, and a portion of the data was processed to engineer features which were inferred to be indicative of purchasing intent. These features were the duration of visits, the count of ‘views’ and ‘carts’ for the most frequently browsed product category, and the quantity of items added to users’ shopping carts. The data was split according to a stratified 80:20 ratio between train-test and validation datasets respectively, and various classification algorithms were evaluated on the train-test dataset based on the KPIs of mean recall scores and mean model training times.

The model with the best recall score of 89.2% employed the XGBoost algorithm from the XGBoost library. The hyperparameters of the XGBoost model were tuned and optimised, and the hyperparameters of 'learning_rate' = 0.01, 'max_depth' = 3, and 'n_estimators’ = 50 were successful in improving the recall score to 92.9% on the train-test data. The XGBoost model with these hyperparameters produced a recall score of 93.0% on the validation dataset, and was concluded to be highly effective in predicting purchasing intent.

Extraction of feature importances from the model revealed that the feature corresponding to the addition of products to users’ carts was highly important in predicting purchasing intent (0.895 out of 1). A discount marketing strategy focused on cart abandonment was proposed as a possible application of the model to convert browsers with high purchasing intent into customers. Simple calculations demonstrated that the additional revenue gained from the strategy was 400% of the cost of implementing the discount, providing a basis to claim that the business could employ the use of such prediction models to increase revenue.

In future, the conclusions of this study could be validated through a real trial to test the robustness of the approach in real conditions. It is also recognised that the engineered features were few and simple due to the rudimentary nature of the raw data, and that the methodology could be further enriched by considering more features such as number of searches performed by the user and clickthrough rates during visits.

8. References

[1] J. Verdon, “Global E-Commerce Sales To Hit $4.2 Trillion As Online Surge Continues, Adobe Reports,” Forbes, 27 April 2021. [Online]. Available: https://www.forbes.com/sites/joanverdon/2021/04/27/global-ecommerce-sales-to-hit-42-trillion-as-online-surge-continues-adobe-reports/?sh=14404f7150fd. [Accessed 23 June 2021].

[2] M. Evans, “Global E-Commerce Market To Expand By $1 Trillion By 2025,” Forbes, 25 May 2021. [Online]. Available: https://www.forbes.com/sites/michelleevans1/2021/03/25/global-e-commerce-market-to-expand-by-us1-trillion-by-2025/?sh=3e3a83e06cc0. [Accessed 23 June 2021].

[3] J. Clement, “COVID-19 impact on global retail e-commerce site traffic 2019-2020,” 3 November 2020. [Online]. Available: https://www.statista.com/statistics/1112595/covid-19-impact-retail-e-commerce-site-traffic-global/. [Accessed 23 June 2021].

[4] Shopee, “What are the short term methods to boost Conversion Rate (CR)?,” [Online]. Available: https://seller.shopee.sg/edu/article/2378. [Accessed 23 June 2021].

[5] Lazada, “Boost Product Visibility To Increase Conversion Rate,” 3 July 2020. [Online]. Available: https://sellercenter.lazada.sg/seller/helpcenter/boost-product-visibility-to-increase-conversion-rate-11280.html. [Accessed 23 June 2021].

[6] B. Requena, G. Cassani, J. Tagliabue, C. Greco and L. Lacasa, “Shopper intent prediction from clickstream e-commerce data with minimal browsing information,” Scientific Reports, 12 October 2020. [Online]. Available: https://www.nature.com/articles/s41598-020-73622-y. [Accessed 23 June 2021].

[7] M. Hendriksen, E. Kuiper, P. Nauts, S. Schelter and M. de Rijke, “Analyzing and Predicting Purchase Intent in E-commerce: Anonymous vs. Identified Customers,” 16 December 2020. [Online]. Available: https://arxiv.org/abs/2012.08777. [Accessed 23 June 2021].

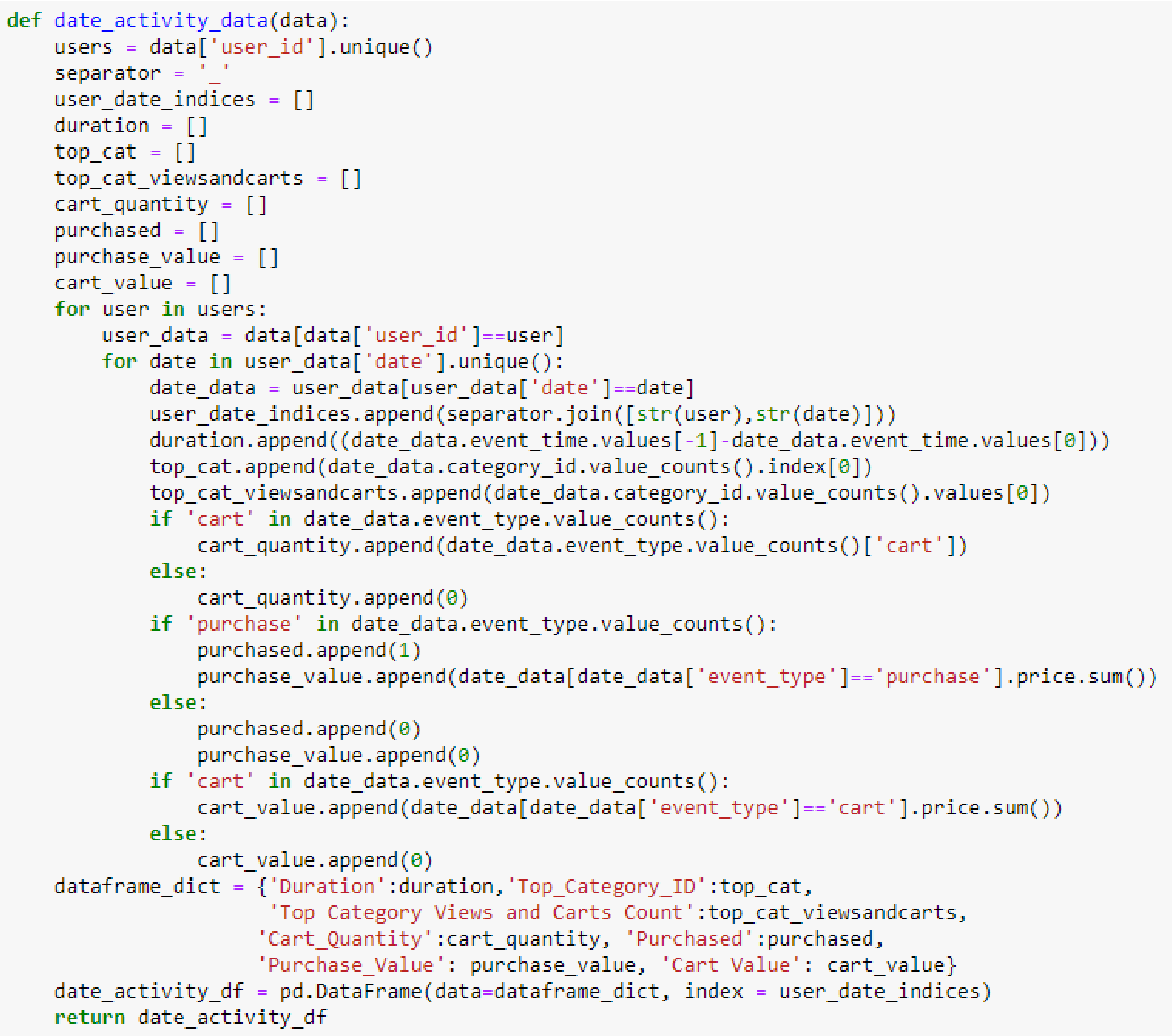
[8] R. Esmeli, M. Bader-El-Den and H. Abdullahi, “Towards early purchase intention prediction in online session based retailing systems,” 19 December 2020. [Online]. Available: https://link.springer.com/article/10.1007/s12525-020-00448-x. [Accessed 23 June 2021].

[9] REES46 Marketing Platform, “eCommerce events history in electronics store,” REES46 Marketing Platform, 29 March 2021. [Online]. Available: https://www.kaggle.com/mkechinov/ecommerce-events-history-in-electronics-store. [Accessed 10 June 2021].

[10] Baymard Institute, “44 Cart Abandonment Rate Statistics,” Baymard Institute, 20 December 2020. [Online]. Available: https://baymard.com/lists/cart-abandonment-rate. [Accessed 23 June 2021].

Appendix

Function for Feature Engineering from Raw Data



```python
def date_activity_data(data):
    users = data['user_id'].unique()
    separator = '_'
    user_date_indices = []
    duration = []
    top_cat = []
    top_cat_viewsandcarts = []
    cart_quantity = []
    purchased = []
    purchase_value = []
    cart_value = []
    for user in users:
        user_data = data[data['user_id']==user]
        for date in user_data['date'].unique():
            date_data = user_data[user_data['date']==date]
            user_date_indices.append(separator.join([str(user),str(date)]))
            duration.append((date_data.event_time.values[-1]-date_data.event_time.values[0]))
            top_cat.append(date_data.category_id.value_counts().index[0])
            top_cat_viewsandcarts.append(date_data.category_id.value_counts().values[0])
            if 'cart' in date_data.event_type.value_counts():
                cart_quantity.append(date_data.event_type.value_counts()['cart'])
            else:
                cart_quantity.append(0)
            if 'purchase' in date_data.event_type.value_counts():
                purchased.append(1)
                purchase_value.append(date_data[date_data['event_type']=='purchase'].price.sum())
            else:
                purchased.append(0)
                purchase_value.append(0)
            if 'cart' in date_data.event_type.value_counts():
                cart_value.append(date_data[date_data['event_type']=='cart'].price.sum())
            else:
                cart_value.append(0)
    dataframe_dict = {'Duration':duration,'Top_Category_ID':top_cat,
                      'Top Category Views and Carts Count':top_cat_viewsandcarts,
                     'Cart_Quantity':cart_quantity, 'Purchased':purchased,
                     'Purchase_Value': purchase_value, 'Cart Value': cart_value}
    date_activity_df = pd.DataFrame(data=dataframe_dict, index = user_date_indices)
    return date_activity_df
```