CS5228
Knowledge Discovery and Data Mining

Final Project – Predicting Online Shoppers' Purchase Intentions

Submission Date: 7th May 2020

Liu Siyuan	A0119450L
Mak Wai Teng	A0114402B
Wang Hao	A0194959A

Link to Github: https://github.com/hwang018/CS5228

Table of Contents

1. Introduction

With the advancement of technologies and growing Internet population in the modern world, there is a significant growth in the online retail sales globally. Between 2014 and 2018, the global retail spending has experienced a substantial growth from 5.9 to 8.8 percent, especially with the increasing trend of digital buyers' penetration (Figure 1.1). Moreover, with the recent global COVID-19 pandemic outbreak, majority of the population has been working remotely from home. As a result, it has changed the buying behavior of customers in the age of telecommuting, where there is a major shift in in-store traffic to online shopping (Abramovich, 2020). This gives us the motivation to embark on this topic in predicting online customers' purchase intentions, as online retail is a highly lucrative industry and it is important to help e-commerce businesses to drive revenue in this competitive industry.

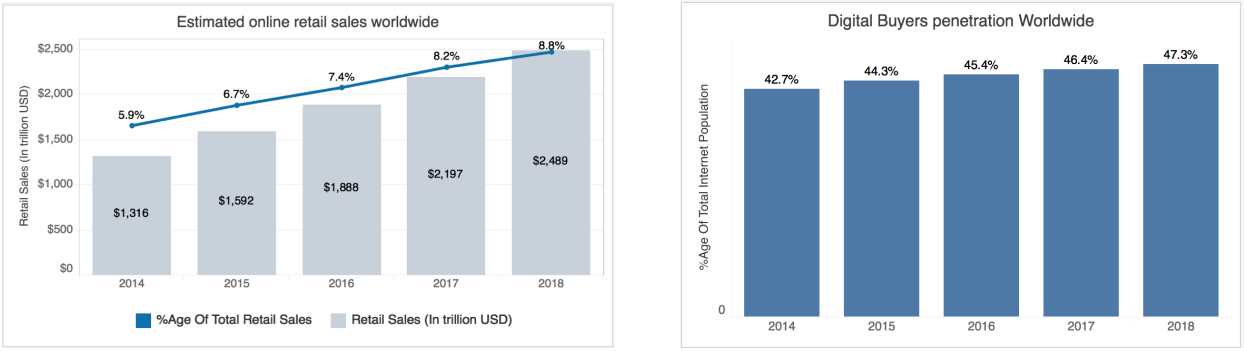


Figure 1.1 – Growth of Online Retail Sales and Digital Buyer Penetration (Saleh, 2018)

Hence, with the aim of predicting the possibility of online shoppers buying from the e-commerce site and significance of variables affecting the buying decision, we have planned and investigated the following approaches to the problem that online businesses faced.

2. Data

2.1 Source of Data

The dataset is sourced from UCI Machine Learning Repository which consists of 12,330 sessions from an online shopping website where each session belongs to a unique individual customer visiting the website (Sakar, Polat and Katircioglu, 2018). In this project, we have only considered data of one-year period for analysis, from February to December, to minimize the effect of time series correlation.

The downloaded dataset consists of 10 numerical and 8 categorical attributes. The 10 numerical attributes include *"Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related", "Product Related Duration", "Bounce Rate", "Exit Rate"*, *"Page Value"* and *"Special Day"*. While the other 8 categorical attributes include *"Operating System", "Browser", "Region", "Traffic Type", "Visitor Type", "Weekend", "Month"* as well as *"Revenue"* as the dependent variable of whether an online shopper will make a purchase and generate revenue for the e-commerce site. The detailed definitions of the variables can be found in Appendix A: Data Definition.

2.2 Data Exploration

Before going into the methodologies and approaches in predicting buyers' purchase intention, we have first conducted descriptive analytics to explore our input data. This allows us to better understand how the input feature correlates with the possibility of whether an online shopper will make a purchase from the website through data visualization.

2.2.1 Dependent Variable

Firstly, we have identified *"Revenue"* as our dependent variable, where it is a binary data element indicating whether revenue will or will not be generated for the website. From the distribution of all sessions in the dataset, we observed that there are only 15.47% of the customers generating revenue with the online website. This shows that the target class is relatively imbalanced, and we will be explaining further in Section 4.1 on how this observation will affect the way we evaluate our model result.

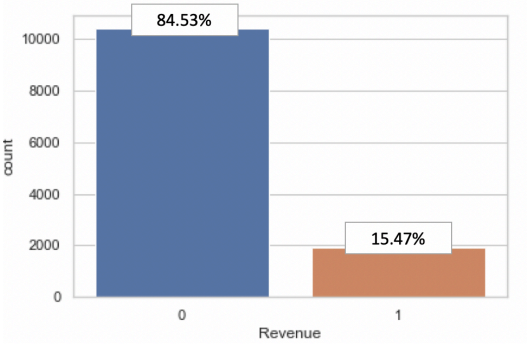


Figure 2.1 – Distribution of Customers with Revenue Generated

2.2.2 Independent Variable

Numerical Variables

Next, we also studied the distribution of the other numerical and categorical independent variables and how they correlate with the dependent variable.

"Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day). Interestingly, from Figure 2.2(b), we observed that customer has a higher tendency to make purchases either slightly ahead of the special event or on the actual day itself. This is probably due to the additional delivery time for the customers to receive the products. However, there is still a group of customers who prefer to do last minute shopping. This could be a potential opportunity for the online businesses to provide same day delivery option to drive more sales on these special days.

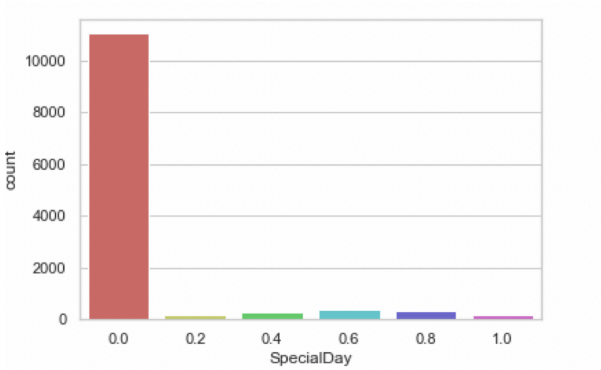


Figure 2.2(a) –
Number of Sessions on Special Day

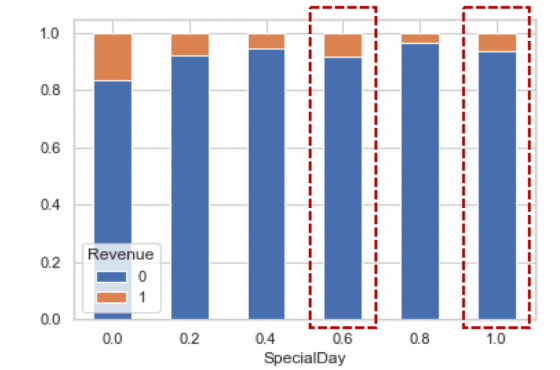


Figure 2.2(b) –
Percentage of Revenue by Special Day

Additionally, we have also plotted the distribution of the other 9 session-related numerical features against the *"Revenue"* dependent variable. Figure 2.3 illustrates a long tail effect and the distribution of data is highly skewed. This is the same observation for *"Special Day"* as shown in Figure 2.2(a). Such skewness may violate model assumptions or may impair the interpretation of feature importance. As a result, it is essential to transform the numerical features before modelling. Details will be elaborated in Section 2.3.3 on how we handle these numeric values.

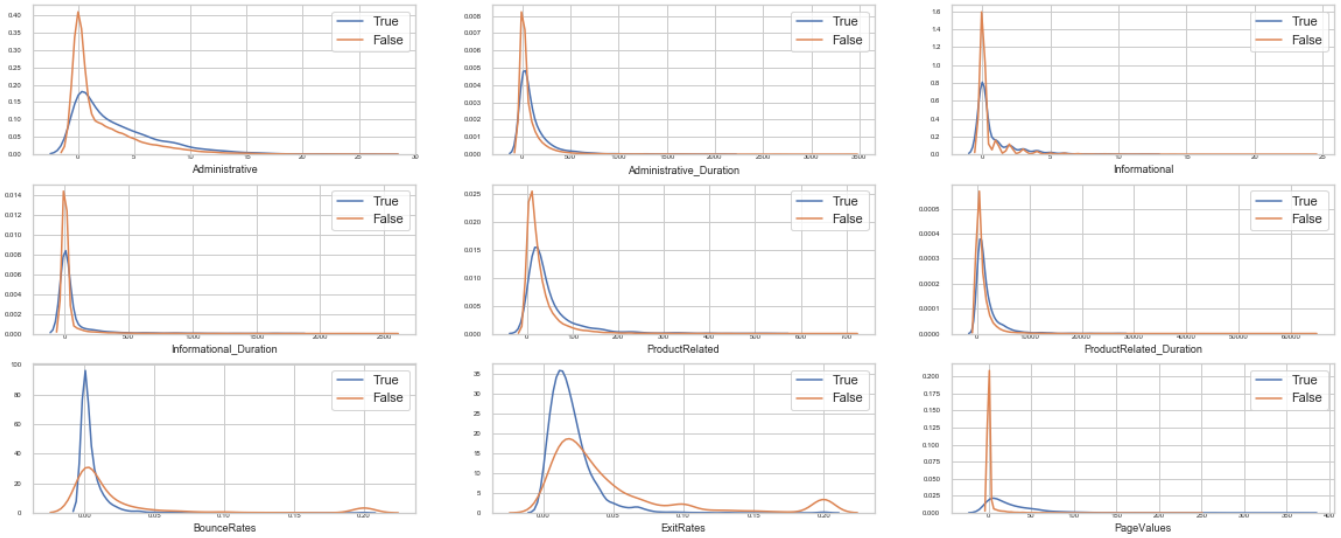


Figure 2.3 – Distribution of the Numerical Features by Revenue

Categorical Variables

"Weekend" is a binary categorical variable that we are interested to explore and identify if the day of the week plays a role in generating revenue. From Figure 2.4(a), it shows that there is a higher traffic volume on weekday than on weekend. However, this is an expected observation as there are more days in weekday than in weekend. Additionally, when we look at the purchase rate, we observed that there is a higher percentage of customers making a purchase on weekend as compared to a weekday (Figure 2.4(b)). This is an important variable as it helps online businesses to decide if they should launch more marketing campaigns or promotional events on a specific day of the week.

Figure 2.4(a) –
Number of Sessions on Weekend

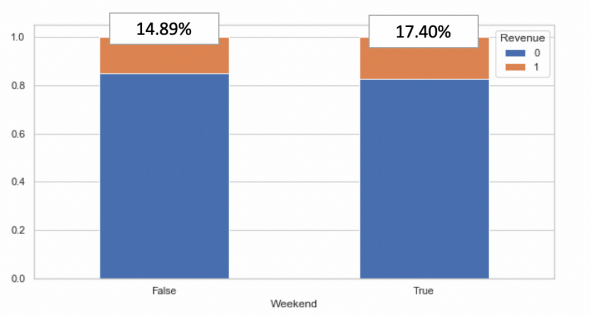


Figure 2.4(b) –
Percentage of Revenue by Weekend

On top of *"Weekend"*, we also plotted out the *"Month"* variable to find out if the period in a year has any impact on the retail sales. Figure 2.5(a) shows that there is no traffic or sales generated in the month of January and April, which could be due to the way data sampling was being conducted. Nonetheless, we still observed meaningful insights where there is a spike in traffic volume in March, May, November and December. This could be due to upcoming events like Mother's Day in May and Thanksgiving Day in November where customers are shopping more often for gift. In terms of the conversion rate, we can see from Figure 2.5(b) that there is a significantly higher percentage of revenue generated by customers in November where 25.35 percent of the sessions eventually got converted into actual sales. This could be attributed to the attractive discounts during Black Friday or Cyber Monday promotions run by the e-commerce store.

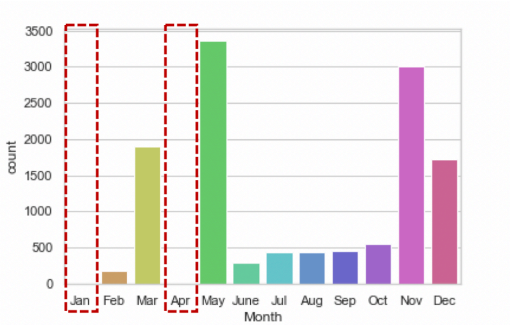


Figure 2.5(a) –
Number of Sessions by Month

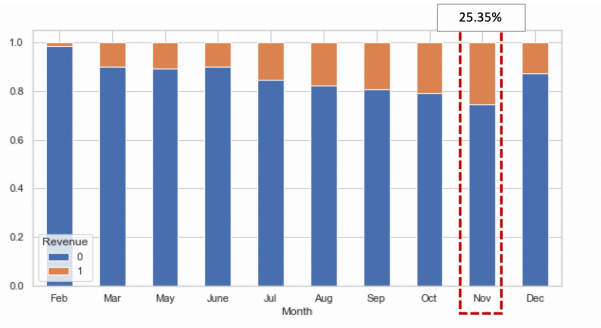


Figure 2.5(b) –
Percentage of Revenue by Month

Next, looking at the *"Visitor Type"* variable, majority of the traffic is driven by returning visitors (Figure 2.6(a)). However, when we compare against the revenue generated, Figure 2.6(b) shows that there is a highly likelihood of 24.91 percent for the new customers to make a purchase in the session than the returning customers, which is merely 13.93 percent. This could be due to new user benefits provided by the online store to attract customers in making their first purchase. This is potentially a good opportunity for online businesses to increase customer base by targeting these new visitors.

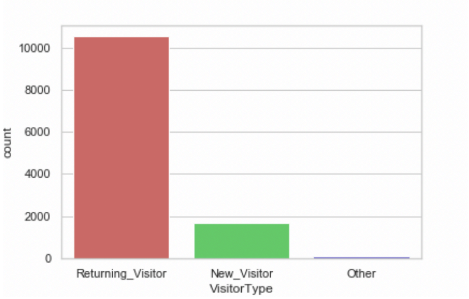


Figure 2.6(a) –
Number of Sessions by Visitor Type

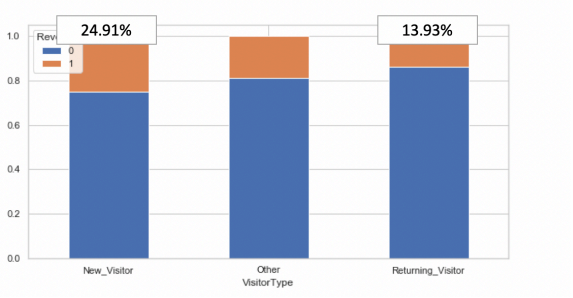


Figure 2.6(b) –
Percentage of Revenue by Visitor Type

We have also done similar descriptive analysis on other variables such as *"Operating System", "Browser", "Region"* and *"Traffic Type"* which can be found in Appendix B: Data Visualization.

Last but not least, we also look at the correlations between the different input variables in the dataset. Figure 2.7 is a correlation plot that shows us that there is a high correlation among the independent variables such as different types of pages the user visited, and the duration spent in those pages. We also observed that that "*Page Value*" has a very high positive correlation of 0.49 with the dependent variable *"Revenue"*. On the other hand, we are seeing a highly negative correlation between *" Exit Rate"* and *" Bounce Rate"* of -0.21 and -0.15 with *"Revenue"* respectively. This shows that the higher the likelihood of users exiting the page, they are less likely for them to make a purchase.

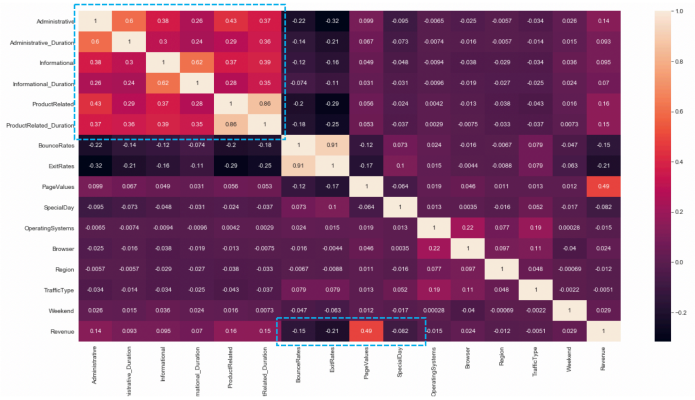


Figure 2.7 – Correlation Plot of Input Features (dependent and independent variables)

2.3 Data Preprocessing

After having a good understanding of the dataset, it is important to first preprocess the data in preparation for the next step of prediction.

2.3.1 Handle Missing Data

In the dataset, we observed that out of the total 12,330 number of user sessions, there are 14 records with missing data values in the dataset. Since there is only 0.1 percent of records with missing values, we have decided to fill those empty values with the columnar mean instead of dropping them.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates
Total	14	14	14	14	14	14	14
Percent	0.113544201135442	0.113544201135442	0.113544201135442	0.113544201135442	0.113544201135442	0.113544201135442	0.113544201135442
Types	float64	float64	float64	float64	float64	float64	float64

Figure 2.8 – Missing Records in Dataset

2.3.2 Handle Categorical Data

As mentioned in the above section, the dataset consists of a total of 8 categorical features, including one dependent variable *"Revenue"*. Hence, to encode and transform the categorical variables, we have experimented two methods, namely one-hot encoding and mean target encoding.

One-Hot Encoding

Firstly, we have implemented the one-hot encoding method to all categorical variables and obtained a total number of 68 features. Each newly transformed binary variable is a unique value of the original categorical feature. These transformed variables are eventually passed as input features to our benchmark model in Section 3.

Mean Target Encoding

On top of the one-hot encoding method, we also perform target encoding to convert categorical columns to numeric by replacing a categorical value with the mean of the target variable. We believed that this could possibly help to improve on the machine learning accuracy, as it is typically challenging for algorithms to deal with high cardinality features.

We performed a cardinality check on the categorical features against the predefined threshold, and we observed that four of the variables with high cardinality, namely *"Operating Systems", "Browser", "Region"* and *"Traffic Type".* Therefore, mean target encoding is applied onto these four variables. The other categorical features such as *"Special Day", "Visitor Type"* and *"Month"* are one-hot encoded. With a combination of the one-hot and mean target encoding techniques, the final dataset yields a total of 35 features. The reduced number of features could possibly avoid potential sparsity in the data. This method ensures that labels are not randomly assigned, and it is correlated to the target variable. As a result, we believed that this will be better than the traditional on-hot and label encoding and will help our prediction models in using the labels more efficiently.

2.3.3 Handle Numerical Data

Based on the analysis done in Section 2.2.2, we realized that most of the numerical variables are highly skewed. Therefore, we have implemented feature normalization on those numerical variables, as some algorithms might be sensitive to the scale of the features. In this study, we have conducted experiments on both Standard Scaler and Min-Max Scalar from Scikit-Learn. However, the model output has shown that standardization with Standard Scaler yields a better result than Min-Max Scalar. The numerical features will therefore be normalized by its columnar mean and standard deviation. These variables will then be passed as the input variables into our final prediction models in Section 3.1.1.

2.3.4 Feature Engineering

With the features available in the dataset, we have created two additional features – *"Special Day Scaled"* as well as *"Mth Conv Prob"*. Firstly, we have created *"Special Day Scaled" from "Special Day"* because online impressions typically decay exponentially (Dotan, 2015). Without transforming this feature, our model will not be able to learn and capture this exponential relationship. As a result, we deliberately applied an exponential decay transformation on the original *"Special Day"* feature, to help the model in learning this relationship.

Secondly, the *"Mth Conv Prob"* feature is transformed from the *"Month"* variable where it represents the corresponding month's conversion rate. We converted the categorical *"Month"* variable to numeric by replacing the binary value with the conversion rate of the target variable. For example, if 20 percent of the training records in December generated a revenue for the business, then the corresponding "*Mth Conv Prob*" with be 0.2 for December records in both training and testing data.

2.4 Train-Test-Split

In order to prevent overfitting, testing should be done on unseen data. Therefore, given that our entire dataset consists of 12,330 user sessions, we have decided to split this data into both training and testing data. With a ratio of 80-20 split, we have obtained a total of 9864 records for training (80 percent) and 2466 records for testing (20 percent) respectively.

3. Methodology

In this project, we have experimented two classification approaches to identify if a customer will make a purchase – the traditional machine learning and deep learning neural network. In both approaches, there is a slight difference in the way we encoded the categorical features.

For the traditional machine learning approach, our key focus was to perform Bayesian search in parameters tuning to obtain the best model with an ideal speed performance. Whereas for the deep learning approach, we aim to identify the optimal neural network structure, as well as the optimal hyperparameters that yield the best result. In both approaches, Area Under the Curve (AUC) scores obtained from testing set will be used as the main evaluation metric due to the relatively imbalanced target class in our dataset. The detailed procedures are illustrated in Figure 3.1.

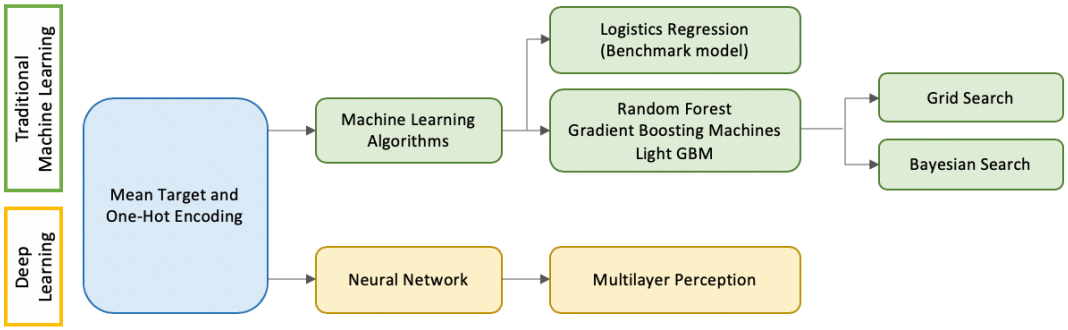


Figure 3.1 – Methodology Summary

3.1 Traditional Machine Learning

As part of the process in building a machine learning model, we first created a simple benchmark model using Logistic Regression with the default parameters. In this model, only one-hot encoding is applied on the categorical features, with no additional transformation of other features. Without any parameters tuning and regularization, our benchmark model has achieved an AUC score of 0.85 (Figure 3.2).

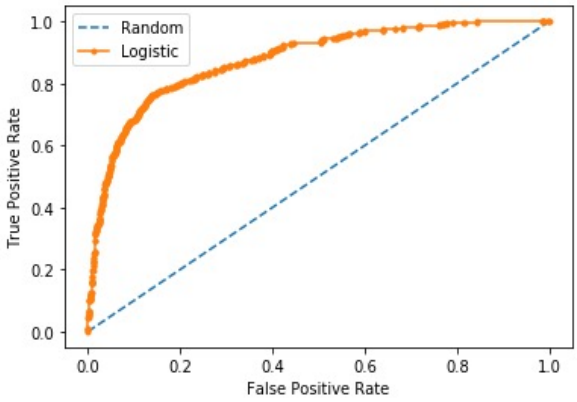


Figure 3.2 – AUC of Logistic Regression classifier (Test AUC = 0.85)

3.1.1 Model Selection and Parameters Tuning

After obtaining a baseline accuracy, we then proceed to explore other alternative models with higher complexity as well as to improve on the result by tuning the parameters using Grid Search.

Firstly, instead of using the one-hot encoded features, we leveraged on the categorical features that we have previously transformed in Section 2.3.2 using mean target encoding. Additionally, the two transformed features that we have created in the feature engineering step are also passed as an input variable into model.

Next, besides data transformation, we have also explored other possible tree-based algorithms such as Random Forest, Gradient Boosting Machines (GBM) and LightGBM (LGBM). Within each of these models, we have also performed parameters tuning using Grid Search and 5-Fold Cross Validation to obtain the best model result. Result in Table 3.1 has shown that LGBM yields the highest AUC score among the other models. However, it is also the slowest algorithm where the cost to run the LGBM model is almost 20 times higher than that of Random Forest. Especially with the increased complexity of the model, the cost of build the model could be significantly more expensive. The traditional Grid Search algorithm is a brute force method that search through all different possible combinations of the parameters which could be inefficient when running large and complex model. Therefore, to improve on the performance, we have explored other more efficient approaches in tuning the model parameters.

3.1.2 Performance Improvement

Given that the LGBM model produces the best outcome from the previous section, we tried to perform Bayesian search based on that model. The Bayesian optimization methodology is a much more efficient algorithm in parameter search, where it is capable of finding the optimal search space and avoid spending unnecessary time in regions that gives poor performance in the objective function. Result from Table 3.1 and Figure 3.3(a) has shown that the LGBM model is still able to achieve the best AUC score of 0.932 using Bayesian tuning. Additionally, the parameter searching cost has reduced by more than half from 428 to 205 seconds in Figure 3.3(b).

Models	Best AUC (test)	Grid Search Cost	Bayes Search Cost
Logistic Regression	0.85	N.A.	N.A.
Random Forest	0.926	28 s	N.A.
GBM	0.931	358 s	N.A.
LGBM	0.932	428 s	205 s

Table 3.1 – AUC scores and Speed performance across models

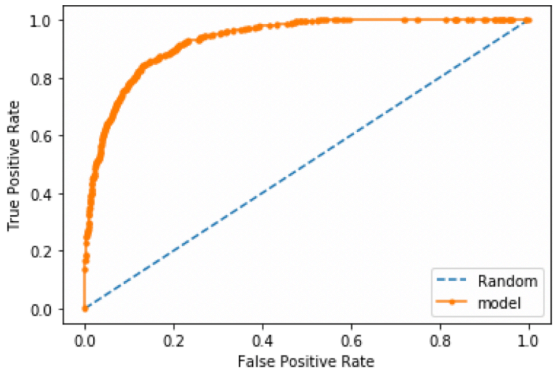


Figure 3.3(a) –
AUC of tuned LGBM (Test AUC = 0.932)

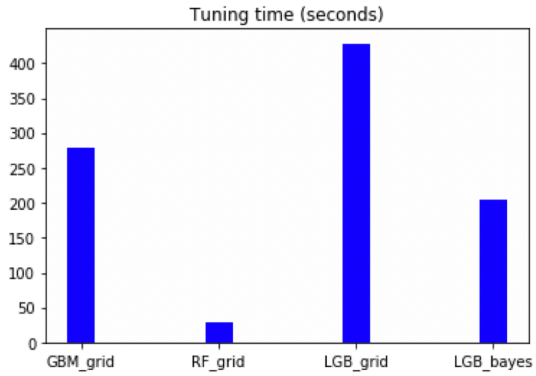


Figure 3.3(b) –
Tuning Cost in Seconds by Classifier

Thus, we have adopted the best performing LGBM model that is tuned using Bayesian search as our final machine learning model. This model is capable in producing the highest accuracy with the best AUC score of 0.932 and is able to run at a much efficient and optimized manner.

3.2 Deep Learning and Neural Network

Besides the traditional machine learning algorithms, we also explored the possibility of using neural networks for classification, which allows for a higher degree of flexibility in fitting the prediction models. The input features used in this Multilayer Perception (MLP) model were pre-processed in the previous section where numerical features were transformed using Standard Scalar and categorical features being one-hot encoded. Standardization of the numerical features is important as the weights of the neurons are highly sensitive to the scale of the features.

3.2.1 Model Selection and Parameters Tuning

To decide on the optimal network architecture, we have tried with N-layers of networks with N ranging from 2 to 5. However, our experiment shows that networks with more than three layers did not result in a significant performance improvement, but it is taking a much longer time to train. Given the trade-off between the network complexity and training speed, our final neural network model consists of three layers as shown in Figure 3.4. The first layer takes in input of 35 dimensions and possesses 50 neurons, and the second layer consists of 10 neurons, followed by the last output layer with 1 neuron. Rectified Linear Units (ReLU) activation function was applied to the input and hidden layers, while the Sigmoid function was applied to the output layer. This produces an output with a value between 0 and 1, representing the probability of whether revenue will be generated.

We have tested and tuned several commonly used optimizers, such as RMSProp, Adam, Adamax and Nadam for comparison. Result has shown that Nadam, which is a combination of RMSProp with Nesterov momentum, achieves the best performance compared to other optimizers. Therefore, we will be adopting Nadam in our neural network experiments.

While neutral network models provide a higher degree of flexibility, we could also face the problem of overfitting. To address this issue, 20 percent of the training set is being set aside as validation and will not involve in the model training. With this, we could examine how losses behave with different parameter settings. Moreover, we also tried to tune the learning rate of the optimizers, as well as the batch size to see how these hyperparameters affect the performance. Other than that, an early stopping mechanism, with 20 epochs patience and dropout are also incorporated in the model to improve performance and reduce overfitting. The model weights are also restored to the best weights at the end of training.

We explored several possible combinations of batch size and learning rate, where learning rate takes value from 1e-3, 1e-4 and 1e-5, while batch size takes value from 10, 25, 100. From our trials, we

observed that smaller batch size and lower learning rate could help to mitigate overfitting, though at the expense of longer training time. Figure 3.4(a) depicts a fluctuating validation loss despite showing a decreasing training loss. It shows that the model is severely overfitting with learning rate equal to 1e-3 and batch size 100. When we decrease the training rate to 1e-5, and reduce the batch size to 10, much smoother and consistent decline in both training and validation loss are observed (Figure 3.4(b)). However, it will also require more epochs to converge. Taking the performance gain and time cost into consideration, we decided to fix the learning rate at 1e-4 and the batch size at 10.

Figure 3.4(a) – Model loss with learning rate 1e-3 and batch size 100

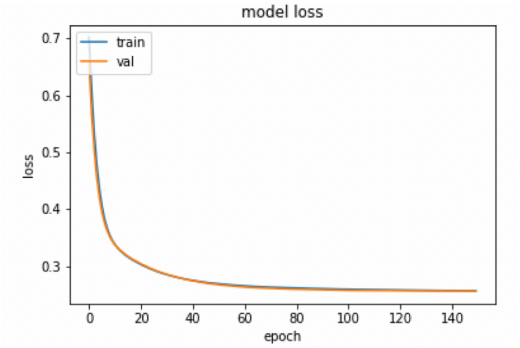


Figure 3.4(b) – Model loss with learning rate 1e-5 and batch size 10

As a result, the final neural network model has obtained a high AUC score of 0.924 on the unseen testing set (Figure 3.5). With the model outcome from the traditional machine learning method as well as the deep learning neural network model, we have decided to build an ensemble model on top of it. Result will be evaluated in Section 4.1.

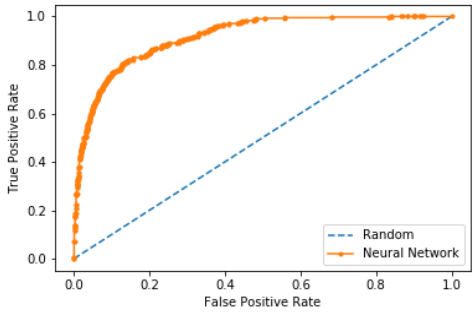


Figure 3.5 – AUC of the best neural network model (Test AUC = 0.924)

4. Result

4.1 Model Evaluation

From the data exploration stage, we understand that our target class is relatively imbalanced. As our objective in this study is to correctly classify the minority class of sessions with revenue generated, we will be leveraging on the AUC score as the performance metrics, which is a combination of sensitivity and specificity, on top accuracy.

Our final ensemble models of the LGBM and MLP neural network, yield an AUC score of 0.927, which is relatively similar to the previous results. Although the score did not improve significantly, the ensemble methods could reduce the generalization error of the prediction on future unseen data by minimizing the errors caused by noise, bias and variance.

4.2 Feature Importance

From Figure 4.1, we observed that 5 out of the top 10 important features from the LGBM model are page-related information – *"Administrative", "Product Related", "Administrative Duration", "Product Related*

Duration" and *"Informational Duration".* This shows that there is a high correlation between the pages and duration that the shoppers visited against the revenue. The e-commerce owners could focus on enhancing the aesthetic and content of their site to increase their attractiveness and encourage more users to browse and stay on the page. This could potentially help them in generating more revenue.

Additionally, the feature importance chart also tells us that the metrics from Google Analytics such as *"Bounce Rate", "Exit Rate"* and *"Page Value"* are very important in predicting buyers' purchase intention. This will give the online businesses more context in determining how their webpages perform as compared to their visitors' expectation. With this insight, e-commerce sites can better target their customers with the page that is well-suited for them.

Other important features identified are *"Mth Conv Prob"* and *"Traffic Type".* From this result, we understand the importance of the month of visit in predicting revenue generation. This is probably due to the attractive promotions driven by mid-year or year-end sales. This gives the online site owners an indication of the optimal period to invest in their marketing campaigns. The different types of traffic coming from social media, referral or even direct traffic also indicates the successfulness of the campaigns launched by online businesses. This provides insights to the businesses in investing on more targeted marketing to effectively reach their online target audience.

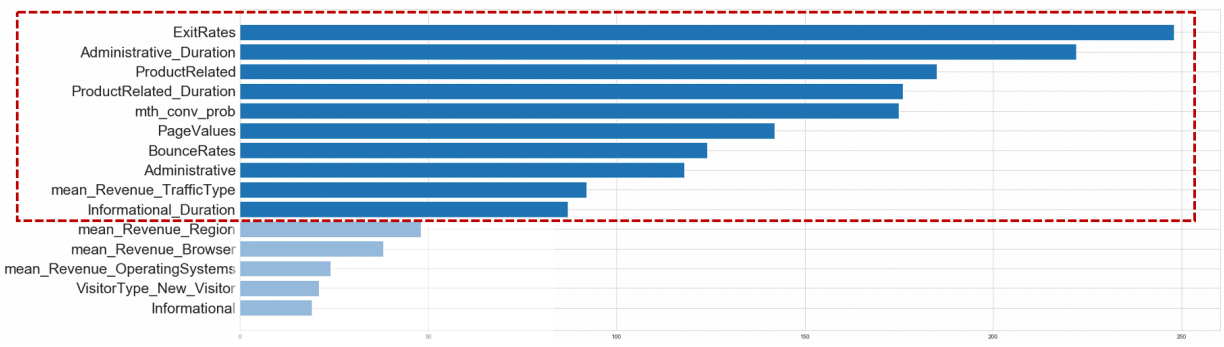


Figure 4.1 – Feature Importance based on tuned LGBM

5. Limitations and Future Improvements

Despite having a thorough experiments on various algorithms and extensive parameter tuning, there are several limitations that our study failed to address.

Firstly, due to the way sampling was conducted in the downloaded dataset, there are missing records for certain period of the year, such as January and April, which could be an important indicator of revenue generation. Through our data exploration, we observed that the volume of data in each month is also drastically different. Thus, to further improve our model as an enhancement, we could possibly try stratified sampling in our train-test-split process instead of splitting randomly or consider the time series component in our study.

Additionally, since this data source is published publicly online, a lot of the sensitive information such as user-related and merchant-related information are being masked out due to privacy issue. As a result, we are not able to make a lot of meaningful insights without those contexts. We could possibly source for alternative data source or do web scrapping to enhance the current dataset as a future enhancement.

6. Conclusion

In conclusion, in this project we have experimented different approaches such as various traditional machine learning and deep learning methodologies to predict if a customer will or will not generate revenue for the online site. With a high accuracy and AUC scores, our method demonstrates the feasibility of predicting online shoppers' purchase intention using the supervised classification approach.

Reference

Abramovich, G. (March 2020). How COVID-19 is Impacting Online Shopping Behavior. Available: https://theblog.adobe.com/how-covid-19-is-impacting-online-shopping-behavior/

Dotan, G. (November 2015). What is eCPM Decay and How to Keep Ad Revenue High. Available: https://blog.soomla.com/2015/11/ecpm-decay.html

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018). Available: https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset

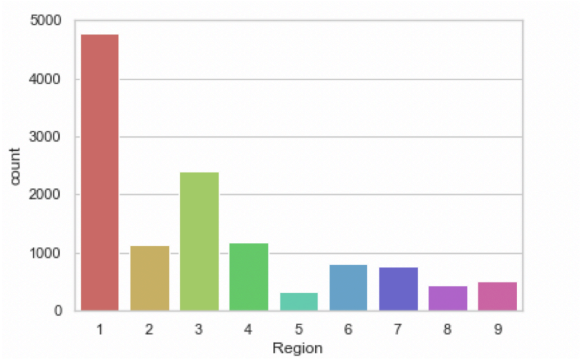
Saleh, K. (2018). Global Online Retail Spending Statistics and Trends. Available: https://www.invespcro.com/blog/global-online-retail-spending-statistics-and-trends/

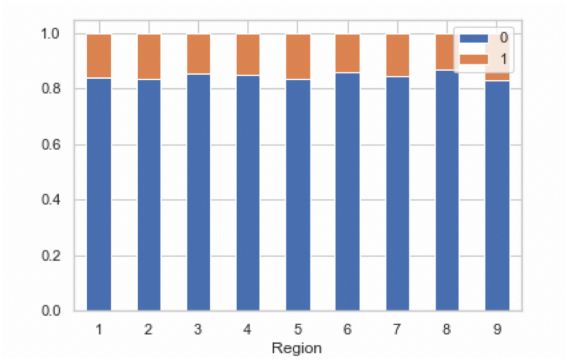
Appendix A: Data Definition

	Data Type	Variable Name	Description
Dependent Variable	Categorical	Revenue	Binary feature whether revenue will or will not be generated
Independent Variables	Categorical	Operating System	N/A
		Browser	N/A
		Traffic Type	N/A
		Region	N/A
		Month	Month of the session is visited (e.g Jan, Feb, Mar)
		Weekend	Binary feature whether the session is visited on weekend
		Visitor Type	Type of customer visiting the webpage (e.g. Returning Visitor, New Visitor and Other)
	Numerical	Administrative	Number of administrative type of pages visited by the visitor in that session
		Administrative Duration	Duration of administrative type of pages visited by the visitor in that session
		Informational	Number of informational type of pages visited by the visitor in that session
		Informational Duration	Duration of informational type of pages visited by the visitor in that session
		Product Related	Number of product-related type of pages visited by the visitor in that session
		Product Related Duration	Duration of product-related type of pages visited by the visitor in that session
		Page Value	Average value for a web page that a user visited before completing an e-commerce transaction
		Exit Rate	Is a feature for a specific web page is calculated as for all pageviews to the page, the percentage that were the last in the session.
		Bounce Rate	Percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session
		Special Day	The closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction.

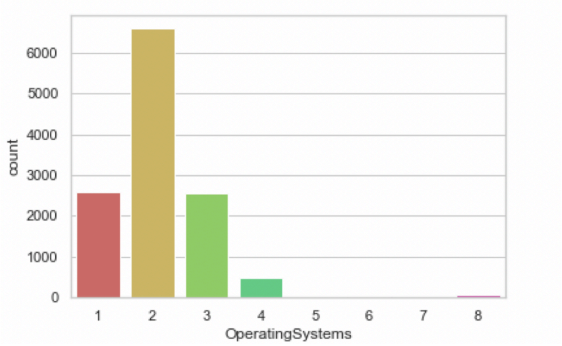
Appendix B: Data Visualization

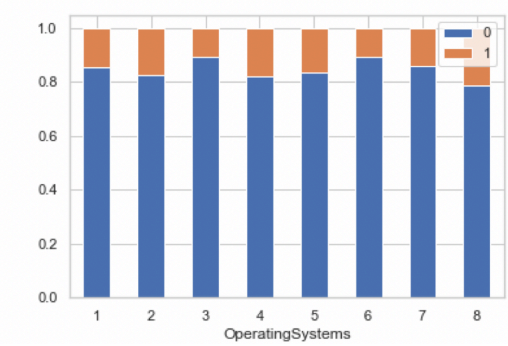
Region



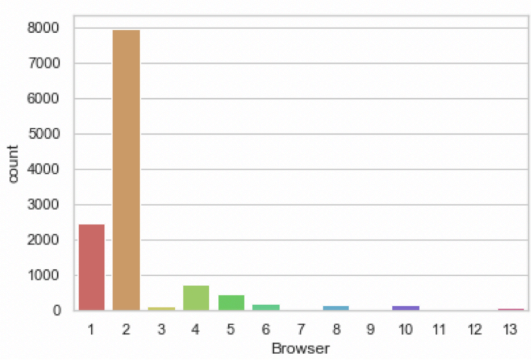


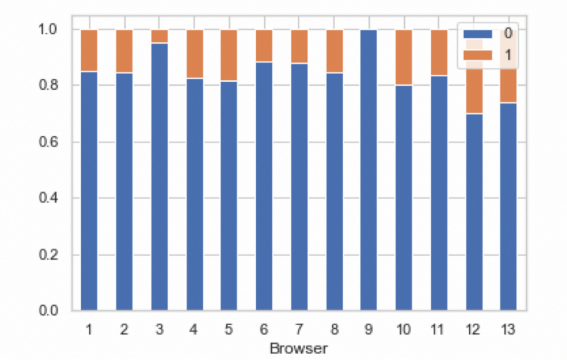
Operating System





Browser





Traffic Type