Assignment 2: Data Modelling and Presentation

Statement of solution: The session storage of the online customers for a course of a year are captured, which includes the result of shopping outcome as both positive and negative. The data captured consists of relevant information about the pages and their durations the customer had spent surfing, their exit rates, page values, and bounce rates, Operating system, Browsers, type of the day etc., Using these features and values, machine learning models are constructed to predict the customers purchasing intention such as buy or no buy.

Title: Prediction of Online Shopper's Purchasing Intention

Student ID: s3959200

Student Name: Krishnakanth Srikanth

Email (contact info): s3959200@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

Abstract

Most of the people visiting the shopping sites may or may not have an intention to buy due to various reasons. The aim of this report was to predict the intention of customer in online shopping, whether they would buy a product or not. In order to eliminate any inclination to a particular campaign, special day, user profile, or timeframe, the dataset was created so that each session would belong to a different user over the course of a year. Using the session's history of the customers logged in, predictions were being made on whether it ended up in a purchase. The report concludes by finalizing which model suits the best for predicting the intensions of the customer. It is recommended that with the equal splits of positive and negative class samples in the dataset, it would result in a proper and complete prediction of the shopper's intention.

Introduction

For this assignment, the dataset has been obtained from 'UCI Machine Learning Repository: Online Shoppers Purchasing Intention Dataset Data set' and it contains the results of **12,330 sessions**, of which **10,422** were **negative class samples with no shopping outcomes** and the remaining **1908** were **positive class samples with successful shopping outcomes**, with 17 descriptive and 1 target features. *The main goal of this project is to design a classification model, that is able to predict the intention of an online shopper (buy or no buy), based on the values of the given features*. The dataset is imported to the python environment using pandas function **read_csv().** It is completely pre-processed by checking for missing values, presence of any incorrect or duplicated values, and presence of any extra spaces. Finally dealt with presence of outliers by removing them using the IQR and statistical methods. Once after the dataset is cleaned and free from errors, it is explored using seaborn and matplotlib. At first, each attribute is explored individually and then pairs of attributes are explored with a plausible hypothesis being defined for each exploration. Post the cleaning and exploratory analysis, we proceed with data modelling where the data types are converted as needed, dataset are split into train and test datasets using necessary libraries. This dataset chosen for assignment, has the target feature with two classes and hence, selecting classification models namely the **K Nearest Neighbors Classification and Decision Tree Classification** to predict the online customer's purchasing intention whether they would end up buying a product or not, in short, buy or no buy. For each model, feature selection, parameter tuning, confusion matrix and cross validation were being calculated and performed. Lastly, by comparing the outcomes of both the models, the best or better one is selected.

Methodology

The procedures that were undertaken to build a model to predict the outcome is described in detail here.

Data Preparation

Data Retrieving

The first and foremost step is Data Retrieving. Here, the data containing 12,330 attributes and 18 features is downloaded from 'UCI Machine Learning Repository: Online Shoppers Purchasing Intention Dataset Data set' and is stored in the same folder along with the python file. Once dowloaded, the data has been imported into the python environment using **read_csv()** from pandas and stored in **orig_data** variable, checked for the data types using **dtypes** and printed out the first 5 rows from the dataset using **head()**. Before proceeding with other steps, a copy of original data (orig_data) is saved in **data** variable using **copy()** and used for the next steps.

Check for errors

Check for typos/incorrect values/duplicates:
Next, the check for any incorrect values or typos and duplicates were undertaken to ensure that the data is free from these errors using **value_counts()**.

Check for missing values:
Once after the result from above check was positive, the next check was to see for any missing or NA or NaN values. This is done by using **.isna().sum()** function of pandas. **The count of missing values were found to be 0** and hence, proceeded to next check which is the most important one, the outliers.

Check for outliers:
Outliers are values that occur very rare in a dataset. This check was done on all **numerical features** using **box plot** to visualise and **IQR** is calculated along with upper bound and lower bound limits. To remove the outliers, we select only the values that lie withing the range of upper bound and lower bound.

Below are the formulas that are used:

- IQR = Q3 – Q1

  where, Q3 = Third quartile (75%)

  Q1 = First quartile (25%)

- upperbound = Q3 + (1.5 * IQR)
- lowerbound = Q1 - (1.5 * IQR)

Outliers check on Exit Rates:

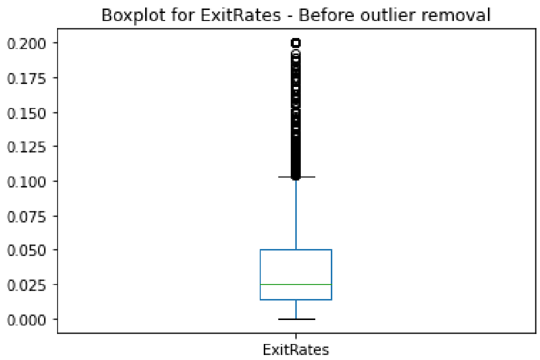


Fig 1: Before removing outliers

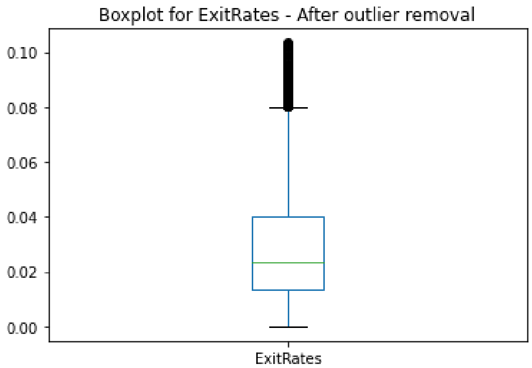


Fig 2: After removing outliers

Outliers check on Administative_Duration, Informational_Duration, ProductRelated_Duration:

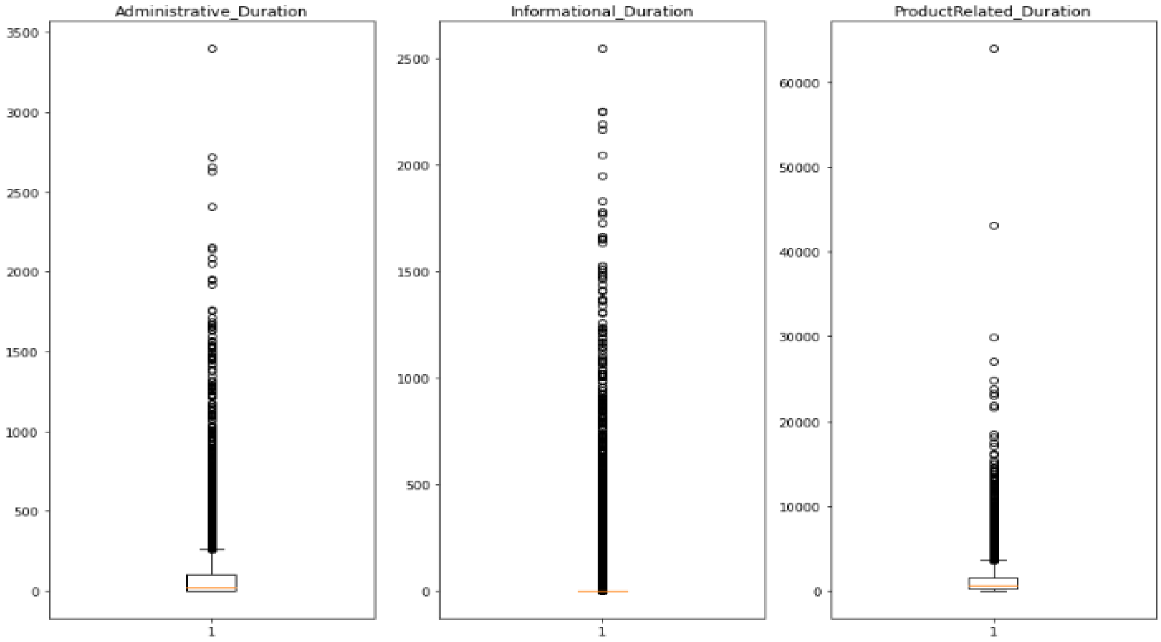


Fig 3: Before removing outliers

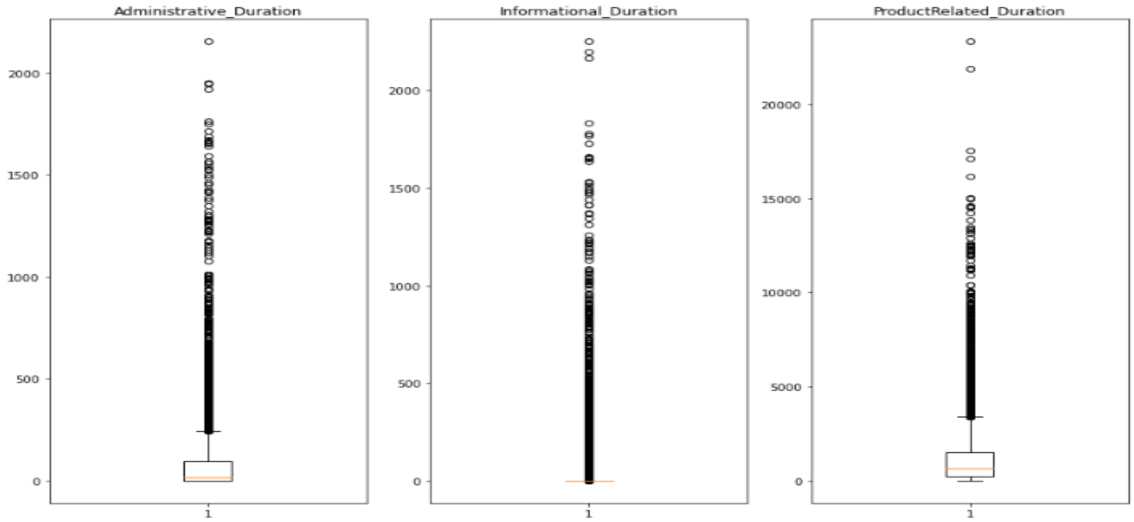
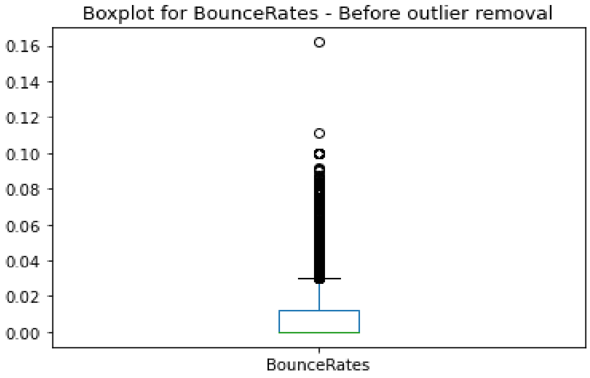


Fig 4: After removing outliers

Outliers check for BounceRates:



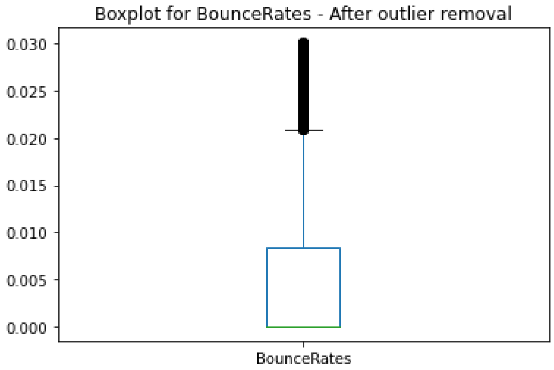


Fig 5: Before removing outliers **Fig 6: After removing outliers**

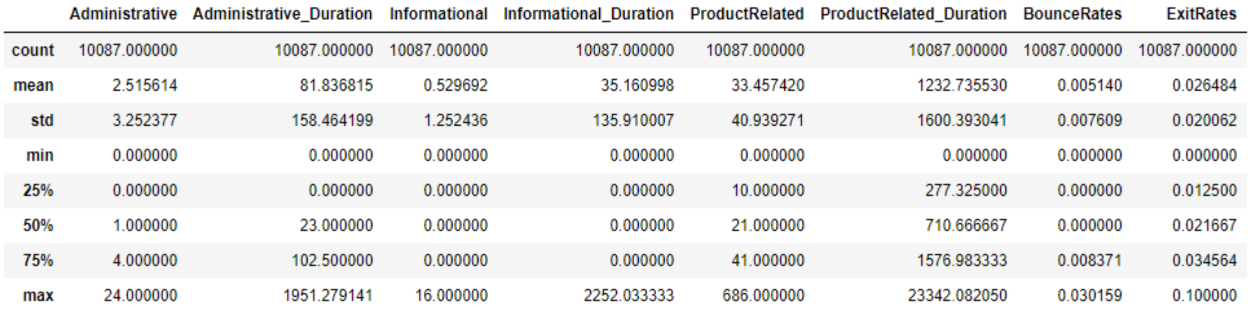
Finally, after removing the outliers, the index is reseted using **reset_index()**

Data Exploration

Task 1

Here, descriptive statistics and visualizations for each column is generated and observations are noted.

Descriptive statistics for numeric features:

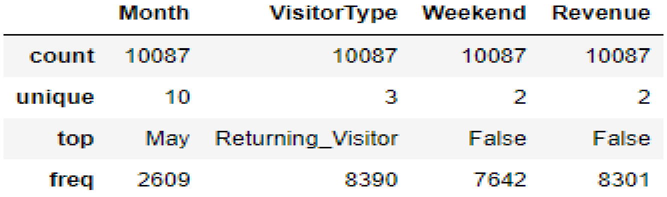


	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates
count	10087.000000	10087.000000	10087.000000	10087.000000	10087.000000	10087.000000	10087.000000	10087.000000
mean	2.515614	81.836815	0.529692	35.160998	33.457420	1232.735530	0.005140	0.026484
std	3.252377	158.464199	1.252436	135.910007	40.939271	1600.393041	0.007609	0.020062
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	10.000000	277.325000	0.000000	0.012500
50%	1.000000	23.000000	0.000000	0.000000	21.000000	710.666667	0.000000	0.021667
75%	4.000000	102.500000	0.000000	0.000000	41.000000	1576.983333	0.008371	0.034564
max	24.000000	1951.279141	16.000000	2252.033333	686.000000	23342.082050	0.030159	0.100000

Fig 7: Descriptive statistics of numeric columns

From the above statistics, there are some values that deviates abruptly from the data group which confirms the presence of outliers.

Descriptive statistics for non-numeric features:



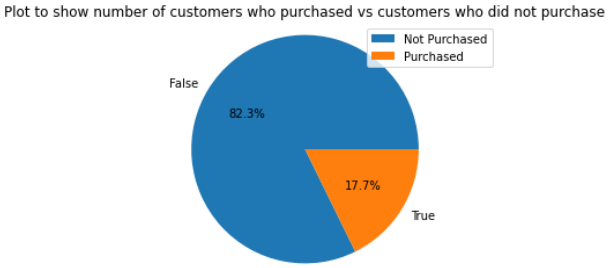
	Month	VisitorType	Weekend	Revenue
count	10087	10087	10087	10087
unique	10	3	2	2
top	May	Returning_Visitor	False	False
freq	2609	8390	7642	8301

Fig 8: Descriptive statistics for non-numeric features

From the above statistics, we can see that our target variable has majority of negative class samples (8301)

Visualizations:

Below are the visualizations of each column attributes and their observations.



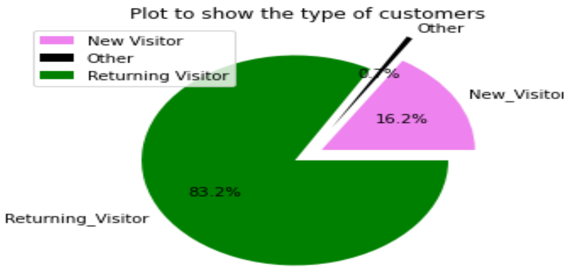
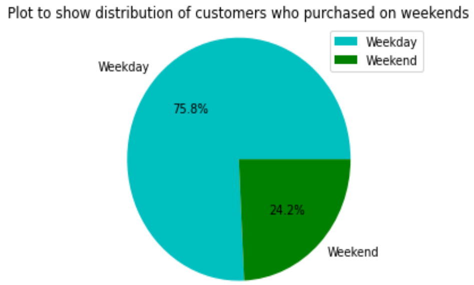


Fig 9: Revenue

Fig 10: VisitorType

From fig 9, majority of customers (82.3%) did not purchase. From fig 10, majority (83.2%) of customers are old customers.



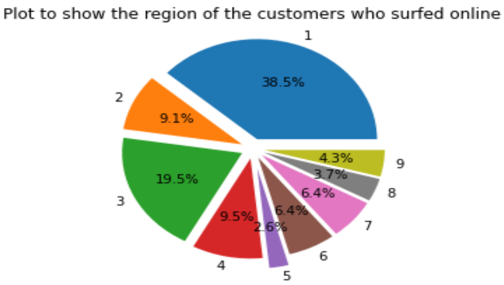
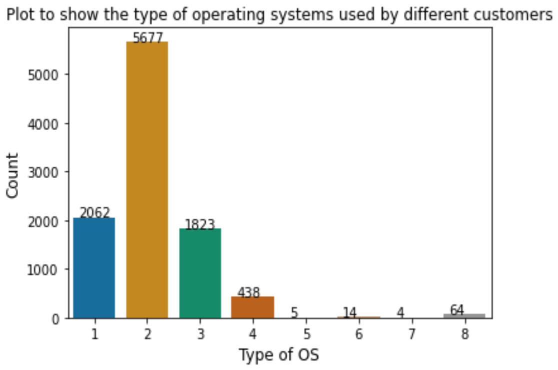


Fig 11: Weekend

Fig 12: Region

From fig 11, the percentage of customers visited the page on a weekends is less than the percentage of customers who visited the page on weekdays. From fig 12, most customers who surfed online are more from Region 1 (38.5%), followed by Region 3 (19.5%) and rest does not have that many online shoppers.



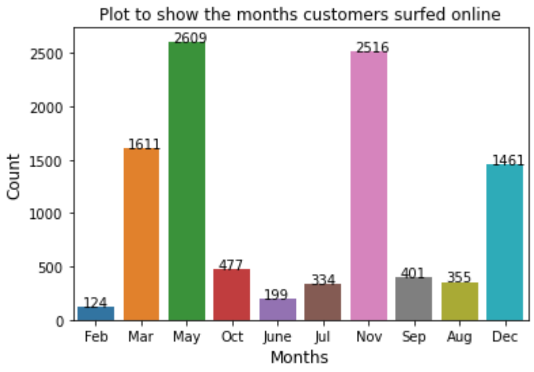
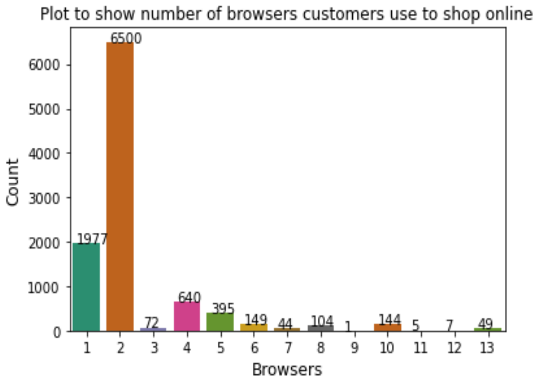


Fig 13: OperatingSystems

Fig 14: Month

From fig 13, customers use OS type '2' the most (5677). From fig 14, most surfed month is May (2609), followed by November (2516).



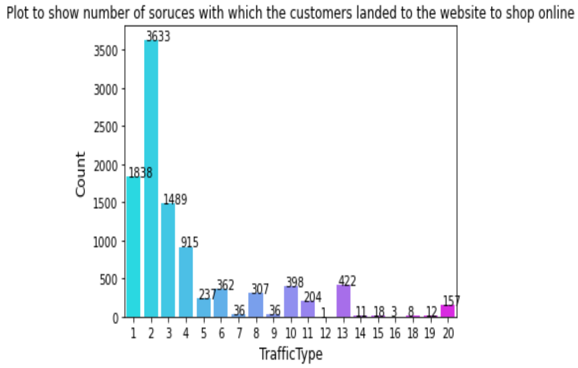
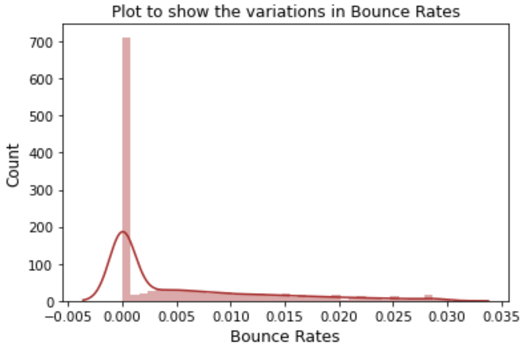


Fig 15: Browser

Fig 16: TrafficType

From fig 15, most customers who surfed online used Browser 2 (6500) the most. From fig 16, customers landed to the shopping site mostly through the source 2 (3633).



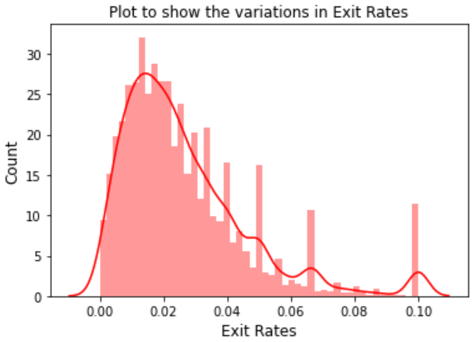


Fig 17: BounceRates

Fig 18: ExitRates

From fig 17 and 18, the average bounce rate and average exit rate value of the pages visited by the visitor is at the point of distribution which starts from 0.00.

Task 2

Here, visualisations are generated to show relationships between pairs of attributes definning a plausible hypothesis.

Hypothesis 1: People prefer to shop more on weekends than on weekdays

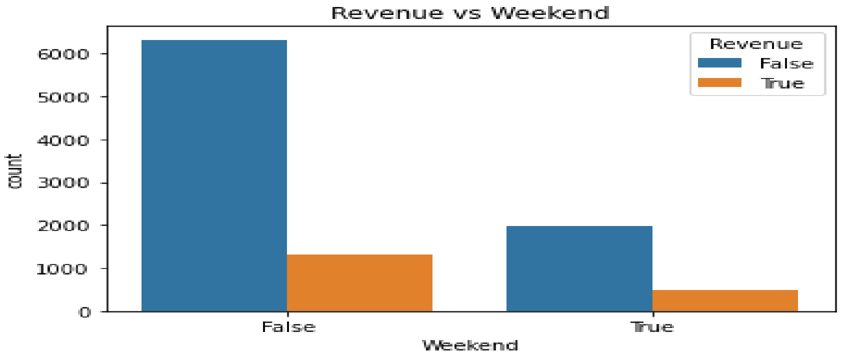


Fig 19: Hypothesis 1

From the above plot, we can see that number of customers who visited the shopping site pages are more on weekdays. Though majority of the visits did not end up buying, the count of customers shopping on weekdays is more when compared to customers shopping on weekends. **This proves our hypothesis to be incorrect.**

Hypothesis 2: Browser type plays a major role in customer's purchase

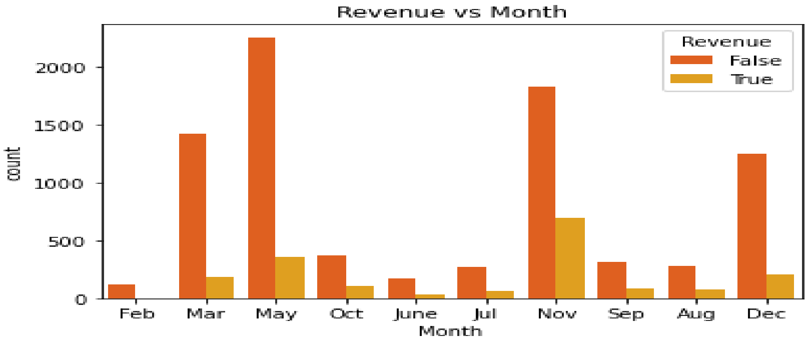


Fig 20: Hypothesis 2

From the above plot, it is observed that month of November is where the count of 'Purchase' is high when compared to other months. And the month of May has large number of page visits without any purchase by the customers. **This shows our hypothesis to be correct**.

Hypothesis 3: Old customers prefer to shop more than others

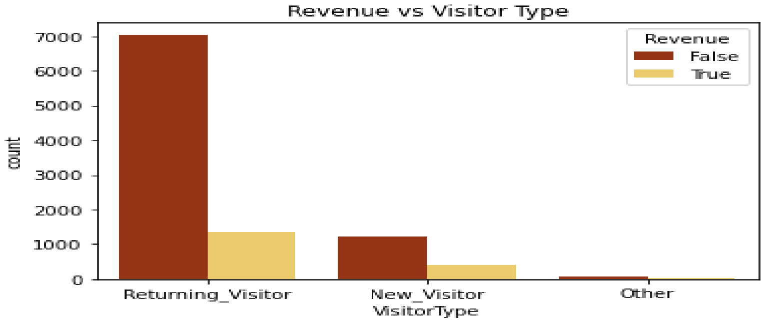


Fig 21: Hypothesis 3

From the above plot, we can observe that the count of 'Returning Visitor' is high when compared to any other type of visitors with respect to making a Purchase. **This shows our hypothesis to be correct**.

Hypothesis 4: People's shopping increases as special days approach

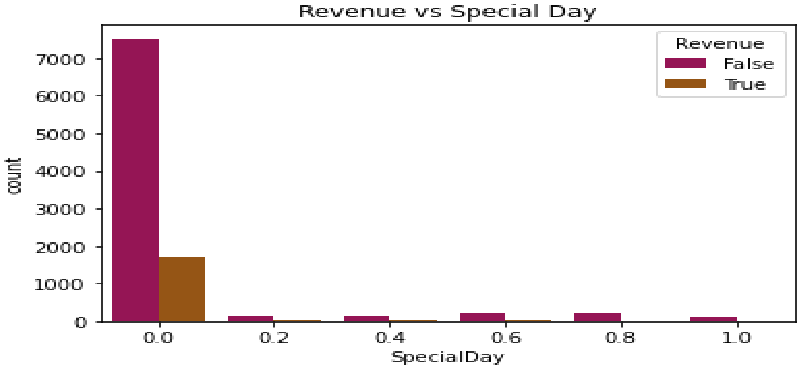


Fig 22: Hypothesis 4

From the above plot we can see that Special Day has no impact in visitors making a purchase. **This shows our hypothesis to be incorrect.**

Hypothesis 5: People from specific regions do not shop online

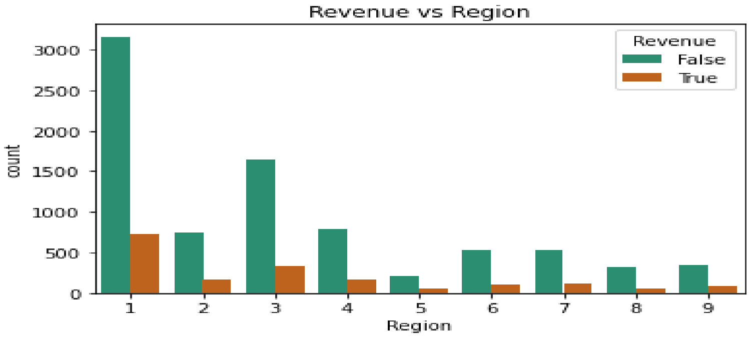


Fig 23: Hypothesis 5

From the above plot, we can confirm that the people from all regions use online shopping. **Hence, our hypothesis is incorrect.**

Hypothesis 6: Exit rates has impact on the purchase factor

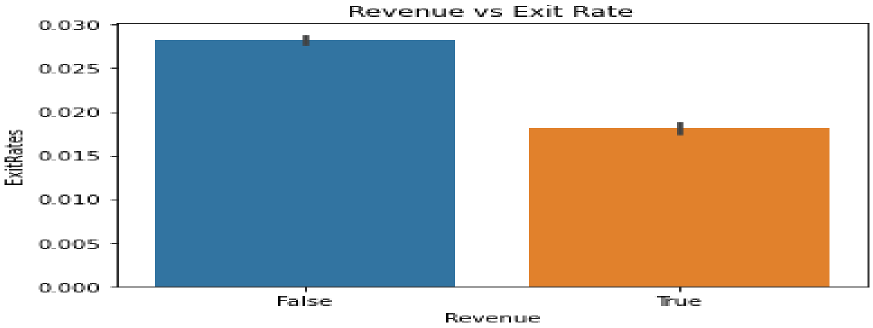


Fig 24: Hypothesis 6

From the above plot, we can observe that the exit rates are low whenever there is a 'Purchase' made. This shows that exit rates has impact on purchase factor and **hence, our hypothesis is correct**.

Hypothesis 7: Online shoppers are spread across various regions

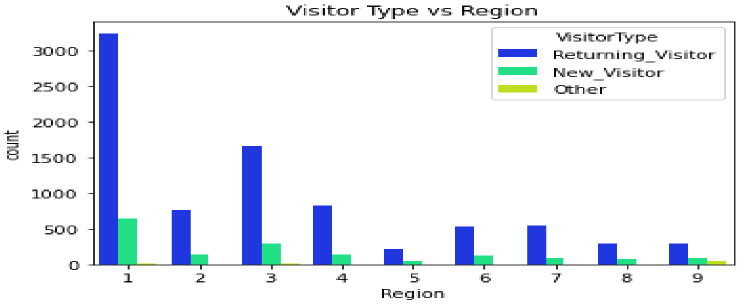
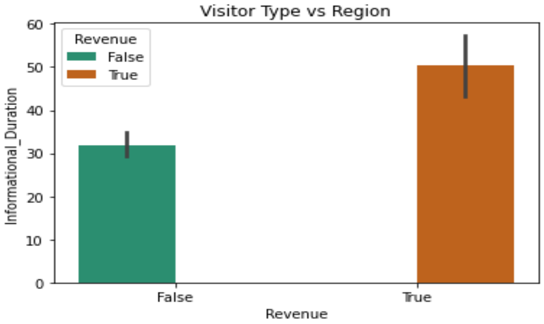


Fig 25: Hypothesis 7

From the above plot, we can confirm that the online shoppers are spread across all the regions with majority of them are from Region 1. **This confirms our hypothesis to be correct.**

Hypothesis 8: Purchase rate depends upon the amount of website's information provided in the shopping sites

Hypothesis 9: Purchase rate depends upon the amount of product related information provided in the shopping sites



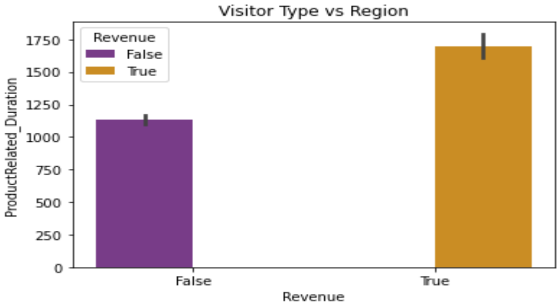


Fig 26: Hypothesis 8

Fig 27: Hypothesis 9

From the above two graphs, we can confirm that, as the duration in the product or informational pages are more, it is more likely that there is a purchase made. **Hence, this confirms our hypothesis 8 and 9 to be correct**.

Hypothesis 10: Purchase factor depends on page value of the pages visited by the customer

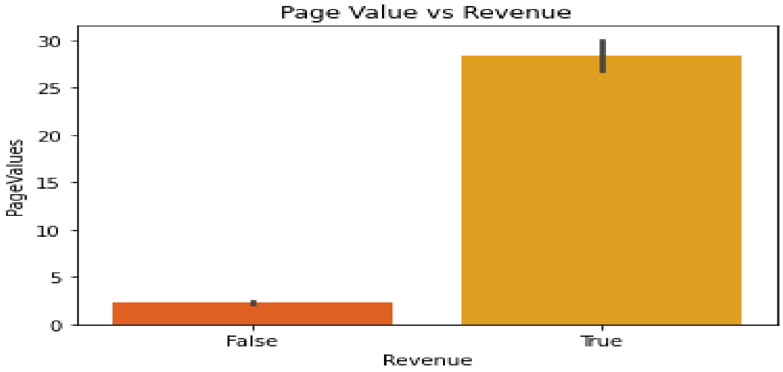


Fig 28: Hypothesis 10

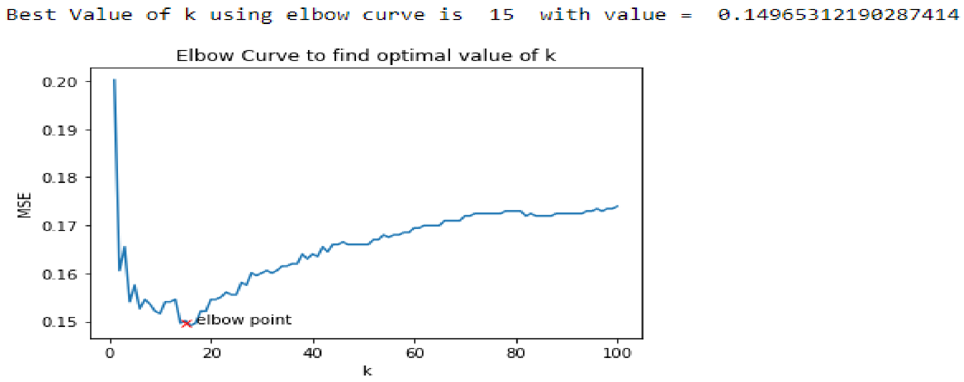
From the above plot, we can observe that whenever there is a purchase, the average page value of the pages visited by the visitor is larger. **This confirms our hypothesis to be correct**.

Data Modelling

First, the non-numeric values are mapped to numeric, data types are changed accordingly and appropriately. Now, to create a model, two datasets (**target and descriptive**) are created and converted to numpy arrays using **to_numpy()**. These arrays are then divided into training and testing datasets using **train_test_split** from sklearn.model_selection. **80% of data for training and 20% for testing datasets**.

Choosing k value

For KNN model, the maximum value of k is calculated by sqrt(n) i.e. sqrt(10087) = 100.43 and so, k value can be any value within the range of 1 to 100. Hence, k value is chosen by plotting the MSE and k values (**Elbow curve**) from which we could see that **when k=15, the error score seems to very low** and hence choosing **k=15** would be reasonable.

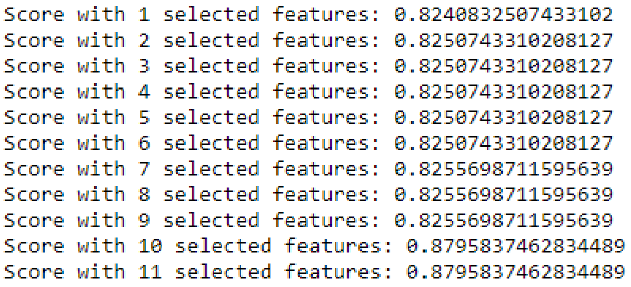


```
Best Value of k using elbow curve is  15  with value =  0.14965312190287414
```

Fig 29: Choosing k value

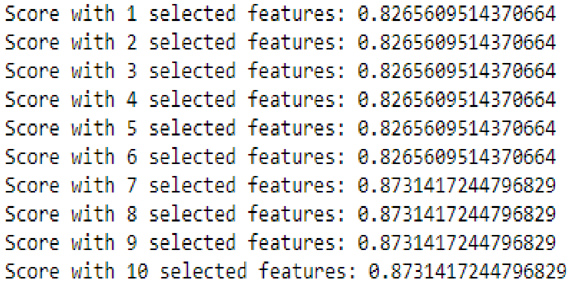
Feature Selection

Here, the best features are selected using **Hill climbing** where a for loop is created to run and predict the model accuracy of both KNN and Decision Tree models with selected features combination and scores are printed along with number of features selected.



```
Score with 1 selected features: 0.8240832507433102
Score with 2 selected features: 0.8250743310208127
Score with 3 selected features: 0.8250743310208127
Score with 4 selected features: 0.8250743310208127
Score with 5 selected features: 0.8250743310208127
Score with 6 selected features: 0.8250743310208127
Score with 7 selected features: 0.8255698711595639
Score with 8 selected features: 0.8255698711595639
Score with 9 selected features: 0.8255698711595639
Score with 10 selected features: 0.8795837462834489
Score with 11 selected features: 0.8795837462834489
```

Fig 30: KNN Feature selection



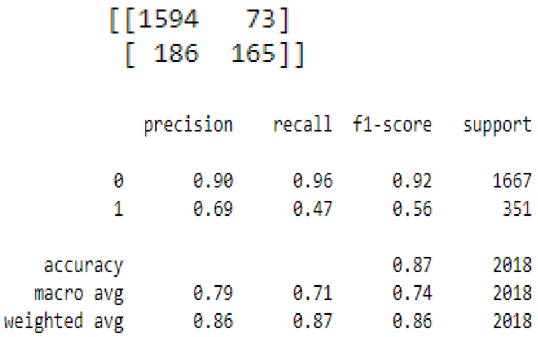
```
Score with 1 selected features: 0.8265609514370664
Score with 2 selected features: 0.8265609514370664
Score with 3 selected features: 0.8265609514370664
Score with 4 selected features: 0.8265609514370664
Score with 5 selected features: 0.8265609514370664
Score with 6 selected features: 0.8265609514370664
Score with 7 selected features: 0.8731417244796829
Score with 8 selected features: 0.8731417244796829
Score with 9 selected features: 0.8731417244796829
Score with 10 selected features: 0.8731417244796829
```

Fig 31: Decision Tree Feature selection

From above figs 30, 31, **KNN** has **more accuracy** when **10, 11 features** are selected and for **decision tree, accuracy** is **more** when **7, 8, 9** and **10 features** are selected. The highest accuracy obtained by feature selection for **KNN** is **87.9%** and for **Decision Tree** the score is **87.3%**.

Parameter Tuning

For KNN, **weight** is chosen as '**distance**' as the data is not uniform and '**p**' value which I chose as **1** because of the dataset being high in dimension, smaller p value would suit the best to avoid overfitting. For Decision Tree, default parameters are used except for **max_features** which I chose as '**auto**' because it chooses square root of max_features and as per rule-of-thumb it works well. (As a rule-of-thumb, the square root of the total number of features works well) and **max_depth=3** as lower the max depth, lower is over fitting of the model and **random_state as 0**.



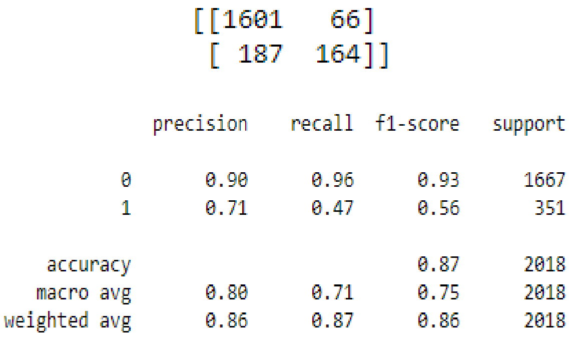
```
[[1594   73]
 [ 186  165]]

              precision    recall  f1-score   support

           0       0.90      0.96      0.92      1667
           1       0.69      0.47      0.56       351

    accuracy                           0.87      2018
   macro avg       0.79      0.71      0.74      2018
weighted avg       0.86      0.87      0.86      2018
```

Fig 32: CM and CR for KNN



```
[[1601   66]
 [ 187  164]]

              precision    recall  f1-score   support

           0       0.90      0.96      0.93      1667
           1       0.71      0.47      0.56       351

    accuracy                           0.87      2018
   macro avg       0.80      0.71      0.75      2018
weighted avg       0.86      0.87      0.86      2018
```

Fig 33: CM and CR for Decision Tree

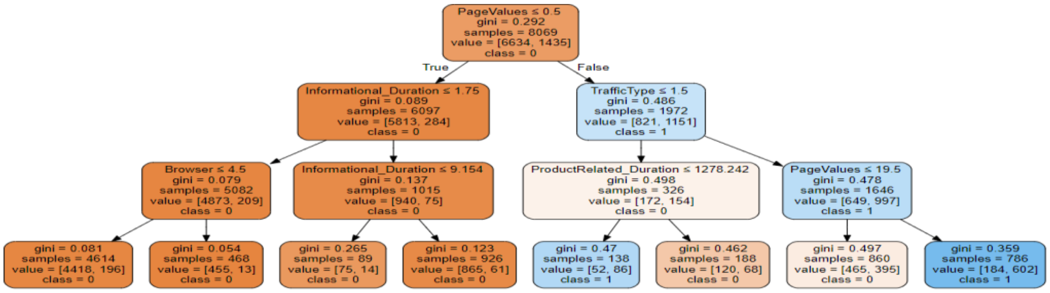


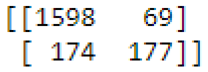
Fig 34: Decision Tree generated for the created model after parameter tuning

Results

The results are in the form of confusion matrix, classification reports that contains accuracy, precision, recall, f1-score.

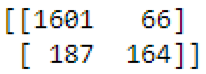
Confusion Matrix

Below are the confusion matrix obtained from both the constructed models.



```
[[1598   69]
 [ 174  177]]
```

Fig 35: KNN Confusion Matrix



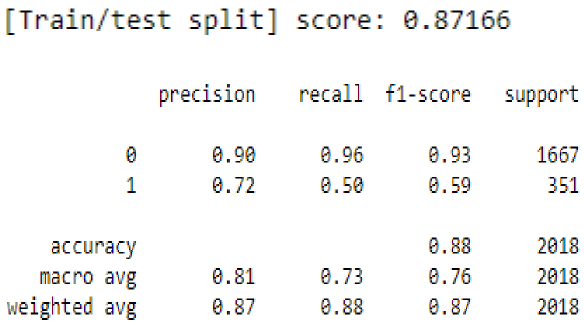
```
[[1601   66]
 [ 187  164]]
```

Fig 36: Decision Tree Confusion Matrix

From the above matrices, the FN (False Negative) and FP (False Positive) values are comparatively low in both the models.

Accuracy Score and Classification Report

From the classification report we could get the results of precision, recall, F1, and support scores for the model we constructed. Below are the classification reports of KNN and Decision Tree classification models obtained after feature selection.



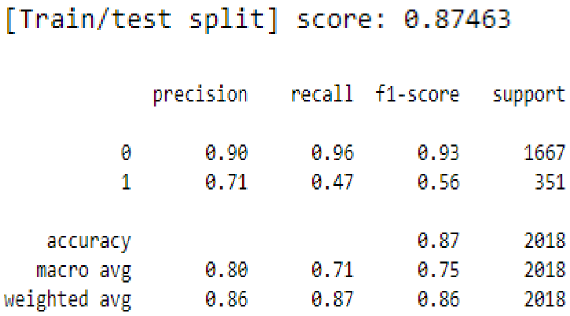
```
[Train/test split] score: 0.87166

              precision    recall  f1-score   support

           0       0.90      0.96      0.93      1667
           1       0.72      0.50      0.59       351

    accuracy                           0.88      2018
   macro avg       0.81      0.73      0.76      2018
weighted avg       0.87      0.88      0.87      2018
```

Fig 37: Score & Classification report of KNN



```
[Train/test split] score: 0.87463

              precision    recall  f1-score   support

           0       0.90      0.96      0.93      1667
           1       0.71      0.47      0.56       351

    accuracy                           0.87      2018
   macro avg       0.80      0.71      0.75      2018
weighted avg       0.86      0.87      0.86      2018
```

Fig 38: Score & Classification report of Decision Tree

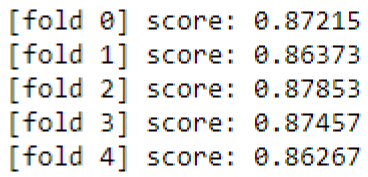
Below table is used to compare the accuracy and F1-score of the models.

Scores / Models	K Nearest Neighbours	Decision Tree
Accuracy Score	0.871 (ie) 87.1%	0.874 (ie) 87.4%
F1-score	0.59	0.56

Table 1: Final results/scores

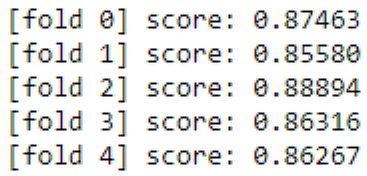
K-Fold Cross Validation

Once after the model is fully constructed it is cross validated with different test data to avoid overfitting and get similar results. Using **KFold()** from package **sklearn.model_selection**, the data is split into 5 sets (**n_splits=5**) and so using for loop, 5 iterations each with a new test data is used for the prediction. The accuracy scores of the models when **K-Fold Cross Validation** is performed is below:



```
[fold 0] score: 0.87215
[fold 1] score: 0.86373
[fold 2] score: 0.87853
[fold 3] score: 0.87457
[fold 4] score: 0.86267
```

Fig 39: Accuracy scores of KNN



```
[fold 0] score: 0.87463
[fold 1] score: 0.85580
[fold 2] score: 0.88894
[fold 3] score: 0.86316
[fold 4] score: 0.86267
```

Fig 40: Accuracy scores of Decision Tree

Discussion

We could see that, after K-Fold Cross Validation is performed on both the models after selecting appropriate features by feature selection, both the KNN and Decision Tree models actively learn, provides reliable output and also, the accuracy scores increases as folds increases with smaller bias (Arya, 2022). **K Nearest Neighbors and Decision Tree both has an equal and higher accuracy score of 86.2%** after KFold cross validation with 5 folds. Also, for a model to be successful and efficient, it should have the counts of both False Positives (FP) and False Negatives (FN) low. **The data used here is unbalanced**, i.e. negative class samples is super high than the positive class samples, which actually should be of equal splits for a model to predict more exactly the results. Hence, with the accuracy score, it is difficult to interpret which model suits the best for this particular problem of predicting the customer's purchase intention in online shopping.

Conclusion

The aim of the assignment was to build a solution to predict the purchasing intention of the customer with accuracy as high as possible. For this, from the above two constructed models – K-Nearest Neighbor Classifier and Decision Tree Classifier, with the accuracy score obtained predicting the test data, the **score of KNN Classifier (0.871) is greater than the score of Decision Tree Classifier (0.874)** (though nearly the same). Hence, I conclude by saying that with the features and parameters I have opted to use in these models, and considering the advantages and disadvantages of both the classifier models, precision, recall, f1-scores and validation scores as well into account, **Decision Tree Classifier model would best suit to predict the intention of the online shoppers (buy or no buy)**. It may be observed that the models perform better with various types of data after testing the algorithm with different parameters.

References

1. Sakar, C.O. et al. (2018) Real-time prediction of online shoppers' purchasing intention using Multilayer Perceptron and LSTM recurrent neural networks - neural computing and applications, SpringerLink. Available at: https://link.springer.com/article/10.1007/s00521-018-3523-0 (Accessed: 09 May 2023).
2. Detect and remove the outliers using Python (2023) GeeksforGeeks. Available at: https://www.geeksforgeeks.org/detect-and-remove-the-outliers-using-python/ (Accessed: 10 May 2023).
3. Sklearn.neighbors.kneighborsclassifier (no date) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (Accessed: 10 May 2023).
4. Band, A. (2023) How to find the optimal value of K in Knn?, Medium. Available at: https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb (Accessed: 10 May 2023).
5. Sklearn.tree.decisiontreeclassifier (no date) scikit. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (Accessed: 10 May 2023).
6. Arya, N. (2022) Why use K-fold cross validation?, KDnuggets. Available at: https://www.kdnuggets.com/2022/07/kfold-cross-validation.html#:~:text=K%2Dfold%20Cross%2DValidation%20is,data%20sample%20is%20split%20into . (Accessed: 23 May 2023).
7. ARAT, M.M. (2019) A complete guide to K-nearest-neighbors with applications in Python, Mustafa Murat ARAT. Available at: https://mmuratarat.github.io/2019-07-12/k-nn-from-scratch#:~:text=K%2DNN%20algorithm%20is%20an,samples%20in%20the%20training%20dataset. (Accessed: 23 May 2023).
8. Ren, Yongli (2023) 'Introduction/What is Data Science?' [PowerPoint slides, COSC2670], RMIT University, Melbourne.
9. Ren, Yongli (2023) ' Classification (1)' [PowerPoint slides, COSC2670], RMIT University, Melbourne.
10. Ren, Yongli (2023) 'Classification (2)' [PowerPoint slides, COSC2670], RMIT University, Melbourne.