

# Capstone Project Report

Analysis and prediction of online shoppers' purchasing intention using various algorithms



Submitted towards partial fulfillment of the criteria for award of Post Graduate Program in Business Analytics and Business Intelligence by Great Lakes Institute of Management

Submitted to



Name	Roll Number
Sahil Sachdeva	BABIGAPR1937
Amitesh Bajpai	BABIGAPR1921
Kushal Maheshwari	BABIGAPR1903
Amit Sharma	BABIGAPR1902
Neeraj Singh	BABIGAPR1922

PGP-BABI Gurgaon Apr19  
Project Mentor: Mr. Suvajit Mukhopadhyay  
Date of Completion – May 2020

## **CERTIFICATE OF COMPLETION**

I hereby certify that the project titled “**Analysis and prediction of online shoppers’ purchasing intention using various algorithms**” was undertaken and completed under my supervision by Sahil Sachdeva, Amitesh Bajpai, Kushal Maheshwari, Amit Sharma and Neeraj Singh students of the Postgraduate Program in Business Analytics & Business Intelligence (PGPBABI Gurgaon Apr19).

Date: 03<sup>rd</sup> May 2020  
Place: Gurgaon

Mr. Suvajit Mukhopadhyay  
Mentor

## ABSTRACT

In this project, we aim to construct a real-time prediction machine learning system for online shopping environment. We use an online retailer data to perform the experiments. In order to predict the purchasing intention of the visitor, we use aggregated page view data kept track during the visit along with some session and user information as input to machine learning algorithms. Oversampling/Undersampling and feature selection pre-processing techniques are applied to improve the success rates and scalability of the models.

Our findings support the argument that the features extracted from clickstream data during the visit convey important information for online purchasing intention prediction. The findings show that choosing a minimal subset of combination of clickstream data aggregated statistics and session information results in a more accurate and scalable system.

Keeping last mile goal of having predictive model in mind, we used different data cleansing, data balancing imputation methods to arrive at dataset fit for development of predictive models. The project includes creation of various analytical models for predicting and classifying users as Revenue or Non-revenue generating using Logistic Regression, Ensemble methods, Bagging, Boosting techniques etc. The base models are benchmarked against Hyperparameter tuned model outputs and considering all factors and final goal in mind best model is advised.

Techniques: Predictive Modelling, Machine learning, Hyperparameter tuning

Tools: Python, Tableau

Domain: Marketing and Retail Analytics

<b>ABSTRACT</b> .....	3
<b>1. EXECUTIVE SUMMARY</b> .....	8
<b>2. PROBLEM STATEMENT, SCOPE AND OBJECTIVE</b> .....	9
2.1 PROBLEM STATEMENT .....	9
2.2 OBJECTIVE.....	9
2.3 SCOPE.....	9
2.4 RESEARCH OBJECTIVES.....	9
<b>3. DATA SOURCE AND DESCRIPTION</b> .....	10
3.1 SCREEN-SHOT OF DATA SET .....	10
3.2 DATA DESCRIPTION.....	10
3.3 DATA OVERVIEW .....	11
<b>4. LITERATURE REVIEWS</b> .....	14
<b>5. DATA PREPROCESSING</b> .....	16
5.1 MISSING VALUE IMPUTATION .....	16
5.2 NEGATIVE VALUE TREATMENT.....	18
5.3 OUTLIERS TREATMENT.....	19
5.4 HANDLING CATEGORICAL VARIABLES .....	21
<b>6. EXPLORATORY DATA ANALYSIS</b> .....	22
6.1 UNIVARIATE/BIVARIATE ANALYSIS .....	23
6.1.1 Page Visits vs Revenue.....	23
6.1.2 Share of web page visits w.r.t Revenue .....	24
6.1.3 Duration spent on pages Vs Revenue .....	25
6.1.4 Bounce Rates and Exit Rates vs Revenue.....	25
6.1.5 Exploring Google Analytics Metrics .....	26
6.1.6 Special Day vs Revenue .....	26
6.1.7 Revenue Vs Visitor Type.....	27
6.1.8 Weekend vs Revenue .....	27
6.1.9 Month vs Revenue.....	28
6.1.10 Region vs Revenue.....	28
6.1.11 Traffic Type vs Revenue.....	29
6.1.12 Operating System vs Revenue.....	29
6.1.13 Browser vs Revenue .....	30
6.1.14 Traffic Type coming on website.....	30
6.1.15 Effect of Special Day on website .....	31
6.1.16 Monthly visits to website by Users.....	31
6.1.17 Share of Visitor Types w.r.t Revenue.....	32
6.1.18 Share of completed Purchases.....	32
6.1.19 Month Vs Special Days .....	33
6.2 MULTIVARIATE ANALYSIS .....	34
6.2.1 Multivariate Analysis -1 .....	34

6.2.2 Multivariate Analysis -2 .....	35
6.2.3 Multivariate Analysis -3 .....	36
6.3 CORRELATION MATRIX .....	37
<b>7. FEATURE SELECTION .....</b>	<b>38</b>
7.1 PRINCIPAL COMPONENT ANALYSIS CHECK .....	41
<b>8. MODELLING APPROACH .....</b>	<b>42</b>
8.1 TRAIN TEST SPLIT .....	43
8.2 HANDLING CLASS IMBALANCE.....	44
8.3 MODELS ON BASE DATA AND COMPARISION .....	45
8.4 MODELS WITH FEATURE SELECTION, UPSAMPLING and HYPERPARAMETER TUNING AND COMPARISION .....	46
8.5 MODELS WITH FEATURE SELECTION, SMOTE AND HYPERPARAMETER TUNING AND COMPARISION .....	47
8.6 MODELS WITH FEATURE SELECTION, SMOTEEN AND HYPERPARAMETER TUNING AND COMPARISION .....	48
8.7 STACKING OF MODELS .....	48
8.8 STRATGEY FOR SELECTING BEST MODEL .....	51
8.9 MODELS: A DETAILED ANALYSIS .....	51
8.9.1 LOGISTIC REGRESSION.....	51
8.9.2 NAÏVE BAYES .....	53
8.9.3 K NEAREST NEIGHBOUR .....	54
8.9.4 SUPPORT VECTOR MACHINE.....	55
8.9.5 ADA BOOST.....	56
8.9.6 GRADIENT BOOSTING.....	58
8.9.7 BAGGING TREE .....	59
8.9.8 DECISION TREE .....	60
8.9.9 RANDOM FOREST .....	61
8.9.10 XGBOOST .....	62
8.9.11 VOTING CLASSIFIER .....	63
<b>9. ACTIONABLE INSIGHTS AND RECOMMENDATIONS .....</b>	<b>65</b>
9.1 RECOMMENDATIONS AND CONCLUSIONS POST EDA.....	65
9.2 RECOMMENDATIONS AND CONCLUSIONS POST MODELLING .....	66
<b>10. FUTURE SCOPE OF WORK .....</b>	<b>67</b>
10.1 APPLICATIONS OF WORK.....	67
<b>11. REFERENCES AND BIBLIOGRAPHY .....</b>	<b>68</b>
<b>12. APPENDIX .....</b>	<b>69</b>

Figure 1: Dataset excel screenshot .....	10
Figure 2: Dataset overview .....	11
Figure 3: List of attributes.....	11
Figure 4: Number of rows/columns.....	11
Figure 5: Fields with their datatype.....	12
Figure 6: 5-point summary.....	13
Figure 7: NULL entries in the dataset .....	16
Figure 8: Filling NAs with mean/median/zero/random values.....	17
Figure 9: Median imputation distribution .....	17
Figure 10: Random imputation distribution .....	17
Figure 11: Zero imputation distribution.....	18
Figure 12: Entries with imputed median values. ....	18
Figure 13: Entries with negative values.....	18
Figure 14: Boxplots of numerical variables .....	19
Figure 15: Histogram for column Informational_Duration.....	19
Figure 16: Kernel Density Estimate plot for column Informational_Duration.....	20
Figure 17: IQR check for extreme outliers.....	20
Figure 18: Different values considered on Top-Coding.....	20
Figure 19: One Hot Encoding of categorical data .....	21
Figure 20: Label encoding of response variable.....	21
Figure 21: Pairwise plots of variables w.r.t Revenue .....	22
Figure 22: KDE plot Revenue vs Page visits .....	23
Figure 23: Catplot Revenue vs Page visits.....	23
Figure 24: Avg visits on each page w.r.t. Revenue .....	24
Figure 25: Share of web page visits w.r.t. Revenue.....	24
Figure 26: Avg duration spent on web page w.r.t. Revenue.....	25
Figure 27: Share of duration spent on webpage w.r.t Revenue .....	25
Figure 28: KDE plots Bounce/Exit Rates vs Revenue .....	25
Figure 29: Boxplots of Google Analytics Metrics.....	26
Figure 30: KDE plot Special Day vs Revenue .....	26
Figure 31: Barplot Revenue vs Visitor type.....	27
Figure 32: Barplot Weekend vs Revenue.....	27
Figure 33: Barplot Month vs Revenue.....	28
Figure 34: Barplot Region vs Revenue .....	28
Figure 35: Barplot Traffic Type vs Revenue.....	29
Figure 36: Barplot OS vs Revenue .....	29
Figure 37: Barplot Browser vs Revenue .....	30
Figure 38: Catplot Traffic type hitting the website.....	30
Figure 39: Barplot Effect of special day on Revenue .....	31
Figure 40: Barplot monthly visits to website .....	31
Figure 41: No of visitor types w.r.t. Revenue .....	32
Figure 42: Share of Visitor types w.r.t. Revenue .....	32
Figure 43: Barplot completed purchases w.r.t. Revenue.....	32
Figure 44: Month and closeness to Special Days .....	33
Figure 45: Multivariate analysis part 1 .....	34
Figure 46: Multivariate analysis part 2 .....	35
Figure 47: Multivariate analysis part 3 .....	36
Figure 48: Correlation Matrix .....	37
Figure 49: Correlated features to be dropped (Brute force approach).....	37
Figure 50: Correlated features to be dropped (Second approach).....	38
Figure 51: Various feature selection techniques.....	40
Figure 52: Final selected variables and their frequencies .....	40
Figure 53: PCA check.....	41
Figure 54: Modelling approach .....	43
Figure 55: Shape of train and test data.....	43
Figure 56: Proportion of values for Revenue .....	44
Figure 57: Random oversampling minority class.....	44

Figure 58: SMOTE oversampling .....	45
Figure 59: SMOTEEN oversampling and undersampling .....	45
Figure 60: Stacking of Models .....	49
Figure 61: Stacking (Random upsampling) .....	49
Figure 62: Stacking (SMOTE) .....	49
Figure 63: Stacking (SMOTEEN) .....	50
Figure 64: Logistic Regression model metrics .....	52
Figure 65: Naive Bayes model metrics .....	53
Figure 66: KNN model metrics .....	54
Figure 67: SVM Model metrics.....	55
Figure 68: AdaBoost model metrics .....	56
Figure 69: Gradient Boosting model metrics .....	58
Figure 70: Bagging tree model metrics.....	59
Figure 71: Decision Tree model metrics.....	60
Figure 72: Random Forest model metrics.....	61
Figure 73: XGBoost model metrics.....	62
Figure 74: Voting classifier model metrics .....	63

## 1. EXECUTIVE SUMMARY

- E-commerce, the activity of buying and selling products online, is one of the many fields revolutionized by data science. One of the essential goals for e-commerce companies is to increase purchase conversion rates, i.e. the percentage of website visitors who complete the purchase at online stores. To achieve this goal, e-commerce companies as well as researchers in academia have devoted efforts in analyzing and modelling the behaviors of webpage users. Especially in recent year, there has been a trend in research to use machine learning methods to predict the behavior of users.
- One of the most popular activities on the Web is shopping. E-commerce became possible in 1991 when the Internet was opened to commercial use. Since that date thousands of businesses have taken up residence at web sites. Even though E-commerce industry is experiencing perennial growth since its inception, one of the crucial problems is that most of the visitors still do not complete their online shopping process. This leads to loss of revenues for the online retailers. This study is done in order to provide a solution for the above-mentioned problem by evaluating the actions taken by the visitors on E-commerce environment in real time and predicting the visitor's shopping intent.
- The information provided by the visits of users is fed to machine learning classification algorithms to build a predicting model. In the process of refining the model and making it better to provide more insightful results, oversampling and feature selection pre-processing steps are employed.
- For this project we used the Online Shoppers Purchasing Intention Dataset Data Set, obtained from the UCI Machine Learning repository. The goal is to build a predictive machine learning model that could categorize users as either, revenue generating, and non-revenue generation based on their behavior while navigating a website.
- The intention is to develop a prediction model using with selected variables for predicting the purchasing intention of users. The model can help e-commerce businesses identify customers who are more likely to complete transactions and adjust marketing strategies accordingly.
- With e-commerce becoming more and more prevalent in today's economy it is important for businesses within this sector to understand what factors into a site visitor making a purchase and being able to put their attention on potential customers. We thought it would be interesting to look into if it's possible to predict the buying behavior of a site visitor as this can have many implications such as E-commerce website able to better target ads or figure out factors that may lead to increased sales.



## 2. PROBLEM STATEMENT, SCOPE AND OBJECTIVE

### 2.1 PROBLEM STATEMENT

The online shoppers purchasing intention dataset consists of feature vectors belonging to 12,330 sessions. The dataset is formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period. More than 85% online sessions did not result in users completing the purchase and consequently did not generate any sort of revenue for the e-commerce website. This is a very serious issue which can affect the company's market share in future in a negative way.

### 2.2 OBJECTIVE

Our overall objective is to analyze the given data and predict/put forward a reasonable action plan for the company which it can employ around its marketing strategies so that increase number of online visitors complete their purchase successfully. This will result in increased revenue generation and greater market share for the company.

To achieve this objective, we will perform below steps: -

- Predict online visitor's shopping intent and whether online visitor will complete their transactions.
- Predict online visitor's web site abandonment likelihood.
- Separate users as either, revenue generating, and non-revenue generation based on their behavior while navigating a website.
- Identify customers who are more likely to complete transactions than others.

### 2.3 SCOPE

Scope of the project includes following: -

- Understanding the existing online visitors' shopping intention and their behavior.
- Focus will be on factors for which authentic data is available.
- Coding will be done using Python.
- Derive insights around key objectives – purchasing intention, revenue generation, website navigation behavior, website abandonment likelihood etc.

### 2.4 RESEARCH OBJECTIVES

- Online shopper's website visiting behavior depends on many things like total no of pages visited, time spent on product related pages, bounce rates, exit rates, operating system of the computer, significance of day etc. We will look to understand these factors and their influence.
- In many cases like our dataset, the final model will be a binary classifier meaning that there will be only two outcomes. The successful value is obtained only for 15% of the records. Such target incidence considering the amount of data in the dataset could work, however the proportion is still very small, and we consider it to be imbalanced. However, if this doesn't work, we will look to explore various data augmentation techniques and which ones can be implemented.

- There might be a no of other factors which might influence online visitor's behavior more than the ones already given in dataset. After model build, we may look to incorporate these factors into the model and check whether it improves the prediction better.
- One of our main objectives of our research is to run various algorithms mentioned in the Analytical Approach section to see which one will perform the best on our dataset. We will rigorously simulate numerous algorithms taught in class to see which one gives most accurate prediction.
- In addition to point 2 above, we will try to put forward a comprehensive end to end approach on how to deal with imbalanced data and build a reasonable accurate prediction model around it.

### 3. DATA SOURCE AND DESCRIPTION

The dataset is obtained from open UCI Machine Learning repository. The dataset consists of 12330 records, each containing metrics of web visits of a user within a one-year timeframe. 85.4% (10422) of the customers did not complete the transaction. Positive examples, i.e. those who completed transactions, only take up 15.5% (1908) of the dataset. The dataset contains 18 columns, a total of which 17 are features and 1 is the target variable, in this instance, 'Revenue'. There are both continuous and categorical features in the data. Continuous variables include the total number of visits and time spent in three types of websites. There is also the average Google Analytics web metrics for all the websites that the users visit. Categorical variables record the user profiles and session information, including the type of operation systems and browsers, time and locations of the visits, etc.

#### 3.1 SCREEN-SHOT OF DATA SET

Administrative	Administrative_Dur	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValue	SpecialDay	Month	Operating	Browser	Region	TrafficTyp	VisitorType	Weekend	Revenue
0	0	0	0	1	0	0.2	0.2	0	0	Feb	1	1	1	1	Returning	FALSE	FALSE
0	0	0	0	2	64	0	0.1	0	0	Feb	2	2	1	2	Returning	FALSE	FALSE
0	0	0	0	1	0	0.2	0.2	0	0	Feb	4	1	9	3	Returning	FALSE	FALSE
0	0	0	0	2	2.66666667	0.05	0.14	0	0	Feb	3	2	2	4	Returning	FALSE	FALSE
0	0	0	0	10	627.5	0.02	0.05	0	0	Feb	3	3	1	4	Returning	TRUE	FALSE
0	0	0	0	19	154.216667	0.015789474	0.024561	0	0	Feb	2	2	1	3	Returning	FALSE	FALSE
0	0	0	0	1	0	0.2	0.2	0	0.4	Feb	2	4	3	3	Returning	FALSE	FALSE
1	0	0	0	0	0	0.2	0.2	0	0	Feb	1	2	1	5	Returning	TRUE	FALSE
0	0	0	0	2	37	0	0.1	0	0.8	Feb	2	2	2	3	Returning	FALSE	FALSE
0	0	0	0	3	738	0	0.222222	0	0.4	Feb	2	4	1	2	Returning	FALSE	FALSE
0	0	0	0	3	395	0	0.066667	0	0	Feb	1	1	3	3	Returning	FALSE	FALSE
0	0	0	0	16	407.75	0.01875	0.025833	0	0.4	Feb	1	1	4	3	Returning	FALSE	FALSE
0	0	0	0	7	280.5	0	0.028571	0	0	Feb	1	1	1	3	Returning	FALSE	FALSE
0	0	0	0	6	98	0	0.066667	0	0	Feb	2	5	1	3	Returning	FALSE	FALSE
0	0	0	0	2	68	0	0.1	0	0	Feb	3	2	3	3	Returning	FALSE	FALSE
2	53	0	0	23	1668.285119	0.008333333	0.016313	0	0	Feb	1	1	9	3	Returning	FALSE	FALSE
0	0	0	0	1	0	0.2	0.2	0	0	Feb	1	1	4	3	Returning	FALSE	FALSE
0	0	0	0	13	334.966667	0	0.007692	0	0	Feb	1	1	1	4	Returning	TRUE	FALSE
0	0	0	0	2	32	0	0.1	0	0	Feb	2	2	1	3	Returning	FALSE	FALSE
0	0	0	0	20	2981.166667	0	0.01	0	0	Feb	2	4	4	4	Returning	FALSE	FALSE
0	0	0	0	8	136.166667	0	0.008333	0	1	Feb	2	2	5	1	Returning	TRUE	FALSE
0	0	0	0	2	0	0.2	0.2	0	0	Feb	3	3	1	3	Returning	FALSE	FALSE
0	0	0	0	3	105	0	0.033333	0	0	Feb	3	2	1	5	Returning	FALSE	FALSE
0	0	0	0	2	15	0	0.1	0	0.8	Feb	2	4	1	3	Returning	FALSE	FALSE
0	0	0	0	1	0	0.2	0.2	0	0	Feb	2	2	4	1	Returning	TRUE	FALSE
0	0	0	0	5	156	0	0.04	0	0	Feb	1	1	9	3	Returning	FALSE	FALSE
4	64.6	0	0	32	1135.444444	0.002857143	0.009524	0	0	Feb	2	2	1	3	Returning	FALSE	FALSE
0	0	0	0	4	76	0.05	0.1	0	0	Feb	1	1	1	3	Returning	FALSE	FALSE

Figure 1: Dataset excel screenshot

#### 3.2 DATA DESCRIPTION

Variable	Description
Administrative	Number of 'administrative' pages viewed
Administrative_Duration	Time spent looking at 'administrative' pages
Informational	Number of 'informational' pages viewed

<b>Informational_Duration</b>	Time spent looking at ‘informational’ pages
<b>ProductRelated</b>	Number of ‘product related’ pages viewed
<b>ProductRelated_Duration</b>	Time spent looking at ‘product related’ pages
<b>BounceRates</b>	The percentage of visitors who enter the site from that page and then leave (“bounce”) without triggering any other requests to the analytics server during that session
<b>ExitRates</b>	It is calculated as for all page views to the page, the percentage that were the last in the session
<b>PageValues</b>	Represents the average value for a web page that a user visited before completing an e-commerce transaction
<b>SpecialDay</b>	Indicates the closeness of the site visiting time to a specific special day (e.g. Mother’s Day, Valentine’s Day) in which the sessions are more likely to be finalized with transaction
<b>Month</b>	Month of the year for the session
<b>OperatingSystems</b>	Operating system used for the session
<b>Browser</b>	Browser used for the session
<b>Region</b>	Region of the user
<b>TrafficType</b>	Traffic Type
<b>VisitorType</b>	Types of Visitor
<b>Weekend</b>	Session occurred on a weekend or not
<b>Revenue</b>	Represents whether the user generated revenue or not

Table 1: Data description of fields

### 3.3 DATA OVERVIEW

Below is the broad overview of the dataset with table structure, column details, 5 point summary and brief overview of the data.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues
0	0.0	0.0	0.0	0.0	1.0	0.000000	0.20	0.20	0.0
1	0.0	0.0	0.0	0.0	2.0	64.000000	0.00	0.10	0.0
2	0.0	-1.0	0.0	-1.0	1.0	-1.000000	0.20	0.20	0.0
3	0.0	0.0	0.0	0.0	2.0	2.666667	0.05	0.14	0.0
4	0.0	0.0	0.0	0.0	10.0	627.500000	0.02	0.05	0.0

Figure 2: Dataset overview

```
Index(['Administrative', 'Administrative_Duration', 'Informational',
      'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration',
      'BounceRates', 'ExitRates', 'PageValues', 'SpecialDay', 'Month',
      'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType',
      'Weekend', 'Revenue'],
      dtype='object')
```

Figure 3: List of attributes

Number of records : 12330  
Number of Features : 18

Figure 4: Number of rows/columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
Administrative      12316 non-null float64
Administrative_Duration  12316 non-null float64
Informational      12316 non-null float64
Informational_Duration  12316 non-null float64
ProductRelated     12316 non-null float64
ProductRelated_Duration  12316 non-null float64
BounceRates        12316 non-null float64
ExitRates          12316 non-null float64
PageValues         12330 non-null float64
SpecialDay         12330 non-null float64
Month              12330 non-null object
OperatingSystems   12330 non-null int64
Browser            12330 non-null int64
Region             12330 non-null int64
TrafficType        12330 non-null int64
VisitorType        12330 non-null object
Weekend            12330 non-null bool
Revenue            12330 non-null bool
dtypes: bool(2), float64(10), int64(4), object(2)
memory usage: 1.5+ MB
```

Figure 5: Fields with their datatype

### **Brief Overview of the Data:**

- **Data Structure:** The dataset has users' web session details over one-year period. It consists of feature vectors belonging to 12,330 sessions. Continuous variables include the total number of visits and time spent in three types of websites. There is also the average Google Analytics web metrics for all the websites that the users visit. Categorical variables record the user profiles and session information, including the type of operation systems and browsers, time and locations of the visits, etc.
- **No. of rows :12330, No. of columns: 18**
- First 8 columns i.e. **Administrative, Administrative\_Duration, Informational, Informational\_Duration, ProductRelated, ProductRelated\_Duration, BounceRates, ExitRates** have **NULL** Values. There are 14 Null values in each of the mentioned columns.
  - All of these are numerical columns so missing values imputation will be done by either Median, Mean, Zero or Random value-based method. We will choose the one for which distributions after imputations remain similar to the original distributions.
- Dataset has one line of **record for each session belonging to different user in a 1-year period.**
- The different metrics measured by "Google Analytics" for each page in the e-commerce site as follows:
  - Bounce Rates
  - Exit Rates
  - Page Values
- **Revenue is our Response Variable.** When a user's online session results in a completed purchase the value is TRUE otherwise FALSE.
- There are 10422 users who did not complete the purchase (i.e. Revenue FALSE) and 1908 users completed the purchase (i.e. Revenue TRUE).

### **FIVE POINT Summary of the Data:**

Table below represents a 5-point summary of the data:

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	Pa
count	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000	12316.000000
mean	2.317798	80.906176	0.503979	34.506387	31.763884	1196.037057	0.022152	0.043003	0.043003
std	3.322754	176.860432	1.270701	140.825479	44.490339	1914.372511	0.048427	0.048527	0.048527
min	0.000000	-1.000000	0.000000	-1.000000	0.000000	-1.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000	185.000000	0.000000	0.014286	0.014286
50%	1.000000	8.000000	0.000000	0.000000	18.000000	599.766190	0.003119	0.025124	0.025124
75%	4.000000	93.500000	0.000000	0.000000	38.000000	1466.479902	0.016684	0.050000	0.050000
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230	0.200000	0.200000	0.200000

PageValues	SpecialDay	OperatingSystems	Browser	Region	TrafficType
12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000
5.889258	0.061427	2.124006	2.357097	3.147364	4.069586
18.568437	0.198917	0.911325	1.717277	2.401591	4.025169
0.000000	0.000000	1.000000	1.000000	1.000000	1.000000
0.000000	0.000000	2.000000	2.000000	1.000000	2.000000
0.000000	0.000000	2.000000	2.000000	3.000000	2.000000
0.000000	0.000000	3.000000	2.000000	4.000000	4.000000
361.763742	1.000000	8.000000	13.000000	9.000000	20.000000

Figure 6: 5-point summary

- Summary clearly shows the count of records, mean, minimum values, maximum values, 25-50-75 quantile values for variables in the given dataset.
- First 6 columns show number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. All of these have low 25-50-75 quantile values but very extreme maximum values. These might be outliers, but we can't be sure as high values could mean a good chance of completed purchase.
- Bounce Rates and Exit rates have very low 5-point values as these are basically percentages. Average Bounce Rate is less than average exit rate. Both rates have same maximum percentage values.
- Columns like Operating Systems, Browser, Region, Traffic Type are categorical and thus quantile values have little significance.
- Page Values average value is 6. It has an abnormally extreme maximum value which we will check later.

## 4. LITERATURE REVIEWS

Our dataset is publicly available on UCI machine learning repository, so it was apparent that analytics work would already have been carried out on this dataset. We researched online to find out about this. We did find couple of papers which talked about predicting purchasing intent, but they were carried out on different datasets containing different variables and with separate parameters.

### **1. Customer Purchase Intent Prediction Under Online Multi-Channel Promotion: A Feature-Combined Deep Learning Framework**

This paper tried to predict customer purchase intent prediction under online multi-channel promotion. In the context of joint promotions on multiple online channels, customers can access and compare prices and services by navigating between channels to aid their purchase decision. The interactions between customer and promotion channels offer another angle to predict their purchase intent during promotions.

Techniques used: Long short-term memory - Fully connected (LSTM-FC) neural network

### **2. Predicting purchasing intent: Automatic Feature Learning using Recurrent Neural Networks**

This paper tried to additionally predict which elements of the product catalogue of the e-commerce website do users prefer i.e. rank content along with usual prediction of users most likely to purchase (predict purchasing intent).

Techniques used: Multi-layer recurrent neural networks

### **3. Investigating the Shopping Orientations on Online Purchase Intention in the e-Commerce Environment: A Malaysian Study**

This paper has first objective to evaluate the impact of shopping orientations on customer online purchase intention. The second objective is to identify which construct has the greatest impact on purchase intention

Techniques used: Principal component factor analysis, Multiple linear regression analysis

### **4. Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks**

This project is to determine the visitors which have purchasing intention but are likely to leave the site in the prediction horizon and take actions accordingly to improve the Web site abandonment and purchase conversion rates. The findings support the feasibility of accurate and scalable purchasing intention prediction for virtual shopping environment using clickstream and session information data.

Techniques used: Random forest, Support vector machines, Multilayer perceptron.

### **5. SMOTE-NC in ML Categorization Models for Imbalanced Datasets**

This blogpost focusses on SMOTE-NC, and its effect on the machine learning models' scores used to categorize the data. SMOTE-NC is a great tool to generate synthetic data to oversample a minority target class in an imbalanced dataset. The parameters that can be tuned are k-neighbors, which allow to determine the number of nearest neighbors to create the new sample, and sampling strategy, which allows to indicate how many new samples to create.

Techniques used: SMOTE-NC, Logistic regression, Random forest, Gradient boosting, Decision tree.

## **6. How to Combine Oversampling and Undersampling for Imbalanced Classification**

This blogpost explains how to combine oversampling and undersampling techniques for imbalanced classification. It also shows how to define a sequence of oversampling and undersampling methods to be applied to a training dataset or when evaluating a classifier model, how to manually combine oversampling and undersampling methods for imbalanced classification and how to use pre-defined and well-performing combinations of resampling methods for imbalanced classification.

Techniques used: Decision tree, SMOTE and Tomek Links Undersampling, SMOTE and Edited Nearest Neighbors Undersampling

## **7. ROC Curves and Precision-Recall Curves for Imbalanced Classification**

This blogpost discovers ROC Curves and Precision-Recall Curves for imbalanced classification. It explains how ROC Curves and Precision-Recall Curves provide a diagnostic tool for binary classification models, ROC AUC and Precision-Recall AUC provide scores that summarize the curves and can be used to compare classifiers and ROC Curves and ROC AUC can be optimistic on severely imbalanced classification problems with few samples of the minority class.

Techniques used: Logistic regression

## **8. A Survey of Predictive Modelling under Imbalanced Distributions**

Many real-world data mining applications involve obtaining predictive models using data sets with strongly imbalanced distributions of the target variable. Frequently, the least common values of this target variable are associated with events that are highly relevant for end users (e.g. fraud detection, unusual returns on stock markets, anticipation of catastrophes, etc.). Moreover, the events may have different costs and benefits, which when associated with the rarity of some of them on the available training data creates serious problems to predictive modelling techniques.

This paper presents a survey of existing techniques for handling these important applications of predictive analytics. It describes methods designed to handle similar problems within regression tasks (numeric target variables). It also discusses the main challenges raised by imbalanced distributions, describe the main approaches to these problems, propose a taxonomy of these methods and refer to some related problems within predictive modelling

## **9. Predicting online purchase intention: An Empirical Study**

Purpose of this research is to explore the factors that affect consumer purchase intention in shopping online. The factors that affect online customer intention are perceived benefit, perceived risk, hedonic motivation, trust, and attitude toward online shopping. The primary data was obtained using questionnaire with the sample size of 200 respondents.

Techniques used: Factor analysis, Correlation analysis, Structural equation modelling.

Other articles/blogs/posts talked about any one aspect of the analysis and that too majority was R driven.

So, the idea exists, and it cannot be denied. But there was no specific case study which explored all our objectives in Python. We perceive our project to be unique, in that, we are treading a territory that may have been possibly partially explored or not explored at all. We will be delving into complete end to end Python based online purchase intent model prediction with imbalanced dataset.



## 5. DATA PREPROCESSING

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. For our dataset we will do relevant imputations for NULL(NA) values and treat outliers as well as negative values.

### 5.1 MISSING VALUE IMPUTATION

We will analyze the NULL values and impute them appropriately. Let's look at the columns and rows which contain NULL values.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates
1063	3.0	43.5	0.0	0.0	30.0	2264.333333	0.0	0.010417
1064	4.0	76.0	0.0	0.0	28.0	866.000000	0.0	0.021839
1065	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1132	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1133	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1134	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1135	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1473	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1474	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1475	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1476	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2037	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2038	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2039	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2753	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2754	0.0	0.0	0.0	0.0	7.0	82.000000	0.0	0.057143
2755	4.0	61.0	0.0	0.0	16.0	502.500000	0.0	0.009524

Figure 7: NULL entries in the dataset

Administrative, Administrative\_Duration, Informational, Informational\_Duration, ProductRelated, ProductRelated\_Duration, BounceRates, ExitRates columns have NULL values in same 14 records.

For each of these 8 columns we will create 4 new columns using below strategy.

- [Variable]\_Median - Fill N/a with median [Variable] value
- [Variable]\_Zero- Fill N/a with 0
- [Variable]\_Mean- Fill N/a with mean [Variable] value
- [Variable]\_Random- Fill N/a with random value of [Variable]

Here we have shown for Administrative column only. Same thing has been done for all 8 columns in consideration.



VisitorType	Weekend	Revenue	Administrative_median	Administrative_zero	Administrative_mean	Administrative_random
Returning_Visitor	True	False	3.0	3.0	3.000000	3.0
Returning_Visitor	False	False	4.0	4.0	4.000000	4.0
Returning_Visitor	False	False	1.0	0.0	2.317798	0.0
Returning_Visitor	False	False	1.0	0.0	2.317798	0.0
Returning_Visitor	False	False	1.0	0.0	2.317798	6.0
Returning_Visitor	False	False	1.0	0.0	2.317798	8.0
Returning_Visitor	False	False	1.0	0.0	2.317798	4.0
Returning_Visitor	False	False	1.0	0.0	2.317798	2.0
Returning_Visitor	True	False	1.0	0.0	2.317798	3.0
Returning_Visitor	True	False	1.0	0.0	2.317798	3.0
Returning_Visitor	False	False	1.0	0.0	2.317798	0.0
Returning_Visitor	False	False	1.0	0.0	2.317798	0.0
Returning_Visitor	False	False	1.0	0.0	2.317798	1.0
Returning_Visitor	False	False	1.0	0.0	2.317798	0.0
Returning_Visitor	True	False	1.0	0.0	2.317798	0.0
Returning_Visitor	False	False	1.0	0.0	2.317798	5.0
Returning_Visitor	False	False	0.0	0.0	0.000000	0.0
Returning_Visitor	True	True	4.0	4.0	4.000000	4.0

Figure 8: Filling NAs with mean/median/zero/random values

**Standard Deviation Check:** It is evident that there is no major change in the variance after imputation except mean imputation.

Original Standard Deviation: 3.3227538813737088  
Standard Deviation after median imputation: 3.3211633492176733  
Standard Deviation after zero imputation: 3.32178410615674  
Standard Deviation after random imputation: 3.3219763431542875  
Standard Deviation after mean imputation: 3.320866795328869

Next, we will also consider if there is any change in distribution after these imputations. We can see clearly below that distribution has not changed with median, zero or random value imputations.

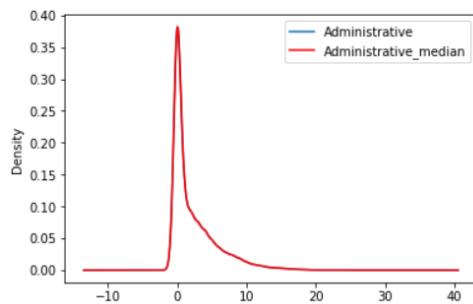


Figure 9: Median imputation distribution

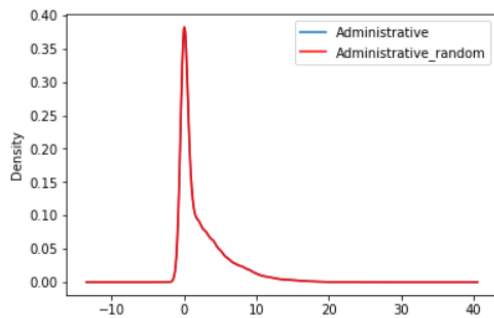


Figure 10: Random imputation distribution

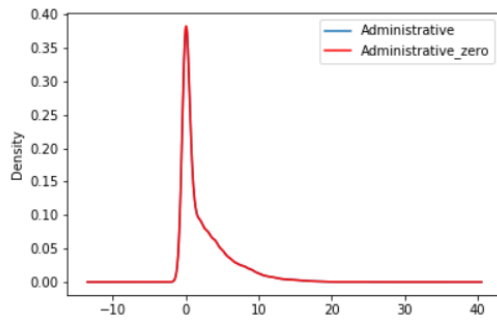


Figure 11: Zero imputation distribution

Therefore, we can impute using any of these methods as it is not changing the original distribution and will not make a major difference. But since data is skewed, we have decided to go with median imputations for all 8 columns.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates
1063	3.0	43.5	0.0	0.0	30.0	2264.333333	0.000000	0.010417
1064	4.0	76.0	0.0	0.0	28.0	866.000000	0.000000	0.021839
1065	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1132	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1133	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1134	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1135	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1136	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1473	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1474	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1475	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
1476	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
2037	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
2038	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
2039	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124
2753	1.0	8.0	0.0	0.0	18.0	599.766190	0.003119	0.025124

Figure 12: Entries with imputed median values.

## 5.2 NEGATIVE VALUE TREATMENT

We have negative values for Administrative\_Duration, Informational\_Duration and ProductRelated\_Duration columns in our dataset.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration
2	0.0	-1.0	0.0	-1.0	1.0	-1.0
6	0.0	-1.0	0.0	-1.0	1.0	-1.0
7	1.0	-1.0	0.0	-1.0	1.0	-1.0
16	0.0	-1.0	0.0	-1.0	1.0	-1.0
21	0.0	-1.0	0.0	-1.0	1.0	-1.0
...	...	...	...	...	...	...
5124	1.0	-1.0	0.0	-1.0	1.0	-1.0
6260	0.0	-1.0	0.0	-1.0	1.0	-1.0
7210	0.0	-1.0	0.0	-1.0	1.0	-1.0
8052	0.0	-1.0	0.0	-1.0	1.0	-1.0
8636	0.0	-1.0	0.0	-1.0	1.0	-1.0

Figure 13: Entries with negative values

These columns represent the time spent on the web page which can't be negative. These values can be put to 0 as there is no meaning of negative values. 0 simply means user not visiting that page. This would also be like all other majority of entries which have 0 values in these columns. Therefore, we decide to impute these values with 0.

### 5.3 OUTLIERS TREATMENT

Boxplots of all the numerical variables in dataset is given below. From the figure we deduce that Administrative\_duration, Informational\_Duration and ProductRelated\_Duration have extreme outliers.

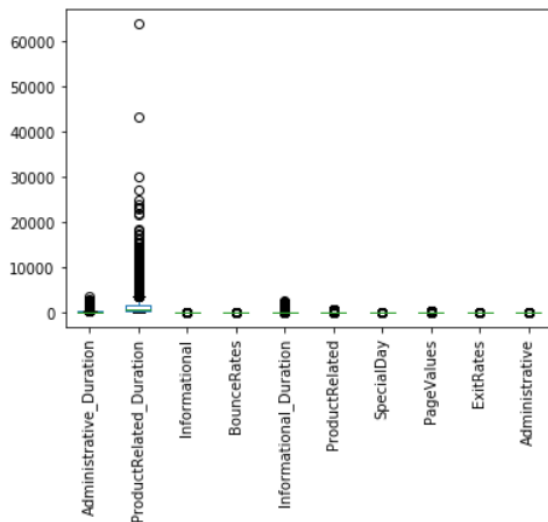


Figure 14: Boxplots of numerical variables

Other variables also seem to have outliers if we examine boxplots individually but upon carefully looking at the dataset values it doesn't feel right to cap/delete them as they provide valuable information.

For the 3 duration variables i.e. Administrative\_duration, Informational\_Duration and ProductRelated\_Duration, we further plot kdeplots and histograms to get an idea of the distribution. This will help us to determine what strategy we can use – Top coding, bottom coding, zero coding, IQR method etc.

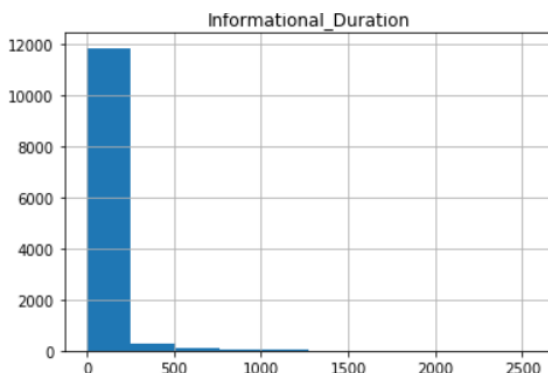


Figure 15: Histogram for column Informational\_Duration

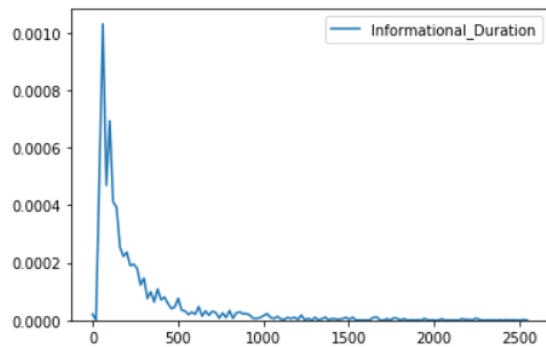


Figure 16: Kernel Density Estimate plot for column Informational\_Duration

Because the distribution of all 3 duration variables is skewed, we should estimate outliers using the quantile method.

The outliers, according to the box plots, lie at the right side of the distribution. Therefore, in these variable, only extremely high values will affect the performance of our machine learning models, and we need to do therefore **top-coding**.

Doing the IQR check for all 3 variables reveals that there are many valuable entries and percentage of outliers is <1%.

```
# Lets Look at the actual number of sessions that have product related duration more than 5311
#print('total sessions: {}'.format(mydata.shape[0]))
#print('number of sessions that have product related duration more than 5310: {}'.format(mydata[mydata.ProductRelated_Duration>5310].shape[0]))
#print('percentage of outliers: {}'.format(mydata[mydata.ProductRelated_Duration>5311].shape[0]/np.float(mydata.shape[0])))
#total sessions: 12330
#number of sessions that have product related duration more than 5310: 402
#percentage of outliers: 0.03260340632603406

# Lets Look at the actual number of sessions that have admin duration more than 374
#print('total sessions: {}'.format(mydata.shape[0]))
#print('number of sessions that have admin duration more than 374: {}'.format(mydata[mydata.Administrative_Duration>374].shape[0]))
#print('percentage of outliers: {}'.format(mydata[mydata.Administrative_Duration>374].shape[0]/np.float(mydata.shape[0])))
#total sessions: 12330
#number of sessions that have admin duration more than 374: 552
#percentage of outliers: 0.04476885644768856

# Lets Look at the actual number of sessions that have INFORMATIONAL duration more than 0
print('total sessions: {}'.format(mydata.shape[0]))
print('number of sessions that have Informational_Duration duration more than 0: {}'.format(mydata[mydata.Informational_Duration>0].shape[0]))
print('percentage of outliers: {}'.format(mydata[mydata.Informational_Duration>0].shape[0]/np.float(mydata.shape[0])))
#total sessions: 12330
#number of sessions that have Informational_Duration duration more than 0: 2404
#percentage of outliers: 0.19497161394971613
```

Figure 17: IQR check for extreme outliers

Finally, we decide to do top coding of Administrative\_Duration at 2500, Informational\_Duration at 2200 and ProductRelated\_Duration at 20000.

ProductRelated_Duration		
43171.23338		
21857.04648		
23050.10414		
23342.08205		
23888.81000		
Administrative_Duration		
63973.52223	2629.253968	
24844.15620	2720.500000	
27009.85943		2252.033333
21672.24425	3398.750000	2549.375000
29970.46597	2657.318056	2256.916667

Figure 18: Different values considered on Top-Coding

## 5.4 HANDLING CATEGORICAL VARIABLES

In our dataset there are 8 categorical attributes. Therefore, we have performed **one hot encoding** of categorical attributes and **label encoding** of Response variable i.e. Revenue. Output of both is shown below. Encoded categorical variables can be seen in image below.

```
Month_Aug          12330 non-null uint8
Month_Dec          12330 non-null uint8
Month_Feb          12330 non-null uint8
Month_Jul          12330 non-null uint8
Month_June         12330 non-null uint8
Month_Mar          12330 non-null uint8
Month_May          12330 non-null uint8
Month_Nov          12330 non-null uint8
Month_Oct          12330 non-null uint8
Month_Sep          12330 non-null uint8
OperatingSystems_1 12330 non-null uint8
OperatingSystems_2 12330 non-null uint8
OperatingSystems_3 12330 non-null uint8
OperatingSystems_4 12330 non-null uint8
OperatingSystems_5 12330 non-null uint8
OperatingSystems_6 12330 non-null uint8
OperatingSystems_7 12330 non-null uint8
OperatingSystems_8 12330 non-null uint8
Browser_1          12330 non-null uint8
Browser_2          12330 non-null uint8
Browser_3          12330 non-null uint8
Browser_4          12330 non-null uint8
Browser_5          12330 non-null uint8
Browser_6          12330 non-null uint8
Browser_7          12330 non-null uint8
Browser_8          12330 non-null uint8
Browser_9          12330 non-null uint8
Browser_10         12330 non-null uint8
Browser_11         12330 non-null uint8
Browser_12         12330 non-null uint8
Browser_13         12330 non-null uint8
Region_1           12330 non-null uint8
Region_2           12330 non-null uint8
Region_3           12330 non-null uint8
Region_4           12330 non-null uint8
Region_5           12330 non-null uint8
Region_6           12330 non-null uint8
Region_7           12330 non-null uint8
Region_8           12330 non-null uint8
Region_9           12330 non-null uint8
VisitorType_New_Visitor 12330 non-null uint8
VisitorType_Other  12330 non-null uint8
VisitorType_Returning_Visitor 12330 non-null uint8
Weekend_False      12330 non-null uint8
Weekend_True       12330 non-null uint8
Revenue            12330 non-null bool
```

Figure 19: One Hot Encoding of categorical data

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

Figure 20: Label encoding of response variable

## 6. EXPLORATORY DATA ANALYSIS

The data has numerical as well as categorical variables. As part of visual exploratory data analysis, we will study the relationship between variables, note down the important patterns/insights found for further analysis, study the correlation between variables and collate initial results from Exploratory Data Analysis.

We start off with pairwise plots shown below. Please zoom in the doc for clearer image.

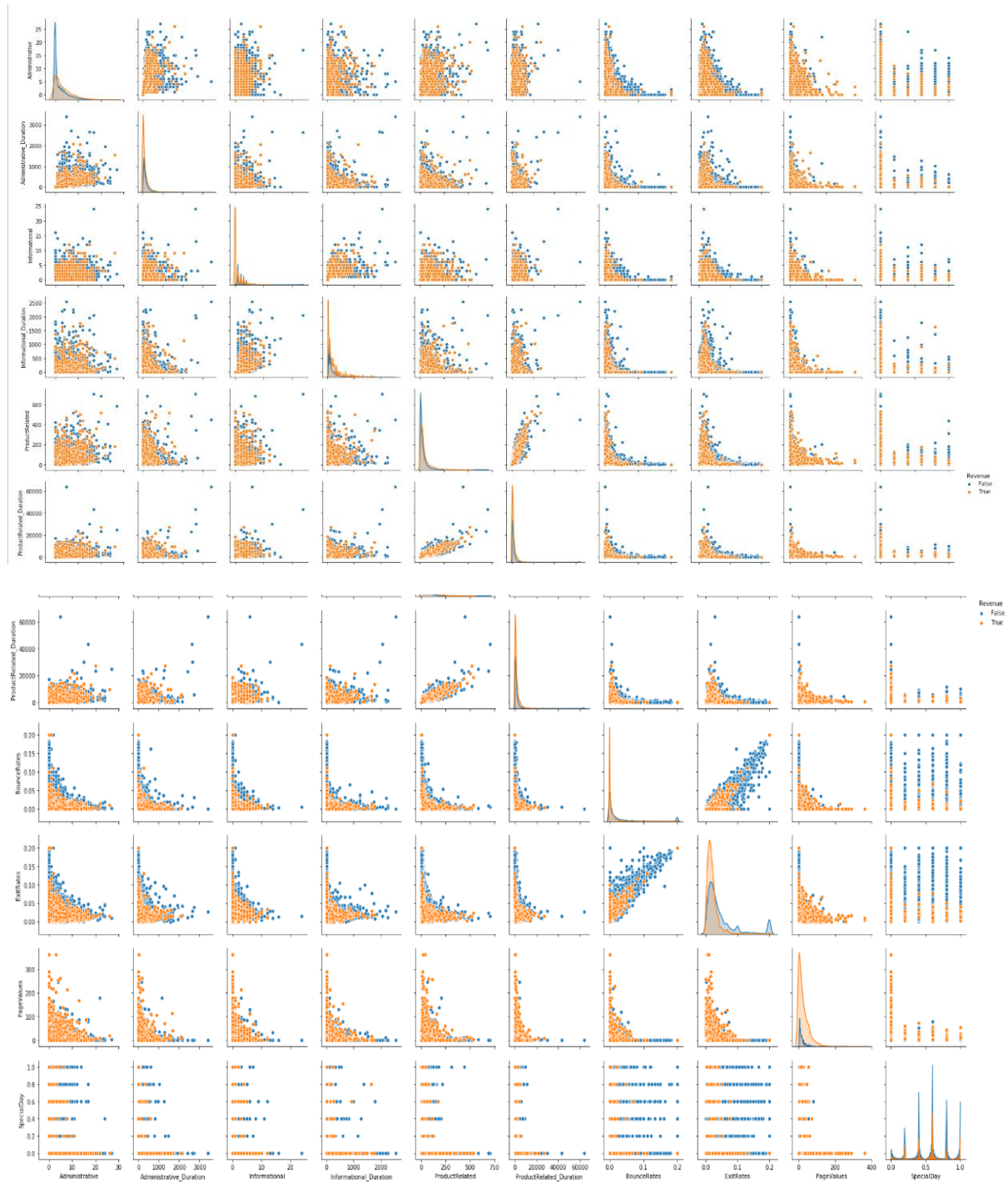


Figure 21: Pairwise plots of variables w.r.t Revenue

### Business Insights

- 1) There is positive correlation between BounceRates and ExitRates.
- 2) Customers with low BounceRates and low ExitRates made the purchase.
- 3) If the page value is less 100, users are more likely to make the purchase.
- 4) ExitRates and BounceRates are less on ProductRelated page as compared to Administrative and Informational.
- 5) Special Day is not having much effect on customer's purchase intension.

## 6.1 UNIVARIATE/BIVARIATE ANALYSIS

### 6.1.1 PAGE VISITS VS REVENUE

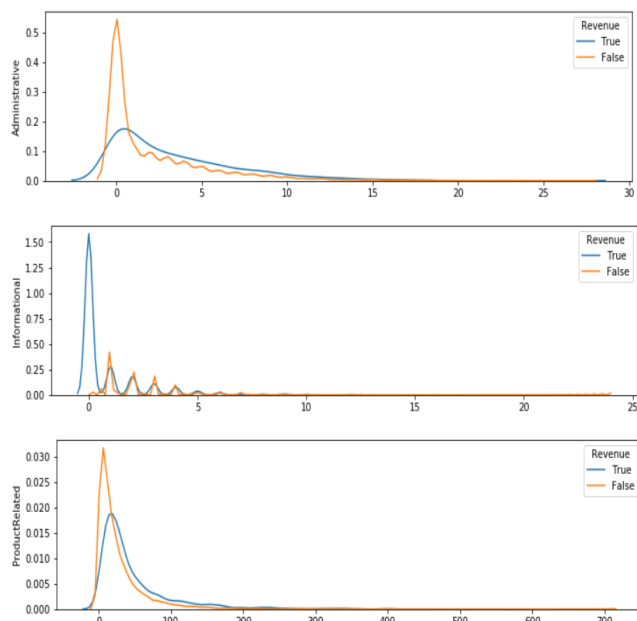


Figure 22: KDE plot Revenue vs Page visits

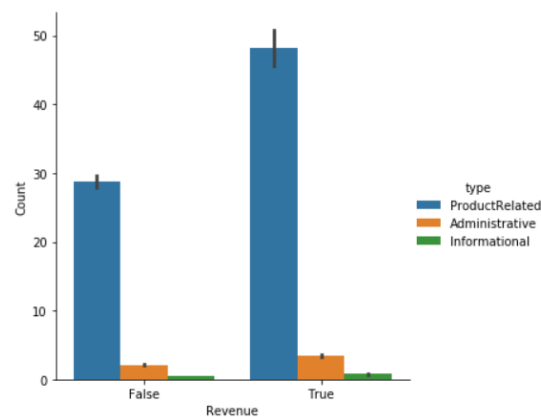


Figure 23: Catplot Revenue vs Page visits

### Business Insights

- 1) Customers visited ProductRelated web page more as compared to Administrative and Informational.
- 2) At this point of visual EDA, it is becoming seemingly clear that Admin page is HOME page, Informational is CONTACT US and ProductRelated is the actual webpage for the product in interest. This exclusive info was not provided in the dataset.

#### 6.1.2 SHARE OF WEB PAGE VISITS W.R.T REVENUE

	Administrative	Informational	ProductRelated
<b>Revenue</b>			
False	2.120580	0.452248	28.748943
True	3.393606	0.786164	48.210168

Figure 24: Avg visits on each page w.r.t. Revenue

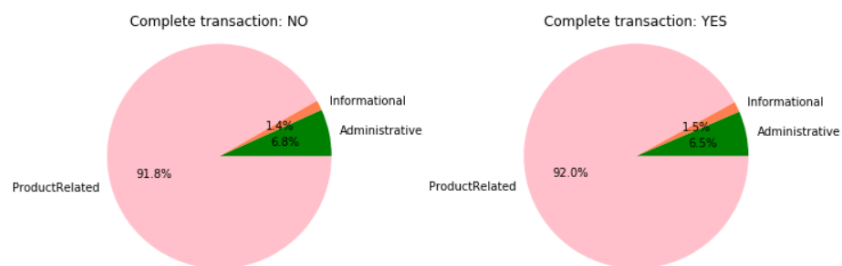


Figure 25: Share of web page visits w.r.t. Revenue

### Business Insights

- 1) People visit product related pages more rather than Administrative or Informational which is obvious as user intends to purchase rather than just read admin and info pages on the website.



### 6.1.3 DURATION SPENT ON PAGES VS REVENUE

	Administrative_Duration	Informational_Duration	ProductRelated_Duration
Revenue			
False	73.834208	30.270759	1071.347468
True	119.483244	57.611427	1876.209615

Figure 26: Avg duration spent on web page w.r.t. Revenue

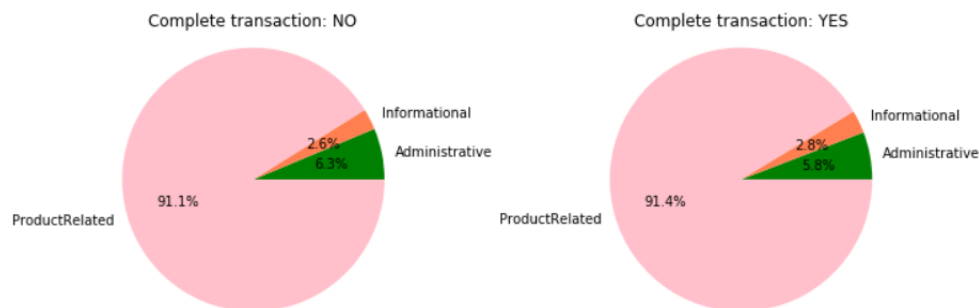


Figure 27: Share of duration spent on webpage w.r.t Revenue

#### Business Insights

- 1) On average, people who complete transactions visit more webpages, and spend more time on these webpages.
- 2) Doesn't seem to be a clear difference in the type of webpage visited between people who complete the purchase and those who don't.

### 6.1.4 BOUNCE RATES AND EXIT RATES VS REVENUE

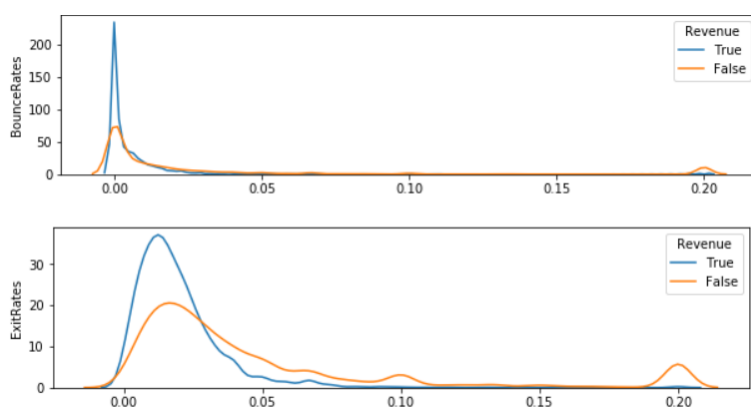


Figure 28: KDE plots Bounce/Exit Rates vs Revenue

#### Business Insights

- 1) Customers with low exit rate and low bounce rate generated more revenue.

### 6.1.5 EXPLORING GOOGLE ANALYTICS METRICS

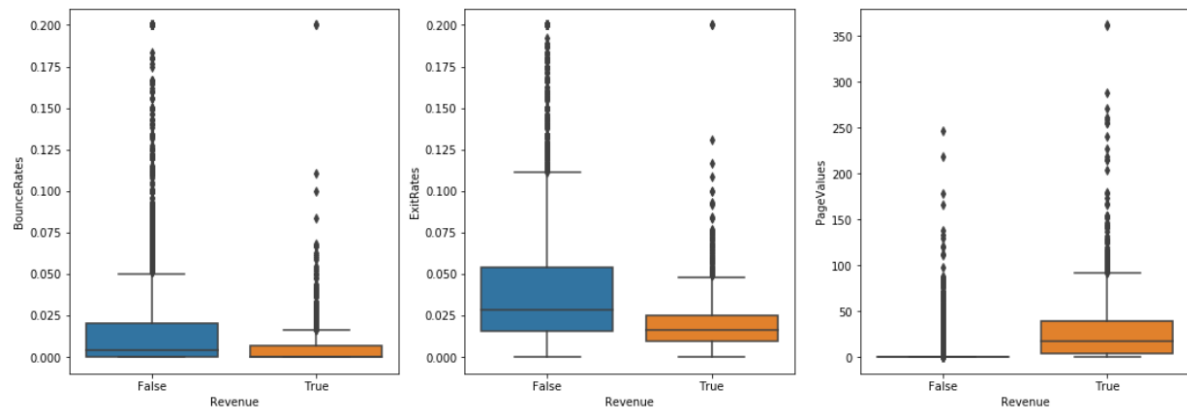


Figure 29: Boxplots of Google Analytics Metrics

#### Business Insights

- 1) For effective revenue generation low bounce rates, low exit rates and high page values hold the key.
- 2) Focus on this combo can give good dividends as part of business strategy.

### 6.1.6 SPECIAL DAY VS REVENUE

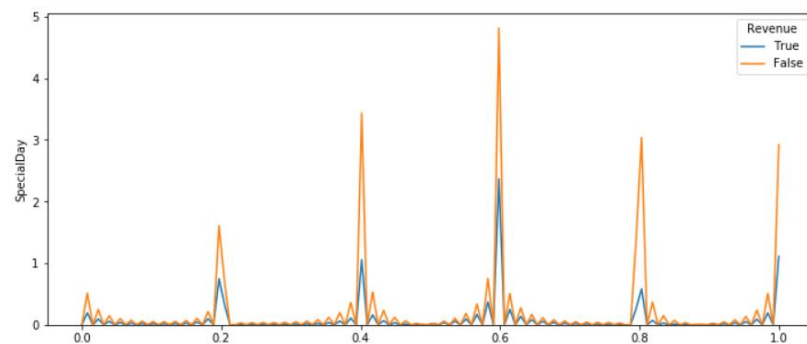


Figure 30: KDE plot Special Day vs Revenue

#### Business Insights

- 1) Closeness of the online session to Special Day is not having much effect on revenue generation.

### 6.1.7 REVENUE VS VISITOR TYPE

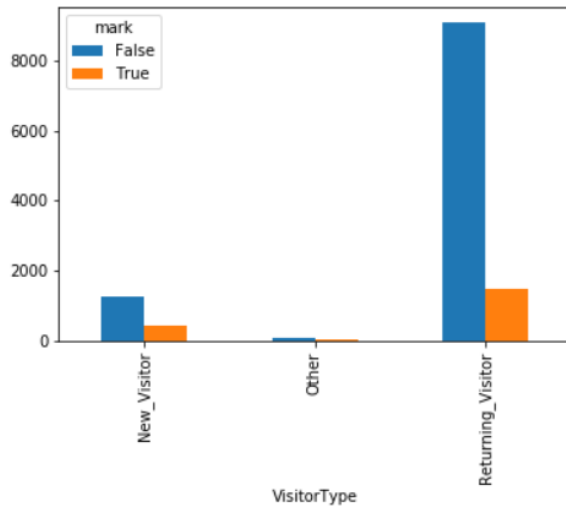


Figure 31: Barplot Revenue vs Visitor type

#### Business Insights

- 1) Most of the customers visiting the website are Returning visitors, contributing to most number of purchases.
- 2) About 1/4th of New\_visitors made the purchase as compared to ~15 % of Returning visitors.

### 6.1.8 WEEKEND VS REVENUE

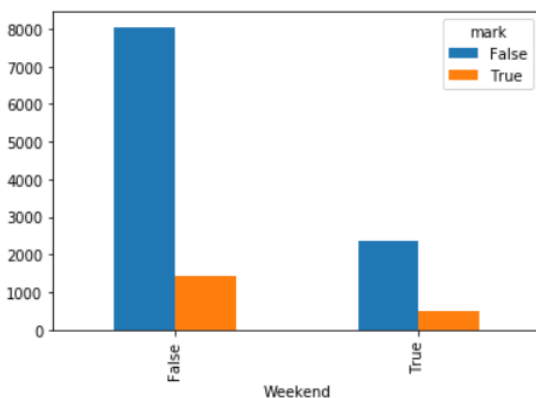


Figure 32: Barplot Weekend vs Revenue

#### Business Insights

- 1) It is evident that even though the no of visitors in weekdays are more compared to weekends but there is no significant difference in conversion to Revenue rates.

### 6.1.9 MONTH VS REVENUE

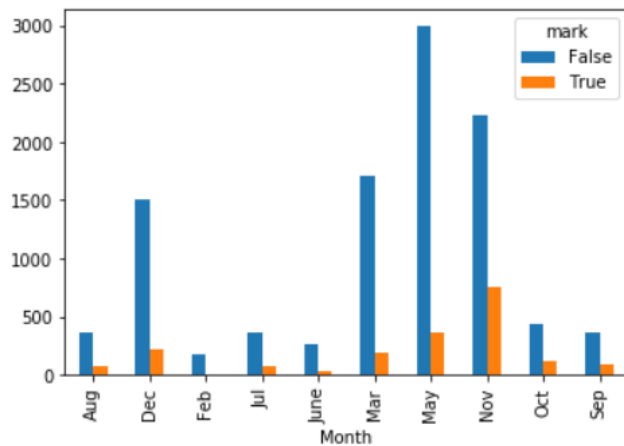


Figure 33: Barplot Month vs Revenue

#### Business Insights

- 1) Most of the customers visited the website in the month of May and Nov with pretty much same conversion to Revenue ratio.
- 2) Highest number of purchases have been made in the month of November.
- 3) Website is least visited by customers in the month of Feb and also the purchase is negligible as compared to other months.

### 6.1.10 REGION VS REVENUE

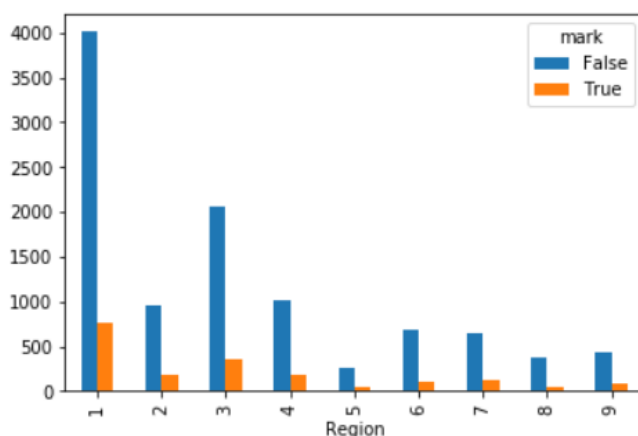


Figure 34: Barplot Region vs Revenue

#### Business Insights

- 1) Most revenue is generated from Region 1 followed by region 3.
- 2) Regions 5 and 8 have negligible contribution in revenue generation.

### 6.1.11 TRAFFIC TYPE VS REVENUE

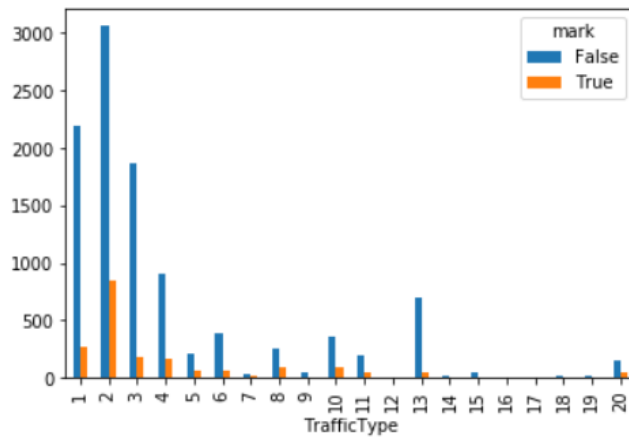


Figure 35: Barplot Traffic Type vs Revenue

#### Business Insights

- 1) Most revenue is generated by traffic type 2, followed by 1 and 3.
- 2) Traffic type 14-20 have very less contribution to Revenue.

### 6.1.12 OPERATING SYSTEM VS REVENUE

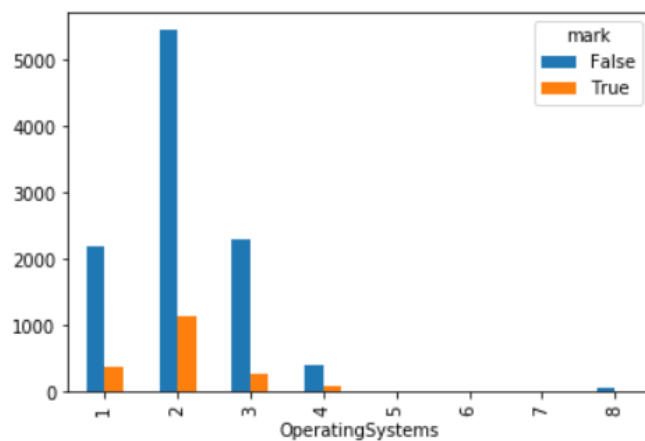


Figure 36: Barplot OS vs Revenue

#### Business Insights

- 1) People using Operating System type 2 generate majority of revenue.

### 6.1.13 BROWSER VS REVENUE

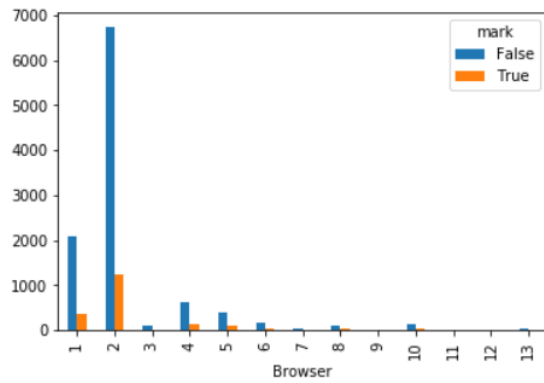


Figure 37: Barplot Browser vs Revenue

#### Business Insights

- 1) Most revenue is generated from browser 2 and 1. No contribution from browsers 3,9,11,12,13.

### 6.1.14 TRAFFIC TYPE COMING ON WEBSITE

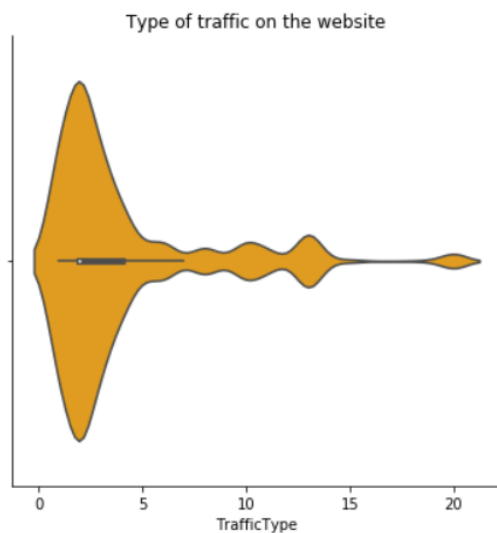


Figure 38: Catplot Traffic type hitting the website

#### Business Insights

- 1) Most of the traffic hitting the website belongs to type 1-5 bracket.

### 6.1.15 EFFECT OF SPECIAL DAY ON WEBSITE

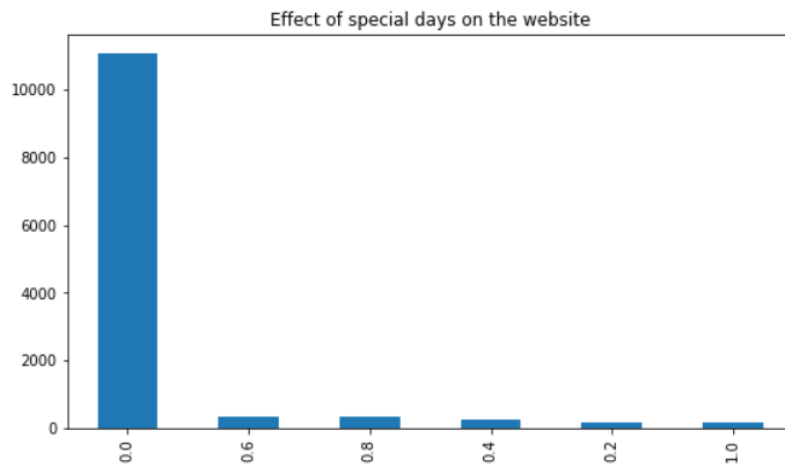


Figure 39: Barplot Effect of special day on Revenue

#### Business Insights

- 1) Majority of the interactions or session generation happened on Non-Special Days. There appears to be no affinity for Special Days to Revenue generation.

### 6.1.16 MONTHLY VISITS TO WEBSITE BY USERS

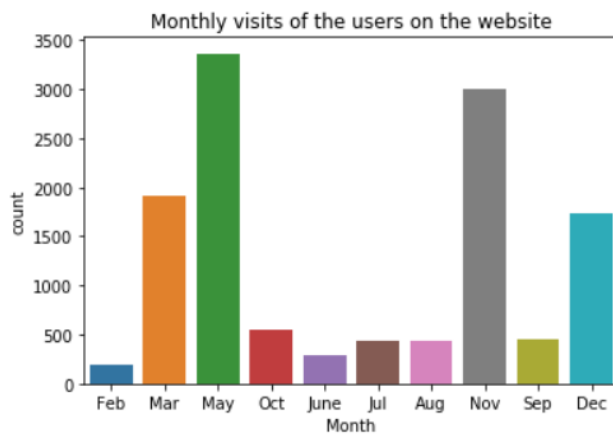


Figure 40: Barplot monthly visits to website

#### Business Insights

- 1) People visited the website in the month of May and Nov the most.

### 6.1.17 SHARE OF VISITOR TYPES W.R.T REVENUE

VisitorType	New_Visitor	Other	Returning_Visitor
Revenue			
False	1272	69	9081
True	422	16	1470

Figure 41: No of visitor types w.r.t. Revenue

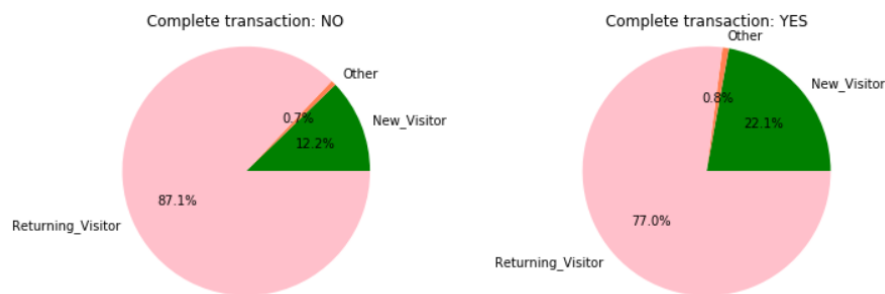


Figure 42: Share of Visitor types w.r.t. Revenue

#### Business Insights

- 1) New visitors take up a larger percentage in those who complete purchase.
- 2) There are more returning visitors among those who do not complete the purchase.

### 6.1.18 SHARE OF COMPLETED PURCHASES

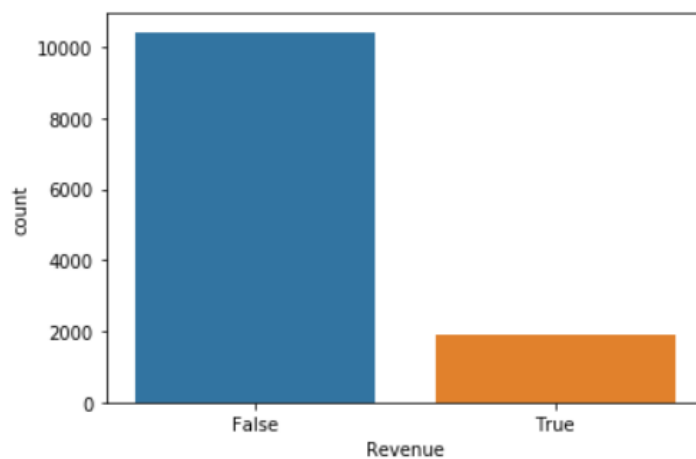


Figure 43: Barplot completed purchases w.r.t. Revenue

#### Business Insights

- 1) Roughly around 20% of the sessions only resulted in completed purchase.
- 2) It's clear that this is imbalanced dataset, we might have to do something about it before model building.



## 6.1.19 MONTH VS SPECIAL DAYS

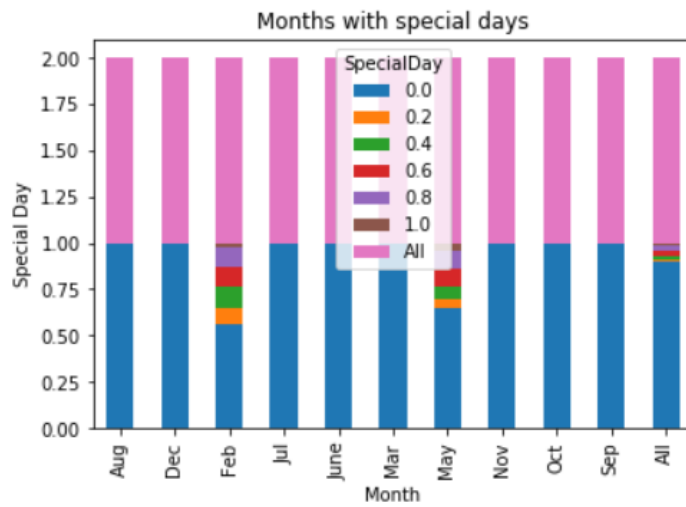


Figure 44: Month and closeness to Special Days

### Business Insights

- 1) Special day only accounts for in the month of February and May.

## 6.2 MULTIVARIATE ANALYSIS

### 6.2.1 MULTIVARIATE ANALYSIS -1

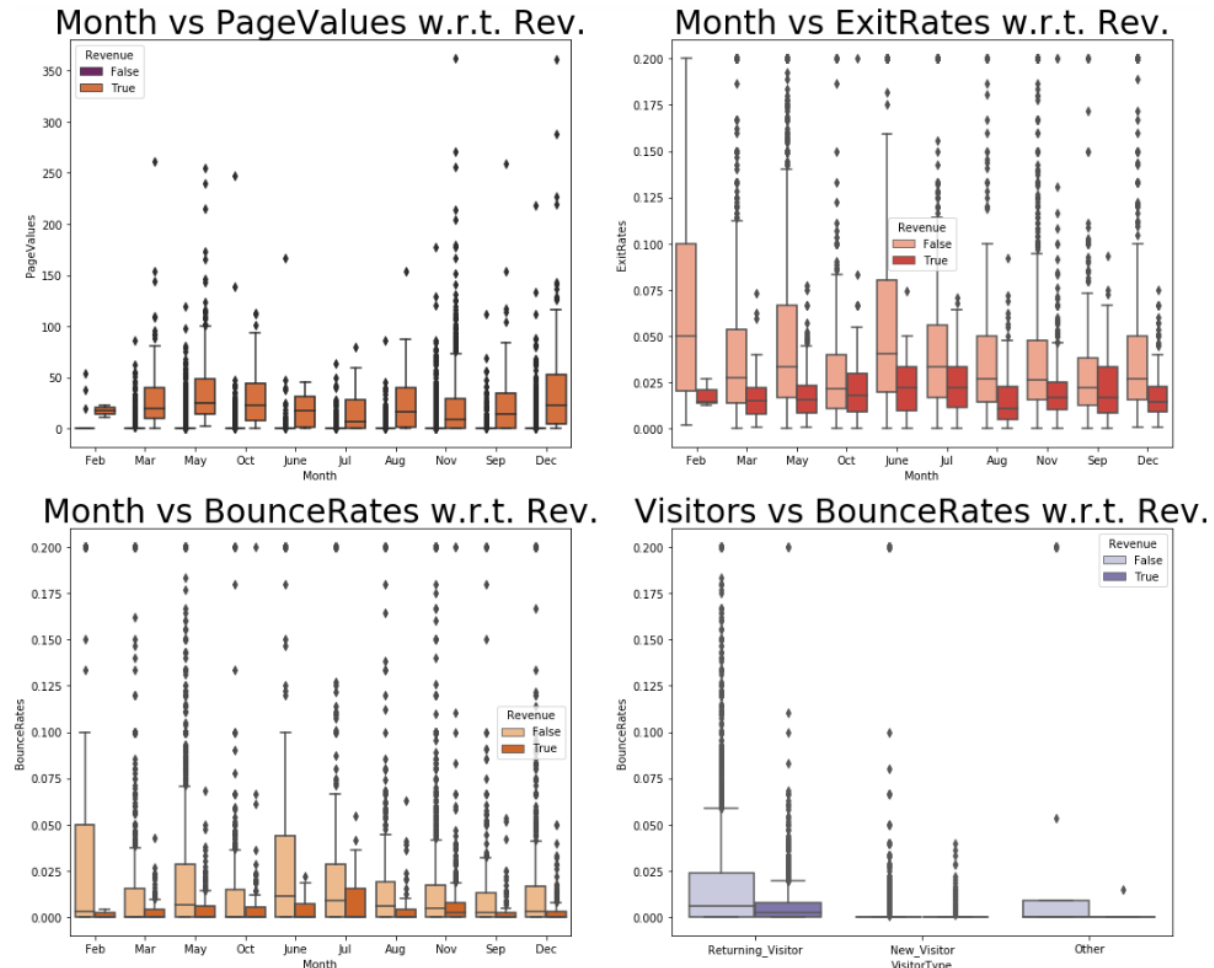


Figure 45: Multivariate analysis part 1

#### Business Insights

- 1) It is evident that if page values are less than or around 50 there is a chance of revenue will be generated, however page values being less than 50 is no guarantee for revenue generation.
- 2) For revenue generation exit rates should be on the lower side i.e. 0.025 and below.
- 3) Higher exit rates are sure shot recipe for revenue loss.
- 4) In case of bounce rates, they should be negligible if at all, way less the 0.0125.
- 5) Company should put all efforts towards minimizing exit and bounce rates.
- 6) If possible, try to minimize bounce rates for only returning visitors as they are the ones generating majority of revenue.

## 6.2.2 MULTIVARIATE ANALYSIS -2

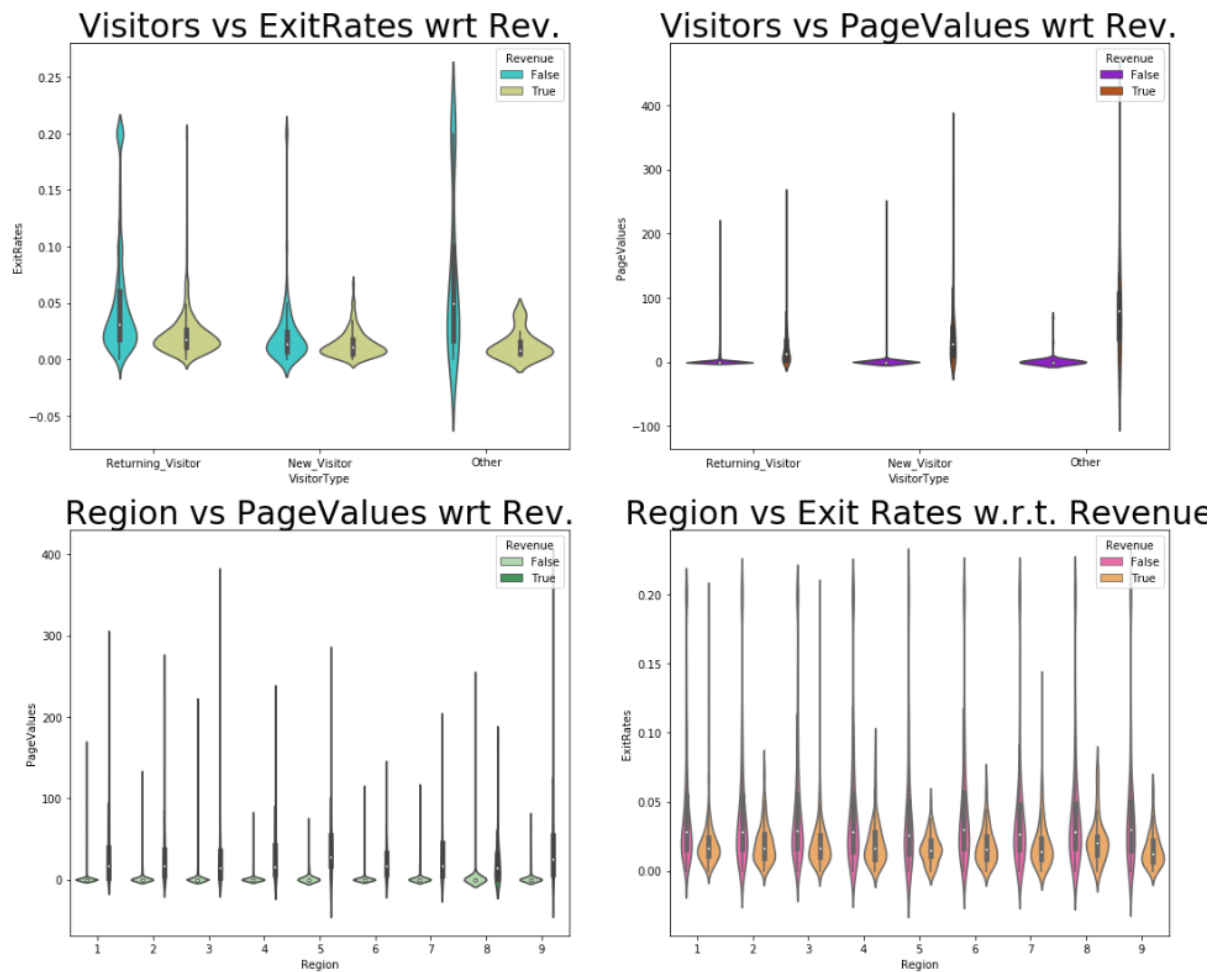


Figure 46: Multivariate analysis part 2

### Business Insights

- 1) Distributions for Exit rates for all 3 visitor types were close to normal when there was a completed purchase.
- 2) In case of non-purchases, distributions were left skewed and there was moderate probability for extreme Exit rates.
- 3) Distributions for Page values follow similar trend irrespective of visitor type.
- 4) All regions follow similar distributions of Page values in case of revenue generation.
- 5) Revenue generation with respect to most attributes follow low probability distributions.
- 6) Most number of regions have Exit rates densely populated if there is a revenue generation otherwise it's just low probability distribution.

## 6.2.3 MULTIVARIATE ANALYSIS -3

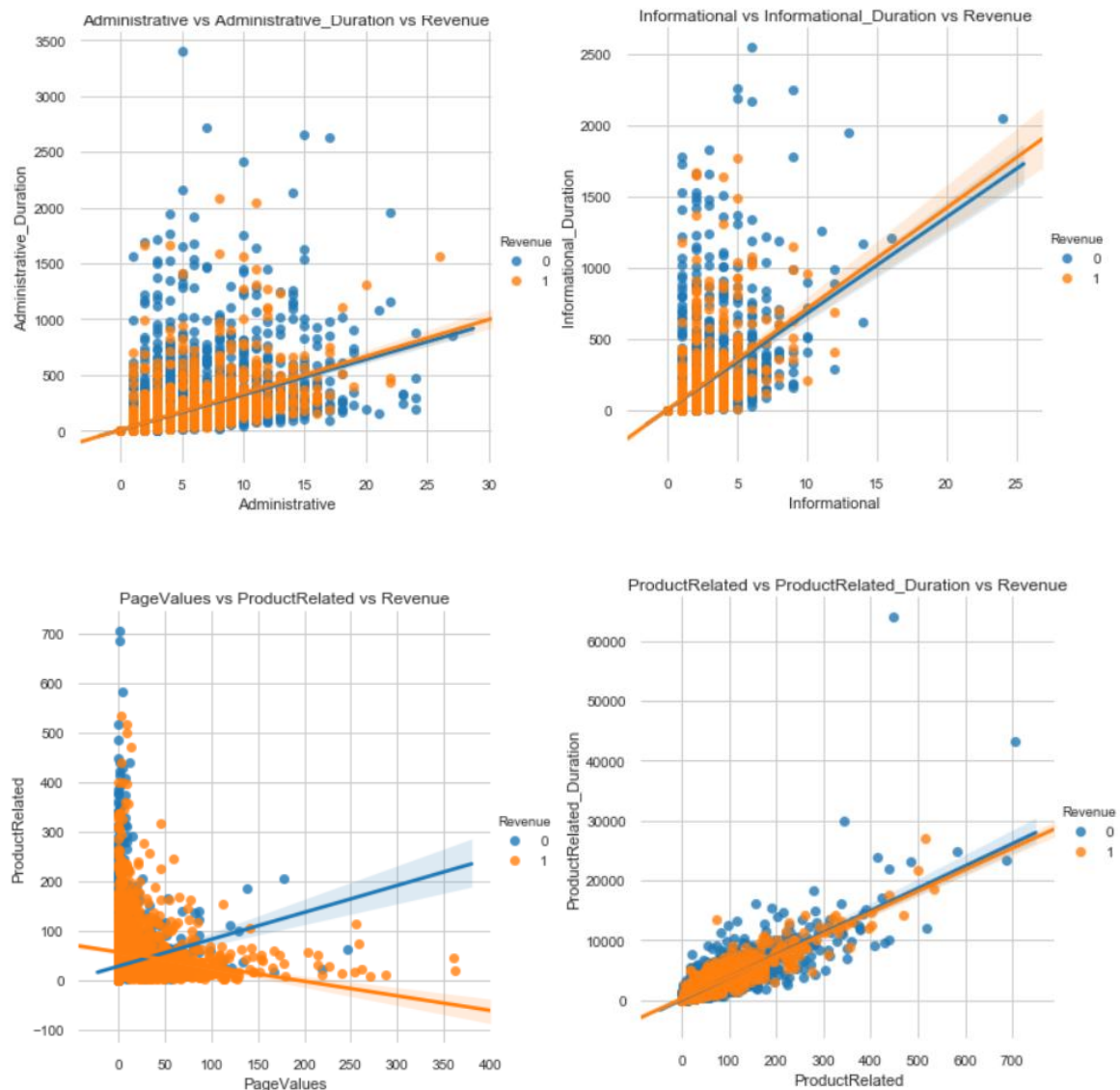


Figure 47: Multivariate analysis part 3

### Business Insights

- 1) Administrative and Administrative Duration, Informational and Informational Duration, Product Related and Product Related Duration are positively correlated.
- 2) On Administrative pages 2 to 15 (probably pages like login, logout, password recovery, profile, email wish list etc.), visitors have spent more than 500 seconds (approx. 8 minutes) which is generally quite higher than normal. It suggests that visitors are having trouble logging in or it's taking too much time to process the request.
- 3) Even though customers/visitors have spent a large amount of time on Product related pages, but the revenue generation is very low. There are certain outliers who spent more than 30000 seconds (approx. 8 Hours) but still didn't make any transaction (possibly screen left idle).
- 4) With increasing Page Values, the revenue generation is more. There are certain pages which have very less page values which need to be improved in order to generate revenue.

## 6.3 CORRELATION MATRIX

From the below correlation matrix, we find out various highly correlated variables in the dataset and look to remedy that before full-fledged model building process. Note that we are checking correlation on training set only. We will be using this approach where we check everything on training and propagate the changes over to testing set. This is to avoid overfitting.

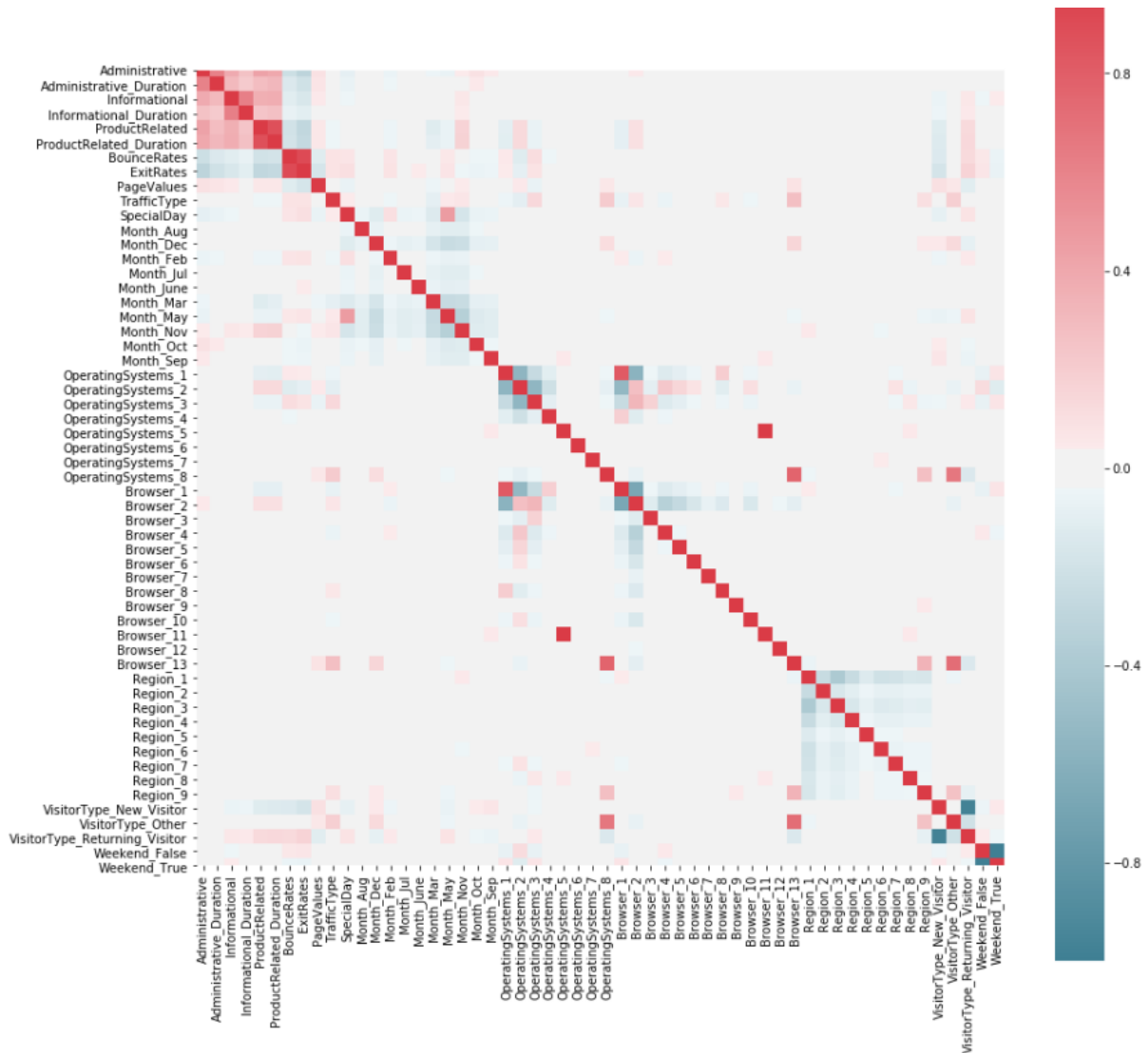


Figure 48: Correlation Matrix

Delving further into correlation we use two python coding approaches – Brute force wherein we loop through all features and remove the first feature that is correlated with anything else and so on.

```
{'Browser_1',
'Browser_11',
'ExitRates',
'ProductRelated_Duration',
'VisitorType_Returning_Visitor',
'Weekend_True'}
```

Figure 49: Correlated features to be dropped (Brute force approach)

Second approach looks to identify groups of highly correlated features. And then, we can make further investigation within these groups to decide which feature we keep and which one we remove.

	feature1	feature2	corr
0	VisitorType_Returning_Visitor	VisitorType_New_Visitor	0.971133
1	VisitorType_New_Visitor	VisitorType_Returning_Visitor	0.971133
2	ExitRates	BounceRates	0.912278
3	BounceRates	ExitRates	0.912278
4	ProductRelated	ProductRelated_Duration	0.876104
5	ProductRelated_Duration	ProductRelated	0.876104
6	Browser_1	OperatingSystems_1	0.824153
7	OperatingSystems_1	Browser_1	0.824153

Figure 50: Correlated features to be dropped (Second approach)

For both approaches, we have taken 0.8 as correlation cut-off. Checking the two approaches, we took a decision to drop **Browser\_1**, **BounceRates**, **ProductRelated** and **VisitorType\_Returning\_Visitor**. After dropping these columns, we don't see any significant correlation left in the training set.

## 7. FEATURE SELECTION

This is a very important step towards final model building. After all the preprocessing and feature engineering we are left with around 50 features which are way too many from a machine learning's perspective. Feature selection enables ML algorithm to train faster, reduces the complexity of the model, reduces overfitting and increases the accuracy.

Keeping our holistic view in mind, we employ many techniques to select most relevant features. Our idea is to take out features which turn out important in all the techniques. This approach is exhaustive, and we hope it produces best results. The techniques used for feature selection has been listed below. Also given below are most important features according to various techniques.

- Information Gain (Mutual information & SelectKBest)
- Fisher Score (Categorical Variables)
- Univariate ROC\_AUC
- Step Forward, Step Backward and Exhaustive Feature Selection
- Random forest feature importance
- Random forest recursive feature elimination
- Feature shuffling
- Hybrid recursive feature elimination(XGBoost)
- Hybrid recursive feature addition(XGBoost)
- Gradient boosting importance

InfoGain-Mutual Information	Fisher Score-Chi Square(Only for categorical features)	StepForward Sel	StepBackward Sel	Exhaustive Sel
Features to Keep	Features to Keep (Order high to low)	Features to Keep	Features to Keep	Features to Keep
Administrative	Month_Nov	Administrative	Administrative_Duration	Administrative
Administrative_Duration	VisitorType_New_Visitor	Administrative_Duration	Informational_Duration	Informational_Duration
Informational_Duration	OperatingSystems_3	Informational_Duration	ProductRelated_Duration	ProductRelated_Duration
ProductRelated_Duration	Month_Mar	ProductRelated_Duration	ExitRates	ExitRates
ExitRates	Month_May	ExitRates	PageValues	PageValues
PageValues	Month_Feb	PageValues	SpecialDay	TrafficType
TrafficType	OperatingSystems_2	TrafficType	Month_Dec	SpecialDay
SpecialDay	Month_Oct	Month_Feb	Month_Jul	
Month_Dec	Month_Dec	Month_Jul	Month_May	
Month_Mar	Month_Sep	Month_Mar	Month_Nov	
Month_May	Month_June	Month_May	Month_Oct	
Month_Nov	Browser_3	Month_Nov	Month_Sep	
Month_Oct	Weekend_True	Browser_2	OperatingSystems_1	
OperatingSystems_2	Region_8	Browser_3	OperatingSystems_2	
OperatingSystems_3	Browser_6	Browser_4	OperatingSystems_5	
OperatingSystems_6		Browser_6	OperatingSystems_6	
Browser_4		Browser_8	Browser_7	
Browser_7		Browser_9	Browser_8	
Browser_9		Browser_12	Browser_9	
Browser_10		Browser_13	Browser_10	
Browser_11		Region_3	Browser_11	
Browser_13		Region_4	Browser_12	
Region_2		Region_6	Region_2	
Region_7		VisitorType_Other	Region_4	
Region_8		Weekend_False	Weekend_False	

RF Feature Imp	RF RFE	Feature Shuffling	Hybrid RFE(XGB)	Hybrid RFA(XGB)	GradBood Imp
Features to Keep	Features to Keep	Features to Keep	Features to Keep	Features to Keep	Features to Keep
Administrative	Administrative	PageValues	PageValues	PageValues	Administrative
Administrative_Duration	Administrative_Duration	Month_Nov	Month_Nov	Month_Nov	ProductRelated_Duration
Informational	Informational	ExitRates	VisitorType_New_Visitor	VisitorType_New_Visitor	ExitRates
Informational_Duration	Informational_Duration	VisitorType_New_Visitor	Month_May	Month_May	PageValues
ProductRelated_Duration	ProductRelated_Duration	Month_May	Month_Mar	Month_Mar	Month_Nov
ExitRates	ExitRates	SpecialDay	ProductRelated_Duration	Administrative	
PageValues	PageValues	TrafficType	Administrative_Duration	ProductRelated_Duration	
TrafficType	TrafficType	OperatingSystems_2	TrafficType	Administrative_Duration	
Month_Nov	Month_Dec	Month_Oct	Month_Sep	Weekend_False	
	Month_Mar	OperatingSystems_3	Informational		
	Month_May	Month_Sep	Region_1		
	Month_Nov	Month_Mar	Browser_2		
	OperatingSystems_1	Month_Dec	Region_6		
	OperatingSystems_2	Month_Aug	Browser_6		
	OperatingSystems_3	Region_5			
	Browser_2	Browser_4			
	Browser_4	Region_1			
	Region_1	Month_June			
	Region_2	Region_4			
	Region_3	Month_Feb			
	Region_4	Browser_6			
	Region_6	Browser_12			
	VisitorType_New_Visitor	Month_Jul			
	Weekend_False	Weekend_True			
	Weekend_True	Browser_11			
		Browser_2			
		Region_6			





## 7.1 PRINCIPAL COMPONENT ANALYSIS CHECK

The main idea of principal component analysis (PCA) is to reduce the dimensionality of the data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent.

Here, we have already fixed the correlation in data. By doing a PCA check we can get an idea of how many variables can we take with respect to feature selection which is to be done next.

```
Number of Fetaures: 10 Total Explained Variance: 79.07 %
Number of Fetaures: 15 Total Explained Variance: 89.68 %
Number of Fetaures: 20 Total Explained Variance: 94.67 %
Number of Fetaures: 25 Total Explained Variance: 97.15 %
Number of Fetaures: 30 Total Explained Variance: 98.76 %
Number of Fetaures: 40 Total Explained Variance: 99.94 %
Number of Fetaures: 50 Total Explained Variance: 100.0 %
```

Figure 53: PCA check

From the above result it seems that any number of features between 20-25 are enough to explain the variation in the data while reducing the feature space.

## 8. MODELLING APPROACH

Models will be built using a training dataset and tested on a test dataset, using the various algorithms. We have already performed feature engineering and feature selection. It's a binary classification problem. We will use the below modelling techniques to predict whether online session of a user results into revenue generation or not.

- Logistic Regression
- Naive Bayes
- K Nearest Neighbor
- Support Vector Machine
- AdaBoost
- Gradient Boosting
- Bagging Tree
- Decision Tree
- Random Forest
- XGBoost
- Voting Classifier
- Stacking

Models will be evaluated using Accuracy, Precision, Recall, F1 Score, AUC ROC with K-Fold Cross Validation. Model with the best performance (F1 Score) will be used for classifying revenue generating user sessions from non-revenue generating ones.

Since our problem inherently has class imbalance, we will address this with upsampling, SMOTE minority oversampling and SMOTEEN over and under sampling and check the performance of models.

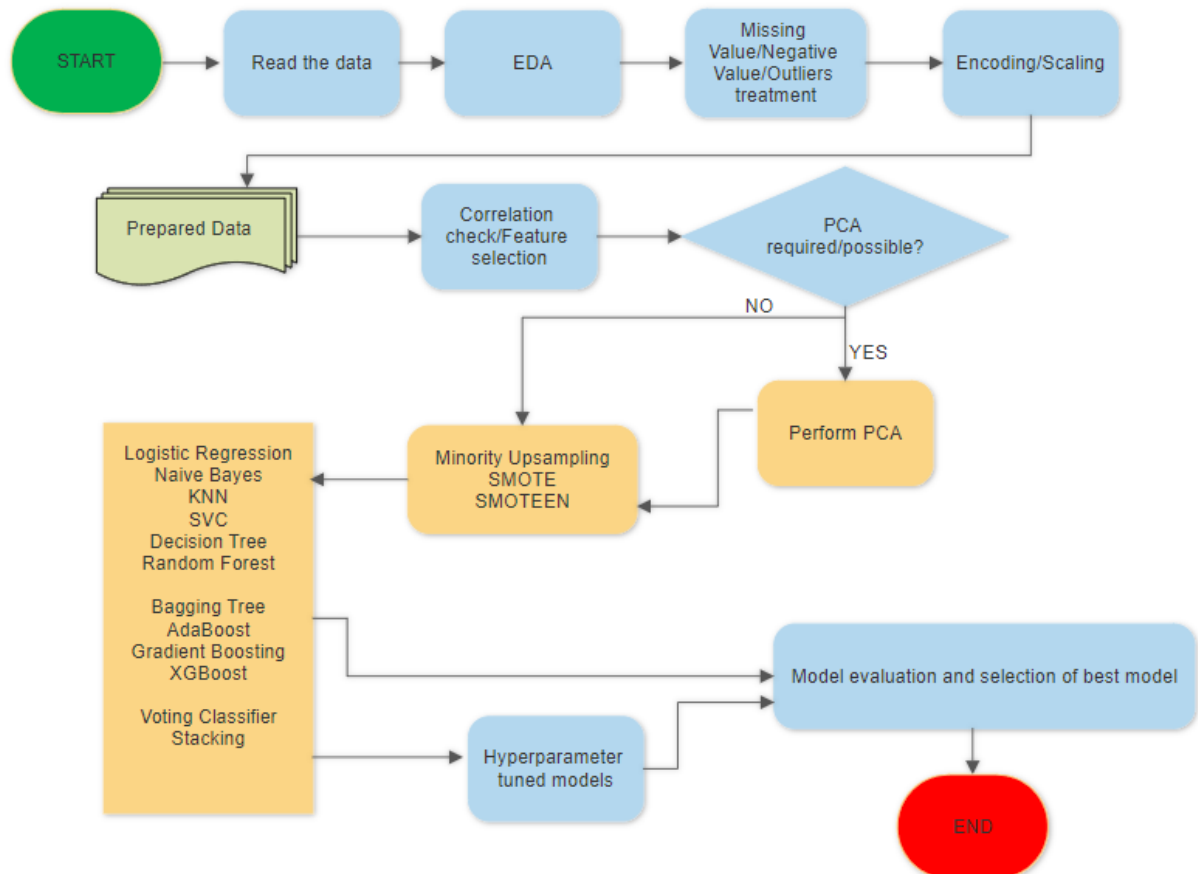


Figure 54: Modelling approach

## 8.1 TRAIN TEST SPLIT

We separate independent and dependent variables. We split the data into 70:30 train test ratio. We check the number of records in train and test.

```

Shape of x_train : (8631, 23)
Shape of y_train : (8631,)
Shape of x_test : (3699, 23)
Shape of y_test : (3699,)
  
```

Figure 55: Shape of train and test data

Feature variables that we have selected for modelling are: -

- PageValues
- Month\_Nov
- ProductRelated\_Duration
- Month\_May
- ExitRates
- TrafficType

- Month\_Mar
- Administrative\_Duration
- Administrative
- Informational\_Duration
- VisitorType\_New\_Visitor
- OperatingSystems\_2
- Month\_Dec
- Weekend\_False
- SpecialDay
- Region\_6
- Region\_4
- OperatingSystems\_3
- Month\_Sep
- Month\_Oct
- Browser\_6
- Browser\_4
- Browser\_2

Target variable is Revenue.

```
0    0.845209
1    0.154791
```

Figure 56: Proportion of values for Revenue

## 8.2 HANDLING CLASS IMBALANCE

This is a major challenge in our project. The minority class percentage is only around 15 in the whole dataset. Therefore, we tried three different techniques to counter this. Note that we have applied these methods only on the training data as this is the ideal way. Let's look at these one by one and see how proportion of both classes is affected by this. We will be creating models with all these methods.

### Random oversampling

Random oversampling involves randomly selecting examples from the minority class (Revenue = 0), with replacement, setting the number of samples to match that of the majority class (Revenue = 1) and adding them to the training dataset. Ratio of two classes becomes 1:1.

```
True      10422
False     10422
Name: Revenue,
```

Figure 57: Random oversampling minority class

### Synthetic Minority Oversampling Technique (SMOTE)

SMOTE works by selecting examples from the minority class to synthesize new examples. These new synthetic examples are created by slightly perturbing feature values. We can play with sampling strategy to decide the ratio of majority and minority class. Below is SMOTE applied in our case

```
0    0.625
1    0.375
Name: Revenue,
```

Figure 58: SMOTE oversampling

### Oversampling and Undersampling SMOTE and Edited Nearest Neighbors (SMOTEEN)

This method is a combination of first oversampling of minority class using SMOTE and then undersampling the majority class using ENN to reduce the number of overall examples. Like SMOTE, sampling strategy can be adjusted to fix the ratio of majority and minority class. Below is SMOTEEN applied in our case.

```
0    0.692373
1    0.307627
Name: Revenue,
```

Figure 59: SMOTEEN oversampling and undersampling

## 8.3 MODELS ON BASE DATA AND COMPARISON

Below is a comparison of models built on categorical encoded and scaled data. This is without applying any oversampling technique.

MODEL	Train Accuracy	Test Accuracy	True Positive (Non-revenue)	True Negative (Revenue)	CV Accuracy	CV Precision	CV Recall	CV F1 Score	Remarks
Logistic Regression	89	88	3059	204	88	74	38	50	
Random Forest	99	90	3038	281	89	74	48	58	
Decision Tree	100	86	2880	315	86	54	56	55	Overfit
Naïve Bayes	36	36	802	547	35	19	96	31	Worst
KNN	90	87	3005	211	88	68	38	49	
SVM	88	88	3072	179	88	77	33	46	
AdaBoost	90	89	2972	316	89	68	58	62	
XGBoost	92	90	2979	347	91	75	62	68	Best
Gradient Boosting	92	89	2971	338	90	73	61	66	

Table 2: Comparison of Models (Base data)

### Technical Summary

- These models were run with all features.
- Models were trained on original data i.e. no under or over sampling has been done.
- Decision Tree model overfitted as training accuracy is too high than the testing accuracy.
- Naïve Bayes model performed the worst as it has least testing accuracy as well as lowest F1 Score.
- Boosting models fared better than single classifiers which is to be expected.
- XGBoost comes out to be the best model with highest F1 Score, pretty good training/testing accuracy and cross validation accuracy of 91. It is also among the Top 2 to be able to predict largest number of Revenue generating samples (Revenue=1) - 347.

## 8.4 MODELS WITH FEATURE SELECTION, UPSAMPLING AND HYPERPARAMETER TUNING AND COMPARISON

Below is a comparison of models built on categorical encoded and scaled data. Feature selection is done. Random upsampling of minority data is done. Built models were hyperparameter tuned to produce better results.

MODEL	Train Accuracy	Test Accuracy	True Positive (Non-revenue)	True Negative (Revenue)	CV Accuracy	CV Precision	CV Recall	CV F1 Score	CV ROC AUC	Remarks
Logistic Regression	72	72	2117	536	82	85	76	80	90	
Naïve Bayes	70	71	2120	503	75	74	77	76	83	
KNN	83	83	2522	560	89	83	99	90	90	
SVM	87	87	2736	469	95	92	98	95	95	
AdaBoost	36	38	859	559	86	87	84	85	86	
Gradient Boosting	39	40	972	501	93	90	97	93	93	
Bagging Tree	49	50	1340	519	95	92	99	95	95	
Decision Tree	32	33	666	543	70	68	80	73	70	Worst
Random Forest	52	54	1449	548	95	92	99	96	95	
Voting Classifier	81	80	2426	549	96	93	99	96	96	Best
XGBoost	39	41	998	501	93	90	97	93	93	

Table 3: Comparison of Models (Feature selected, random upsampled and hyperparameter tuned)

### Technical Summary

- These models were run with selected 23 features.
- All the models were hyperparameter tuned to get the best set of hyperparameters corresponding to the respective classifier.
- Models were trained on upsampled data i.e. minority class were randomly upsampled.
- None of the models performed bad if we talk just in terms of overfitting.
- Decision Tree model performed the worst as it has least testing accuracy as well as lowest F1 Score.
- Voting Classifier comes out to be the best model with highest F1 Score, pretty good training/testing accuracy and cross validation accuracy of 96. It is also among the Top 3 to be able to predict largest number of Revenue generating samples (Revenue=1) - 549.
- Voting Classifier trained on an ensemble of models with each having best set of hyperparameters (given below in next point) – Logistic Regression, SVM, KNN, Naïve Bayes, Decision Tree and Random Forest.
- Hyperparameters are
  - **Logistic Regression** - C=1.0,solver='newton-cg'
  - **SVM** - C=1000, gamma=1,kernel='rbf'
  - **KNN** - algorithm='auto', leaf\_size = 30, n\_jobs = -1, n\_neighbors = 6, weights = 'distance'
  - **Decision Tree** - criterion = 'gini'

- **Random Forest**  
**n\_estimators=112,min\_samples\_split=2,min\_samples\_leaf=1,max\_features='sqrt', max\_depth=None,bootstrap=True**

- Voting classifier here used hard voting i.e. the predicted output class is a class with the highest majority of votes i.e. the class which had the highest probability of being predicted by each of the classifiers.

## 8.5 MODELS WITH FEATURE SELECTION, SMOTE AND HYPERPARAMETER TUNING AND COMPARISON

Below is a comparison of models built on categorical encoded and scaled data. Feature selection is done. Oversampling of minority data is done using SMOTE. Built models were hyperparameter tuned to produce better results.

MODEL	Train Accuracy	Test Accuracy	True Positive (Non-revenue)	True Negative (Revenue)	CV Accuracy	CV Precision	CV Recall	CV F1 Score	CV ROC AUC	Remarks
Logistic Regression	83	88	2904	368	83	85	66	74	80	
Naïve Bayes	75	73	2248	445	75	63	79	70	76	
KNN	100	81	2647	358	90	80	96	87	91	Overfit
SVM	99	85	2959	186	94	91	94	92	94	Overfit
AdaBoost	92	88	2916	351	90	88	85	86	89	
Gradient Boosting	97	89	2937	366	92	90	89	89	91	
Bagging Tree	87	87	2799	426	86	84	80	81	85	
Decision Tree	78	81	2746	268	76	72	58	64	72	
Random Forest	100	89	2910	376	93	89	91	90	92	
Voting Classifier	97	88	2948	321	91	90	84	87	89	
XGBoost	91	88	2874	402	91	88	87	87	90	Best

Table 4: Comparison of Models (Feature selected, SMOTE and hyperparameter tuned)

### Technical Summary

- These models were run with selected 23 features.
- All the models were hyperparameter tuned to get the best set of hyperparameters corresponding to the respective classifier.
- Models were trained on SMOTE oversampled data i.e. new minority class examples were synthetically generated to increase the number of minority class samples.
- KNN and SVM models overfitted as training accuracy is too high than the testing accuracy.
- XGBoost comes out to be the best model with F1 Score of 87, pretty good training/testing accuracy and cross validation accuracy of 91. It is also among the Top 3 to be able to predict largest number of Revenue generating samples (Revenue=1) - 402.
- Best Hyperparameters for XGBoost are **learning\_rate=0.1, n\_estimators=200, n\_jobs=-1**.

## 8.6 MODELS WITH FEATURE SELECTION, SMOTEEN AND HYPERPARAMETER TUNING AND COMPARISON

Below is a comparison of models built on categorical encoded and scaled data. Feature selection is done. Oversampling of minority data is done using SMOTE and undersampling of majority data using ENN. Built models were hyperparameter tuned to produce better results.

MODEL	Train Accuracy	Test Accuracy	True Positive (Non-revenue)	True Negative (Revenue)	CV Accuracy	CV Precision	CV Recall	CV F1 Score	CV ROC AUC	Remarks
Logistic Regression	91	87	2817	417	91	93	76	84	87	
Naïve Bayes	87	77	2403	472	87	75	84	79	86	
KNN	100	80	2551	393	95	94	91	92	94	Overfit
SVM	100	83	2767	318	96	97	90	93	95	Overfit
AdaBoost	94	88	2833	422	93	92	84	87	90	
Gradient Boosting	100	87	2795	437	94	92	89	90	92	
Bagging Tree	91	88	2827	413	91	91	78	84	87	Close to best
Decision Tree	79	81	2747	254	79	70	57	62	73	
Random Forest	100	88	2796	437	95	92	90	91	93	
Voting Classifier	99	85	2704	437	95	93	90	92	94	
XGBoost	95	88	2809	428	94	92	87	89	92	Best

Table 5: Comparison of Models (Feature selected, SMOTEEN and hyperparameter tuned)

### Technical Summary

- These models were run with selected 23 features.
- All the models were hyperparameter tuned to get the best set of hyperparameters corresponding to the respective classifier.
- Models were trained on SMOTEEN oversampled and undersampled data i.e. minority examples oversampled with SMOTE and majority examples undersampled with ENN.
- KNN and SVM models overfitted as training accuracy is too high than the testing accuracy.
- XGBoost comes out to be the best model with F1 Score of 89, pretty good training/testing accuracy and cross validation accuracy of 94. It is also among the Top 3 to be able to predict largest number of Revenue generating samples (Revenue=1) - 428.
- Best Hyperparameters for XGBoost are **learning\_rate=0.1, n\_estimators=200, n\_jobs=-1**.

## 8.7 STACKING OF MODELS

Since many of the models demonstrate a fair performance, we will try ‘stacking’ the models to see if we can tease out a higher F1 Score and more importantly increase the number of correctly predicted True Negatives i.e. Revenue generating sessions.



Here we have Random Forest, Decision Tree and XGBoost as Base Learners. These are hyperparameter tuned learners. Logistic Regression will be our Meta Learner. Base learners are trained on the normal data. Using the predictions of Base Learners as inputs, the correct responses from the output, we train the Meta Learner.

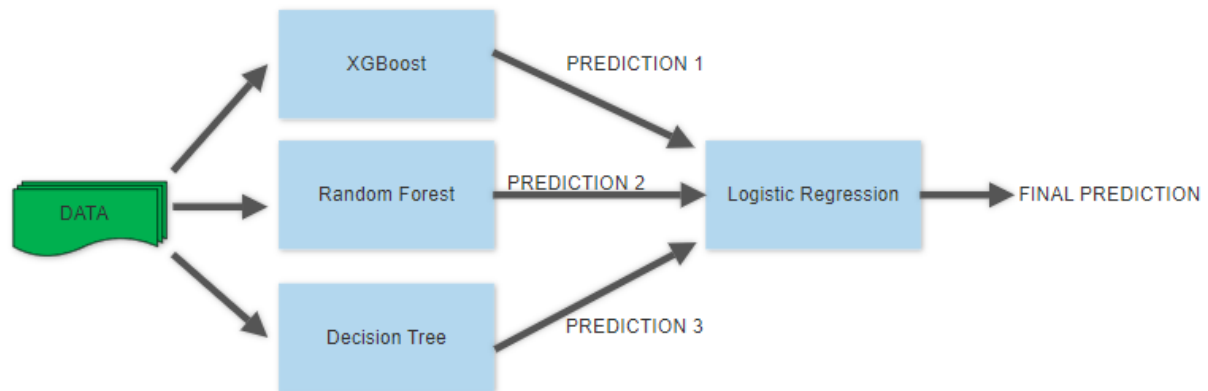


Figure 60: Stacking of Models

We applied this stacking in all three imbalance countering cases. Below is the output for each one of them.

```

[[2841 267]
 [ 10 3136]]
      precision    recall  f1-score   support

     0       1.00      0.91      0.95      3108
     1       0.92      1.00      0.96      3146

 accuracy          0.96
 macro avg          0.96
 weighted avg       0.96

```

Accuracy Stacking= 100.0  
Standard Deviation lr= 0.0

Figure 61: Stacking (Random upsampling)

```

[[2910 217]
 [ 196 376]]
      precision    recall  f1-score   support

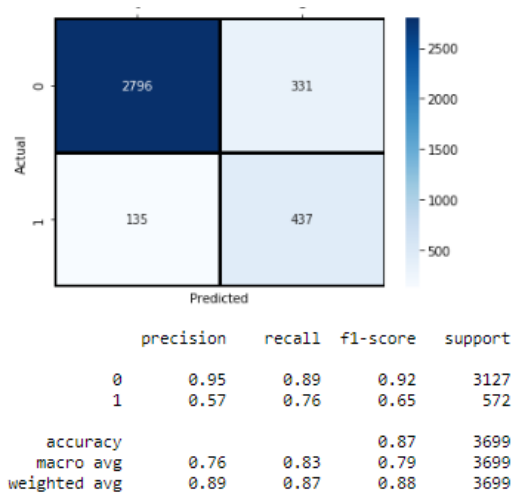
     0       0.94      0.93      0.93      3127
     1       0.63      0.66      0.65       572

 accuracy          0.89
 macro avg          0.79
 weighted avg       0.89

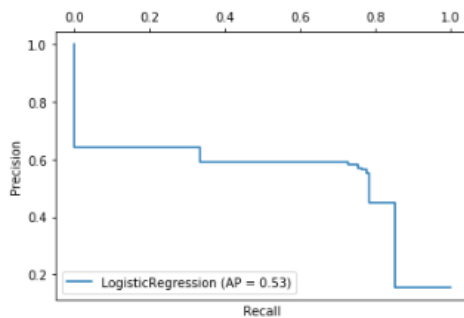
```

CROSS VALIDATION METRICS  
Mean Accuracy : 1.0  
Standard Deviation : 0.0  
Mean precision score : 1.0  
Standard Deviation precision score : 0.0  
Mean recall score : 1.0  
Standard Deviation recall score : 0.0  
Mean f1 score : 1.0  
Standard Deviation f1 score : 0.0  
Mean AUC ROC Score : 1.0  
Standard Deviation AUC ROC Score : 0.0

Figure 62: Stacking (SMOTE)



PRECISION RECALL CURVE



#### CROSS VALIDATION METRICS

Mean Accuracy : 0.9966263606900461  
 Standard Deviation : 0.0028914572953403135  
 Mean precision score : 0.998818848815773  
 Standard Deviation precision score : 0.0018042915069111608  
 Mean recall score : 0.9901960784313726  
 Standard Deviation recall score : 0.009485009115645369  
 Mean f1 score : 0.994463968848945  
 Standard Deviation f1 score : 0.00475736768685506  
 Mean AUC ROC Score : 0.9948370181582706  
 Standard Deviation AUC ROC Score : 0.004705032424319637

Figure 63: Stacking (SMOTEEN)

## Interpretation

In all these cases, we can see that cross-validation accuracy and F1 score are close to 100% without any significant standard deviation. There is also improvement in prediction of True Negatives which we were after. But somehow this doesn't look like generalized predictions. We can see every metric is touching 100% which may be a case of high overfitting.

Due to these reasons we didn't select Stacking as our best model. Instead we will go with our best ensemble classifier i.e. XGBoost as final classifier model.

## 8.8 STRATGEY FOR SELECTING BEST MODEL

Because of so many models it is important to stick to a well-defined strategy of selecting the best model. Ours is given below.

- Remove models which are overfitting i.e. Training accuracy is extreme and/or there is huge difference between Training and Testing accuracy.
- Look for models with decent Training and Testing accuracy but the difference between the two is not huge.
- Look for models which can predict good number of True Negatives i.e. Revenue generating user sessions. Our aim here is to find out the users who are giving Revenue even if that means reducing True Positives i.e. Non-revenue generating user sessions.
- Judge models based on cross validation metrics if we want to check the performance of various models.
- F1 Score which is the harmonic mean of Precision and Recall will be our best indicator as it is a general rule to look for higher F1 Score if our aim is to predict the minority class which here is Revenue generating user sessions more so in imbalanced data.
- After F1 Score if there is any further need to check, we can look for cross validation accuracy and cross validation ROC AUC metrics.

## 8.9 MODELS: A DETAILED ANALYSIS

In this section we will look at all the hyperparameter tuned models. We will analyze their performance metrics as well as draw some observations in detail.

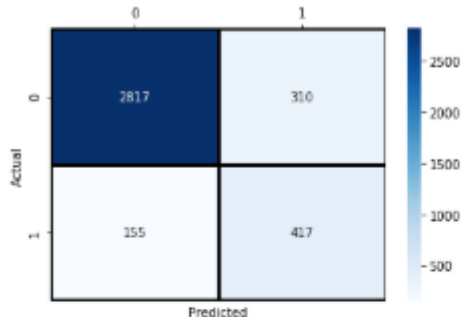
In addition, we have plotted Precision Recall curves because ours is an imbalanced dataset and we are more interested in skill of the model to correctly predict the minority class. Recall is the ability of a classification model to identify all relevant instances and Precision is the ability of a classification model to return only relevant instances. This is the reason we have chosen F1 Score as the accuracy metric.

---

### 8.9.1 LOGISTIC REGRESSION

```
LogisticRegression(C=1, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='newton-cg', tol=0.0001, verbose=0,
warm_start=False)
Training Accuracy : 0.9090251837570792
Testing Accuracy : 0.8742903487429035
ROC AUC Score : 0.8149422132073236
```

\*\*\*\*\*  
CONFUSION MATRIX



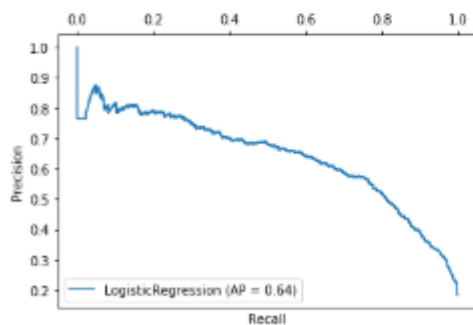
\*\*\*\*\*

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.90	0.92	3127
1	0.57	0.73	0.64	572
accuracy			0.87	3699
macro avg	0.76	0.81	0.78	3699
weighted avg	0.89	0.87	0.88	3699

\*\*\*\*\*

PRECISION RECALL CURVE



\*\*\*\*\*

CROSS VALIDATION METRICS

```
Mean Accuracy : 0.9081805630241109
Standard Deviation : 0.0081642837477491
Mean precision score : 0.9262753546095782
Standard Deviation precision score : 0.014526652244606919
Mean recall score : 0.7622150735294118
Standard Deviation recall score : 0.019963550315585235
Mean f1 score : 0.8361576115250878
Standard Deviation f1 score : 0.015681159295256705
Mean AUC ROC Score : 0.8676203378554497
Standard Deviation AUC ROC Score : 0.011194694811019935
```

Figure 64: Logistic Regression model metrics

## Observations

- 1) Cross validation F1 score of 83%.
- 2) Area under the curve i.e. Average Precision is 0.64. Better the AP better the skill of the model across thresholds.
- 3) Decent model with no overfitting.
- 4) Number of Revenue generating user sessions correctly predicted are 417.
- 5) Best hyperparameters - **C=1,solver='newton-cg'**

## 8.9.2 NAÏVE BAYES

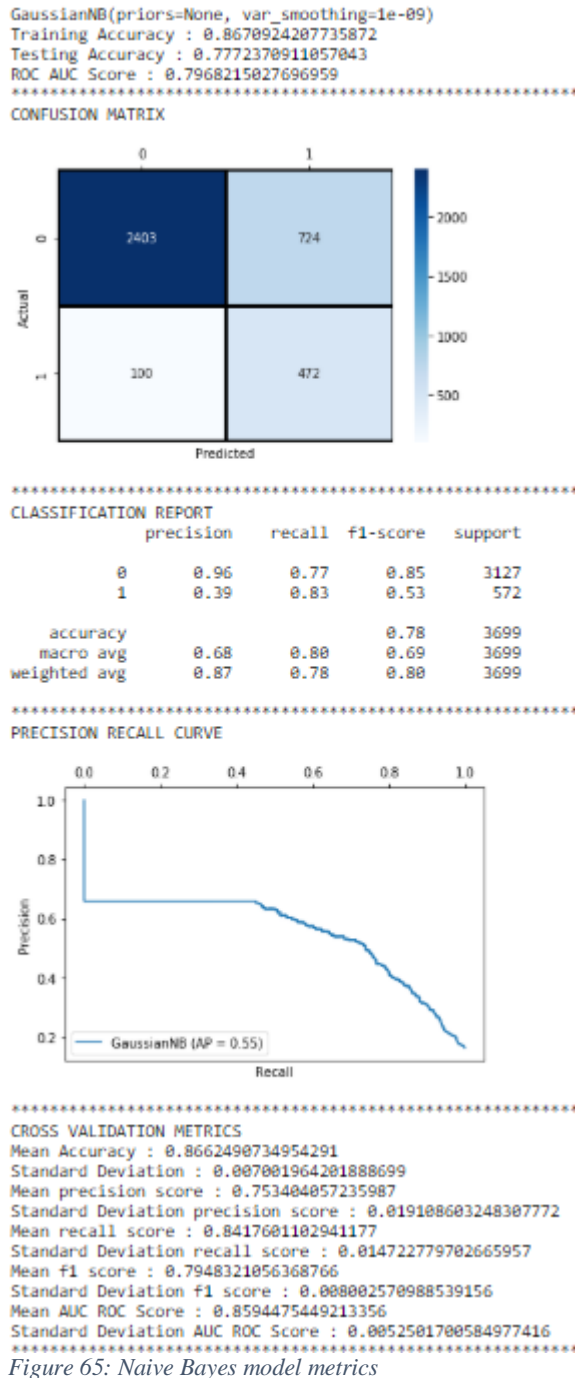


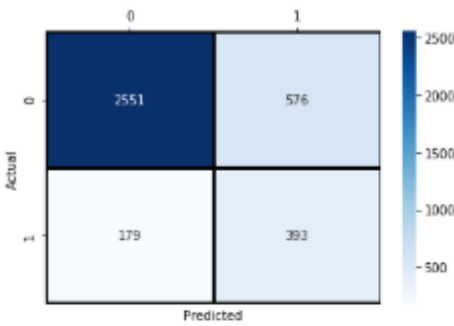
Figure 65: Naive Bayes model metrics

### Observations

- 1) Cross validation F1 score of 79%.
- 2) Area under the curve i.e. Average Precision is 0.55. Skill of the model across thresholds is low.
- 3) Slight Overfitting.
- 4) Number of Revenue generating user sessions correctly predicted are 472.

### 8.9.3 K NEAREST NEIGHBOUR

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                    weights='distance')
Training Accuracy : 1.0
Testing Accuracy : 0.7958987812922411
ROC AUC Score : 0.7514384132868763
*****
CONFUSION MATRIX
```

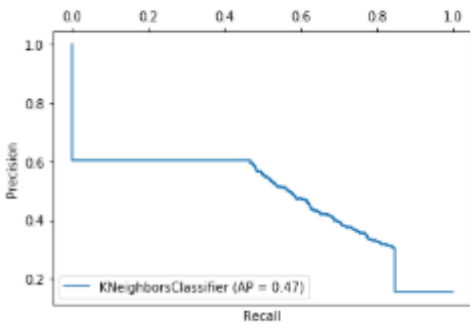


	Actual \ Predicted	0	1
0	2551	576	
1	179	393	

```
*****
CLASSIFICATION REPORT
```

	precision	recall	f1-score	support
0	0.93	0.82	0.87	3127
1	0.41	0.69	0.51	572
accuracy			0.80	3699
macro avg	0.67	0.75	0.69	3699
weighted avg	0.85	0.80	0.82	3699

```
*****
PRECISION RECALL CURVE
```



```
*****
CROSS VALIDATION METRICS
Mean Accuracy : 0.9546954524975657
Standard Deviation : 0.01879514426965914
Mean precision score : 0.941998278324112
Standard Deviation precision score : 0.015414912931459484
Mean recall score : 0.9082720588235293
Standard Deviation recall score : 0.054185527042518514
Mean f1 score : 0.9242871702468729
Standard Deviation f1 score : 0.03311584442999479
Mean AUC ROC Score : 0.9417803257305042
Standard Deviation AUC ROC Score : 0.028490269534295568
*****
```

Figure 66: KNN model metrics

#### Observations

- 1) Cross validation F1 score of 92%.
- 2) Area under the curve i.e. Average Precision is 0.47. Skill of the model across thresholds is poor.
- 3) Highly overfitted.
- 4) Number of Revenue generating user sessions correctly predicted are 393.
- 5) Best hyperparameters - **algorithm='auto',leaf\_size=30,n\_neighbors=5,weights='distance'**

## 8.9.4 SUPPORT VECTOR MACHINE

```
SVC(C=100, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1,
    probability=False, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

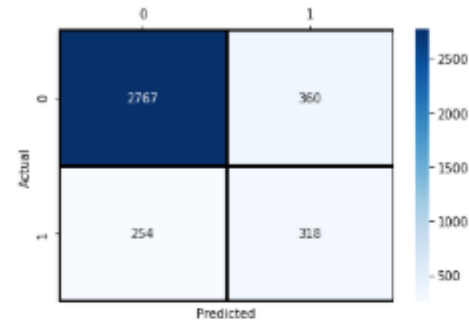
Training Accuracy : 0.9997590071092902

Testing Accuracy : 0.8340091916734252

ROC AUC Score : 0.7204088683941576

\*\*\*\*\*

CONFUSION MATRIX



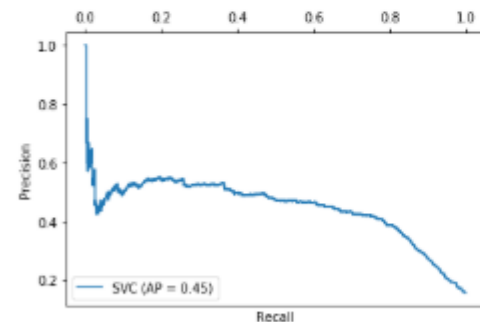
\*\*\*\*\*

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.92	0.88	0.90	3127
1	0.47	0.56	0.51	572
accuracy			0.83	3699
macro avg	0.69	0.72	0.70	3699
weighted avg	0.85	0.83	0.84	3699

\*\*\*\*\*

PRECISION RECALL CURVE



\*\*\*\*\*

CROSS VALIDATION METRICS

Mean Accuracy : 0.9614444751260771  
 Standard Deviation : 0.026630323853240517  
 Mean precision score : 0.969569232897376  
 Standard Deviation precision score : 0.010405055131320014  
 Mean recall score : 0.9023606004901961  
 Standard Deviation recall score : 0.08250006702750098  
 Mean f1 score : 0.9330827128340051  
 Standard Deviation f1 score : 0.04772788647522323  
 Mean AUC ROC Score : 0.9450024484044679  
 Standard Deviation AUC ROC Score : 0.042142002721755396

Figure 67: SVM Model metrics

### Observations

- 1) Cross validation F1 score of 93%.
- 2) Area under the curve i.e. Average Precision is 0.45. Skill of the model across thresholds is poor.
- 3) Overfitted model.
- 4) Number of Revenue generating user sessions correctly predicted are 318.
- 5) Best hyperparameters - **C=100,gamma=1,kernel='rbf'**

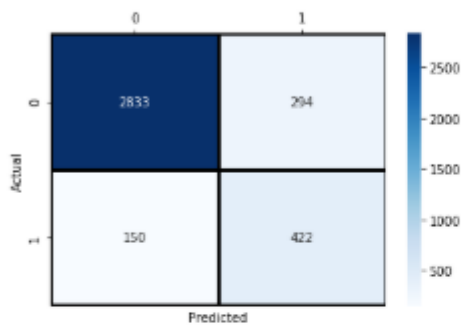
### 8.9.5 ADA BOOST

```
AdaBoostClassifier(algorithm='SAMME.R',
                   base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
                                                           class_weight=None,
                                                           criterion='gini',
                                                           max_depth=1,
                                                           max_features=None,
                                                           max_leaf_nodes=None,
                                                           min_impurity_decrease=0.0,
                                                           min_impurity_split=None,
                                                           min_samples_leaf=1,
                                                           min_samples_split=2,
                                                           min_weight_fraction_leaf=0.0,
                                                           presort='deprecated',
                                                           random_state=None,
                                                           splitter='best'),
                   learning_rate=1, n_estimators=500, random_state=None)
```

Training Accuracy : 0.9383058199783106  
 Testing Accuracy : 0.8799675587996756  
 ROC AUC Score : 0.8218712052258583

\*\*\*\*\*

CONFUSION MATRIX



	0	1
Actual 0	2833	294
Actual 1	150	422

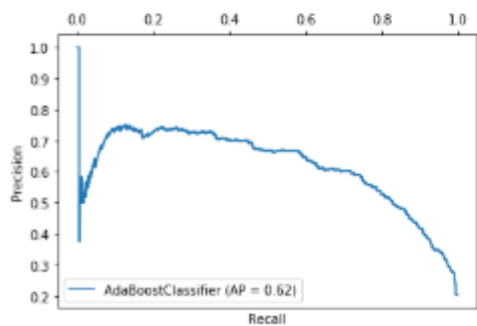
\*\*\*\*\*

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.91	0.93	3127
1	0.59	0.74	0.66	572
accuracy			0.88	3699
macro avg	0.77	0.82	0.79	3699
weighted avg	0.89	0.88	0.89	3699

\*\*\*\*\*

PRECISION RECALL CURVE



\*\*\*\*\*

CROSS VALIDATION METRICS

Mean Accuracy : 0.927945993866903  
 Standard Deviation : 0.03176176780285735  
 Mean precision score : 0.9175665279200377  
 Standard Deviation precision score : 0.02572816489800013  
 Mean recall score : 0.8400658700980392  
 Standard Deviation recall score : 0.09447943353021197  
 Mean f1 score : 0.8749883723365575  
 Standard Deviation f1 score : 0.05946759187310221  
 Mean AUC ROC Score : 0.9035013489257049  
 Standard Deviation AUC ROC Score : 0.04899254828451297

\*\*\*\*\*

Figure 68: AdaBoost model metrics



## Observations

- Cross validation F1 score of 87%.
- Area under the curve i.e. Average Precision is 0.62. Skill of the model across thresholds is OK.
- No overfitting. Good model.
- Number of Revenue generating user sessions correctly predicted are 422.
- Best hyperparameters - **base\_estimator=DecisionTreeClassifier(ccp\_alpha=0.0, class\_weight=None, criterion='gini',max\_depth=1, max\_features=None, max\_leaf\_nodes=None,min\_impurity\_decrease=0.0, min\_impurity\_split=None,min\_samples\_leaf=1, min\_samples\_split=2,min\_weight\_fraction\_leaf=0.0, presort='deprecated',random\_state=None, splitter='best'),learning\_rate=1, n\_estimators=500**

### 8.9.6 GRADIENT BOOSTING

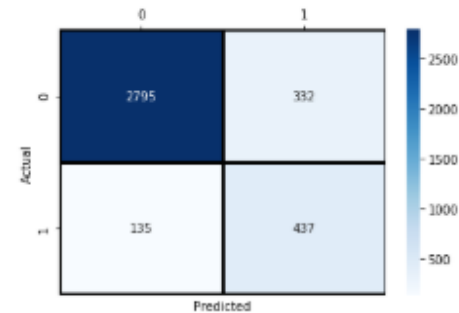
```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=5,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=500,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=None, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)
```

Training Accuracy : 0.9985540426557417

Testing Accuracy : 0.8737496620708299

ROC AUC Score : 0.8289069820489712

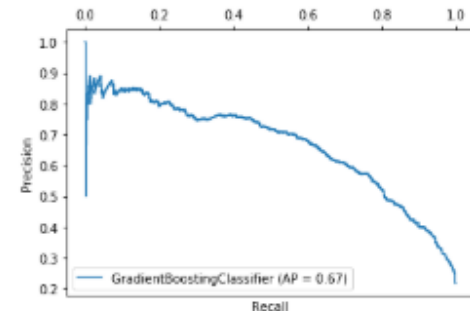
\*\*\*\*\*  
CONFUSION MATRIX



\*\*\*\*\*  
CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.89	0.92	3127
1	0.57	0.76	0.65	572
accuracy				0.87
macro avg	0.76	0.83	0.79	3699
weighted avg	0.89	0.87	0.88	3699

\*\*\*\*\*  
PRECISION RECALL CURVE



CROSS VALIDATION METRICS

Mean Accuracy : 0.9397542401209179

Standard Deviation : 0.02633198059307131

Mean precision score : 0.9150449462369463

Standard Deviation precision score : 0.021412404316068726

Mean recall score : 0.8859145220588236

Standard Deviation recall score : 0.08119188402857638

Mean f1 score : 0.8986411838514774

Standard Deviation f1 score : 0.047582915304086496

Mean AUC ROC Score : 0.9247703196568924

Standard Deviation AUC ROC Score : 0.041311258273904534

\*\*\*\*\*  
Figure 69: Gradient Boosting model metrics

#### Observations

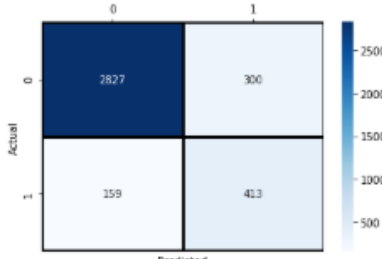
- Cross validation F1 score of ~90%.
- Area under the curve i.e. Average Precision is 0.67. Skill of the model across thresholds is good.
- Slight overfitting.
- Number of Revenue generating user sessions correctly predicted are 437.
- Best hyperparameters - **learning\_rate=0.1, n\_estimators=500, max\_depth=5**

## 8.9.7 BAGGING TREE

```

BaggingClassifier(base_estimator=DecisionTreeClassifier(ccp_alpha=0.0,
class_weight=None,
criterion='gini',
max_depth=None,
max_features=None,
max_leaf_nodes=None,
min_impurity_decrease=0.0,
min_impurity_split=None,
min_samples_leaf=1,
min_samples_split=2,
min_weight_fraction_leaf=0.0,
presort='deprecated',
random_state=1,
splitter='best'),
bootstrap=True, bootstrap_features=False, max_features=23,
max_samples=20, n_estimators=50, n_jobs=None, oob_score=False,
random_state=None, verbose=0, warm_start=False)
Training Accuracy : 0.9073382335221111
Testing Accuracy : 0.8759124087591241
ROC AUC Score : 0.8130446863657609
*****
CONFUSION MATRIX

```



	Actual \ Predicted	0	1
0	2827	300	
1	159	413	

```

*****
CLASSIFICATION REPORT

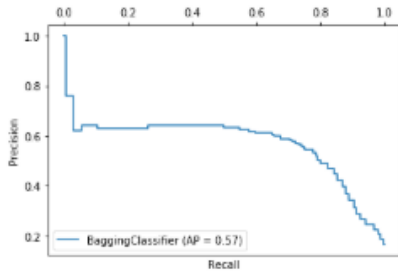
```

	precision	recall	f1-score	support
0	0.95	0.90	0.92	3127
1	0.58	0.72	0.64	572
accuracy			0.88	3699
macro avg	0.76	0.81	0.78	3699
weighted avg	0.89	0.88	0.88	3699

```

*****
PRECISION RECALL CURVE

```



```

*****
CROSS VALIDATION METRICS
Mean Accuracy : 0.9075784440536573
Standard Deviation : 0.008846666418567796
Mean precision score : 0.9064111438602203
Standard Deviation precision score : 0.022956835274487594
Mean recall score : 0.781000306372549
Standard Deviation recall score : 0.02946588385540792
Mean f1 score : 0.8384855774771058
Standard Deviation f1 score : 0.016957014790737218
Mean AUC ROC Score : 0.8724012287808813
Standard Deviation AUC ROC Score : 0.013388939623405203
*****

```

Figure 70: Bagging tree model metrics

### Observations

- Cross validation F1 score of ~84%.
- Area under the curve i.e. Average Precision is 0.57. Skill of the model across thresholds is on the lower side.
- No overfitting.
- Number of Revenue generating user sessions correctly predicted are 413.
- Best hyperparameters - **tree.DecisionTreeClassifier(random\_state=1),max\_features=23, max\_samples=20, n\_estimators=50**

### 8.9.8 DECISION TREE

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                      max_depth=6, max_features='auto', max_leaf_nodes=None,
                      min_impurity_decrease=0.0, min_impurity_split=None,
                      min_samples_leaf=1, min_samples_split=14,
                      min_weight_fraction_leaf=0.0, presort='deprecated',
                      random_state=123, splitter='best')
```

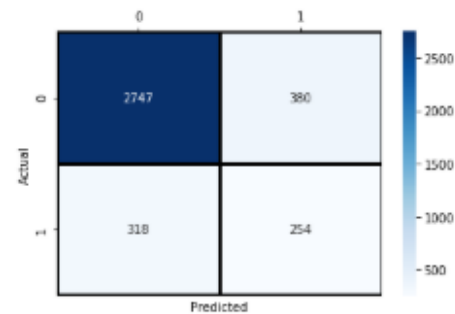
Training Accuracy : 0.7937108855524762

Testing Accuracy : 0.8113083514463368

ROC AUC Score : 0.6612668591402282

\*\*\*\*\*

CONFUSION MATRIX



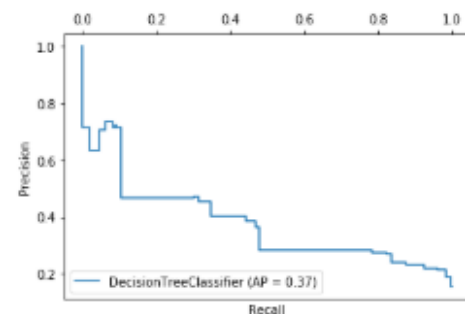
\*\*\*\*\*

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.90	0.88	0.89	3127
1	0.40	0.44	0.42	572
accuracy			0.81	3699
macro avg	0.65	0.66	0.65	3699
weighted avg	0.82	0.81	0.82	3699

\*\*\*\*\*

PRECISION RECALL CURVE



CROSS VALIDATION METRICS

Mean Accuracy : 0.7916627668696499  
 Standard Deviation : 0.013661773524882811  
 Mean precision score : 0.7036840945848555  
 Standard Deviation precision score : 0.03466670737921424  
 Mean recall score : 0.5682751225490196  
 Standard Deviation recall score : 0.09806565207295394  
 Mean f1 score : 0.6221136544039079  
 Standard Deviation f1 score : 0.05272024800208935  
 Mean AUC ROC Score : 0.7295779490945371  
 Standard Deviation AUC ROC Score : 0.035008044417514345

\*\*\*\*\*

Figure 71: Decision Tree model metrics

#### Observations

- Cross validation F1 score of 62% which is very less.
- Area under the curve i.e. Average Precision is 0.37. Skill of the model across thresholds is pathetic.
- No overfitting.
- Number of Revenue generating user sessions correctly predicted are 254.
- Best hyperparameters - **criterion='gini', max\_depth=6,max\_features='auto', min\_samples\_leaf=1, min\_samples\_split=14**

### 8.9.9 RANDOM FOREST

```
RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                        criterion='gini', max_depth=17, max_features='auto',
                        max_leaf_nodes=None, max_samples=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=5,
                        min_weight_fraction_leaf=0.0, n_estimators=175,
                        n_jobs=None, oob_score=False, random_state=None,
                        verbose=0, warm_start=False)
```

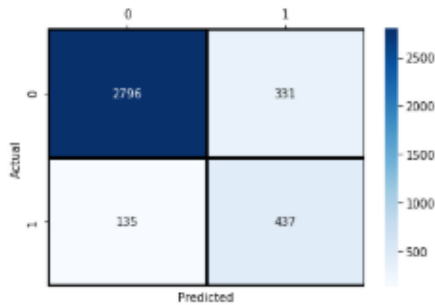
Training Accuracy : 0.9966268995388639

Testing Accuracy : 0.874828854868668

ROC AUC Score : 0.8298668797144652

\*\*\*\*\*

CONFUSION MATRIX



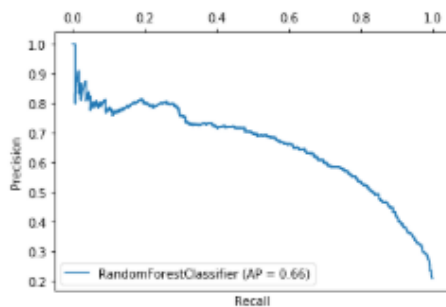
\*\*\*\*\*

CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.89	0.92	3127
1	0.57	0.76	0.65	572
accuracy			0.87	3699
macro avg	0.76	0.83	0.79	3699
weighted avg	0.89	0.87	0.88	3699

\*\*\*\*\*

PRECISION RECALL CURVE



CROSS VALIDATION METRICS

Mean Accuracy : 0.9457789178426614  
Standard Deviation : 0.02182313764511166  
Mean precision score : 0.9243954855697762  
Standard Deviation precision score : 0.02014709195215445  
Mean recall score : 0.8968949142156862  
Standard Deviation recall score : 0.06610165639362582  
Mean f1 score : 0.9093632452005845  
Standard Deviation f1 score : 0.03873835084220197  
Mean AUC ROC Score : 0.9321750741355663  
Standard Deviation AUC ROC Score : 0.03379610780093193

\*\*\*\*\*

Figure 72: Random Forest model metrics

#### Observations

- Cross validation F1 score of ~91%.
- Area under the curve i.e. Average Precision is 0.66. Skill of the model across thresholds is good.
- Slight overfitting.
- Number of Revenue generating user sessions correctly predicted are 437.
- Best hyperparameters  
**n\_estimators=175,min\_samples\_split=5,min\_samples\_leaf=1,max\_features='auto',max\_depth=17,bootstrap=False**

## 8.9.10 XGBOOST

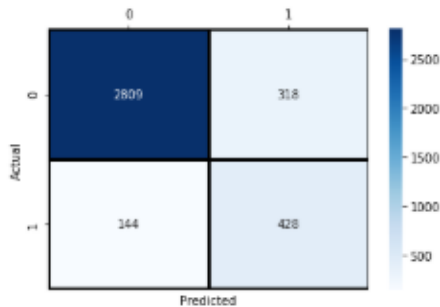
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=200, n_jobs=-1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Training Accuracy : 0.9507169538498614

Testing Accuracy : 0.8751013787510138

ROC AUC Score : 0.8232784164987555

\*\*\*\*\*  
CONFUSION MATRIX



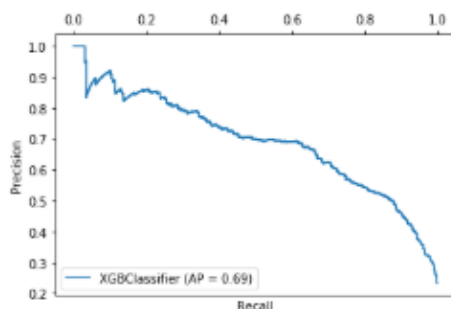
\*\*\*\*\*

### CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.90	0.92	3127
1	0.57	0.75	0.65	572
accuracy			0.88	3699
macro avg	0.76	0.82	0.79	3699
weighted avg	0.89	0.88	0.88	3699

\*\*\*\*\*

### PRECISION RECALL CURVE



### CROSS VALIDATION METRICS

Mean Accuracy : 0.9361392009533913  
 Standard Deviation : 0.023432336706950895  
 Mean precision score : 0.9223367967239311  
 Standard Deviation precision score : 0.020356141346406367  
 Mean recall score : 0.8651577818627452  
 Standard Deviation recall score : 0.07517636773543315  
 Mean f1 score : 0.8912494391510613  
 Standard Deviation f1 score : 0.04348419086508033  
 Mean AUC ROC Score : 0.9163931614964383  
 Standard Deviation AUC ROC Score : 0.037504555687855084

Figure 73: XGBoost model metrics

### Observations

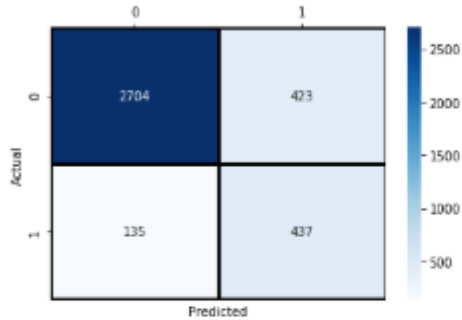
- Cross validation F1 score of 89%.
- Area under the curve i.e. Average Precision is 0.69. Skill of the model across thresholds is pretty good.
- No overfitting.
- Number of Revenue generating user sessions correctly predicted are 428.
- Best hyperparameters - **learning\_rate=0.1, n\_estimators=200, n\_jobs=-1**
- Best model. This model is the chosen one.

### 8.9.11 VOTING CLASSIFIER

```

warm_start=False)),
('lr',
 LogisticRegression(C=1, class_weight=None,
                    dual=False, fit_intercept=True,
                    intercept_scaling=1,
                    l1_ratio=None, max_iter=100,
                    multi_class='auto',
                    n_jobs=None, penalty='l2',
                    random_state=None,
                    solver='newton-cg', tol=0.0001,
                    verbose=0,
                    warm_start=False))),
 flatten_transform=True, n_jobs=None, voting='soft',
 weights=None)>
Training Accuracy : 0.9936136883961924
Testing Accuracy : 0.8491484184914841
ROC AUC Score : 0.8143562944890095
*****
CONFUSION MATRIX

```



	0	1
Actual 0	2704	423
Actual 1	135	437

```

*****
CLASSIFICATION REPORT

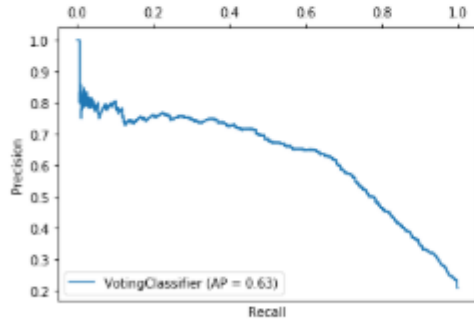
```

	precision	recall	f1-score	support
0	0.95	0.86	0.91	3127
1	0.51	0.76	0.61	572
accuracy			0.85	3699
macro avg	0.73	0.81	0.76	3699
weighted avg	0.88	0.85	0.86	3699

```

*****
PRECISION RECALL CURVE

```



```

*****
CROSS VALIDATION METRICS
Mean Accuracy : 0.9505976136148938
Standard Deviation : 0.01398506406373488
Mean precision score : 0.9347851966121417
Standard Deviation precision score : 0.015202812165616992
Mean recall score : 0.9024157475490195
Standard Deviation recall score : 0.04282359719457717
Mean f1 score : 0.9178045312098242
Standard Deviation f1 score : 0.02490338048854583
Mean AUC ROC Score : 0.9371977237972337
Standard Deviation AUC ROC Score : 0.02168624213973031
*****

```

Figure 74: Voting classifier model metrics

## Observations

- Cross validation F1 score of ~92%.
- Area under the curve i.e. Average Precision is 0.63. Skill of the model across thresholds is OK.
- Slight Overfitting is there.
- Number of Revenue generating user sessions correctly predicted are 437.
- Best Hyperparameters are
  - **Logistic Regression - C=1.0,solver='newton-cg'**
  - **SVM - C=1000, gamma=1,kernel='rbf',probability=True**
  - **KNN - algorithm='auto', leaf\_size = 30, n\_jobs = -1, n\_neighbors = 5, weights = 'distance'**
  - **Decision Tree - criterion='gini',max\_depth=6,max\_features='auto', min\_samples\_leaf=1, min\_samples\_split=14**
  - **Random Forest - n\_estimators=175,min\_samples\_split=5,min\_samples\_leaf=1,max\_features='auto',max\_depth=17,bootstrap=False**
- Soft voting considered.



## 9. ACTIONABLE INSIGHTS AND RECOMMENDATIONS

### 9.1 RECOMMENDATIONS AND CONCLUSIONS POST EDA

Following management level insights and recommendations can be derived from all the EDA done:

- Number of Administrative and Information web pages on the website should be as low as possible as users' interest is on Product related pages and users don't visit other pages.
- Product related web pages can be expanded as users are willing to spend more time on these irrespective of other conditions.
- As part of business strategy, whatever plan is chosen it should always focus on combination of low bounce rates, low exit rates and high page values. This is a must.
- Returning website visitors are contributing more to revenue generation. Therefore, various promotion strategies should focus more on these users.
- Most users visit the website in May and November. In addition, November accounts for maximum number of purchases. This might be due to Christmas and New Year shopping. Business should also look to maximize the conversion of online visits to actual purchases in May.
- Target audience from region 1 and 3 directly as these regions account for maximum revenue generation.
- For Operating System type 2 and Browser type 1-2 revenue is more. This might be due to ease of access and user friendliness of these browsers/OS. Therefore, we can put up something on the website to ask user to access website from these OS/browsers something like – “For best results use OS x and browser y” etc.
- New visitors take up a larger percentage in those who complete purchase. So, it will be a good idea if business form marketing plans to attract new users every day.
- Administrative pages like login, logout, password recovery, profile, email wish list etc. need to be fixed. Users are spending way too much time which is not good. Both the web page and the back end needs to be made more efficient and speedier.
- Automated client-side scripts should be embedded into the web pages so that dormant/idle sessions get logged out after some time of inactivity. Right now, this is absent which is causing wrong data being collected for analysis.

## 9.2 RECOMMENDATIONS AND CONCLUSIONS POST MODELLING

Following recommendations and conclusions can be drawn from the feature selection and subsequent modelling done:

- Most important features from the data are **PageValues, Month\_Nov, ProductRelated\_Duration, Month\_May, ExitRates, TrafficType, Month\_Mar, Administrative\_Duration, Administrative, Informational\_Duration, VisitorType\_New\_Visitor, OperatingSystems\_2, Month\_Dec, Weekend\_False, SpecialDay, Region\_6, Region\_4, OperatingSystems\_3, Month\_Sep, Month\_Oct, Browser\_6, Browser\_4, Browser\_2** and as such company should focus on gathering more data for these variables. Models should be trained as much as on this data with these variables.
- Looking at our results, company should focus on improving mobility between pages to encourage users to browse among different products as Page Value was one of the most important features in determining whether a purchase would be made.
- Certain months such as May and November had a greater frequency in purchases meaning that e-commerce companies should capitalize in these months and provide additional sales and deals to encourage product sales.
- Because our data was highly imbalanced, this bias could influence machine learning algorithms. We had to use multiple oversampling and/or undersampling techniques so that the trained model could predict the minority class i.e. User sessions generating revenue as this is more than important than predicting majority class.
- All throughout the project we maintained our consistent approach of doing everything like over/under sampling, feature selection etc. on training data only. This helped our models to be more efficient and could predict well on unseen data.
- Over/under sampling and feature selection techniques improved the success rates and scalability of the algorithms.
- We used single classifiers as well as ensemble methods for prediction. In addition, stacking was also employed. In general boosting method gave better results. Stacking predictions were too much accurate and therefore not considered as something that could be deployed for general prediction.
- XGBoost model turned out to be the **best model overall**. It predicted higher number of minority class samples without any overfitting. It had one of the top F1 Scores. The model was also most skilled in making predictions at various probability thresholds of majority and minority class.
- Voting classifier was the **second-best model**. It also had good performance metric values. Only thing against this was in some cases it leaned towards slight overfitting though it as not that significant.
- While there are limitations with our data and its application on a larger scale, our analysis shows that it is possible to predict a site visitors buying behavior within a certain level of confidence based on the features we were able to pull from the data set.
- Considering the real time usage of the proposed system, achieving better or similar classification performance with minimal subset of features is an important factor for the e-commerce companies since a smaller number of features will be kept track during the session.

## 10. FUTURE SCOPE OF WORK

- To further improve this project, we believe data that could better shape the context of the site visitor would be important.
- Possible data points that could be used are user buying history, third party data, wish lists, etc.
- It would allow us to better analyze what characteristics and features are important to look at in possible customers.
- It could be useful to add class weights which would add a greater penalty to mis-classifying the under-represented class in order to reduce false positives or false negatives. This would reduce bias on the imbalanced data without “over-training” on the under-represented samples.
- In this project, we have focused mainly on F1 score to evaluate model’s accuracy. F1 score weights precision and recall equally. One another metric which can be explored is the Fbeta-measure which is an abstraction of the F-measure where the balance of precision and recall in the calculation of the harmonic mean is controlled by a coefficient called beta.

$$\text{Fbeta-Measure} = ((1 + \text{beta}^2) * \text{Precision} * \text{Recall}) / (\text{beta}^2 * \text{Precision} + \text{Recall})$$

- There might be an opportunity to explore numerous other algorithms which, while computationally expensive, may be able to provide more accurate predictions.
- If we really want to go really after accuracy, we can look to put different other classifiers in Stacking, tune some more hyperparameters and bring down Stacking accuracy and F1 score to a reasonable value along with predicting good number of minority class samples.

### 10.1 APPLICATIONS OF WORK

Following the areas where are project could prove helpful:

- To assist e-commerce companies, identify which web data and metrics should they focus on to understand online purchasing behavior, so that they can configure their web pages and related Google Analytics accordingly.
- To find out best compatibility of the company’s website in terms of Browser, Operating System etc.
- As our model is able to predict purchasing users and best months for purchase, this can help company to formulate things like targeted ads, offers and schemes, etc.
- This can be a great model for online apparel companies where majority of users do window shopping instead of actually buying.
- This can be applied in usual online shopping for books, household items, electronic, cameras, mobiles etc.

## 11. REFERENCES AND BIBLIOGRAPHY

- Online Shoppers Purchasing Intention Dataset Data Set: <https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset>
- Hoi Piew Tan, Choon Ling Kwek & Teck-Chai Lau. (2010, August). *Investigating the Shopping Orientations on Online Purchase Intention in the e-Commerce Environment: A Malaysian Study*: [https://www.researchgate.net/publication/288582068\\_Investigating\\_the\\_Shopping\\_Orientations\\_on\\_Online\\_Purchase\\_Intention\\_in\\_the\\_e-Commerce\\_Environment\\_A\\_Malaysian\\_Study](https://www.researchgate.net/publication/288582068_Investigating_the_Shopping_Orientations_on_Online_Purchase_Intention_in_the_e-Commerce_Environment_A_Malaysian_Study)
- Fernando Aguilar. (2019, October 9). *SMOTE-NC in ML Categorization Models for Imbalanced Datasets*: <https://medium.com/analytics-vidhya/smote-nc-in-ml-categorization-models-fo-imbalanced-datasets-8adbdcf08c25>
- Yuqing Zhang. (2019, September 23). *Predicting Online Shoppers Purchasing Intention with H2O*: <https://zhangyuqing.github.io/2019/09/predicting-online-shoppers-purchasing-intention-with-h2o/>
- Chen Ling, Tao Zhang & Yuan Chen. (2019, August 27). *Customer Purchase Intent Prediction Under Online Multi-Channel Promotion: A Feature-Combined Deep Learning Framework*: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8795449>
- Humphrey Sheil, Omer Rana & Ronan Reilly. (2018, July 21). *Predicting purchasing intent: Automatic Feature Learning using Recurrent Neural Networks*: <https://arxiv.org/pdf/1807.08207.pdf>
- C. Okan Sakar, S. Olcay Polat, Mete Katircioglu & Yomi Kastro. (2018, May 9). *Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks*: <https://link.springer.com/article/10.1007/s00521-018-3523-0>
- Jason Brownlee. (2020, January 6). *ROC Curves and Precision-Recall Curves for Imbalanced Classification*: <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/>
- Jason Brownlee. (2020, January 22). *Combine Oversampling and Undersampling for Imbalanced Classification*: <https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/>
- Paula Branco, Luis Torgo & Rita Ribeiro. (2015, May 13). *A Survey of Predictive Modelling under Imbalanced Distributions*: <https://arxiv.org/abs/1505.01658>

## 12. APPENDIX

Please refer raw dataset, imputed data and python codes from here.



Raw and imputed  
dataset.zip



Codes.zip