CAPSTONE PROJECT

Strategic Thinking - Semester II

Abstract

This academic project aims to deliver a critical analysis of the knowledge produced, in the course: Higher Diploma in Science in Data Analytics for Business at CCT College.

Student Name: Natalia de Oliveira Rodrigues & Student ID: 2023112

CCT College Dublin

Assessment Cover Page

Module Title:	Strategic Thinking
Assessment Title:	Project Capstone Semester II - Presentation / Report
Lecturer Name:	James Garza
Student Full Name:	Natalia de Oliveira Rodrigues
Student Number:	2023112
Assessment Due Date:	19th September 2023
Date of Submission:	15th November 2023 23:59

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

Introduction

E-commerce is the new route for buying and selling services using the internet. Nowadays, a large part of the population uses this method of purchase to find a product. For that reason, e-commerce and business are now connected since they have a website to offer their products, where you can also place an order.

On the other hand, digital purchasing has increased since the COVID-19 pandemic, as people had to remain at home and had no chance to go anywhere for shopping. This alternative found a way to support more and more online sales, making the experience more convenient for users.

As a result, the companies are improving their websites and investing money to determine whether the product on their websites is the best option for customers.

With this report, we are trying to predict online shoppers' behaviours using machine learning. Three different algorithms were used in this analysis. Methods such as clustering, cross validation, hyperparameters tuning and SHAP were added in this report to enhance its quality.

Business Description

We are trying to train a Machine Learning Model that can predict the purchasing intentions of a visitor to a particular store's website. The data is derived from e-commerce website data. It updates in real time when a user moves from one page to another. This is important as it can have a huge impact on online shops' profitability. This data can be utilized to prompt prospective customers to finish an online transaction in real-time and increase total purchase conversion rates.

Research Question

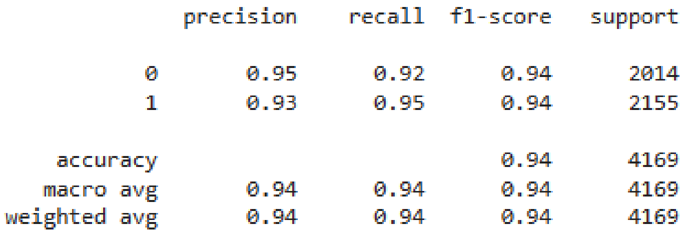
Can we predict if a user will make a purchase on an e-commerce website given their clickstream and session data?

General goal

The main goal of this analysis is to predict if the user will end up generating revenue or not. Online stores can then use the findings to make sure they can continue to be profitable. In this paper, we try to resolve a classification problem.

Success criteria/indicators

After applying 3 different Machine Learning Models. A Random Forest classifier Model appeared to be the model that would best address our classification issue. We have achieved an impressive 94% of accuracy, Precision and Recall through the Random Forest Classification Model. The final results are demonstrated in Figure 1.



```
              precision    recall  f1-score   support

           0       0.95      0.92      0.94      2014
           1       0.93      0.95      0.94      2155

    accuracy                           0.94      4169
   macro avg       0.94      0.94      0.94      4169
weighted avg       0.94      0.94      0.94      4169
```

Figure 1 Classification Report - Accuracy, Precision and Recall Results

Technologies used

Models and machine learning algorithms

Three supervised machine learning models Decision Tree Classifier, Random Forest Classifier, and Support Vector Machine (SVM) that are frequently employed for classification issues were used in this study.

Libraries

To carry out various jobs and model algorithms, numerous libraries have been employed. They might consist of Pandas, Numpy, Seaborn, Matplotlib, Scipy, Statistics, SMOTE, NearMiss, StandardScaler, PCA, Metrics, Counter, dtreeviz, FeatureImportances etc.

Accomplishment

Data

The dataset “online_shoppers_intention” describes if a person is going to buy our products or not and gives us different attributes to analyze; it is composed of 12,330 rows and 18 features of which 14 are numerical and 4 categorical. We are going to analyze the months of frequent visits to the website, type of visitor, and exit days, among other variables that we can take a look at in the Data Dictionary (Appendix 1).

Source

Our data was taken from Kaggle and it was taken from the following link: https://www.kaggle.com/datasets/imakash3011/online-shoppers-purchasing-intention-dataset (Kaggle, 2021)

Attributes

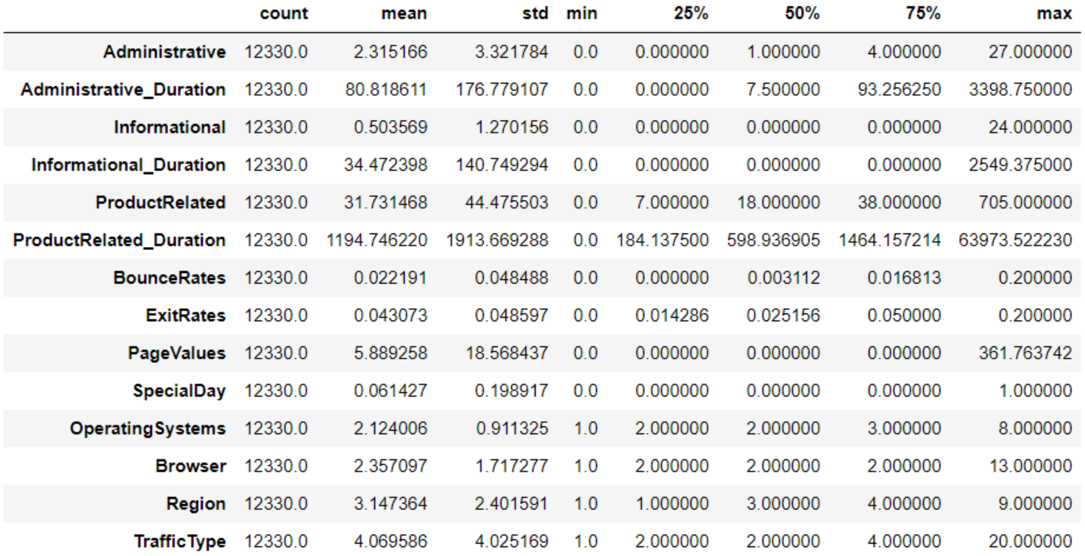
As said previously, we have 18 features. The feature “Revenue” is our dependent variable which means if the user completed the purchase. We are going to analyze 14 features as independent variables.

Dimensions

The shape of our data is 12,330 rows and 18 columns as variables.

Exploratory Data Analysis (EDA)

First, we took a look at our data and the statistics we have to start the analysis of our variables. We are going to see the statistics we found in the numerical variables of our dataset.



	count	mean	std	min	25%	50%	75%	max
Administrative	12330.0	2.315166	3.321784	0.0	0.000000	1.000000	4.000000	27.000000
Administrative_Duration	12330.0	80.818611	176.779107	0.0	0.000000	7.500000	93.256250	3398.750000
Informational	12330.0	0.503569	1.270156	0.0	0.000000	0.000000	0.000000	24.000000
Informational_Duration	12330.0	34.472398	140.749294	0.0	0.000000	0.000000	0.000000	2549.375000
ProductRelated	12330.0	31.731468	44.475503	0.0	7.000000	18.000000	38.000000	705.000000
ProductRelated_Duration	12330.0	1194.746220	1913.669288	0.0	184.137500	598.936905	1464.157214	63973.522230
BounceRates	12330.0	0.022191	0.048488	0.0	0.000000	0.003112	0.016813	0.200000
ExitRates	12330.0	0.043073	0.048597	0.0	0.014286	0.025156	0.050000	0.200000
PageValues	12330.0	5.889258	18.568437	0.0	0.000000	0.000000	0.000000	361.763742
SpecialDay	12330.0	0.061427	0.198917	0.0	0.000000	0.000000	0.000000	1.000000
OperatingSystems	12330.0	2.124006	0.911325	1.0	2.000000	2.000000	3.000000	8.000000
Browser	12330.0	2.357097	1.717277	1.0	2.000000	2.000000	2.000000	13.000000
Region	12330.0	3.147364	2.401591	1.0	1.000000	3.000000	4.000000	9.000000
TrafficType	12330.0	4.069586	4.025169	1.0	2.000000	2.000000	4.000000	20.000000

Figure 2 Statistics of the numerical features in our dataset.

We have a summary of the statistics of every variable showing the quartiles, mean, standard deviation, minimum and maximum values.

Now let’s analyze each group. We are going to take a look at our dependent variable “Revenue”.

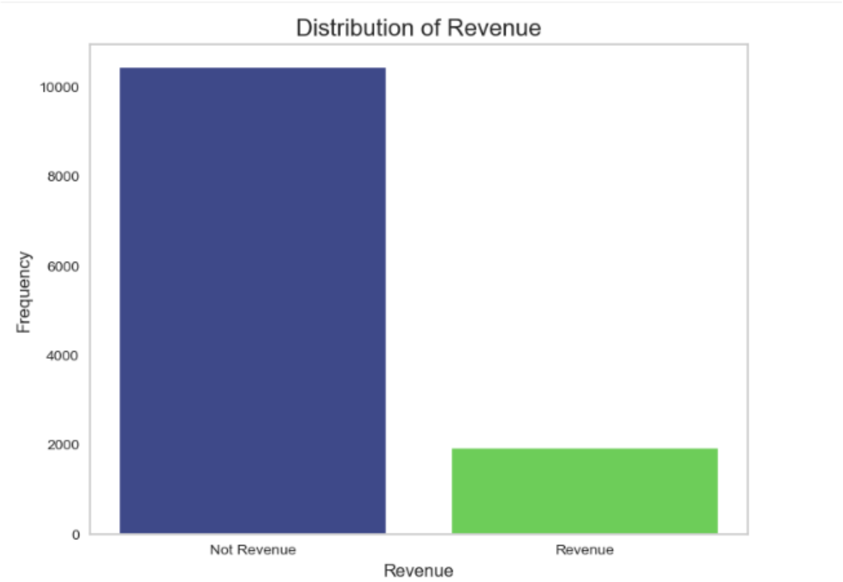


Figure 3 Bar plot target variable “Revenue”

In *Figure* 3, there are around 10,000 users that didn't complete their purchase than the ones that did who are sound 2,000 demonstrating that our data is unbalanced. As we have more False than True values, we are going to have a bias, to avoid that we need to balance it.

Now we are going to analyze the categorical variables.

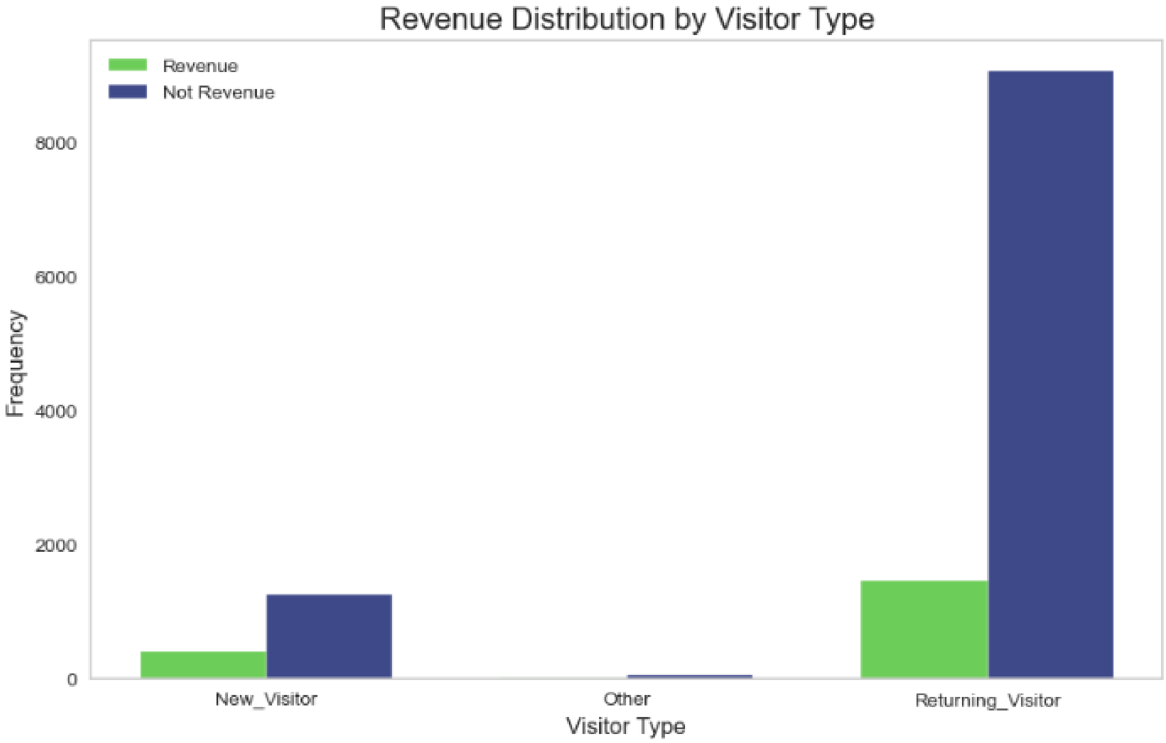


Figure 4 Revenue Distribution by Visitor Type

In *Figure 4*, we can see that most of the users are returning visitors; in *Figure 5,* we can see that fewer people visit the website on weekends (represented by True) than the ones that visit between Monday and Friday (represented by False). These variables have an acceptable distribution to apply to our model.

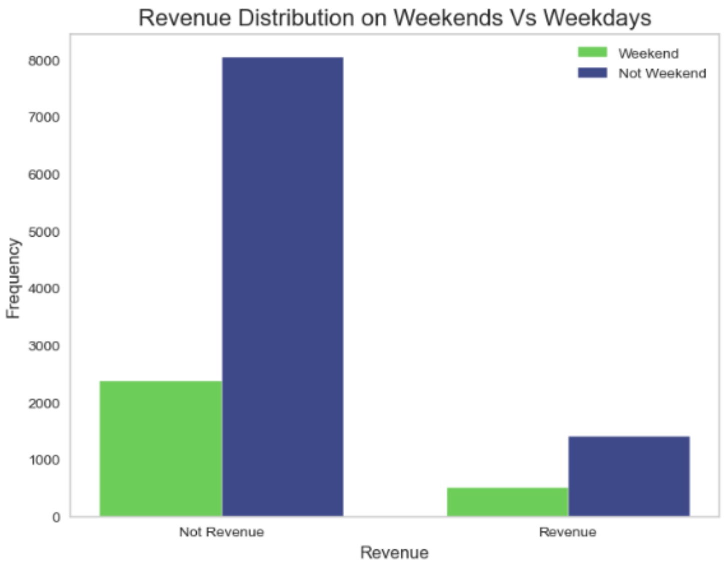


Figure 5 Revenue Distribution on Weekdays Vs Weekends

Figure 6 shows that most of our visitors visit our webpage in March, May, November and December, we could suppose that in those months they celebrate special days and users tend to look at our products.

Let’s visualize the numerical discrete variables.

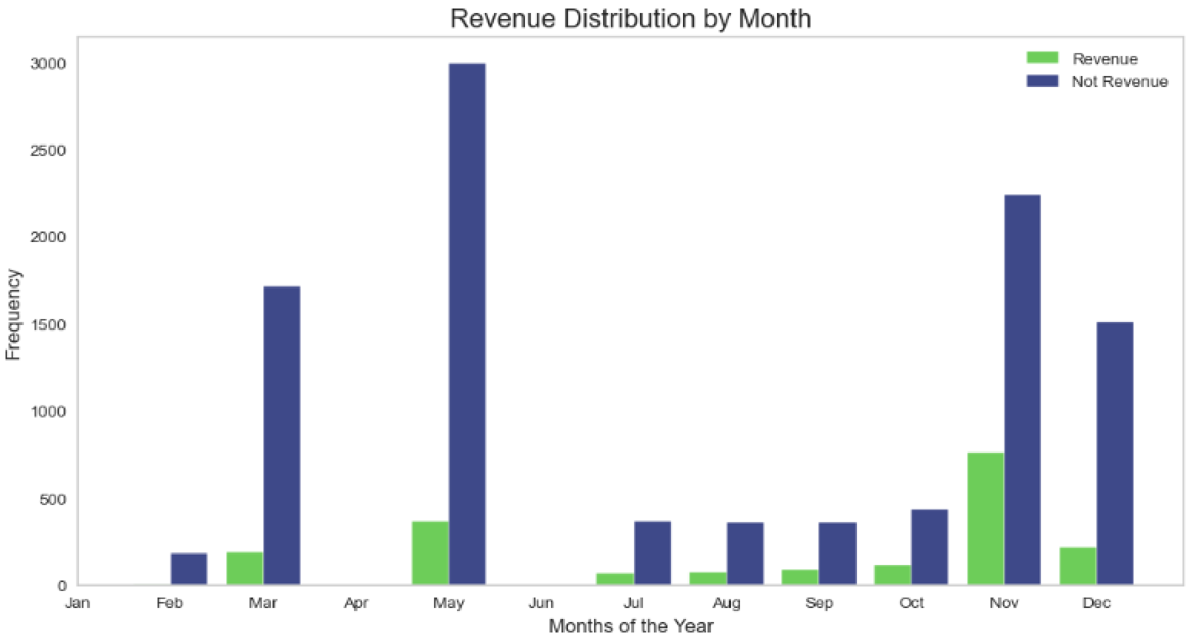


Figure 6 Revenue Distribution by Month

In *Figure 7 and Figure 8*, we observe the histograms of the variables “adm”, “inf”, “prod_rel”, “os”, “browser”, and “reg”. They are numerical variables. We can observe that they are skewed to the right. We also can observe the presence of outliers. Besides, we have comparable frequencies between them. Correlation analysis of each column must be done to determine if all of them are necessary for our analysis.

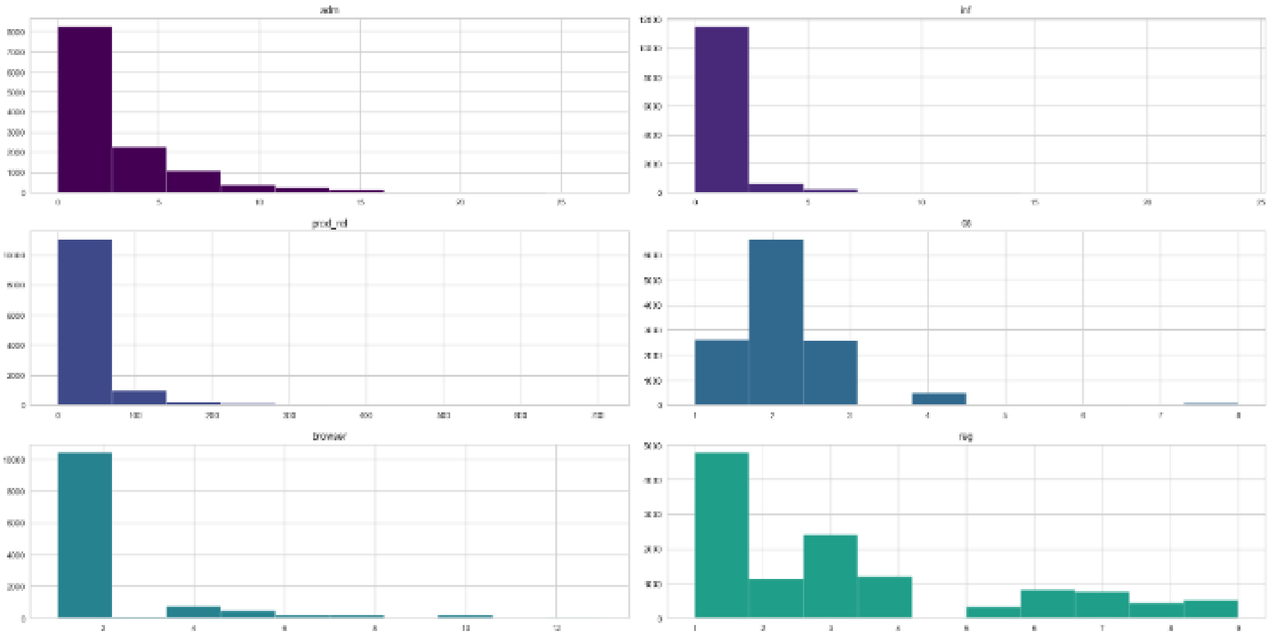


Figure 7 The distribution of numerical attributes

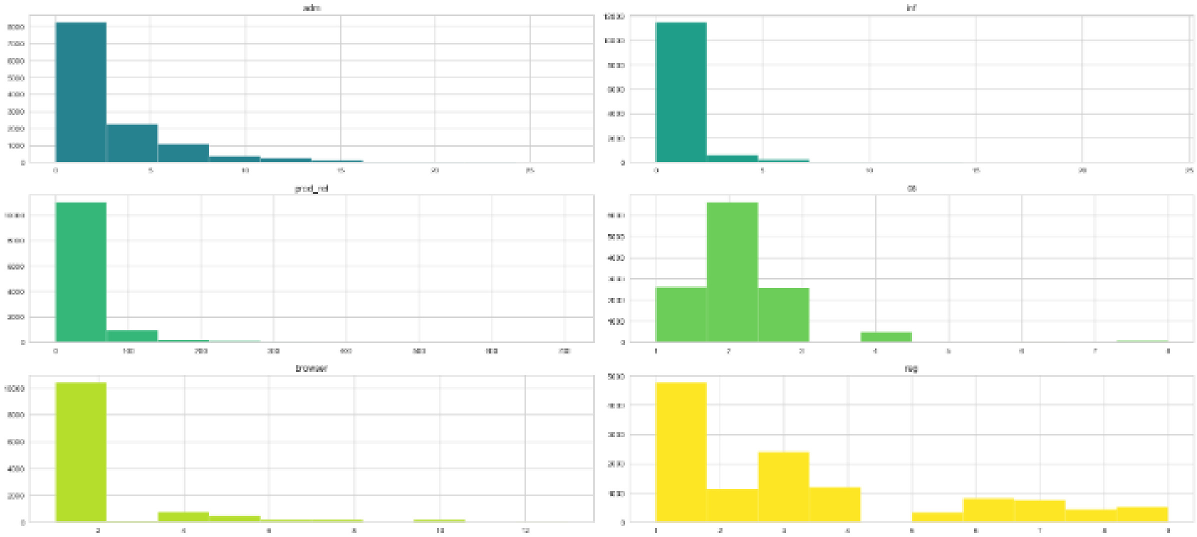


Figure 8 The distribution of numerical attributes

In *Figures 9 and 10*, We notice the existence of outliers and their significant occurrences in the data. To address these outliers, we have decided to use a Robust scaler method to scale the data. Since the outliers are important to consider in the model for our analysis.

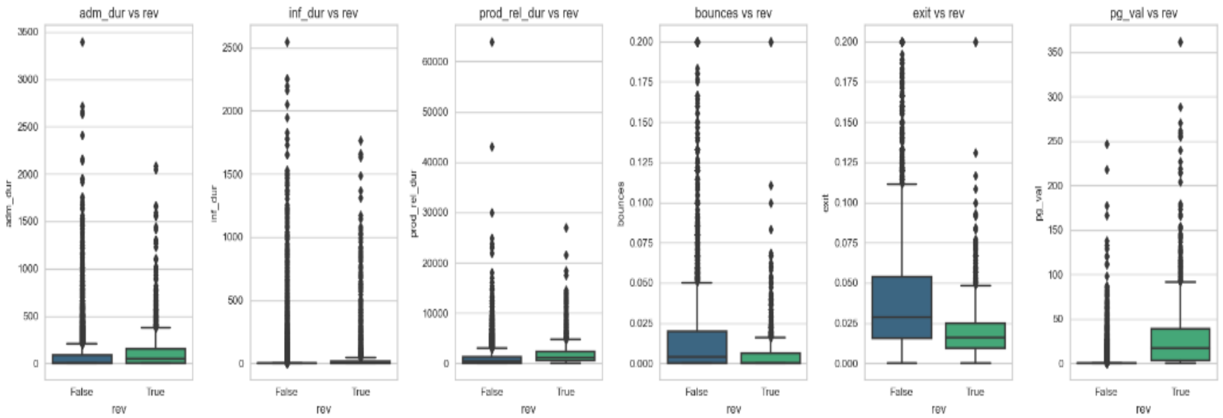


Figure 9 The presence of Outliers

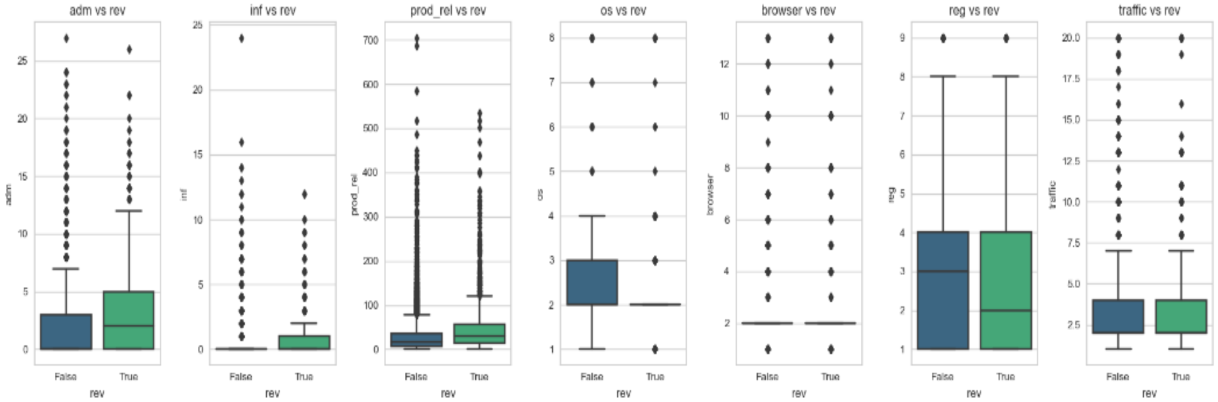


Figure 10 The presence of Outliers

In *Figure 11,* we can compare the IQR between the attributes and the biggest ones are in “prod_rel_dur”, “adm_dur”, and “prod_rel”.

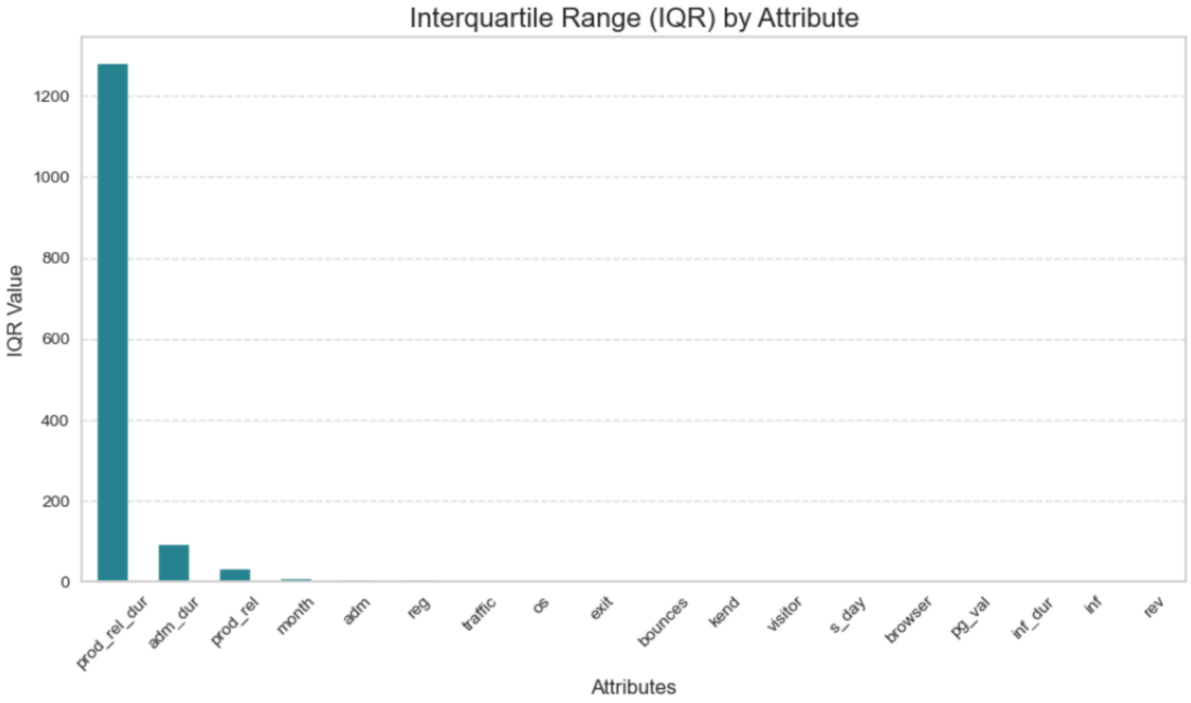


Figure 11 Interquartile Range by Attribute

Previously we saw in histograms that our data is mostly skewed to the right and in *Figure 12*, we can see what features are more skewed. This would be helpful if we had missing values.

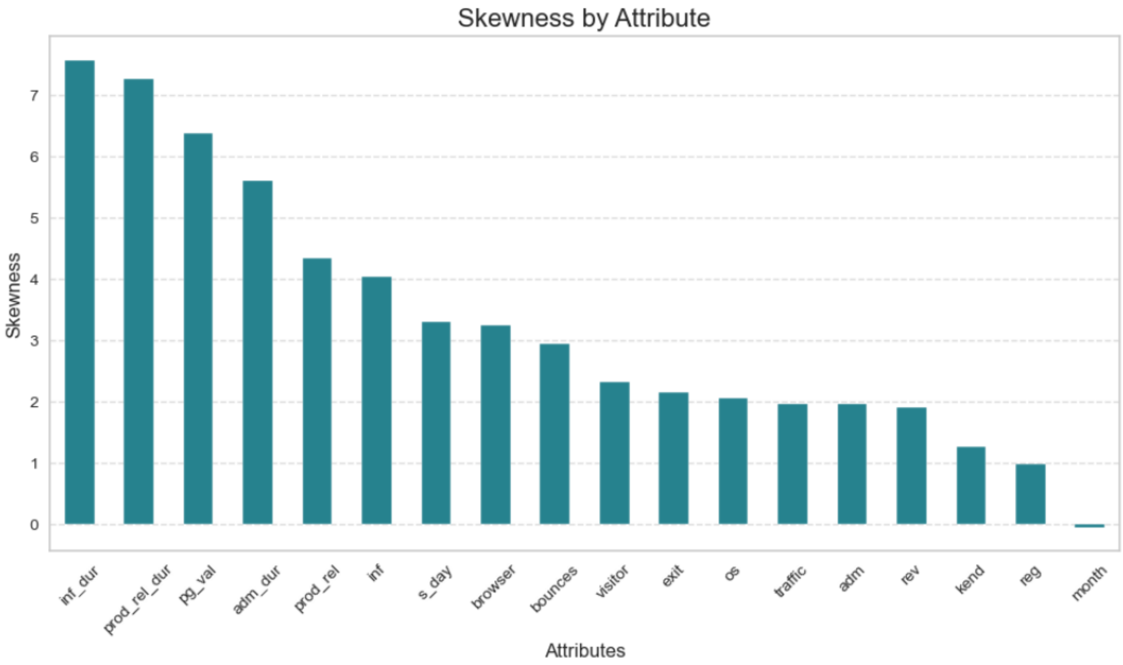
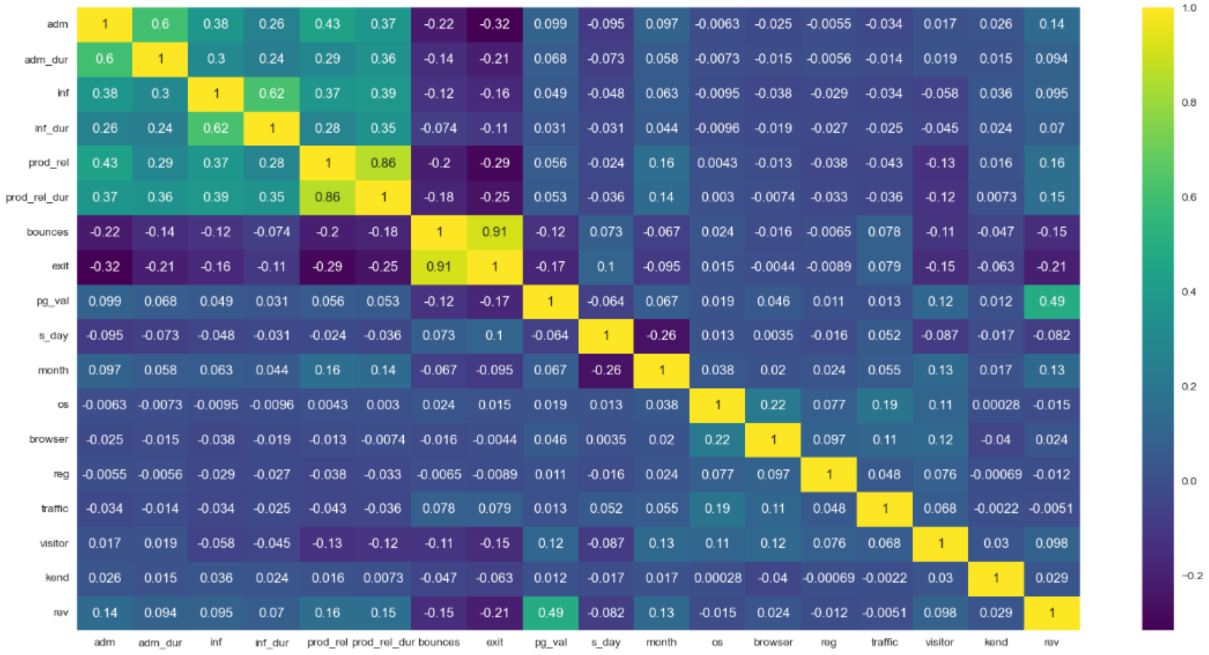


Figure 12 Skewness by Attribute

Heat Map

In the heatmap provided, *Figure 13*, we can confirm if the variables are correlated or not and the column of analysis is the target variable "rev", for continuous variables. The attributes "pg_val" and the Revenue target variable appear to have a moderately positive correlation, as indicated by the correlation coefficient of 0.49.



	adm	adm_dur	inf	inf_dur	prod_rel	prod_rel_dur	bounces	exit	pg_val	s_day	month	os	browser	reg	traffic	visitor	kend	rev
adm	1	0.6	0.38	0.26	0.43	0.37	-0.22	-0.32	0.099	-0.095	0.097	-0.0063	-0.025	-0.0055	-0.034	0.017	0.026	0.14
adm_dur	0.6	1	0.3	0.24	0.29	0.36	-0.14	-0.21	0.068	-0.073	0.058	-0.0073	-0.015	-0.0056	-0.014	0.019	0.015	0.094
inf	0.38	0.3	1	0.62	0.37	0.39	-0.12	-0.16	0.049	-0.048	0.063	-0.0095	-0.038	-0.029	-0.034	-0.058	0.036	0.095
inf_dur	0.26	0.24	0.62	1	0.28	0.35	-0.074	-0.11	0.031	-0.031	0.044	-0.0096	-0.019	-0.027	-0.025	-0.045	0.024	0.07
prod_rel	0.43	0.29	0.37	0.28	1	0.86	-0.2	-0.29	0.056	-0.024	0.16	0.0043	-0.013	-0.038	-0.043	-0.13	0.016	0.16
prod_rel_dur	0.37	0.36	0.39	0.35	0.86	1	-0.18	-0.25	0.053	-0.036	0.14	0.003	-0.0074	-0.033	-0.036	-0.12	0.0073	0.15
bounces	-0.22	-0.14	-0.12	-0.074	-0.2	-0.18	1	0.91	-0.12	0.073	-0.067	0.024	-0.016	-0.0065	0.078	-0.11	-0.047	-0.15
exit	-0.32	-0.21	-0.16	-0.11	-0.29	-0.25	0.91	1	-0.17	0.1	-0.095	0.015	-0.0044	-0.0089	0.079	-0.15	-0.063	-0.21
pg_val	0.099	0.068	0.049	0.031	0.056	0.053	-0.12	-0.17	1	-0.064	0.067	0.019	0.046	0.011	0.013	0.12	0.012	0.49
s_day	-0.095	-0.073	-0.048	-0.031	-0.024	-0.036	0.073	0.1	-0.064	1	-0.26	0.013	0.0035	-0.016	0.052	-0.087	-0.017	-0.082
month	0.097	0.058	0.063	0.044	0.16	0.14	-0.067	-0.095	0.067	-0.26	1	0.038	0.02	0.024	0.055	0.13	0.017	0.13
os	-0.0063	-0.0073	-0.0095	-0.0096	0.0043	0.003	0.024	0.015	0.019	0.013	0.038	1	0.22	0.077	0.19	0.11	0.00028	-0.015
browser	-0.025	-0.015	-0.038	-0.019	-0.013	-0.0074	-0.016	-0.0044	0.046	0.0035	0.02	0.22	1	0.097	0.11	0.12	-0.04	0.024
reg	-0.0055	-0.0056	-0.029	-0.027	-0.038	-0.033	-0.0065	-0.0089	0.011	-0.016	0.024	0.077	0.097	1	0.048	0.076	-0.00069	-0.012
traffic	-0.034	-0.014	-0.034	-0.025	-0.043	-0.036	0.078	0.079	0.013	0.052	0.055	0.19	0.11	0.048	1	0.068	-0.0022	-0.0051
visitor	0.017	0.019	-0.058	-0.045	-0.13	-0.12	-0.11	-0.15	0.12	-0.087	0.13	0.11	0.12	0.076	0.068	1	0.03	0.098
kend	0.026	0.015	0.036	0.024	0.016	0.0073	-0.047	-0.063	0.012	-0.017	0.017	0.00028	-0.04	-0.00069	-0.0022	0.03	1	0.029
rev	0.14	0.094	0.095	0.07	0.16	0.15	-0.15	-0.21	0.49	-0.082	0.13	-0.015	0.024	-0.012	-0.0051	0.098	0.029	1

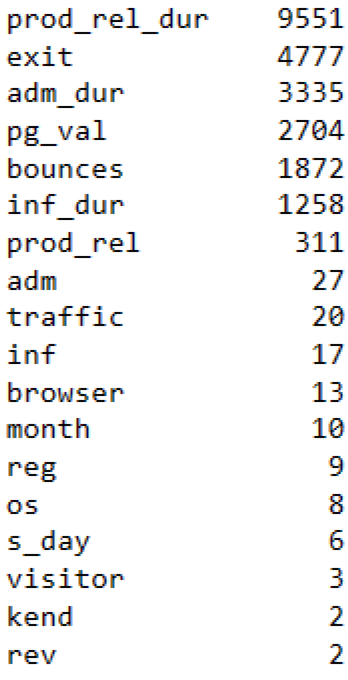
Figure 13 Correlation HeatMap

Correlation

Bruce, Gedeck. and Bruce. (2020, p.30) believe that exploratory data analysis in many modelling projects (whether in data science or in research) involves examining correlations among predictors and between predictors and a target variable. Variables X and y (each with measured data) are said to be positively correlated if high values of X go with high values of y, and low values of X go with low values of y. If high values of X go with low values of y, and vice versa, the variables are negatively correlated.

To deeply analyze the correlation between the data's variables, we decided to use hypothesis tests: ANOVA and Chi-squared. The aim is to know which variables are correlated with our target variable Revenue ("rev").

In line with Hashmi's (2020) Titanic survival prediction case study in Python, we adopt the author's criterion for variable categorization. According to this criterion, a variable is considered categorical if it has fewer than 20 unique values; otherwise, it is classified as continuous. In this project, we adapted the number to 30 unique values, after careful verification of each attribute. We are going to split our data that have less and more than 30 unique values. (*See Figure 14*)



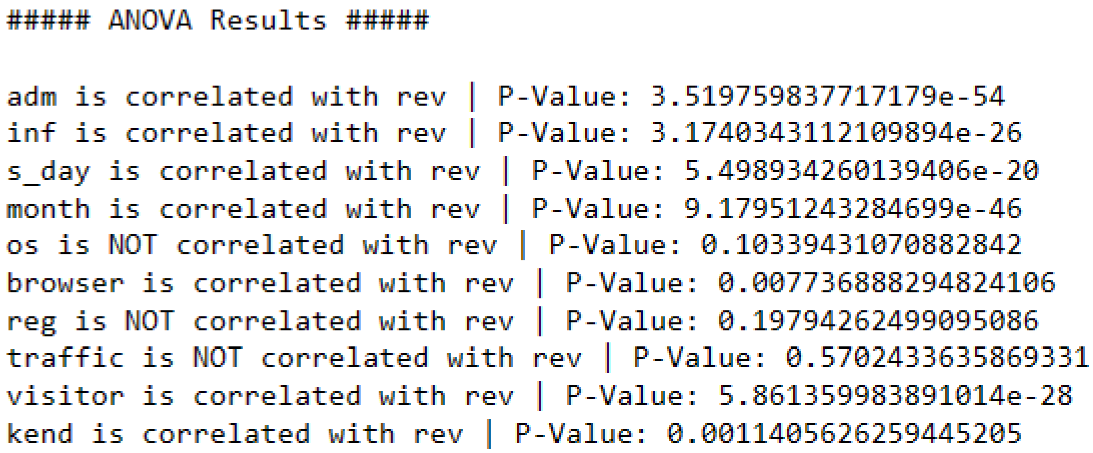
```
prod_rel_dur    9551
exit            4777
adm_dur         3335
pg_val          2704
bounces         1872
inf_dur         1258
prod_rel         311
adm               27
traffic           20
inf               17
browser           13
month             10
reg                9
os                 8
s_day              6
visitor            3
kend               2
rev                2
```

Figure 14 Number of unique values per attribute

ANOVA Test

The ANOVA test is applied to categorical features. As explained before, we are testing the features with less than 30 unique values, and our hypothesis is to prove if they are correlated with our target variable or not.

In *Figure 15*, ANOVA compares the p-value of each column, and if it is less than 0.05, the variable is correlated. The columns “os”, “reg”, and “traffic” are not correlated with our target variable. So, we are not going to drop those columns.



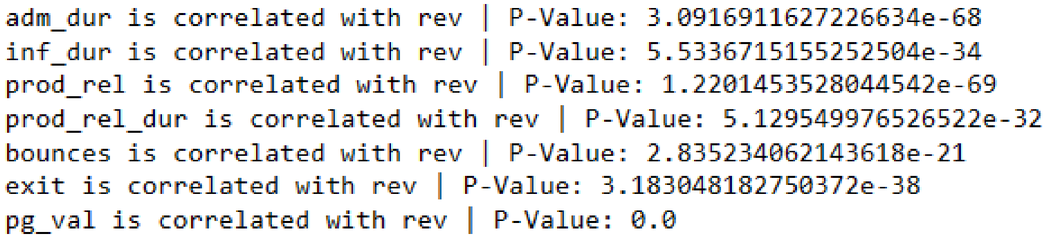
```
##### ANOVA Results #####

adm is correlated with rev | P-Value: 3.519759837717179e-54
inf is correlated with rev | P-Value: 3.1740343112109894e-26
s_day is correlated with rev | P-Value: 5.498934260139406e-20
month is correlated with rev | P-Value: 9.17951243284699e-46
os is NOT correlated with rev | P-Value: 0.10339431070882842
browser is correlated with rev | P-Value: 0.007736888294824106
reg is NOT correlated with rev | P-Value: 0.19794262499095086
traffic is NOT correlated with rev | P-Value: 0.5702433635869331
visitor is correlated with rev | P-Value: 5.861359983891014e-28
kend is correlated with rev | P-Value: 0.0011405626259445205
```

Figure 15 ANOVA Test Results

Chi-Squared Test

The Chi-Squared Test method is applied to numerical variables. As explained before, the columns with more than 30 unique values. The hypothesis analyzes if they are correlated with our target variable Revenue (“rev”).

The results of the Chi-Squared Test show that all of them are correlated for our analysis and application to the machine learning model. The metric is the same as the ANOVA Test, if the p-value < 0.05, the variable is correlated. (*See Figure 16*)



```
adm_dur is correlated with rev | P-Value: 3.0916911627226634e-68
inf_dur is correlated with rev | P-Value: 5.5336715155252504e-34
prod_rel is correlated with rev | P-Value: 1.2201453528044542e-69
prod_rel_dur is correlated with rev | P-Value: 5.129549976526522e-32
bounces is correlated with rev | P-Value: 2.835234062143618e-21
exit is correlated with rev | P-Value: 3.183048182750372e-38
pg_val is correlated with rev | P-Value: 0.0
```

Figure 16 Chi-Squared Test Results

Cluster Analysis: Bounce Rates, Exit Rates, Page Engagement, and Revenue Insights

In this project, we conducted a cluster analysis of the data to identify patterns and groups of objects in the dataset.

Cluster analysis, as defined by Wikipedia Contributors (2019), is the process of organizing a collection of objects so that objects within the same cluster are more like one another than they are to objects outside of it.

K-means clustering was the machine learning model that was used. According to Sharma (2019), it makes use of vector quantization and seeks to allocate each observation to the cluster that has the closest mean or centroid, acting as a prototype for the cluster.

To get this result, was necessary to modify the data until the result achieved the desired properties:

- The pages visited sum attribute was created. It sums the observations for three existing columns: 'adm', 'inf', and 'prod_rel'.
- The duration sum also was created. It sums the observations for 3 existing columns: and 'adm_dur', 'inf_dur', 'prod_rel_dur'. The results were divided by 60, so the results display would be shown in hours to improve clarity.
- The outliers for duration sum were handled using the most logical value to be replaced in place of outliers. Outliers were replaced by the number three hundred. The idea used was seen in the case study Titanic survival prediction case study in Python created by Hashmi (2020).

The Elbow method was used to define the appropriate number of clusters for the data. In this analysis, the number was three. (*Figure 17)*

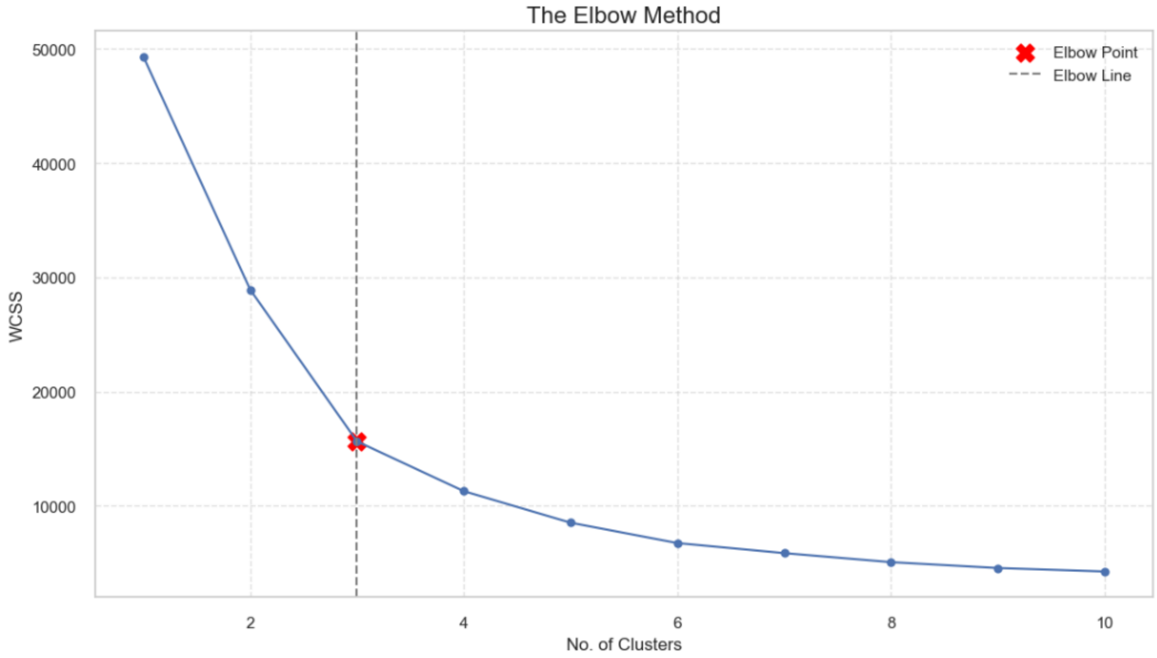


Figure 17 The Elbow Method

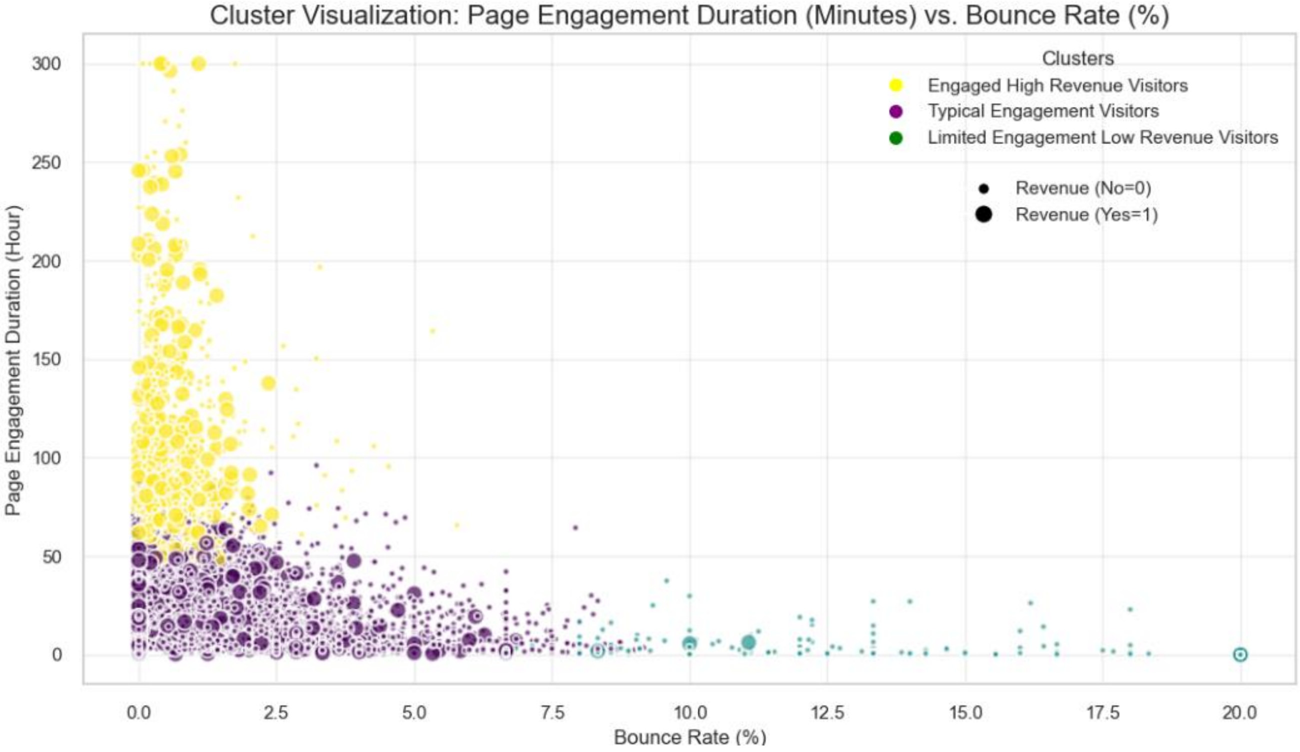
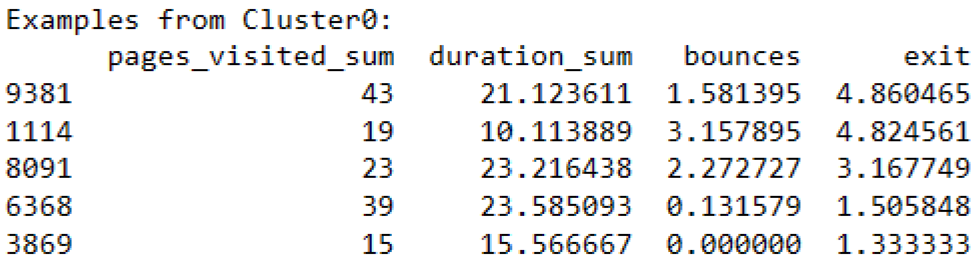


Figure 18 Cluster Analysis

In Figure 18, we notice the cluster formed by the online shoppers' behaviour. The clusters' descriptions are:

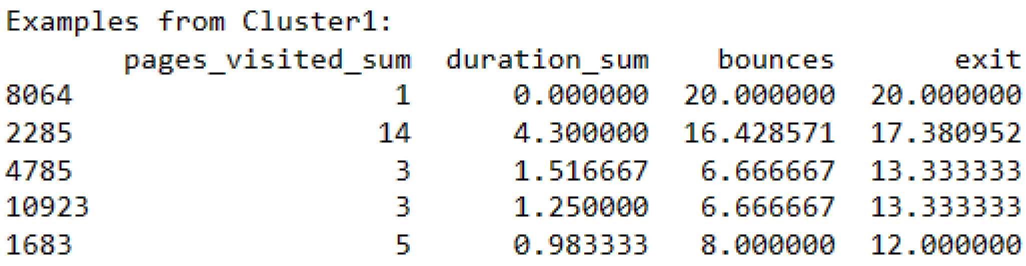
- Cluster 0: Typical Engagement Visitors: The largest cluster with moderate (Duration sum < 50h), and moderate bounce rate (up to 9%), generating a few revenues. *(See Figure 19)*



```
Examples from Cluster0:
      pages_visited_sum  duration_sum   bounces      exit
9381                 43     21.123611  1.581395  4.860465
1114                 19     10.113889  3.157895  4.824561
8091                 23     23.216438  2.272727  3.167749
6368                 39     23.585093  0.131579  1.505848
3869                 15     15.566667  0.000000  1.333333
```

Figure 19 Examples from Cluster 0

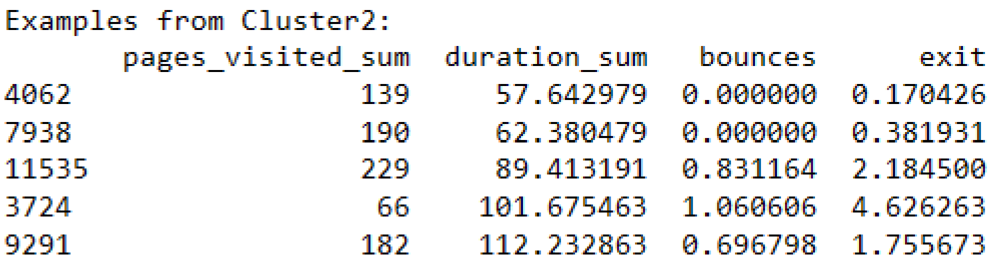
- Cluster 1: Limited Engagement Low Revenue Visitors: Visitors with low engagement (small duration) and high bounce rate (> 8%), with the majority not generating revenue. *(See Figure 20)*



```
Examples from Cluster1:
       pages_visited_sum  duration_sum    bounces       exit
8064                   1      0.000000  20.000000  20.000000
2285                  14      4.300000  16.428571  17.380952
4785                   3      1.516667   6.666667  13.333333
10923                  3      1.250000   6.666667  13.333333
1683                   5      0.983333   8.000000  12.000000
```

Figure 20 Examples from Cluster 1

- Cluster 2: Engaged High Revenue Visitors: Visitors with high engagement (Duration sum > 50h), low bounce rate (< 5%), and most of the revenue generated. *(See Figure 21)*



```
Examples from Cluster2:
       pages_visited_sum  duration_sum   bounces      exit
4062                 139     57.642979  0.000000  0.170426
7938                 190     62.380479  0.000000  0.381931
11535                229     89.413191  0.831164  2.184500
3724                  66    101.675463  1.060606  4.626263
9291                 182    112.232863  0.696798  1.755673
```

Figure 21Example from Cluster 2

Cluster Analysis Conclusion

In conclusion, the cluster analysis has revealed a better comprehension of the dataset's underlying structures. The discovered clusters provide information that can help allocate resources more wisely, enhance customer satisfaction, and ultimately help the company succeed.

Flowchart of Data Preparation and Modelling

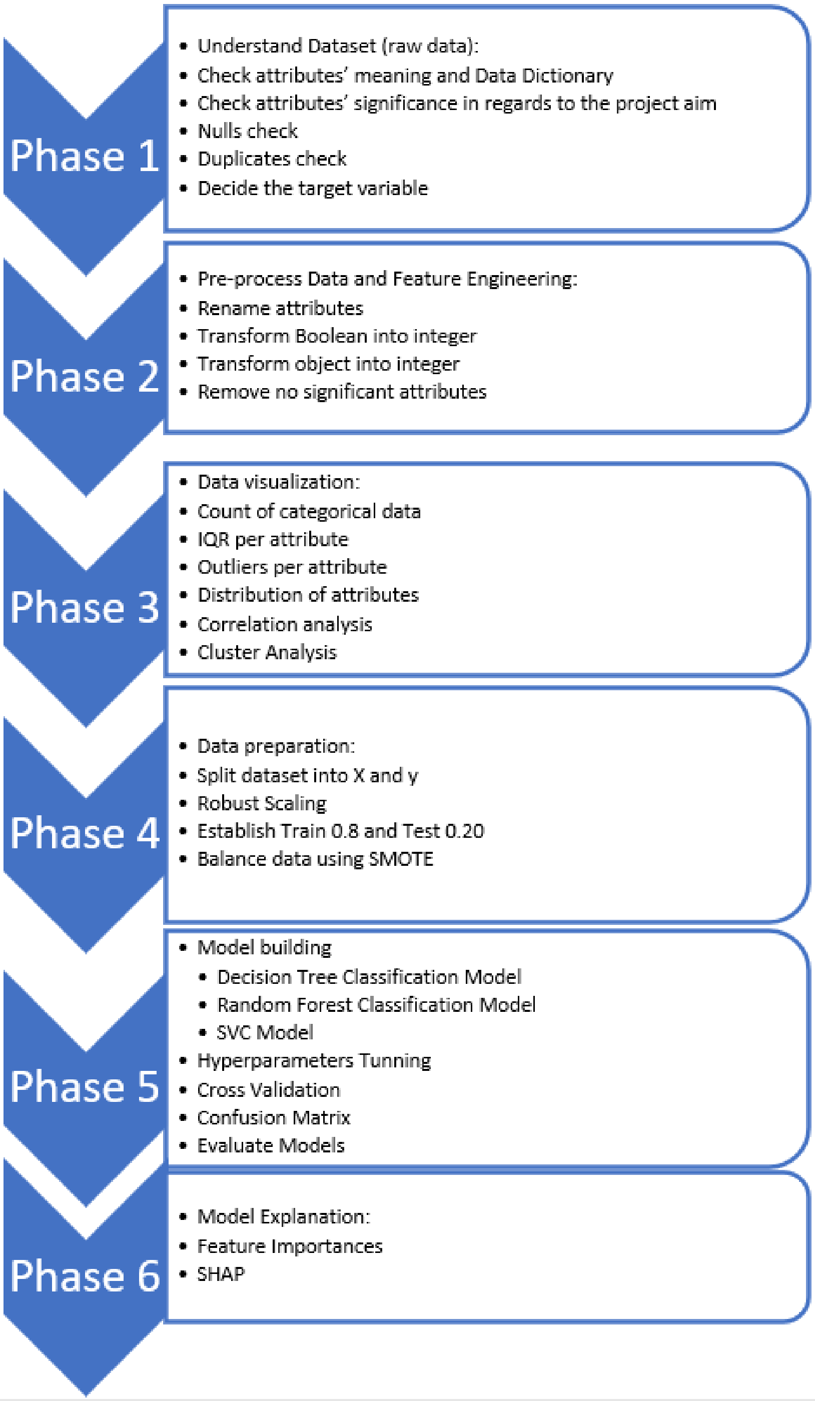

Phase 1
Understand Dataset (raw data):
Check attributes' meaning and Data Dictionary
Check attributes' significance in regards to the project aim
Nulls check
Duplicates check
Decide the target variable
Phase 2
Pre-process Data and Feature Engineering:
Rename attributes
Transform Boolean into integer
Transform object into integer
Remove no significant attributes
Phase 3
Data visualization:
Count of categorical data
IQR per attribute
Outliers per attribute
Distribution of attributes
Correlation analysis
Cluster Analysis
Phase 4
Data preparation:
Split dataset into X and y
Robust Scaling
Establish Train 0.8 and Test 0.20
Balance data using SMOTE
Phase 5
Model building
Decision Tree Classification Model
Random Forest Classification Model
SVC Model
Hyperparameters Tunning
Cross Validation
Confusion Matrix
Evaluate Models
Phase 6
Model Explanation:
Feature Importances
SHAP

Figure 22 Flowchart of Data Preparation and Modelling

Data Preparation and Preprocessing

During the analysis, we used different steps regarding data preparation, such as splitting the data, normalizing the data, and balancing the data.

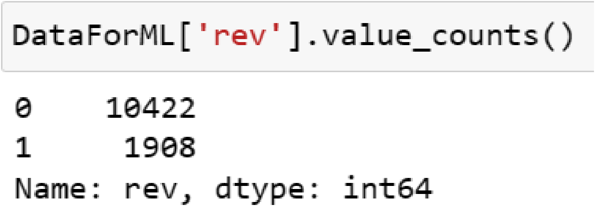
Normalizing the data

To work with this dataset, we used different methods such as Standarscaler, Scale, MinMaxScaler, and RobustScaler to see which one suits the data the best.
However, after applying them, we have chosen the RobustScaler normalization technique. It works better for that dataset due to the number of outliers. Additionally, the other techniques were discarded after we applied them because we observed how our accuracy had decreased significantly.

Balancing the data

In this project, we attempt to use 2 different techniques to balance our data, NearMiss and SMOTE techniques, and see which one would work better giving us the best result.

We proceed to use the SMOTE (Synthetic Minority Over-sampling Technique) to balance the class distribution of the target variable **”rev”** in the DataForML dataset. The method is used to oversample the minority class **(rev = 1)** by creating synthetic samples. See in *Figure 23*, the values count per class. It was also shown in *Figure 3*, as a graph.

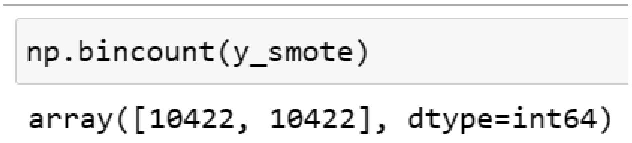


```
DataForML['rev'].value_counts()
```

```
0    10422
1     1908
Name: rev, dtype: int64
```

Figure 23 Values count per class

The output shows that both classes now have the same number of observations 10422 each. (*See Figure 24)*



```
np.bincount(y_smote)
```

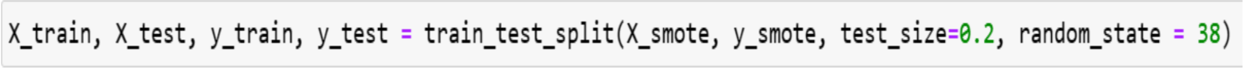
```
array([10422, 10422], dtype=int64)
```

Figure 24 Values count after the SMOTE technique

Splitting the data

As we are attempting to find the prediction from people who would generate revenue, we determined that the target variable is Revenue ("rev"). In the character matrix **"**X_smote" and the target variable "y_smote" are inputs produced by the SMOTE technique.

Furthermore, it is specified that the test size should be 20% of the total data size, and the random state is set to 38 for reproducibility. Test sizes of 10% and 30% were performed. However, the 20% test size delivered the best performance in our model.



```
X_train, X_test, y_train, y_test = train_test_split(X_smote, y_smote, test_size=0.2, random_state = 38)
```

Figure 25 Train and Test Code

Dimensionality Reduction

Additionally, tests were performed, including the Principal Component Analysis (PCA). However, machine learning worked poorly since it reduced our accuracy. We have decided not to apply dimensionality reduction in this paper.

Feature Engineering

To get a better analysis we encode categorical values into numerical, representing each category with a number to make it simple to analyze. Duplicated values and missing data were not found in our data set. Three features were removed based on the results of ANOVA and Chi-squared test. After dropping the columns that are not correlated, the dataset was modified, and now it has 12,330 rows and 15 features.

Models

We experimented using three of the most common models for classification problems to find the best accuracy. They are Decision Tree Classifier, Support Vector Machine, and Random Forest.

Challenges encountered

The process of analyzing the data presented several difficulties. Choosing the best scaler method and weighing whether or not to include outliers was one of the biggest challenges. Furthermore, determining each variable's correlation, resolving data imbalance, and choosing the right hyperparameters for model evaluation were major obstacles. The fact that our data only covered a single year presented another significant obstacle.

Inclusion of strategies to overcome them

We used ANOVA and Chi-squared tests to determine if the features were correlated, and we removed three features from the data.

We also decided to include the outliers in our analysis since we have many of them that are part of our analysis, and most of them are spared because they represent the duration of time.

We also tried different scaling methods to see how the model performed. In the end, we decided to use Robust Scaler to include the outliers as part of our model since they appear because they represent durations of time and are important for our analysis.

We also used SMOTE since our target variable was unbalanced, and instead of reducing observations that could drop important information, we decided to create synthetic data.

Model Building and Evaluation

Three different machine learning models are applied: Decision Tree Classifier (CART), Random Forest Classifier (RF), and Support Vector Machine (SVC). See the step-by-step:

1. First of all, the accuracy of each model was measured by fitting the model using the default hyperparameters.
2. Second, the cross-validation was applied to observe minimum accuracy, maximum accuracy and average accuracy across 10-folds.
3. Third, GridSearchCV was used to identify the best hyperparameters.
4. Fourth, to compare the accuracy improvement, each model was tuned using the optimal hyperparameters.

Cross Validation Method

According to (Daniel, 2019), one of the most popular techniques for resampling data and estimating the true prediction error of models is cross validation, which is also used to adjust model parameters. It is a technique for resampling data to evaluate how well predictive models generalize and to avoid overfitting. Random subsampling is carried out k times in a K-fold cross. No two test sets are sampled in a way that results in overlap. Until every one of the k subsets has functioned as a validation set, this process is repeated.

Using the K-fold cross-validation technique is a practical method that enables estimation of the classifier's generalization error as well as hyperparameter tuning (Anguita et al., 2012). In this paper, the stratified cross-validation method is used due to the unbalanced classes. This technique ensures that the class proportions in each subset accurately reflect the proportions in the learning set (Daniel, 2019).

According to (Jung, 2017) a typical choice of k is between 5 and 10. (Anguita et al., 2012) states that typical values are 5, 10 and 20. In the paper, The 'K' in K-fold Cross Validation the best-performing cross-validation is 10-fold cross-validation for real-world datasets (Daniel, 2019). In this paper, the analysis was done using 10-folds.

Hyperparameters Tuning

In this paper, GridSearchCV was used to identify the optimal hyperparameters. According to (Probst, Wright and Boulesteix, 2019), GridSearchCV is one of the simplest strategies to analyze all possible combinations of parameters using k-fold cross validation. The process of optimizing a learning algorithm's hyperparameters for a given dataset is known as tuning.

According to Weerts, Mueller and Vanschoren (2020), Setting an algorithm's hyperparameters affects how well it performs on a particular learning task. Machine learning practitioners can adjust the hyperparameters to achieve optimal performance. They analysed the importance of hyperparameter tuning in 59 datasets taken from an open source using Random Forest (RF) and Support Vector Machine (SVC) algorithms. Their finding brings to light the importance of tuning max_features of the Random Forest and gamma of the Support Vector Machine depending on the number of features in the dataset. They also conclude that fixing min_samples_leaf to 1, and the high number of trees bring the best results in terms of the performance RF model. A similar conclusion regarding the number of trees is brought by Probst, Wright and Boulesteix (2019).

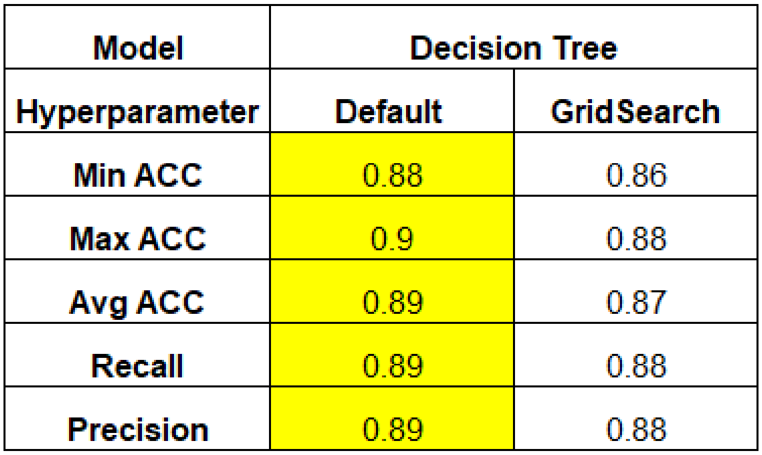
As in this paper, the same models are being used. So, their finding was considered during the tests.

An overview of the Decision Tree Classifier, Support Vector Machine, and Random Forest Classifier Models and the results are provided in the following part of this report, along with a thorough analysis of the evaluation's findings and their implications for the data aim.

Building a Decision Tree Classifier Model

In this report, we import the Decision Tree Classifier from the Sklearn library, and we fit the model using the default parameters. Then we used GridSearchCV to find the optimal hyperparameters. After the results, we fit the model again using the optimal hyperparameters. The results can be observed in *Table 1*.

Table 1 Decision Tree Classifier Results



Model	Decision Tree	
Hyperparameter	Default	GridSearch
Min ACC	0.88	0.86
Max ACC	0.9	0.88
Avg ACC	0.89	0.87
Recall	0.89	0.88
Precision	0.89	0.88

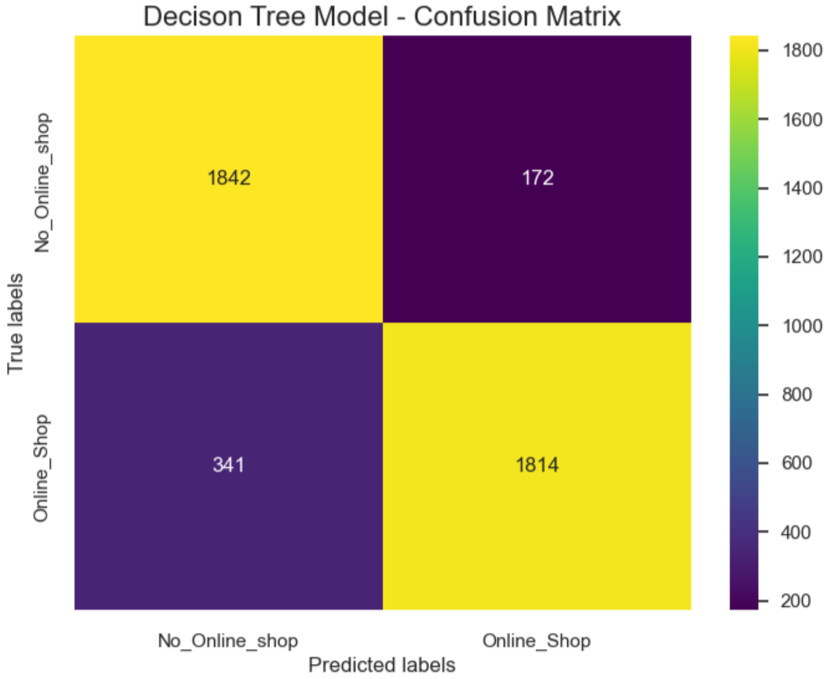


Figure 26 Confusion Matrix: Decision Tree Model

In *Figure 26*, We observe that the model predicted 1842 true negatives (TN) and 1814 true positives (TP) while misclassifying 341 instances as false negatives (FN) and 172 instances as false positives (FP).

Results and analysis: Decision Tree Classifier Model

We notice that the model using default hyperparameters performed better:

The minimum accuracy score is: 0.8848920863309353 and the maximum accuracy score is: 0.9082183563287343 across 10 folds. The average accuracy score is 0.8985312649700277 across 10 folds.

The results using the optimal hyperparameters are: The minimum accuracy score is: 0.8621103117505995 and the maximum accuracy score is: 0.8848230353929214 across 10 folds. The average accuracy score is 0.8712452113893768 across 10 folds.

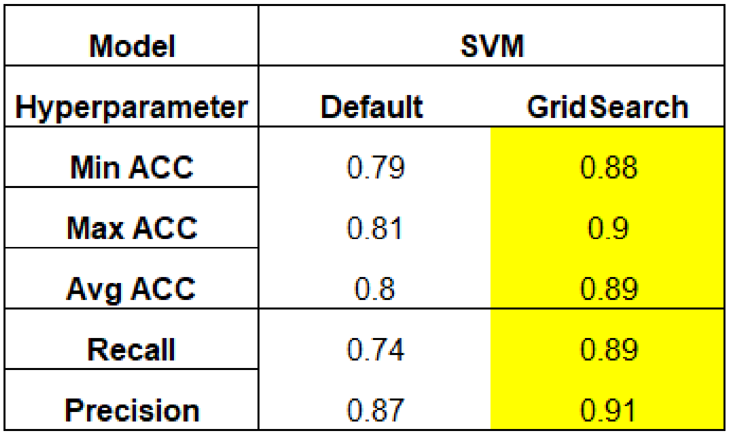
Precision and Recall: Decrease its performance from 89 to 88 when the model was tuned.

Building Support Vector Machine Model

According to (Anguita et al., 2012), within the context of classification problems, the Support Vector Machine (SVM) is one of the most advanced methods available. Nevertheless, the SVM's learning process is not finished by the pursuit of ideal parameters. Actually, to choose the optimal model, a set of extra variables known as the hyperparameters must be adjusted.

Similarly, to the previous model, the same steps were performed. See model results in *Table 2*.

Table 2 Support Vector Machine Results



Model	SVM	
Hyperparameter	Default	GridSearch
Min ACC	0.79	0.88
Max ACC	0.81	0.9
Avg ACC	0.8	0.89
Recall	0.74	0.89
Precision	0.87	0.91

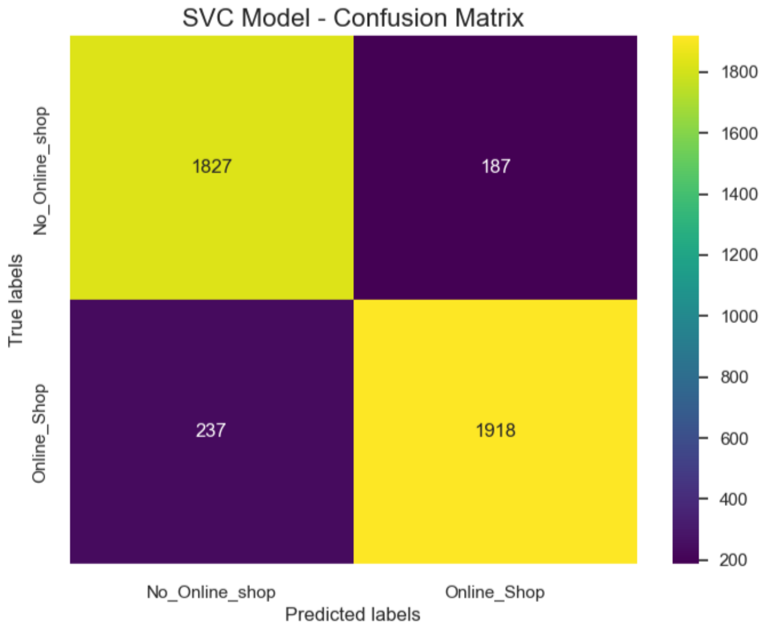


Figure 27 Confusion Matrix: SVC Model

In *Figure 27*, We observe that the model predicted 1827 true negatives (TN) and 1918 true positives (TP) while misclassifying 237 instances as false negatives (FN) and 187 instances as false positives (FP).

Results and analysis: Support Vector Machine Model

We notice that the model using optimal hyperparameters performed better:

The minimum accuracy score is: 0.7985611510791367 and the maximum accuracy score is: 0.8152369526094781. The average accuracy score is 0.8068978290672801 across 10 folds.

The results using the optimal hyperparameters are: The minimum accuracy score is: 0.886622675464907 and the maximum accuracy score is: 0.9010197960407919
The average accuracy score is 0.8937929320610698 across 10 folds.

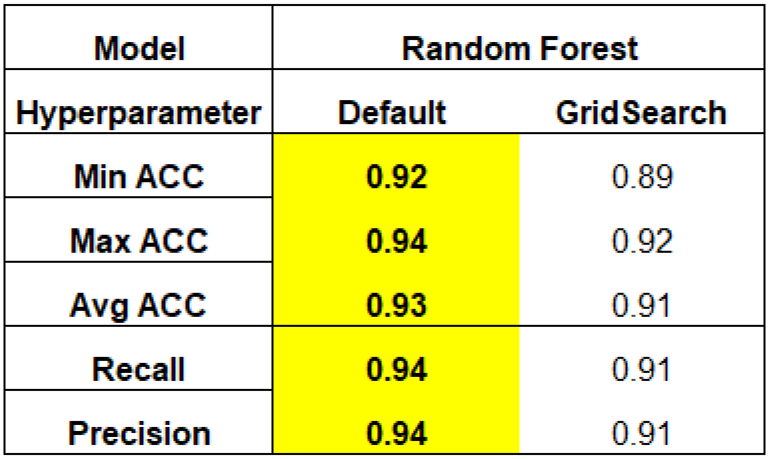
Precision: Improved from 74 to 89 when the model was tuned.

Recall: Improved from 87 to 91 when the model was tuned.
.

Building Random Forest Classifier Model

Random Forest Classifier Model was imported from the Sklearn library. Similarly, to the previous model, we fit the model using the default parameters. Then we used GridSearchCV to find the optimal hyperparameters. After getting the optimal results, we fit the model again using the optimal hyperparameters. See model results in *Table 3*.

Table 3 Random Forest Classifier Results



Model	Random Forest	
Hyperparameter	Default	GridSearch
Min ACC	**0.92**	0.89
Max ACC	**0.94**	0.92
Avg ACC	**0.93**	0.91
Recall	**0.94**	0.91
Precision	**0.94**	0.91

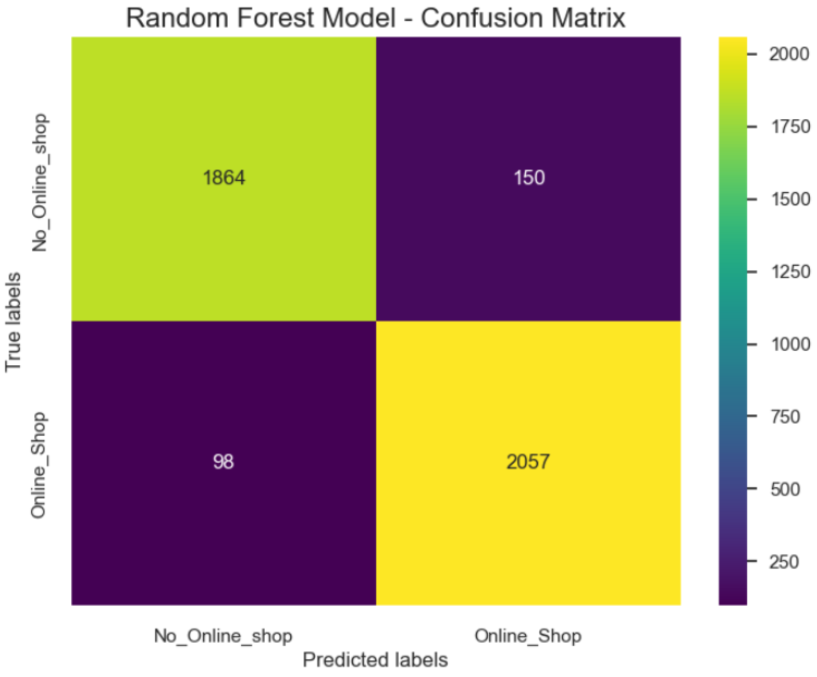


Figure 28 Confusion Matrix: Random Forest Classifier

In *Figure 28*, We observe that the model predicted 1864 true negatives (TN) and 2057 true positives (TP) while misclassifying 98 instances as false negatives (FN) and 150 instances as false positives (FP).

Results and analysis: Random Forest Classifier Model

We notice that the model using default hyperparameters performed better. According to the research (Probst, Wright and Boulesteix, 2019), the RF model works well with the default values of the hyperparameters. When adjusting the settings of a machine learning model, it is important to be aware of overfitting. It occurs when the model becomes too complex. It can lead to complex rules that can cause the memorization of the data too precisely, and it causes poor performance on new, unseen data.
According to Probst, Wright and Boulesteix (2019), a Random Forest is less far-tuneable than a Support Vector Machine. In this analysis, it is proved based on empirical performance. After tuning, The SVC model improves by approximately ten per cent in accuracy, Recall and Precision while RF decreases by three per cent.

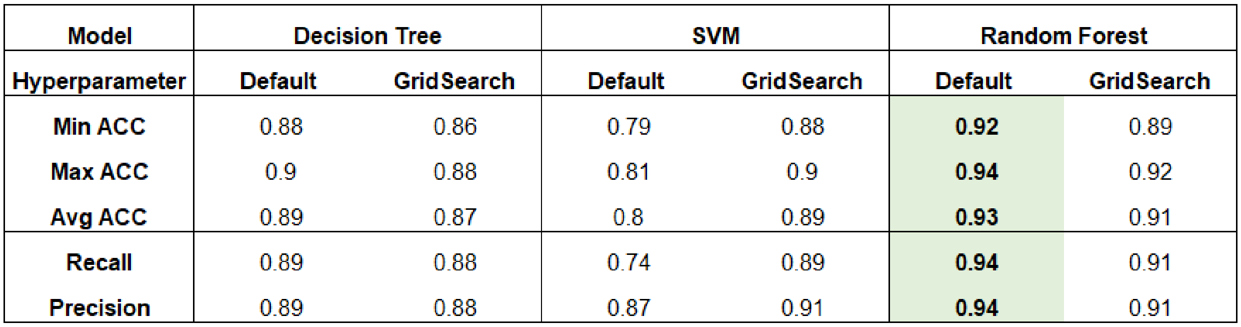
The minimum accuracy score is: 0.920863309352518 and the maximum accuracy score is: 0.9436450839328537. The average accuracy score is 0.9339135050687704 across 10 folds.

The results using the optimal hyperparameters are: The minimum accuracy score is: 0.8926214757048591 and the maximum accuracy score is: 0.922615476904619. The average accuracy score is 0.9113033868046534 across 10 folds.

Precision and Recall: Improved from 91 to 94 when the model was tuned.

Comparison of Machine Learning Models Performance

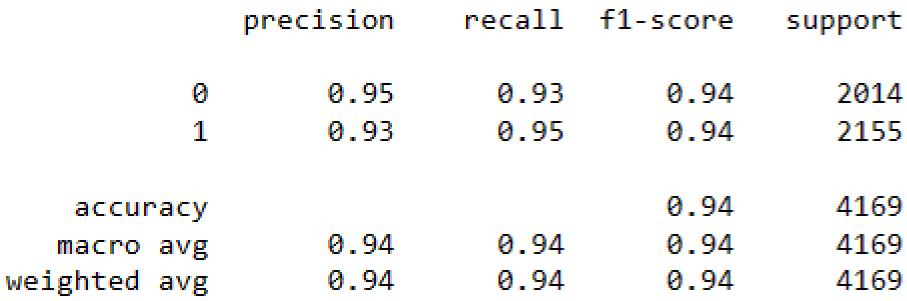
The Random Forest Classifier Model had the best performance when compared with the other two models used in this analysis.



Model	Decision Tree		SVM		Random Forest	
Hyperparameter	Default	GridSearch	Default	GridSearch	Default	GridSearch
Min ACC	0.88	0.86	0.79	0.88	**0.92**	0.89
Max ACC	0.9	0.88	0.81	0.9	**0.94**	0.92
Avg ACC	0.89	0.87	0.8	0.89	**0.93**	0.91
Recall	0.89	0.88	0.74	0.89	**0.94**	0.91
Precision	0.89	0.88	0.87	0.91	**0.94**	0.91

Figure 29 Comparison of Performance: Machine Learning Models

Additionally, both classes' (0 and 1) precision, recall, and f1-score values were excellent, with scores higher than 90%. *(See Figure 30)*



```
              precision    recall  f1-score   support

           0       0.95      0.93      0.94      2014
           1       0.93      0.95      0.94      2155

    accuracy                           0.94      4169
   macro avg       0.94      0.94      0.94      4169
weighted avg       0.94      0.94      0.94      4169
```

Figure 30 RF Classification Report

Feature Importance

After training our machine learning model, we analysed the importance of different features in predicting whether the website visitor would end up shopping or not. The Feature Importance show how important each feature is to the model's prediction in general.

- Pg_value: This feature is the most crucial in predicting the shop.
- Month: Month is the second most important factor. It is not as significant as the first features.
- Exit rate: This is the third most important feature.
-

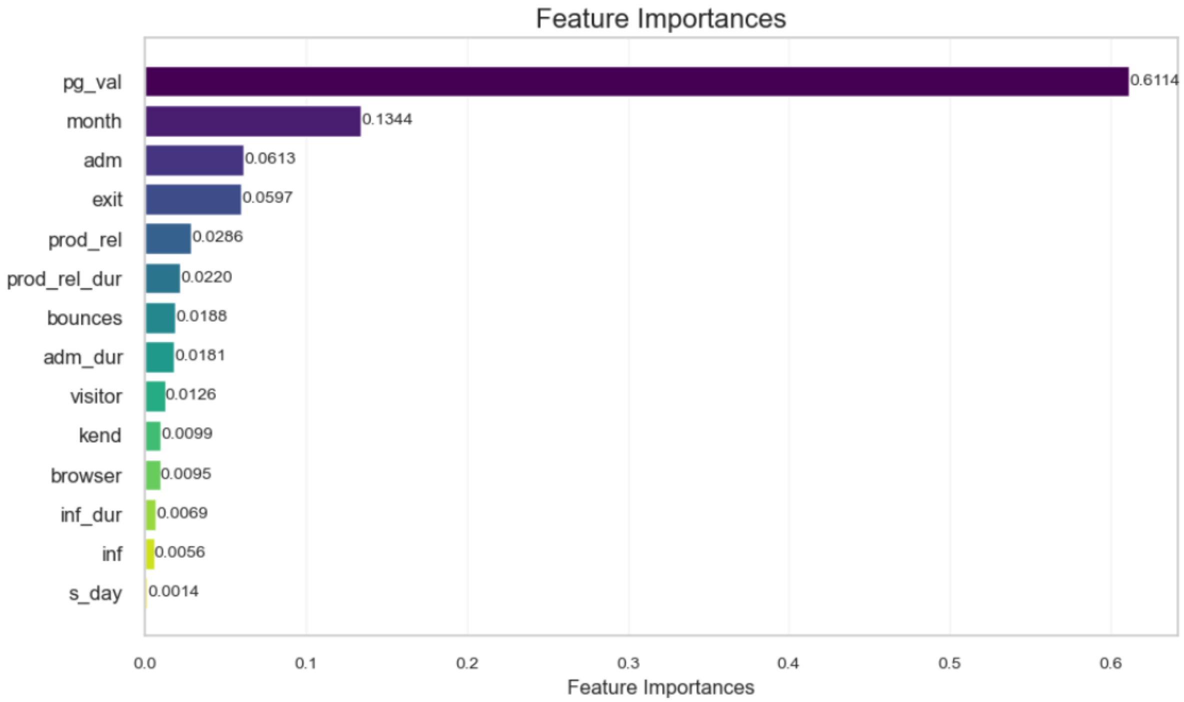


Figure 31 Feature Importances

Shapley Additive exPlanations (SHAP)

According to Awan (2023), It is a method used to explain the output of the machine learning model. It helps to explain how the model gets at its decision for individual prediction.

SHAP take the predictions, removes one variable, and sees how much it impacts the result. Then, SHAP repeats the same process for each one of the variables. By doing that, it is possible to calculate the SHAP values, which shows how each feature impacts its predicted values.

Features that tend to make significant contributions to prediction will have high mean SHAP values (Trevisan, 2022). SHAP also shows the degree of impact, if the feature tends to increase or decrease the prediction.

This project uses a tree model, Random Forest Classifier, and because of that the TreeExplainer is used to calculate the SHAP values due to your ability to handle tree model predictions.

The inspiration for the following plots is drawn from a (Sasaki, 2021) study on Clustering and prediction modelling by PyCaret.

Interpreting Model Decisions Plot

The summary plot brings to the top the most important features for the model's prediction. As we can observe, pg_values, month and exit have the biggest impact on this particular model.

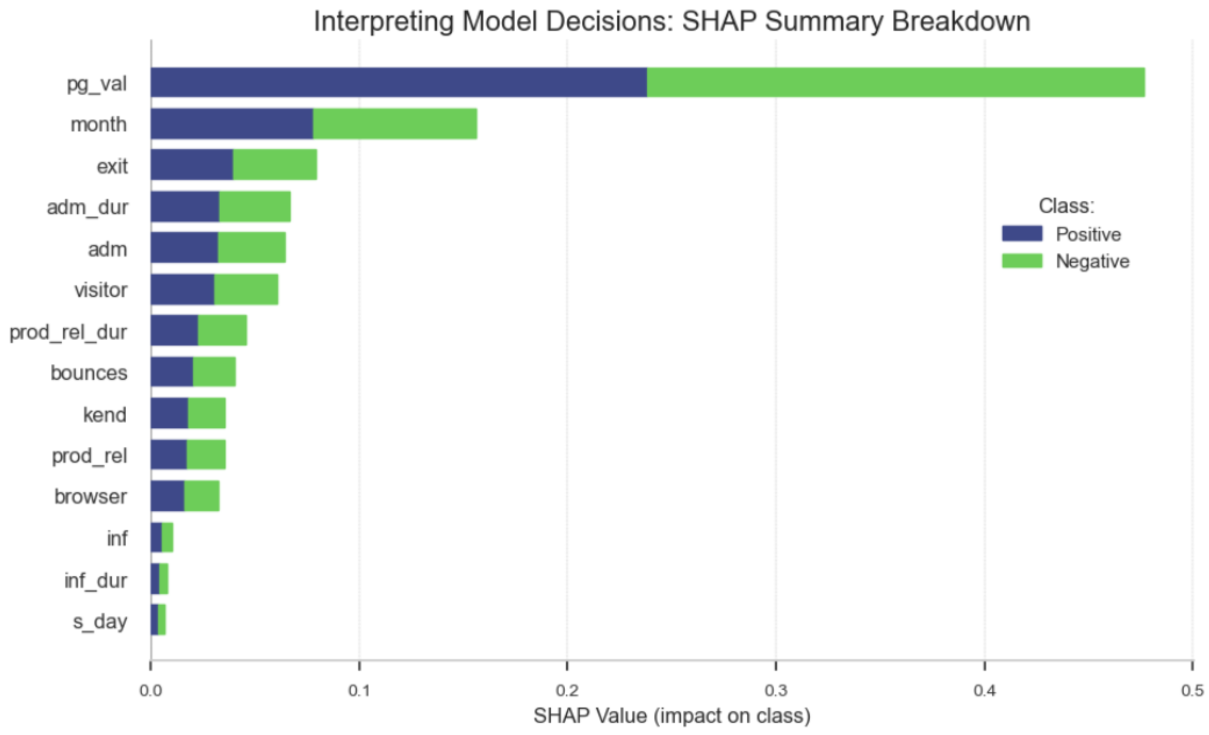


Figure 32 Interpreting Model Decisions Plot

SHAP Summary Plot – Positive Class

According to the researcher Lantos (2021), summary plots sort the features based on their importance in predicting the power of the model. The plot displays pink for high feature values and blue for low feature values. The line in the middle of the plot split Class 0 (Negative) on the left and Class 1 (Positive) on the right side. Going left on the X-axis means that the log odds decrease, and the probability of a positive class decreases as well.

Analysing the 3 features with the highest prediction power, it is possible to observe that:

-As pg_val increases, the probability of the positive class also increases, and as pg_val decreases, the probability of the negative class increases.

-As the month increases, the probability of the positive class also increases, and as the month decreases, the probability of the positive class decreases.

-The opposite is observed for the exit feature. As exit increases, the probability of the negative class increases, and as exit decreases, the probability of the positive class increases. What makes sense, once the exit rates are low, the probability of the site visit ending in shopping is high.

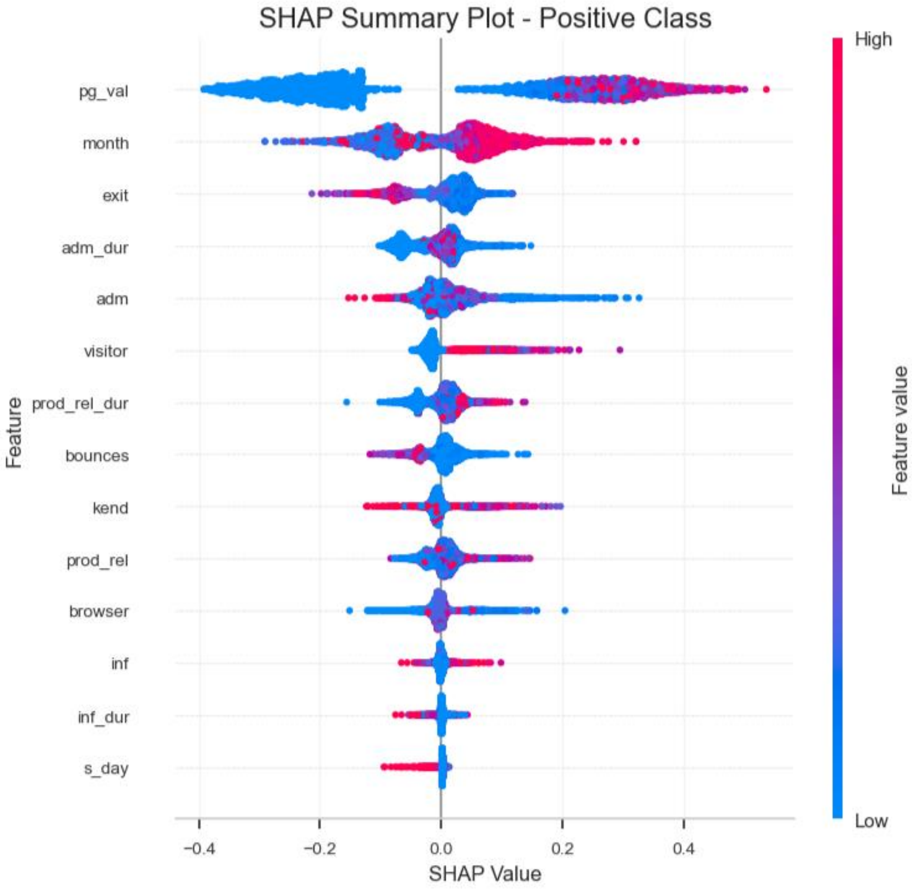


Figure 33 Summary Plot – Positive Class

Force Plot – Positive Class

As said before, SHAP allows us to see how the model arrives at its decision for individual prediction. The index number 8 is being used for demonstration. Analysing the force plot, it is possible to see, for this individual row, how the model predicted the result. The f(x) is 0.75 above the base value of 0.50. It means that this prediction should belong to the positive class. It is confirmed by displaying y_test[8] that as a result shows number 1 (positive class).

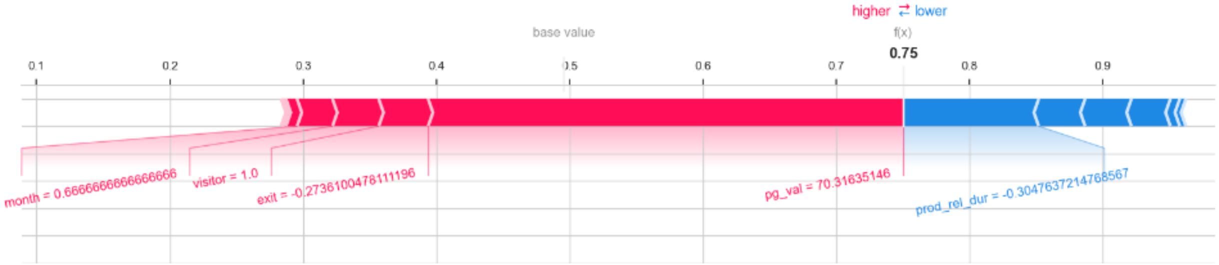


Figure 34 Force Plot – Positive Class - Index 8

SHAP Decision Plot

It makes clear the features contribute the most to the positive prediction for the index number 8. They are pg_val and visitor.

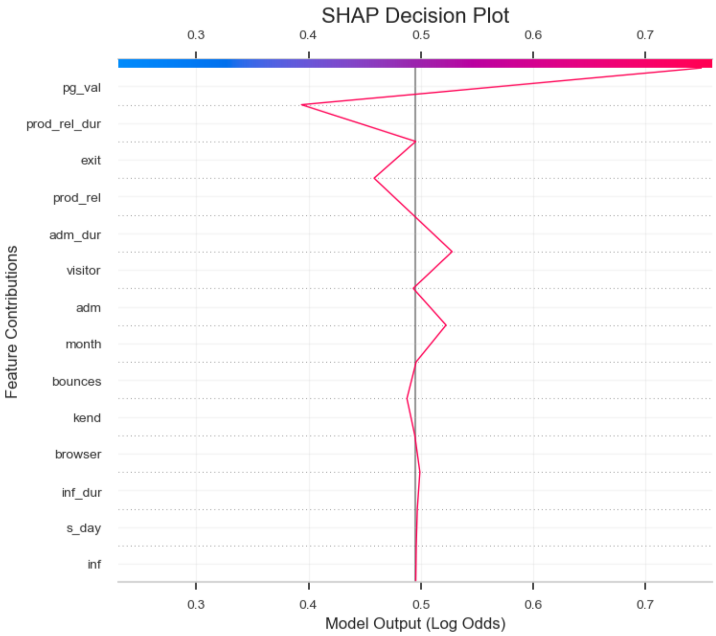


Figure 35 SHAP Decision Plot

Conclusion

- The Random Forest Classifier Model presented the best results when compared with the other two models applied in this paper. The minimum accuracy score is: 0.92 and the maximum accuracy score is: 0.94. The average accuracy score is 0.93 across 10-folds.

- A stratified Cross-validation technique was used to make sure that our samples would have an appropriate amount of both classes.

- Hyperparameter tuning was applied in all models analyzed in this paper. GridSearchCV was used to find the optimal hyperparameters to fit the model. However, the best performance was using default hyperparameters. Further, an analysis must be done to improve the find of optimal hyperparameters, and then, make the model perform even better.

- In this paper, The Cluster Analysis was provided. It brought to light interesting insights regarding the customers' behaviours. Such as “Engaged high revenue visitors” of the website. Visitors with high engagement (Duration sum > 50h), low bounce rate (< 5%), and most of the revenue generated

- Feature Importances were applied to highlight the features that contribute the most to our model performance. Pg_val attribute 61%, month 13%, adm 6% and exit 5%.

- Shapley Additive Explanations (SHAP) helped to explain how the model gets at its decision for individual prediction. SHAP explain that, as pg_val increases, the probability of the positive class also increases, and as pg_val decreases, the probability of the negative class increases. It also explains that as the month increases, the probability of the positive class also increases, and as the month decreases, the probability of the positive class decreases.

- Tests were performed, including the Principal Components Analysis (PCA). However, we decided to continue working without PCA since our model performed better.

- Outliers are important for this analysis performance. They were kept. To handle them, the data was scaled using the Robust Scale technique.

- The SMOTE Technique was used to handle the unbalancing classes.

- The CRISP-DM methodology has been utilized since the beginning of this project. All the steps to deliver this project are documented there.

- We conclude that, effectively, the model is capable of predicting if a user will make a purchase on an e-commerce website given their clickstream and session data.

Reference list

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L. and Ridella, S. (2012). *The ‘K’ in K-fold Cross Validation*. [online] https://www.esann.org/: Esann, pp.441–446. Available at: https://www.esann.org/sites/default/files/proceedings/legacy/es2012-62.pdf.

Awan, A.A. (2023). *Using SHAP Values for Model Interpretability in Machine Learning*. [online] KDnuggets. Available at: https://www.kdnuggets.com/2023/08/shap-values-model-interpretability-machine-learning.html [Accessed 14 Nov. 2023].

Bruce, P.C., Bruce, A. and Gedeck, P. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python*. Sebastopol, Ca: O’reilly Media, Inc.

Daniel, B. (2019). Cross-validation. *Research Gate*, [online] pp.542–545. Available at: https://www.researchgate.net/profile/Daniel-Berrar/publication/324701535_Cross-Validation/links/5cb4209c92851c8d22ec4349/Cross-Validation.pdf [Accessed 1 May 2023].

Hashmi, F. (2020). *Titanic survival prediction case study in Python*. [online] Thinking Neuron. Available at: https://thinkingneuron.com/titanic-survival-prediction-case-study-in-python/ [Accessed 1 May 2023].

Jung, Y. (2017). Multiple predictingK-fold cross-validation for model selection. *Journal of Nonparametric Statistics*, 30(1), pp.197–215. Doi: https://doi.org/10.1080/10485252.2017.1404598.

Kaggle (2021). *Online Shoppers Purchasing Intention Dataset*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/imakash3011/online-shoppers-purchasing-intention-dataset [Accessed 28 Apr. 2023].

Lantos, D. (2021). *How to explain your machine learning model using SHAP?* [online] Advancing Analytics. Available at: https://medium.com/advancing-analytics/how-to-explain-your-machine-learning-model-using-shap-449cf05d5160 [Accessed 14 Nov. 2023].

Mckinney, W. (2018). *Python for data analysis: data wrangling with pandas, NumPy, and IPython*. Sebastopol, Ca: O’reilly Media, Inc., October.

Müller, A.C. and Guido, S. (2017). *Introduction to machine learning with Python: a guide for data scientists*. Beijing: O'reilly.

Probst, P., Wright, M.N. and Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, [online] 9(3). doi:https://doi.org/10.1002/widm.1301.

Sasaki, T. (2021). *Clustering and Predict Modeling by PyCaret*. [online] kaggle.com. Available at: https://www.kaggle.com/code/sasakitetsuya/clustering-and-predict-modeling-by-pycaret [Accessed 14 Nov. 2023].

Sharma, P. (2019). *The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/#How_to_Apply_K-Means_Clustering_Algorithm? [Accessed 1 Nov. 2023].

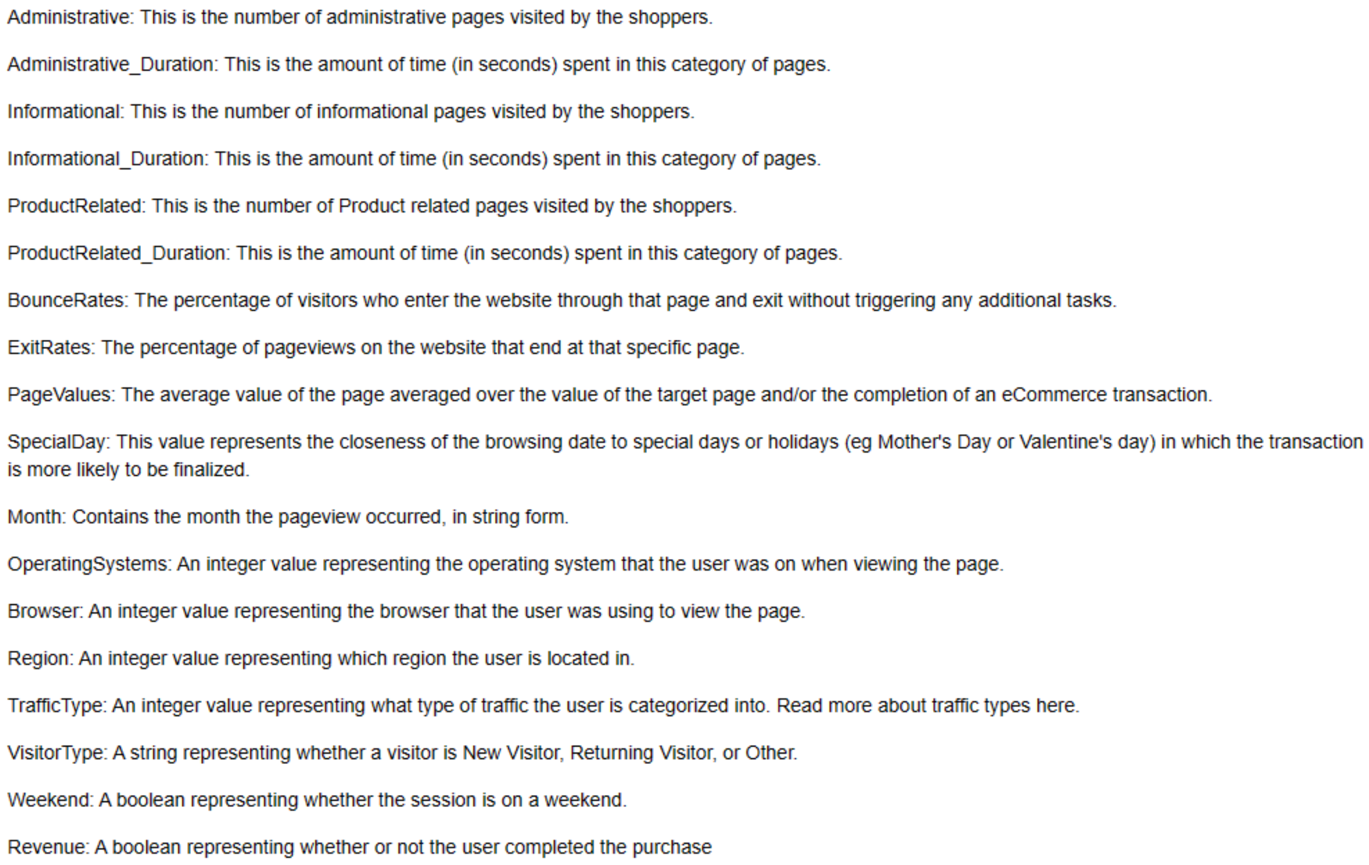
Trevisan, V. (2022). *Using SHAP Values to Explain How Your Machine Learning Model Works*. [online] Medium. Available at: https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137.

Weerts, H.J.P., Mueller, A.C. and Vanschoren, J. (2020). Importance of Tuning Hyperparameters of Machine Learning Algorithms. *arXiv:2007.07588 [cs, stat]*. [online] Available at: https://arxiv.org/abs/2007.07588 [Accessed 5 Nov. 2023].

Wikipedia Contributors (2019). *Cluster Analysis*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Cluster_analysis [Accessed 1 Nov. 2023].

Appendix

Appendix 1: Data Dictionary



Administrative: This is the number of administrative pages visited by the shoppers.

Administrative_Duration: This is the amount of time (in seconds) spent in this category of pages.

Informational: This is the number of informational pages visited by the shoppers.

Informational_Duration: This is the amount of time (in seconds) spent in this category of pages.

ProductRelated: This is the number of Product related pages visited by the shoppers.

ProductRelated_Duration: This is the amount of time (in seconds) spent in this category of pages.

BounceRates: The percentage of visitors who enter the website through that page and exit without triggering any additional tasks.

ExitRates: The percentage of pageviews on the website that end at that specific page.

PageValues: The average value of the page averaged over the value of the target page and/or the completion of an eCommerce transaction.

SpecialDay: This value represents the closeness of the browsing date to special days or holidays (eg Mother's Day or Valentine's day) in which the transaction is more likely to be finalized.

Month: Contains the month the pageview occurred, in string form.

OperatingSystems: An integer value representing the operating system that the user was on when viewing the page.

Browser: An integer value representing the browser that the user was using to view the page.

Region: An integer value representing which region the user is located in.

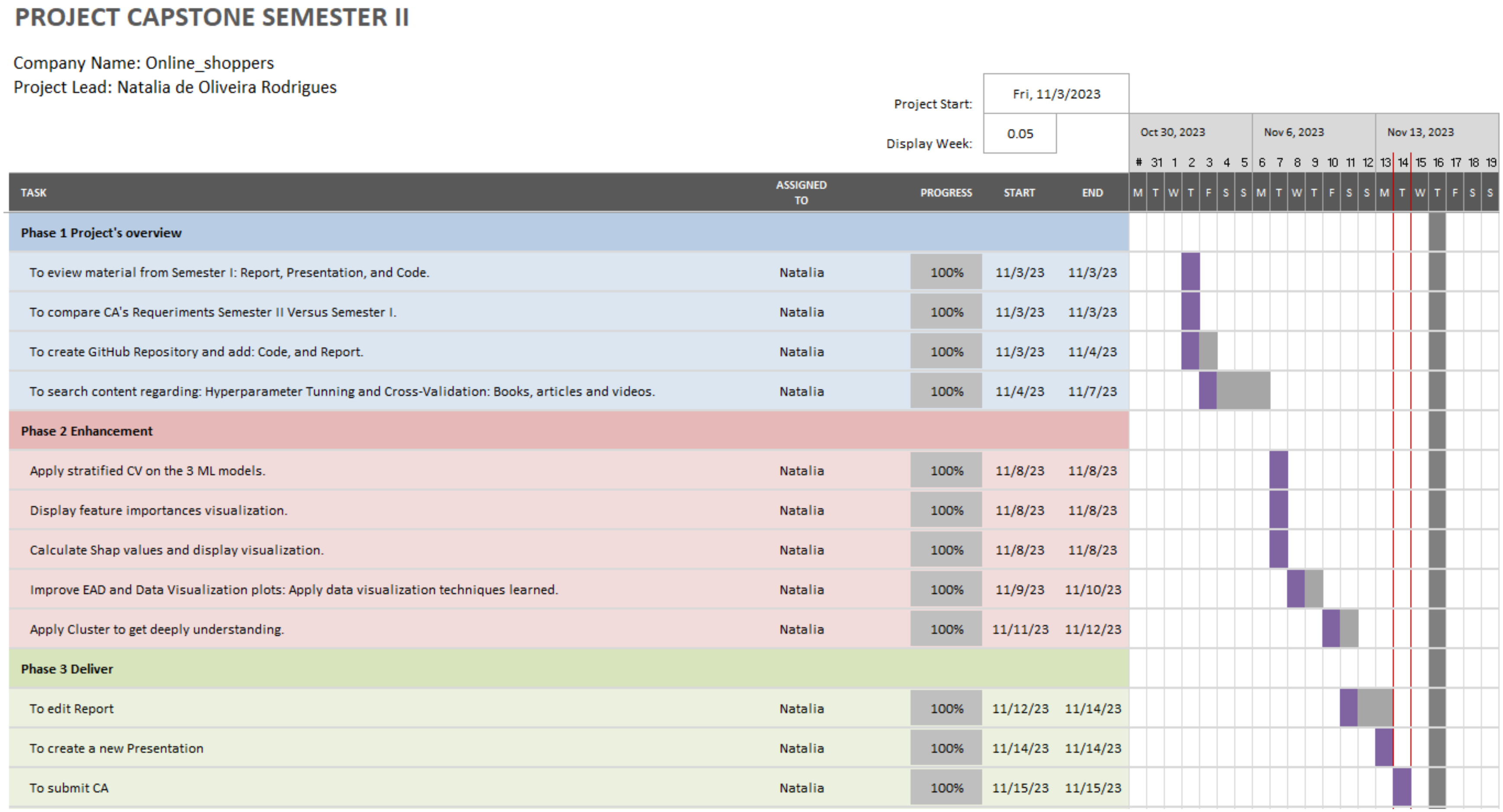
TrafficType: An integer value representing what type of traffic the user is categorized into. Read more about traffic types here.

VisitorType: A string representing whether a visitor is New Visitor, Returning Visitor, or Other.

Weekend: A boolean representing whether the session is on a weekend.

Revenue: A boolean representing whether or not the user completed the purchase

Appendix 2: CRISP-DM



PROJECT CAPSTONE SEMESTER II

Company Name: Online_shoppers
Project Lead: Natalia de Oliveira Rodrigues

Project Start: Fri, 11/3/2023

Display Week: 0.05

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Project's overview				
To eview material from Semester I: Report, Presentation, and Code.	Natalia	100%	11/3/23	11/3/23
To compare CA's Requeriments Semester II Versus Semester I.	Natalia	100%	11/3/23	11/3/23
To create GitHub Repository and add: Code, and Report.	Natalia	100%	11/3/23	11/4/23
To search content regarding: Hyperparameter Tunning and Cross-Validation: Books, articles and videos.	Natalia	100%	11/4/23	11/7/23
Phase 2 Enhancement				
Apply stratified CV on the 3 ML models.	Natalia	100%	11/8/23	11/8/23
Display feature importances visualization.	Natalia	100%	11/8/23	11/8/23
Calculate Shap values and display visualization.	Natalia	100%	11/8/23	11/8/23
Improve EAD and Data Visualization plots: Apply data visualization techniques learned.	Natalia	100%	11/9/23	11/10/23
Apply Cluster to get deeply understanding.	Natalia	100%	11/11/23	11/12/23
Phase 3 Deliver				
To edit Report	Natalia	100%	11/12/23	11/14/23
To create a new Presentation	Natalia	100%	11/14/23	11/14/23
To submit CA	Natalia	100%	11/15/23	11/15/23