Title: Predicting Whether An Online Shopper Purchases An Item Based On Their Behaviour

Student ID: S3853674

Student Name: Vignesh Kumar Sathya Murthy
Email ID :S3853674@student.rmit.edu.au

Affiliations: RMIT University.

Date of Report: 24/05/2023

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show I agree to this honor code by typing "Yes": *Yes*.

Table of Contents

1 Abstract

The purpose of this report is to predict whether online shoppers will make a purchase based on features from the "Online Shoppers Purchasing Intention Dataset." The dataset contains information about browsing behaviours and shopper characteristics. Revenue classification models are developed and compared with the help of K-Nearest Neighbours (KNN) and Decision Tree algorithms Overall, the results contribute to enhance our understanding of customer behaviour in the online shopping platform. Businesses can use these insights to optimize their strategy and increase their revenue. Based on the findings, it is recommended that business focus more on product duration information, BounceRates and pagevalue as it contributes more in deciding whether a user will purchase or not.

2 Introduction

Online shopping has experienced tremendous growth in recent years, transforming the retail landscape and presenting new opportunities and challenges for businesses. As more consumers shift towards online platforms, understanding the factors influencing their purchasing behaviour becomes paramount. This project aims to leverage machine learning algorithms and the "Online Shoppers Purchasing Intention Dataset" to gain insights into the browsing behaviour and characteristics of online shoppers and predict their likelihood of making a purchase. By uncovering patterns and trends in customer behaviour, this study seeks to assist businesses in optimizing marketing strategies, enhancing the online shopping experience, and ultimately improving revenue generation.

3 Methodology

3.1 Data Collection

The data for the project was obtained from the “Online Shoppers Purchasing Intention Dataset” available on the UCI Machine Learning Repository. The dataset provides the information about the online shopping users browsing behaviour and their purchasing intentions.

3.2 Data Description

The dataset contains a total of 12,330 instances and 18 attributes. The attributes have both numerical and categorical datatypes. In the dataset, each session belongs to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period (Sakar and Kastro, 2018).

3.3 Data Retrieving and Pre-processing

Upon downloading the dataset to the location PDS_Assignment2/input/online_shoppers_intention.csv, the data is retrieved using the offline method. After retrieving, the data was prepared/pre-processed to ensure its quality and consistency. This included checking for missing values, extra whitespaces, typos, impossible values and outliers.

Missing Value Handling

There were no missing values found in the dataset.

Extra Whitespaces

There were no extra whitespaces in the dataset.

Impossible Value/Typos

Sanity Check was performed in the dataset to check for impossible values. It was free of errors.

For Example, SpecialDay value should be between 0 and 1 (as per the dataset description). Proper months are available in Month column.

Categorical Value Encoding

The dataset contained categorical variables/object like VisitorType and Month. These feature will cause while training the model, if not converted to numeric. Therefore have applied one-hot encoding to convert them into binary indicators (Verma, 2021).

Data Integrity Check

After pre-processing, I have conducted final integrity check to ensure correctness of transformed data and that all the columns had proper datatypes and there were no anomalies present in them.

Duplicate Values

It was found that there were 125 rows having duplicate values. Therefore have removed them from the dataset. New updated data frame consists of 12205 rows.

```
online_shop = online_shop.drop_duplicates()
online_shop.shape
```

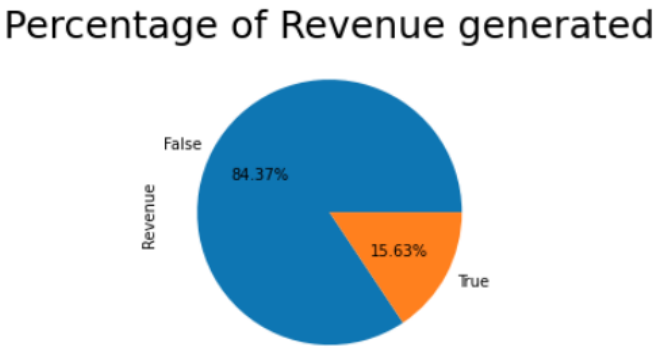
```
(12205, 18)
```

3.4 Data Exploration

I have performed data exploration to gain insight into the datasets structure, distribution and relationship between variables.

3.4.1 Individual Column Analysis

1. **Revenue**



Plotted Pie chart to visualize the percentage of revenue generated through online shopping sessions.

Observation: Around 84 percentage of the people who visit online shopping have not purchased. The percentage of people visiting the website without buying is high.

2. **Weekend**

Bar graph to visualize the number of people shopping on Weekends.

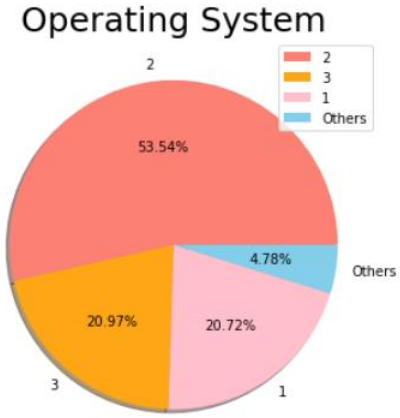
Observation: Since weekday consist of more no. of days, the number of customers are high on weekdays as compared to weekend.

3. Region

Plotting to check the number of users shopping from different regions.
Observation:

- 1st Region has the highest number of customer.
- 3rd Region is half of 1st region.
- All the other regions are relatively low as compared to 1st region
- 5th Region has lowest number of customers.

4. Operating System

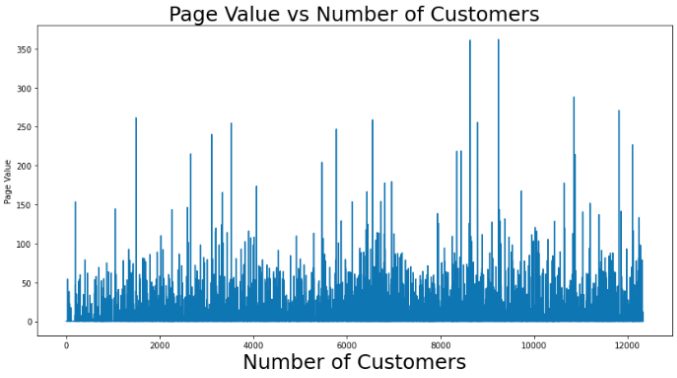


Plotting pie chart to check the types of operating systems used for online shopping.
Observation:

- The top 3 OS contribute the majority percentage of data.

- '2' contributes 50% of the contribution
- '3' and '1' have similar percentage

5. PageValues



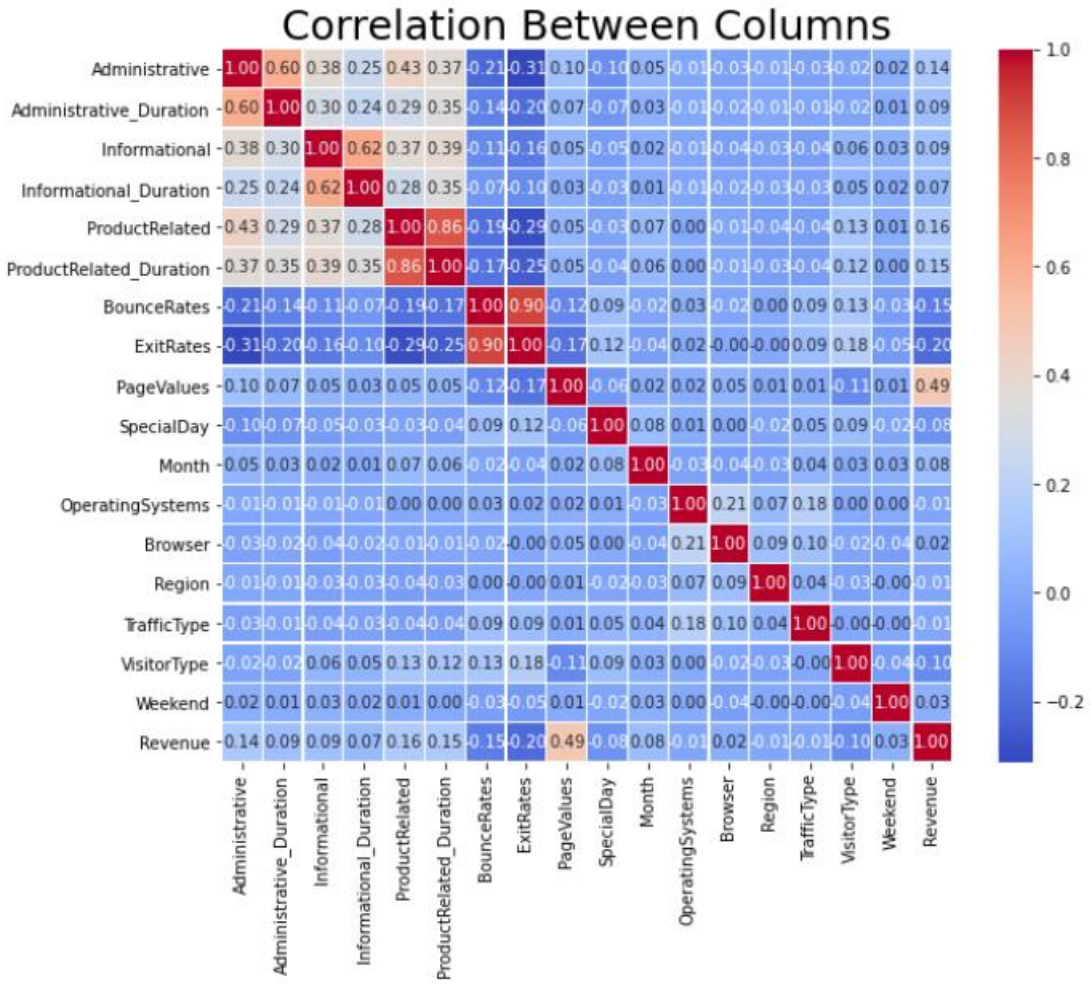
Plotting to check the average time spent by the users before completing a transaction.
Observation:

- The majority of page value are in range 0 to 60.
- There are also customers who have page value more than 100, but it is very low.

3.4.2 Pairwise Column Analysis

This analysis is done to uncover potential relationships between 2 features in the dataset.

1. Heatmap

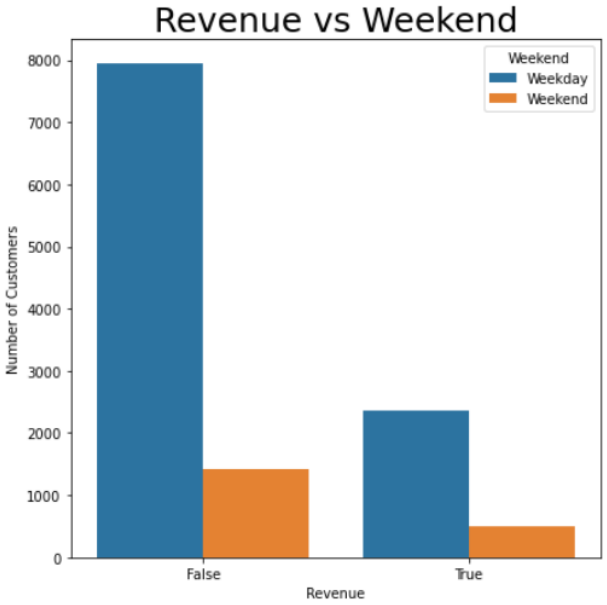


Heatmap is used to visualize the co relationship between different features.

Observation

- The target column Revenue has highest correlation with the PageValues column.
- BounceRates and Exit rate have high dependancy on each other.
- Administrative vs Admministrative_Duration have high corelation.
- Informational vs Informational_Duration have high corelation.
- ProductRelated vs ProductRelated_Duration have high corelation.

2. Revenue vs Weekend

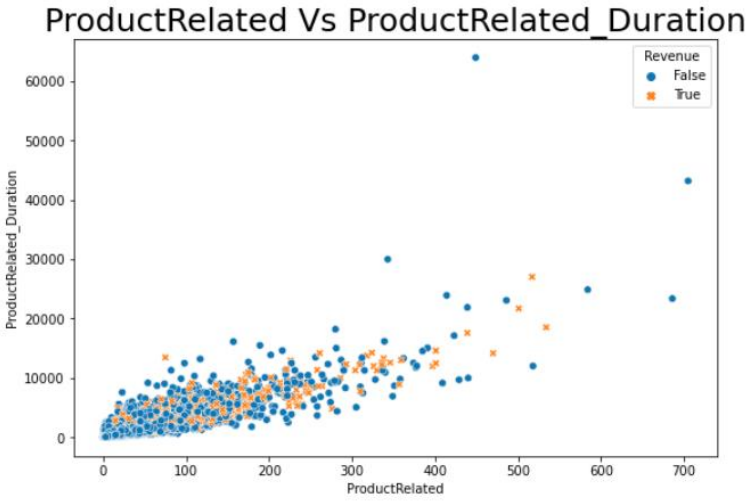


Visualizing to check if the shopping in the weekend effect the chances of purchasing an item.

Observation:

- Users not purchasing an item is always high than purchasing an item whether weekend or not.
- More users shops on weekdays.
- Weekend does not effect a lot in deciding a user purchased an item.

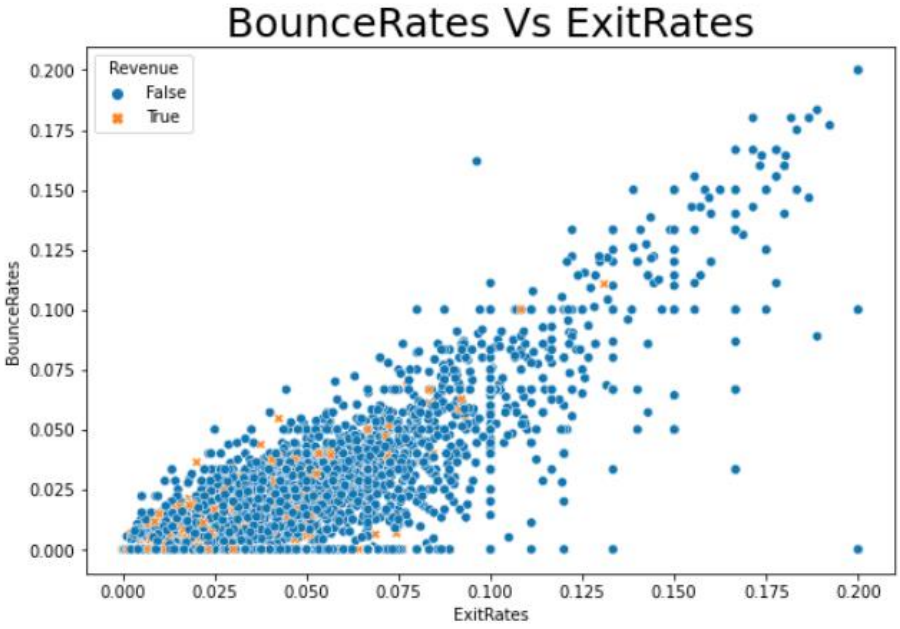
3. ProductRelated vs ProductRelatedDuration



Observation:

- There is a linear relation ship between ProductRelated and its duration. But it cannot linearly predict whether a user will buy a product or not based on only this information.

4. BounceRates vs ExitRates



Observation

- They have linear relation between them.
- It can be said that high bounce rate contributes to high exit rated for certain pages.

3.5 Data Modelling

In this, I have employed K-Nearest Neighbours (KNN) algorithm and Decision Tree for classifying revenue based on the “Online Shoppers Purchasing Intention Dataset”.

3.5.1 K-Nearest Neighbours (KNN) Algorithm

First, I applied the K-Nearest Neighbours algorithm to predict revenue based on the provided dataset. The following steps were performed.

Data Splitting: Before going to the modelling phase, the data is split into 2 parts – Training(70%) and Test(30%). This is done for unbiased evaluation.

Training the model with random k=3 parameters.

I initially trained the KNN model using all columns of the dataset, with a random choice k=3. This allowed me to calculate a baseline performance for classification task. Below have provided the classification report.

```
              precision    recall  f1-score   support

       False       0.89      0.95      0.92      3114
        True       0.56      0.36      0.44       548

    accuracy                           0.86      3662
   macro avg       0.73      0.66      0.68      3662
weighted avg       0.85      0.86      0.85      3662


The accuracy for KNN where k = 3 is 0.8626433642818132
```

The accuracy for this was 86 percent.

Parameter Tuning with Greedy Method for K Value with different weights and p value

To determine the optimal K value for the KNN model, I have employed greedy method by iteratively increasing the k and evaluating the models performance.

With default weight and p Value, below is the accuracy I got for k ranging from 1 and 10

```
Accuracy Scores having k = 1 is: 0.8339705079191698
Accuracy Scores having k = 2 is: 0.8659202621518296
Accuracy Scores having k = 3 is: 0.8626433642818132
Accuracy Scores having k = 4 is: 0.87001638448935
Accuracy Scores having k = 5 is: 0.8681048607318406
Accuracy Scores having k = 6 is: 0.872200983069361
Accuracy Scores having k = 7 is: 0.871381758601857
Accuracy Scores having k = 8 is: 0.8716548334243582
Accuracy Scores having k = 9 is: 0.8705625341343528
Accuracy Scores having k = 10 is: 0.87001638448935
```

With weights='distance' and p=1 value, below is the accuracy I got for k ranging from 1 and 10

```
Accuracy Scores having k = 1 is: 0.8399781540141998
Accuracy Scores having k = 2 is: 0.8399781540141998
Accuracy Scores having k = 3 is: 0.8585472419442928
Accuracy Scores having k = 4 is: 0.8637356635718186
Accuracy Scores having k = 5 is: 0.8702894593118514
Accuracy Scores having k = 6 is: 0.8683779355543418
Accuracy Scores having k = 7 is: 0.8667394866193336
Accuracy Scores having k = 8 is: 0.8675587110868378
Accuracy Scores having k = 9 is: 0.8659202621518296
Accuracy Scores having k = 10 is: 0.8651010376843256
```

Therefore, the best value that I got was when weights='distance' and p=1 and k=5.

Feature Selection with Hill Climbing

Then I performed feature selection using Hill climbing algorithm to find the most relevant features for revenue classification. I initialized the algorithm with the above identified best parameters for KNN. Below is the result I got.

```
Score with 1 selected features: 0.8473511742217368
Score with 2 selected features: 0.8779355543418896
Score with 3 selected features: 0.8809393773894046
Score with 4 selected features: 0.883397050791917
Score with 5 selected features: 0.8844893500819224
Score with 6 selected features: 0.8877662479519388
The best features are in the following indexes [13, 8, 15, 0, 9, 10]
```

The columns that Hill climbing recommended are Region, PageValues, VisitorType, Administrative, SpecialDay, Month.

But this value also changed when I changed my random parameter and ran the model again.

Retraining the model with Selected columns in Hill Climbing

Finally, I retrained the model using the selected subset of columns found in Hill climbing algorithm. The new accuracy that I got is 88 percent (1 percent more than the previous one).

```
The accuracy for KNN after using hill climbing is 0.8896777717094484
```

Classification Report:

	precision	recall	f1-score	support
False	0.92	0.95	0.94	3114
True	0.65	0.56	0.60	548
accuracy			0.89	3662
macro avg	0.79	0.75	0.77	3662
weighted avg	0.88	0.89	0.89	3662

My precision for both the values increased. The precision for True is still less because of the imbalance in the original dataset. The number of True revenue scenarios are very less as compared to ‘True’ scenario.

K- Folds

After using K- Folds to evaluate the performance, the average accuracy that I got is 85.79%

```
[fold 0] score: 0.90741
[fold 1] score: 0.90455
[fold 2] score: 0.84637
[fold 3] score: 0.81278
[fold 4] score: 0.81852

Average accuracy across all folds: 85.79270790659565
```

3.5.2 Decision Tree Classifier

Now, applying decision tree classifier to predict revenue.

Data Splitting

Split the dataset into training and testing set, using same scenario as KNN modelling process.(70 % for training and 30% for testing)

Model Training with default parameters

Initially trained the model with default parameters.

Below is the classification report and accuracy that I got.

```
              precision    recall  f1-score   support

       False       0.93      0.97      0.95      3114
        True       0.78      0.57      0.66       548

    accuracy                           0.91      3662
   macro avg       0.86      0.77      0.80      3662
weighted avg       0.91      0.91      0.91      3662

The accuracy for Decision tree is 0.9120699071545604
```

Parameter Tuning

To find the optimal parameter for the Decision Tree model, I have used GridSearchCv (scikit-learn, 2019). This helps to search through a predefined parameter grid and evaluates models performance using cross validation. It helped to find out maximum depth, minimum number of sample to split a node and criterion for the split.

Below is the result that I got. But the accuracy was similar to the one with default parameter.

```
Best Accuracy of Decision Tree: 0.9120699071545604
Best Parameters: {'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 1
0}
Best Score: 0.8885630968710903
```

Feature Importance Analysis

Just for analysis, I have also used feature Importance to analyse the importance of different features in the Decision Tree. These scores help to tell the significance of each feature. We got the below result.

	Feature	Importance
8	PageValues	0.421996
5	ProductRelated_Duration	0.077383
6	BounceRates	0.077112
7	ExitRates	0.069093
4	ProductRelated	0.067660
1	Administrative_Duration	0.060812
10	Month	0.043868
0	Administrative	0.037475
13	Region	0.031921
14	TrafficType	0.027863
3	Informational_Duration	0.019047
11	OperatingSystems	0.017763
12	Browser	0.012377
2	Informational	0.011894
15	VisitorType	0.009946
16	Weekend	0.009001
9	SpecialDay	0.004788

From this, we know that PageValues contributes the highest percentage of significance in the dataset. We got the same result while data exploration and Hill climbing. Also have create visualization for decision tree which helps to understand better about the features relationship with target column.

3.6 Results and Evaluation – KNN and Decision Tree

According to the results, the Decision Tree algorithm had a better accuracy than the KNN Algorithm. The reason behind this is that Decision Tree is able to find complex interactions and dependencies between the features. From the data exploration, it is understood that many features have nonlinear relationships with the target variable, therefore decision tree works best to find nonlinear boundaries. And also we able to find the best features in the dataset using feature selection i.e PageValues, ProductRelated_Duration, BounceRates and ExitRates.

3.6.1 KNN Modelling Results

Initial Accuracy: The KNN model trained with all features and random k value of 3, achieved an accuracy of 86%.

Parameter Tuning: After applying greedy method for parameter tuning, the optimal k value was determined to be k=5 and weights =’distance’ and p=1. This resulted in accuracy of 87.

Feature selection: The hill climbing algorithm helped to identify columns to improve model’s performance. List of columns recommended by Hill climbing were BounceRates, PageValues, VisitorType, Administrative, SpecialDay, Month.

Retrained Model: The newly trained model achieved the accuracy of 88% (2 percent more than the initial value)

3.6.2 Decision Tree Modelling Results

Initial Accuracy: The KNN model trained with all features and default parameter, achieved an accuracy of 91%.

Parameter Tuning: GridSearchCV helped to find best parameters (max_depth:5, min_samples_leaf:4,min_samples_split:10), but there was no significant improvement in accuracy.

Feature importance: It helped us to find the best features in the dataset. This was used only for analysis. With this we could conclude that the PageValues feature has the highest importance among all features. Other columns are Bounce Rates, ProductRelated_Duration and ExitRate. This was also noticed in the data exploration phase with the help of heat map.

Retrained Model: The retrained trained model achieved the accuracy of 91%.

3.7 Conclusion

In conclusion, my analysis demonstrated the effectiveness of both the KNN and Decision Tree Algorithm for revenue classification using the “Online Shoppers Purchasing Intension Dataset”. The Decision Tree algorithm outperformed KNN, achieving a higher accuracy of 91% compared to 88%. Decision Tree performed better as it could effectively capture the underlying patterns and dependencies in the dataset, providing valuable insights to customer data. By accurately predicting purchase likelihood, businesses can tailor their marketing strategies, personalize recommendations, and optimize the online shopping experience to enhance revenue generation. The interpretability of the Decision Tree algorithm further aids in understanding the important features and decision rules driving customer classification, empowering businesses to make data-driven decisions and maximize their performance in the e-commerce domain.

3.8 References

Sakar, C.O. and Kastro, Y. (2018). *UCI Machine Learning Repository: Online Shoppers Purchasing Intention Dataset Data Set*. [online] archive.ics.uci.edu. Available at: https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset [Accessed 16 May 2023].

Sakar, C.O., Polat, S.O., Katircioglu, M. and Kastro, Y. (2018). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. *Neural Computing and Applications*, [online] 31(10), pp.6893–6908. doi:https://doi.org/10.1007/s00521-018-3523-0.

scikit-learn (2019). *sklearn.model_selection.GridSearchCV — scikit-learn 0.22 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed 22 May 2023].

scikit-learn (2019). *sklearn.model_selection.GridSearchCV — scikit-learn 0.22 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed 22 May 2023].