# Classification of Online Shoppers Purchasing Intention

Sahil Mathur
Faculty of Engineering and Computing
*Coventry University*
Coventry, UK
mathurs2@uni.coventry.ac.uk

Sukhmani Saini
Faculty of Engineering and Computing
*Coventry University*
Coventry, UK
sainis9@uni.coventry.ac.uk

**Abstract— *The primary objective of this paper is to analyse the intention/behaviour of online shoppers using machine learning techniques on the given dataset. Utilising Classification methods including K-nearest Neighbour, Naïve Bayes, Decision Trees and Random Forest , a comparison is drawn from the results and a conclusion is formulated on the most suitable method. Python programming language was used to implement techniques and algorithms utilizing machine learning repositories, prediction results and algorithm performance measures were obtained, and visualized for comparison and discussion.***

***Keywords— machine learning; Classification; python; categorical data; decision tree; k-nearest neighbour; random forest, k- nearest.***

***Key: ML = Machine Learning, KNN = k-Nearest Neighbor, Random Forest = RF, PLT = Matplotlib***

## I. INTRODUCTION

With the boom in internet access across the world, the normal retail shopping has been rapidly shifting to E-commerce or online shopping and as a result, the shopping dynamics have been continuously changing and has already become a major part of the retail market. A typical customer prefers shopping online in their own comfort as compared to travelling to a physical shop and buying the products. Resulting in Businesses to start implementing various online marketing techniques to lure customers to their shopping portal. Customers who visit these web portals might not make any purchases at all. This could be for a variety of reasons, like high product pricing or window shopping. There have been multiple studies (Santini 2018) that imply that retention strategies such as an appropriate recommendation system play a critical role in converting sales. For example, if the ML solution predicts a strong customer purchase intention, the recommend system may suggest a higher quality or more expensive product because it can be inferred that the user is ready to consider a better or more expensive product if their intent to buy a particular item is very strong. If the solution forecasts a lower intention to buy, the recommendation system may suggest discounted products or products with special deals. Cambridge Analytica used similar strategies, although on a larger scale, to influence voter decisions in American elections (The Guardian, 2018). This demonstrates the strength of a proper machine learning technique if applied effectively could vary a majority population's intent.

## II. OBJECTIVE

This project intends to make advantage of information that customers may leave behind in the form of browsing history data or user information when they browse an ecommerce platform. Utilizing clickstream and activity data information, the study seeks to forecast online buyers' purchasing intentions using this information. The research intends to build a machine learning model based on this data to forecast client purchasing intentions. The project's goal is to create a Machine Learning model that can anticipate customer purchase intent as accurate as possible.

## III. DATASET

The dataset used in the project was obtained from UCI machine learning. The dataset has been provided by Authors Sakar and Y. Kastro (Sakar et al., 2018).

The data has 12330 instances, each with 18 characteristics and no missing data. Each instance represents a single person's visit to the site. When a repeat consumer visits the site several times, only the initial visit from the time frame examined is taken into account, with subsequent visits being excluded. The first 17 attributes are visitor-related features that include 10 numerical values and 7 categorical elements. The last 18th attribute that is 'Revenue' reflects the category if the visit resulted in revenue or not, which means that if the visitor made a purchase, it is labelled as True or False.

The categorical and numerical features used in the prediction model are shown in Tables 1 and 2, respectively.

*Table 1. Categorical Features*

| Attribute Name | Description | Number of values |
|---|---|---|
| OperatingSystems | Visitor's Operating System | 8 |
| Browser | Visitor's Browser type | 19 |
| Region | Geographical Region of the visitor | 9 |
| TrafficType | Source of redirect by visitor | 20 |
| VisitorType | Type of Visitor ("New", "Returning", "Other") | 3 |
| Weekend | Indicating if the day of visit is a weekend or not by True/False | 2 |
| Month | Month of visit | 10 |
| Revenue | Boolean to indicate if revenue was generated | 2 |

*Table 2. Numerical Values*

| Attribute Name | Description | Number of values |
|---|---|---|
| Administrative | Number of pages visited by the visitor | 12330 |
| Administrative_Duration | Total amount of time (in seconds) spent by the visitor | 12330 |
| Informational | Number of pages visited by the visitor about Web site, communication and address information of the shopping site | 12330 |
| Informational_Duration | Total amount of time (in seconds) spent by the visitor on informational pages | 12330 |
| ProductRelated | Number of pages visited by visitor about product related pages | 12330 |
| ProductRelated_Duration | Total amount of time (in seconds) spent by the visitor on product related pages | 12330 |
| BounceRates | Average bounce rate value of the pages visited by the visitor | 12330 |
| ExitRates | Average exit rate value of the pages visited by the visitor | 12330 |
| PageValues | Average page value of the pages visited by the visitor | 12330 |
| SpecialDay | - | 12330 |

## IV. Data Cleaning

The dataset is available in a good condition with no null values and some minor issues which we decided not to handle beforehand.

## V. Exploratory Data Analysis (EDA)

In this section we have thoroughly explored and analysed the data to summarize the main characteristics. We have divided this section into three parts: 1. Univariate Analysis – analyses of every column in the dataset, we managed to extract some useful information about each column such as the skewness, data distribution etc., 2. Bi-variate analysis – In this section we have compared each variable or column to the target variable ('Revenue') and managed to extract some useful information such as the distribution, outliers etc., 3. Multi-variate analysis – in this section we have tried to encompass the simultaneous observation and analysis of more than one outcome variable. The dataset has both numerical and categorical variables. The "BounceRates," "ExitRates," and "PageValues" features, among others, depict the metrics for every session. In this section Our data analysis yielded the following major findings:
The sample is imbalanced, with only 15% of visits resulting in a purchase. There is huge number of outliers in the data.
PageValues may be the most critical feature in predicting purchase conversion.

Data Imbalance –

This dataset is skewed, with only 15% of visits resulting in a purchase. While 15% is not a severely imbalanced dataset, we can still investigate strategies or algorithms which are more effective at dealing with it.

PageValue Importance –

PageValues is described as the average page value of the user's pages visited. Values are typically allocated to essential pages in an e-commerce setting, such as checkout pages or pages following the checkouts.

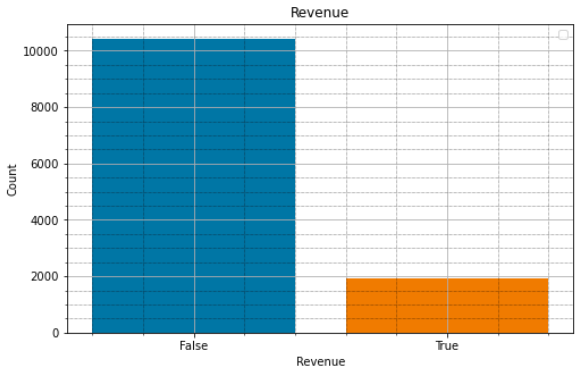


*Figure 1. Distribution of Revenue generation*

As illustrated in Fig 2, a PageValues greater than 5 enhances the likelihood of sales conversion. As a result, the PageValues feature gives an excellent indication of whether the visitor will buy something.

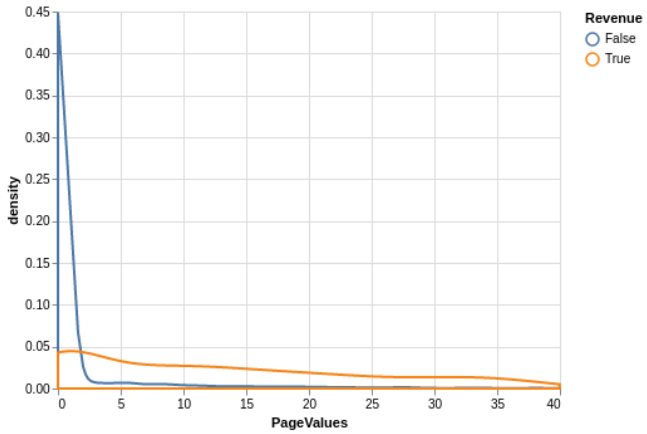


*Figure 2. PageValues Density Plot*

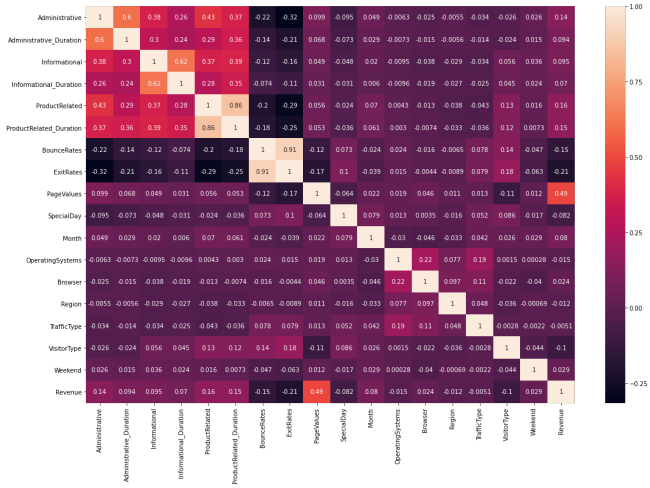

*Figure 3. Heatmap of the dataset*

From Fig 3, We can depict that Administrative and Administrative_Duration are correlated. ExitRates, Information , ProductRelated and BounceRates show similar properties. Page Value appears to have a stronger relationship with Revenue.

Right-Skewness of the data –

Finally, we discovered that most of numerical features have right-skewed ends. This is typical in e-commerce environments, where some individuals have a disproportionately high utilisation rate. We can test whether removing outliers or using feature transformations like Box-Cox can improve model performance.

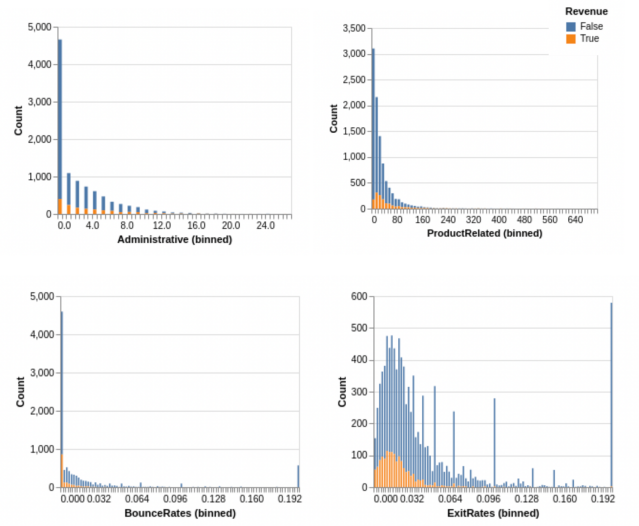


*Figure 4. Right Skewness of the data*

## VI. STATISTICAL TESTS

In this section we have performed various statistical tests to analyse if the variable is significant or not. This section is divided into two sections :

1. Categorical column vs target column

For comparing categorical columns with the target column we are performing chi-squared test. A chi-squared test is a hypothesis tests which is used to compare the observed results with expected result, the main purpose of this test is to determine if a difference between observed data and expected data is due to chance or if it is due to a relationship between the variables that we are studying. Basically, we will check if our target column 'Revenue' is getting influenced by any categorical column or not. Our null hypothesis (Ho) is that the proportion of revenue across the category is same and our alternative hypothesis (Ha) is proportion of revenue at least in two categories is different. We reject the null hypothesis if p-value < 0.05. Here are the observations:

*Table 3. P-value of Categorical Features*

| | Feature | P-Value | Significance |
|---|---|---|---|
| 0 | Administrative | 0.0000 | * |
| 1 | Informational | 0.0000 | * |
| 2 | ProductRelated | 0.0000 | * |
| 3 | Month | 0.0000 | * |
| 4 | OperatingSystems | 0.0000 | * |
| 5 | Browser | 0.0061 | * |
| 6 | Region | 0.3214 | ** |
| 7 | TrafficType | 0.0000 | * |
| 8 | VisitorType | 0.0000 | * |
| 9 | Weekend | 0.0013 | * |
| 10 | SpecialDay | 0.0000 | * |

From table 3 we can conclude that only one column i.e., Region is not significant, therefore, we can say that the proportion of revenue across the categories are not the same.

2. Numerical column vs Target column

For comparing numerical columns with target columns we will be performing Shapiro-wilk test. It is a statistical test that is used to check if the continuous variables/columns follows the normal distribution or not. Here, our null hypothesis (Ho) is that data is normally distributed and alternative hypothesis (Ha) is that the data is not normally distributed. We reject the null hypothesis if p-value < 0.05. Here are the observations:

*Table 4. P-value of Numerical features*

| | Feature | P-Value | Significance |
|---|---|---|---|
| 0 | Administrative_Duration | 0.0 | * |
| 1 | Informational_Duration | 0.0 | * |
| 2 | ProductRelated_Duration | 0.0 | * |
| 3 | BounceRates | 0.0 | * |
| 4 | ExitRates | 0.0 | * |
| 5 | PageValues | 0.0 | * |

We can conclude that our data is not normally distributed.

Before moving ahead and transforming our data we are also going to check the outliers. Outliers are the data points that significantly differs from other observations, in other word the data that lies outside the other values in the set.

*Table 5. Outliers in the dataset*

| | Feature | % Of Outliers |
|---|---|---|
| 4 | ExitRates | 5.782644 |
| 3 | BounceRates | 5.742092 |
| 5 | PageValues | 2.100568 |
| 0 | Administrative_Duration | 1.881590 |
| 1 | Informational_Duration | 1.865369 |
| 2 | ProductRelated_Duration | 1.776156 |

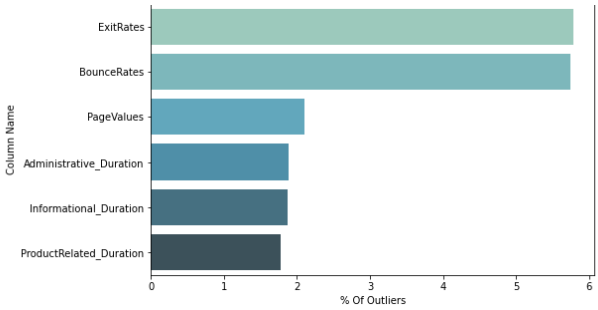


*Figure 5. Outlier percentage*

As per our above observation we can see that most pf the outliers are in the column 'ExitRates' and 'BounceRates', therefore, we are going to handle it manually.

We are going to transform our data using a method called Box-Cox. The Box-Cox method helps to address the non-normally distributes data by transforming to normalize the data.

Distribution before transformation:

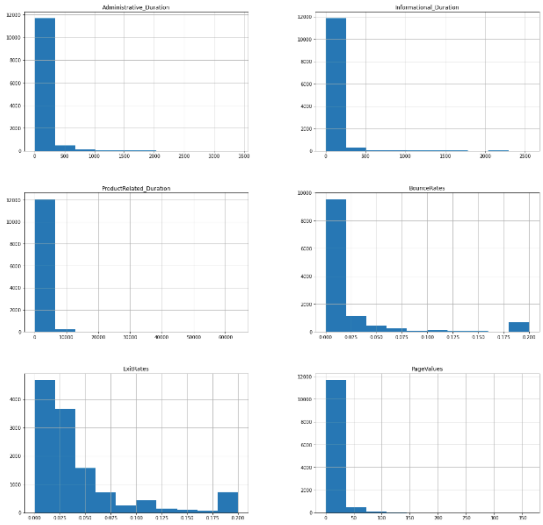

*Figure 6. Data Distribution (Pre-Transformation)*

Distribution after transformation:

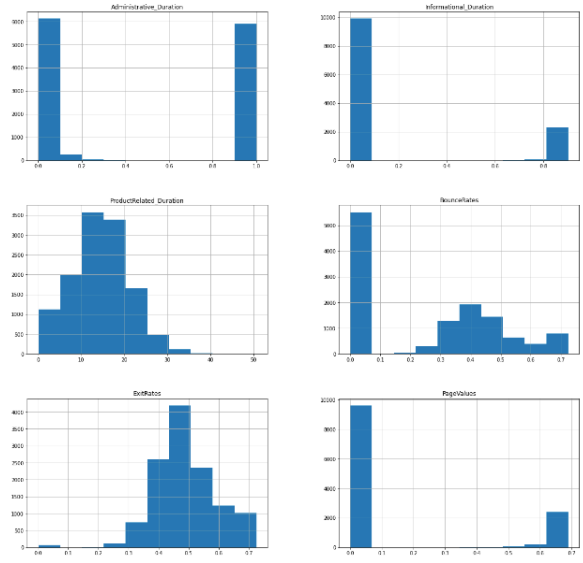

*Figure 7. Data Distribution (Post-Transformation)*

So we can conclude, there are significant changes in the distribution of the data.

## VII. DATA PRE-PROCESSING

Data pre-processing is manipulation of the data before feeding into any model to enhance the performance. This includes cleaning and transforming the data. As per our previous observations there are no null values in our data, therefore, we will go ahead and transform the necessary column. In our dataset we have 10 numerical columns and 8 categorical. We will convert our categorical columns into numerical because machine learning models requires all input variables to be numeric. We are going to use LabelEncoder to convert it, LabelEncoder replaces the categorical value with a numeric value between 0 and number of classes-1.

## VIII. FEATURE ENGINEERING

Feature engineering also known as feature extraction is the process of transforming the raw data into features that better represents the underlying problem to the predictive models. Firstly, we are going to scale our raw data. Scaling is an important method that is mainly performed in order to standardize the functionality of the input dataset, usually it fit within a specific scale like 0-100 or 0-1. We are going to apply StandardScalar on our data because our dataset differ greatly between the ranges. It removes the mean and scales the data to the unit variance. The basic idea behind it is that the variables that are measured at different scales do not contribute equally to the fit of the model and the learning function of the model and could end up creating bias, thus, it's necessary to scale it before integrating it into the machine learning model.

## IX. DATA MODELLING

We have divided our scaled data into train and test. Now we are going to feed it into our machine learning models but before that there's an important point of observation. Our target variable 'Revenue' is highly imbalance as per the figure below.

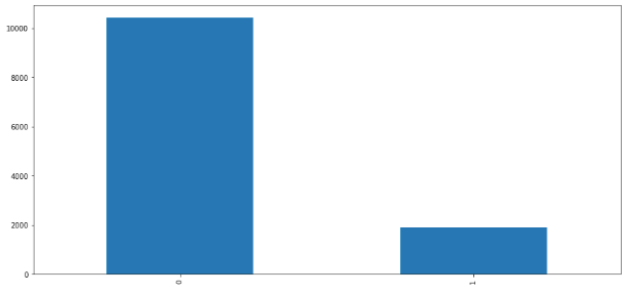


*Figure 8. Data modelling (Revenue)*

Therefore, we are going to use a technique called SMOTE (Synthetic Minority Oversampling Technique) to solve this problem. SMOTE is basically an improved alternative for oversampling that performs data augmentation by creating synthetic data points based on the original data points. In this section we are going to feed our scaled data, firstly without

using SMOTE and then after using SMOTE and we will compare the accuracy.

1. Without SMOTE

*A.* Logistic Regression

"Logistic Regression models the probabilities for classification problems with two possible outcomes." ("Logistic Regression.docx - Logistic Regression Logistic...") Basically, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1. ("Logistic Regression: Theory. Introduction | by Dhruv Khanna - Medium")

```
Train score: 0.8693083072645117
Test score: 0.8680724520140578
```

*Figure 9. Accuracy for Logistic regression*

```
F1 Score: 0.3527851458885942
Kappa Score: 0.3005713648047649
Classification report:
              precision    recall  f1-score   support

           0       0.88      0.98      0.93      3127
           1       0.73      0.23      0.35       572

    accuracy                           0.87      3699
   macro avg       0.80      0.61      0.64      3699
weighted avg       0.85      0.87      0.84      3699
```

*Figure 10. Metrics for Logistic Regression*

We have got a train score of 86% and a test score of 86% as well but the F1-score is only 35% and Kappa score is 30% which is not very good.

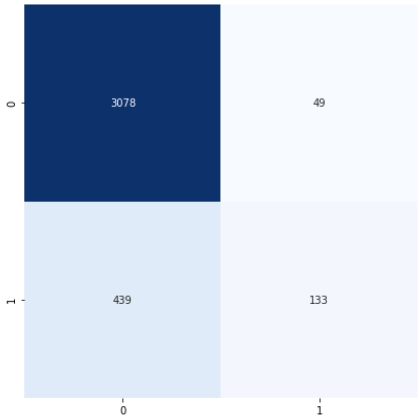


*Figure 11. Confusion matrix*

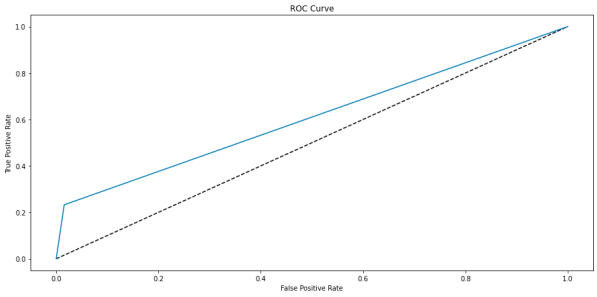


*Figure 12. ROC Curve*

The Confusion Matrix and ROC curve indicates the moderate performance of our algorithm.

*B. Decision Tree*

Decision Trees are a non-parametric supervised learning method used for classification and regression. This creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
Below are the outcomes of decision tree.

```
Train score: 0.9139149577105782
Test score: 0.8988915923222492
```

*Figure 13. Accuracy for Decision Tree*

```
F1 Score: 0.6382978723404256
Kappa Score: 0.5803015437598862
Classification report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94      3127
           1       0.71      0.58      0.64       572

    accuracy                           0.90      3699
   macro avg       0.82      0.77      0.79      3699
weighted avg       0.89      0.90      0.89      3699
```

*Figure 14. Metrics for Decision Tree*

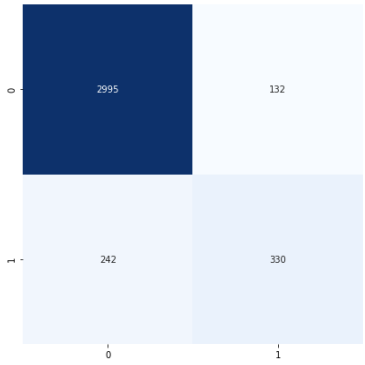


*Figure 15. Confusion Matrix*

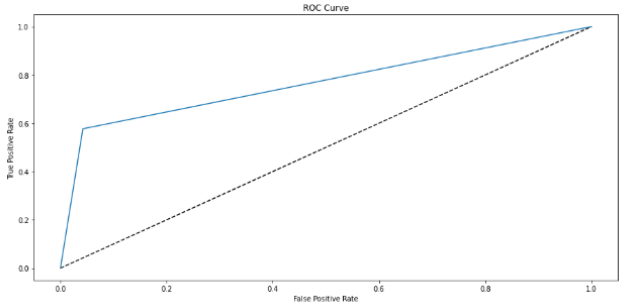


*Figure 16. ROC Curve*

*C. Random Forest*

Random forest is a classification algorithm consisting on various decision trees. It is great with high dimensional data and faster to train than decision tree since it only works on subsets of data.
Below are the outcomes of Random Forest

```
Train score: 0.9142625419997683
Test score: 0.8978102189781022
```

*Figure 17. Accuracy for Random Forest*

```
F1 Score: 0.5970149253731343
Kappa Score: 0.5417105053560226
Classification report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      3127
           1       0.77      0.49      0.60       572

    accuracy                           0.90      3699
   macro avg       0.84      0.73      0.77      3699
weighted avg       0.89      0.90      0.89      3699
```

*Figure 18. Metrics for Random Forest*

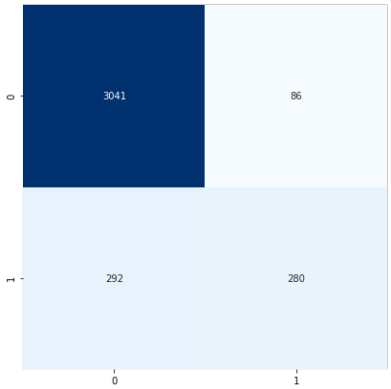


*Figure 19. Confusion Matrix*

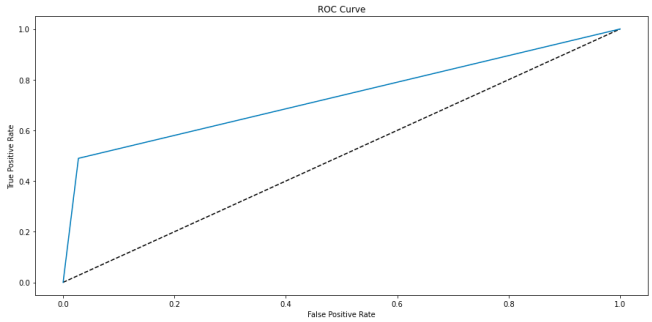


*Figure 20. ROC Curve*

### *D. Naïve Bayes*

The Naïve Bayes classification algorithm is a probabilistic classifier that is based on the Bayes Theorem strong independence assumptions. It is a very fast and simple classification algorithms suitable for a very high-dimensional datasets.
Below are the outcomes of the algorithm:

```
Train score: 0.8367512455103696
Test score: 0.8432008650986753
```

*Figure 21. Accuracy for Naive Bayes*

```
F1 Score: 0.5970149253731343
Kappa Score: 0.5417105053560226
Classification report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      3127
           1       0.77      0.49      0.60       572

    accuracy                           0.90      3699
   macro avg       0.84      0.73      0.77      3699
weighted avg       0.89      0.90      0.89      3699
```

*Figure 22. Metrics for Naive Bayes*

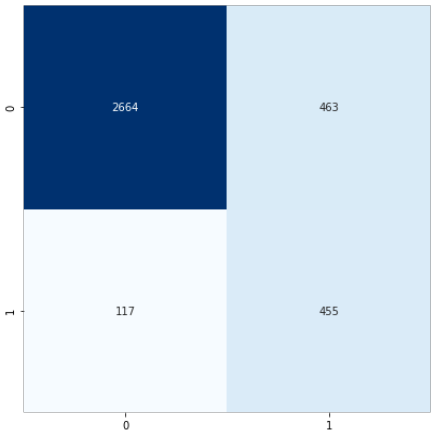


*Figure 23. Confusion Matrix*

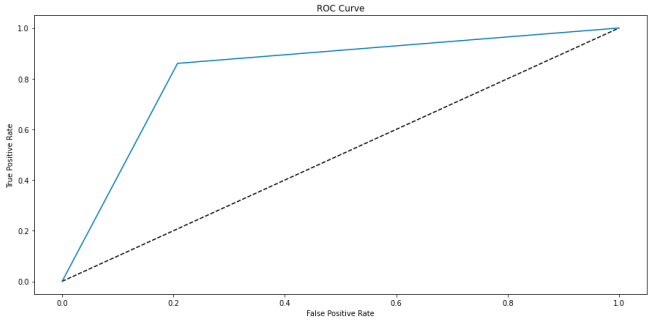


*Figure 24. ROC Curve*

2. With SMOTE

Developing predictive Machine learning models on classification datasets with severe class imbalance is what imbalanced classification entails.
The issue of working with imbalanced datasets is that most ML algorithms will disregard, and so perform poorly on, the minority class, despite the fact that performance on the minority class is generally the most essential.

Oversampling the minority class is one method for dealing with imbalanced datasets. The most basic method involves duplicating examples from the minority class, even though these examples add no new information to the model. Instead, new examples can be created by combining existing ones. The SMOTE(Synthetic Minority Oversampling Technique), is a type of data augmentation for the minority class.
Let's apply SMOTE on our dataset here.

```
0    7295
1    1336
Name: Revenue, dtype: int64
```

*Figure 25. Train datasets count pre-SMOTE*

```
0    7295
1    7295
Name: Revenue, dtype: int64
```

*Figure 26. Train datasets count post-SMOTE*

A. Logistic Regression –

Accuracy of the model -
Train score: 0.8603740330950749
Test score: 0.8578935343842815

Metrics of the Model -
F1 Score: 0.8551467163483931
Kappa Score: 0.7157997522206814

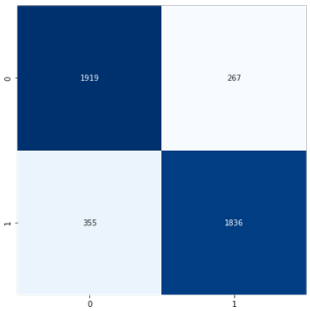


*Figure 27. Confusion Matrix*

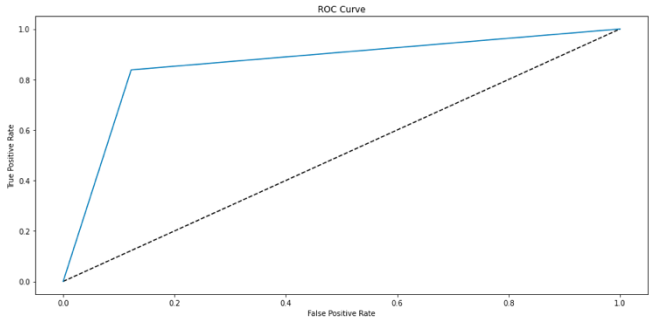


*Figure 28.ROC Curve*

B. Naïve Bayes –

Accuracy of the model -

```
Train score: 0.8272789581905414
Test score: 0.8265935572309802
```

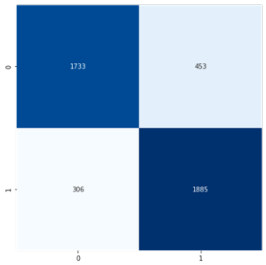


*Figure 29. Confusion Matrix*

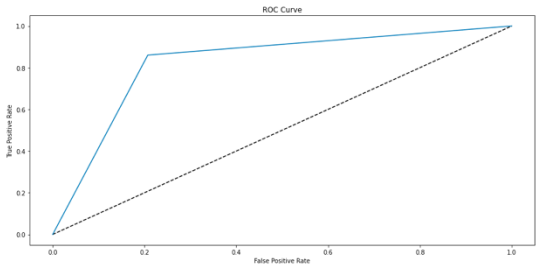


*Figure 30. ROC Curve*

After creating a model using different classification algorithms Post SMOTE, we got following result –

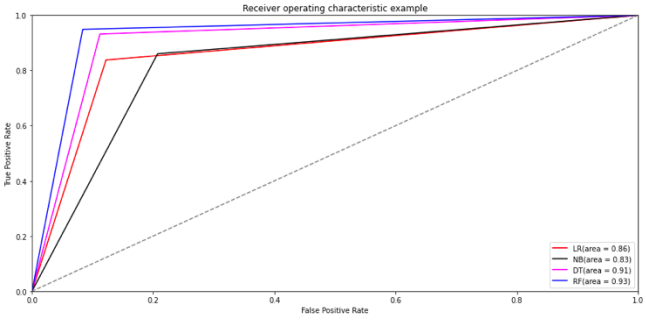


*Figure 31. Combined ROC curve of all algorithms*

From Fig 31. We have the red line for Logistic Regression , Black line represents Naïve Bayes , Pink represents Decision tree, and the Blue is for Random Forest algorithm.
Random Forest is best among all as it gives 98% area under, F1 score as 0.90
Transformation improved accuracies for Logistic and Naïve Bayes but it hardly improved Random Forest and Decision Tree. So, we go ahead without transformations as it hardly improves any accuracy.
KNN is a classification technique but due to less computational power we were able to use it. Also, KNN takes more time to
Random Forest was used for feature selection and the important features given by Random Forest are supported by the EDA analysis done.

## X. RESULTS AND CONCLUSION

As per our above observation through accuracy metrics and ROC, we have managed to conclude that Random Forest is giving us the best result, it's 93% area under the curve and F1-score as 0.90. Although, transformation improved certain accuracies for logistic regression and Naïve Bayes, but Random Forest and Decision tree were almost the same.

We can suggest some business insights:

a. To maximise sales, we advise offering a discount three to four days before the special day.

b. To make some money, Jan. & April. need special attention. Although February contains special days, overall revenue is quite modest, thus it is important to take use of them.

c. Since there are users of non-mainstream browsers, advertising on those browsers will help to reach new consumers.

d. Region - Because we have a wide geographic reach, a large market will aid in creating income in low-producing areas.

e. We must create strategies to capitalise on the new tourists' spending power.

f. Due of the extremely low weekend revenue, special weekend offers must be promoted.

g. Since we currently have a high exit rate in the months of July, June, May, and August, lowering the bounce rate during those months could boost revenue.

h. Special planning must be created for additional channels in addition to the typical traffic generators.

i. Many pages have low page values, hence we propose improving the product search algorithm to raise page values.

References


[1] Henry Sue, H. (n.d.). *Henry's data science page - Online shopper's classification project*. https://henrysue.dev/shoppers/shoppers.html

[2] The Guardian. (2018, May 7). *Cambridge Analytica: How did it turn clicks into votes?* the Guardian. https://www.theguardian.com/news/2018/may/06/cambridge-analytica-how-turn-clicks-into-votes-christopher-wylie

[3] Santini, R. (2018). Recommender systems as "tastemakers": Collaborative filtering as a market strategy for online cultural products. Observatorio (OBS*). https://doi.org/10.15847/obsobs1222018847

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

[5] *UCI machine learning repository*. (n.d.). https://archive-beta.ics.uci.edu/ml/datasets/online+shoppers+purchasing+intention+dataset#Descriptive

[6] Nico Van den Hooff, Ting Zhe (TZ) Yan, & Arijeet Chatterjee. (n.d.). *Online shoppers purchasing intention*. UBC MDS. https://ubc-mds.github.io/online-shoppers-purchasing-intention/data_analysis.html

[7] *SMOTE for Imbalanced classification with Python*. (2021, March 16). Machine Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[8] *Classification: ROC curve and AUC*. (n.d.). Google Developers. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[9] S. B. Kotsiantis, & P. E. Pintelas. (2007, November 10). *Machine learning: A review of classification and combining techniques*. SpringerLink. https://link.springer.com/article/10.1007/s10462-007-9052-3

[10] Gandhi, R. (2018, May 17). *Naive Bayes classifier*. Medium. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[11] Onesmus Mbaabu. (n.d.). *Introduction to random forest in machine learning*. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

[12] Maran Chandrasekaran. (n.d.). *Logistic regression for machine learning*. Capital One. https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/


## XI. Appendix –

Source Code and the dataset have been uploaded on OneDrive - https://livecoventryac-my.sharepoint.com/:f:/g/personal/mathurs2_uni_coventry_ac_uk/Eo_om7Y-lJpBt8xpfc2s5qABdG1P5hqkTOy0kqkGQsSCUg?e=kO2q1C