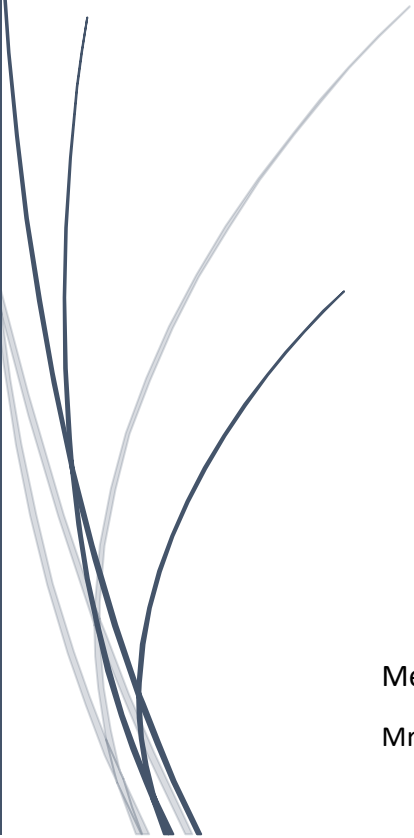
A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

14/10/2020

Project Report on Predicting Customer Purchase Intention

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Mentored by,
Mr. Romil Gupta

Submitted by,
S Revanth Shalon Raj
Kumar Ashutosh
Vishak Subramanyan A

Contents

| | |
|--|----|
| Impact of Technology in Online Shopping: | 4 |
| Smartphone Shopping | 4 |
| Dynamic Pricing Strategies | 4 |
| Behavioural Analytics | 4 |
| Dataset Description | 5 |
| Features and its Description | 5 |
| Models to be used | 5 |
| Exploratory Data Analysis Report | 7 |
| Target Feature (Revenue) | 7 |
| Administrative Feature | 7 |
| Description and Summary Statistics | 7 |
| Observation of Administrative Page Counts Plot | 8 |
| Observation of Administrative Page against Revenue | 9 |
| Administrative Duration | 9 |
| Description and Summary Statistics | 9 |
| Observation of Administrative Duration as Box plot | 10 |
| Observation of Administrative Duration against Revenue with Administrative Page Category | 11 |
| Informational Feature | 11 |
| Description and Summary Statistics | 11 |
| Observation of Informational Page Category Counts | 12 |
| Observation of Informational Pages against Revenue | 13 |
| Informational Duration | 13 |
| Description and Summary Statistics | 13 |
| Observation of Informational Duration as Box plot | 14 |
| Observation of Informational Duration against Revenue based in Informational Page Categories | 15 |
| Product Related Feature | 15 |
| Description and Summary Statistics | 15 |
| Observation on the Product Related Page Counts | 16 |
| Observation on Revenue with Product Related Pages | 17 |
| Product Related Duration | 17 |
| Description and Summary statistics | 17 |
| Observation of Product Related Duration Box plot | 18 |
| Observation of Product Related Duration with Respect to Revenue based on Product Related Pages | 19 |
| Bounce Rate Feature | 19 |
| Description and Summary Statistics | 19 |
| Exit Rate Feature | 20 |
| Description and Summary Statistics | 20 |
| Page Values Feature | 21 |

| | |
|--|----|
| Description and Summary Statistics | 21 |
| Operating Systems Features | 21 |
| Description and Summary Statistics | 21 |
| Visitor Type | 22 |
| Description and Summary Statistics | 22 |
| Weekend | 23 |
| Description and Summary Statistics | 23 |
| Region | 24 |
| Description and Summary Statistics | 24 |
| Traffic Types | 26 |
| Description and Summary Statistics | 26 |
| Browser | 28 |
| Description and Summary Statistics | 28 |
| Month | 30 |
| Description and Summary Statistics | 30 |
| Bounce Rate vs Exit Rates | 31 |
| Special Days | 31 |
| Product Related Duration of New Customers vs returning Customers. | 32 |
| February | 33 |
| May | 33 |
| Product Duration, Administrative Duration and Informational Duration | 34 |
| Product Duration | 34 |
| Administrative Duration | 35 |
| Informational Duration | 36 |
| Important Note | 36 |
| Data Preparation | 37 |
| Binning Columns | 37 |
| Treatment of the Independent Features | 37 |
| Pipelining the Data | 37 |
| Models | 37 |
| Logistic Regression (Base Estimator) | 37 |
| Decision Tree Classifier | 40 |
| Random Forest Classifier | 42 |
| Adaptive Boosting Classifier | 43 |
| Gradient Boosting Classifier | 45 |
| Support Vector Machines Classifier | 45 |
| K Neighbours Classifier | 46 |
| Bagging Classifier | 47 |
| Gaussian Naïve Bayes | 47 |

| | |
|--|----|
| XGBoost Random Forest Classifier | 48 |
| Model Selection | 49 |
| Comparison of Cross Validated Scores of all the Models | 49 |
| Accuracy Scores | 49 |
| Recall Scores | 50 |
| Applying Validation Set to Classifiers | 50 |
| Conclusion | 51 |

Impact of Technology in Online Shopping:

The lucrative nature of the e-commerce market is attracting an increasing number of businesses to this domain. In order to thrive amidst the cut-throat competition and make their business successful, marketers need to focus on offering an unparalleled shopping experience to their customers.

Technology has changed the manner in which retailers and customers interact, enabling marketers to build their online brand image and equity.

Smartphone Shopping

Market Research revealed that more than 86 million Americans use their smartphones for online shopping. The study found that four out of five smartphone users do a thorough research on the products and services available online before making a purchase.

Smartphones have become the default screen for brand engagement and e-commerce transactions, making it crucial for marketers to maintain a good online reputation and offer fair pricing strategies. To drive their business growth, marketers must strive to make the mobile shopping experience enjoyable, informative, and convenient for their customers.

Dynamic Pricing Strategies

The online retail market is highly price-sensitive and competitive. Dynamic pricing is a strategy used by e-retailers whereby the price of the products or services offered are changed depending upon the supply and demand.

Dynamic pricing also enables firms to monitor their competitors' pricing strategies, helping them make sound pricing decisions. For instance, if its competitor's stocks are low, a firm can choose to increase the prices, boosting its sales and profits.

Behavioural Analytics

With mobile users becoming increasingly comfortable with online shopping, web analytics and customer behavioural analytics are gaining importance.

Customers prefer to do an online research on products and services, however, they expect e-retail stores to offer them an array of options with respect to their preferences and buying behaviour. Online business analytics offer rich data on the customer behaviour trends, helping retailers improve merchandising, supply chain, marketing, advertising, and other strategic decisions.

Behavioural analytics tracks the shoppers' search and purchase history and their interactions with the customer care professionals, offering a wealth of information to online marketers. This data enables retailers to predict and suggest the relevant products and services to their target customers.

This is where we as Data Scientists try to predict the customer intention, by identifying the important factors and try to maintain those factors in a range so that we can increase the customer's intention of buying a product.

Dataset Description

Understanding Customer Product Purchase Intention project is done with the aim of producing a customer purchase intention model, which can be used as a binary classifier to measure the user's intention of purchasing a product. The dataset consists of 12,300 sessions, where 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. The values "Bounce Rate", "Exit Rate", and "Page Value" features represent the metric measures by "Google Analytics" for each page.

Features and its Description

- Administrative - Number of pages visited by the visitor about account management.
- Administrative Duration - Total amount of time (in seconds) spent by the visitor on account management related pages.
- Informational - Number of pages visited by the visitor about Website, communication and address information on the shopping site.
- Informational Duration - Total amount of time spent (in seconds) by the visitor on informational pages.
- Product Related - Number of pages visited by visitors about product related pages.
- Product Related Duration - Total amount of time (in seconds) spent by the visitor on product related pages.
- Bounce Rate - It represents the percentage of visitors who enter the site and then leave ("bounce") rather than continuing to view other pages within the same site. Bounce rate is calculated by counting the number of single page visits and dividing that by the total visits. It is then represented as a percentage of total visits.
- Exit Rate - Exit rate as a term used in web site traffic analysis is the percentage of visitors to a page on the website from which they exit the website to a different website. The visitors just exited from that specific page.
- Page value - Average page value of the pages visited by the visitor.
- Special day - Closeness of the site visiting time to a special day. (Categorical)
- Operating Systems - Operating system of the visitor. (Categorical)
- Browser - Browser of the visitor. (Categorical)
- Region - Geographic region from which the session has been started by the visitor. (Categorical)
- Traffic Type - Traffic source by which the visitor has arrived at the Web site (e.g., banner, SMS, direct). (Categorical)
- Visitor Type - Visitor type as "New Visitor", "Returning Visitor" and "Other". (Categorical)
- Weekend - Boolean value indicating whether the date of the visit is weekend. (Categorical)
- Month - Month value of the visit date. (Categorical)
- Revenue - Class label indicating whether the visit has been finalized with a transaction. (Target)

Models to be used

Since the customer purchase intention model is a binary predictor model, we would be applying different algorithms such as

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machines
- Boosting Techniques
- Bagging Techniques
- Stacked Classifier
- Voted Classifier
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- K-Means Clustering
- Agglomerative Clustering
- Bernoulli Naïve Bayes Classifier

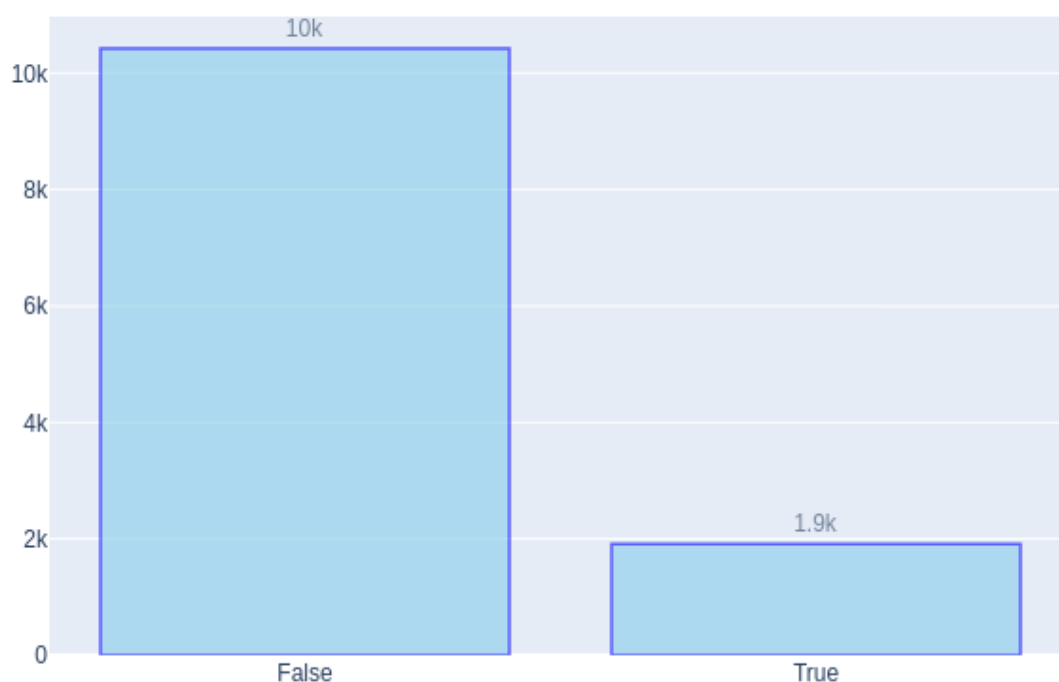
- Gaussian Naive Bayes Classifier
- Multinomial Naïve Bayes Classifier
- K-Nearest Neighbours Classifier

Exploratory Data Analysis Report

Target Feature (Revenue)

- This column consists of Boolean values (True or False).
- When true, the visitor purchased the product from the website.
- When false, the visitor did not purchase any product from the website.
- We have a total of 10,422 false values and 1908 true values.
- The False values alone constitute up to 84.5% observations.
- The True values constitute only 15.5% of the total observations.
- This dataset is heavily imbalanced, and we might have to use *Synthetic Minority over Sampling Technique* (SMOTE) for balancing the dataset for higher accuracy in prediction.

Total Revenue Counts from the Data



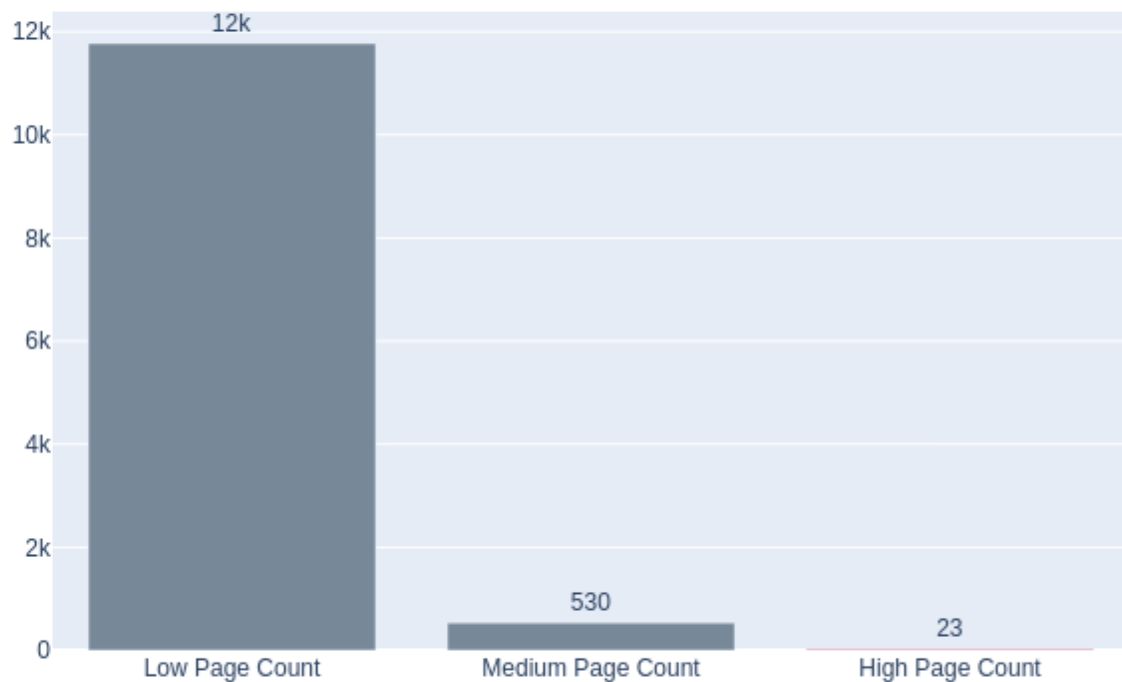
Administrative Feature

Description and Summary Statistics

- This feature represents the number of pages visited by the user related to account management.
- Summary Statistics of the 'Administrative' Column are,
 - Min value: 0
 - Max value: 27
 - Mean: 2.3151
 - Median: 1
 - Standard Deviation: 3.32
- We know that the pages cannot be partial.
- It will always be a discrete value.
- We also know that the Administrative values ranges from 0-27.
- We can split the data into 3 categories, i.e.,
 - Low Administrative Page Views Category.
 - Medium Administrative Page Views Category.

- High Administrative Page Views Category.

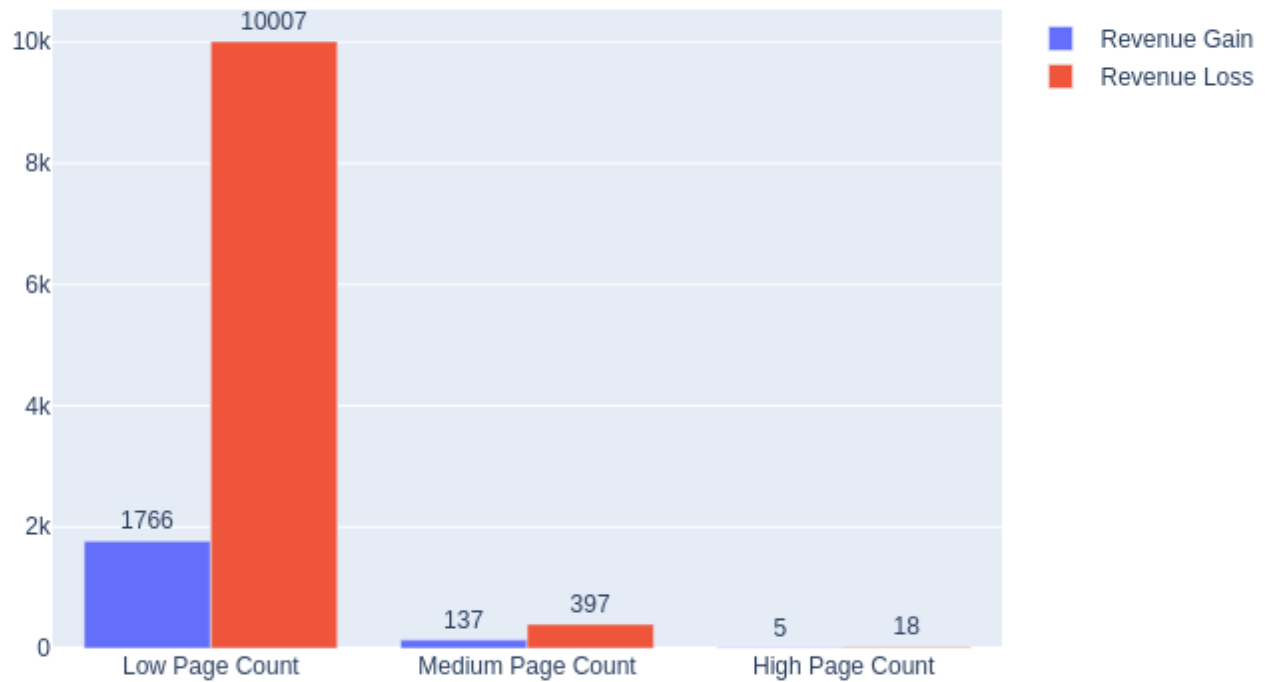
Administrative Page Counts



Observation of Administrative Page Counts Plot

- The number of observations in each category are,
 - Low Administrative Page counts: 11,773
 - Medium Administrative Page counts: 534
 - High Administrative Page counts: 23
- This is in accordance with normal distribution and practical observation.
- People don't usually keep checking out the account pages repeatedly, unless there is an issue related or when tracking an order.
- The outlier count is also very low.

Comparison of Revenue with Administrative Page visits by the user



Observation of Administrative Page against Revenue

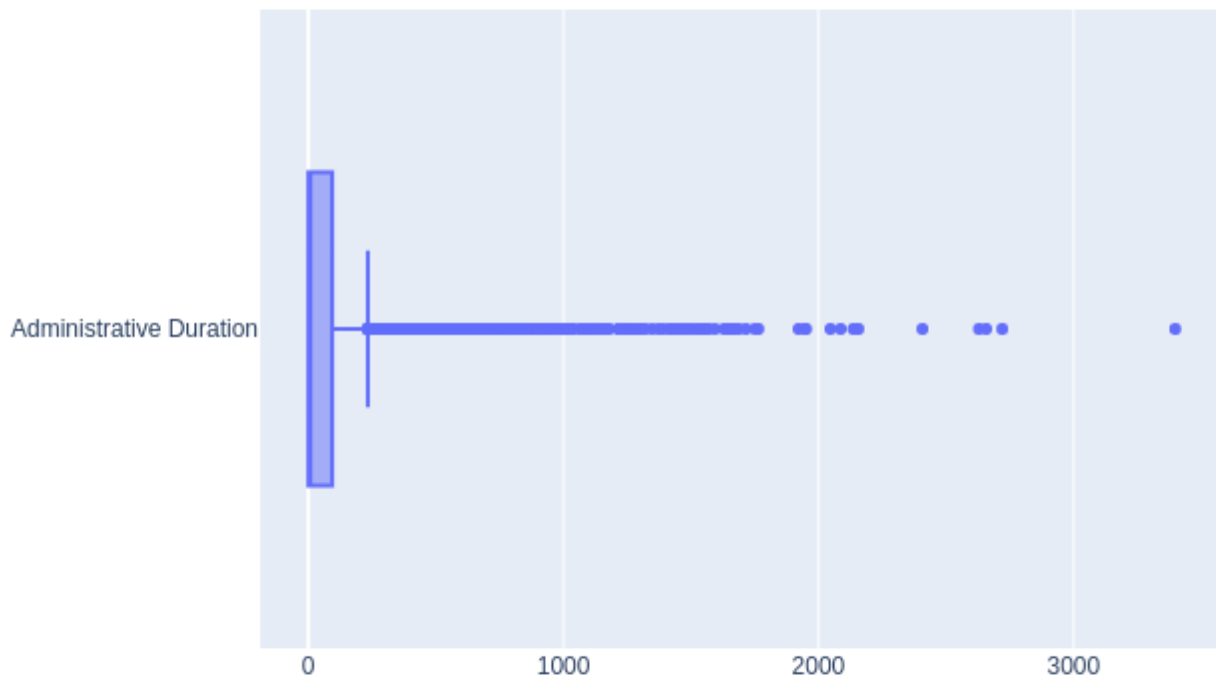
- We can observe that for people with high administrative page count values, the revenue gain percentage is higher compared with the low administrative page count category.

Administrative Duration

Description and Summary Statistics

- This feature represents the duration spent by the visitor in Administrative pages (in seconds).
- Summary Statistics of Administrative Duration Column are,
 - Min value: 0
 - Max value: 3398.750
 - Mean: 80.8186
 - Median: 7.5
 - Standard Deviation: 176.779
- Time is a continuous value.
- We are constructing a box plot to check the data visually.

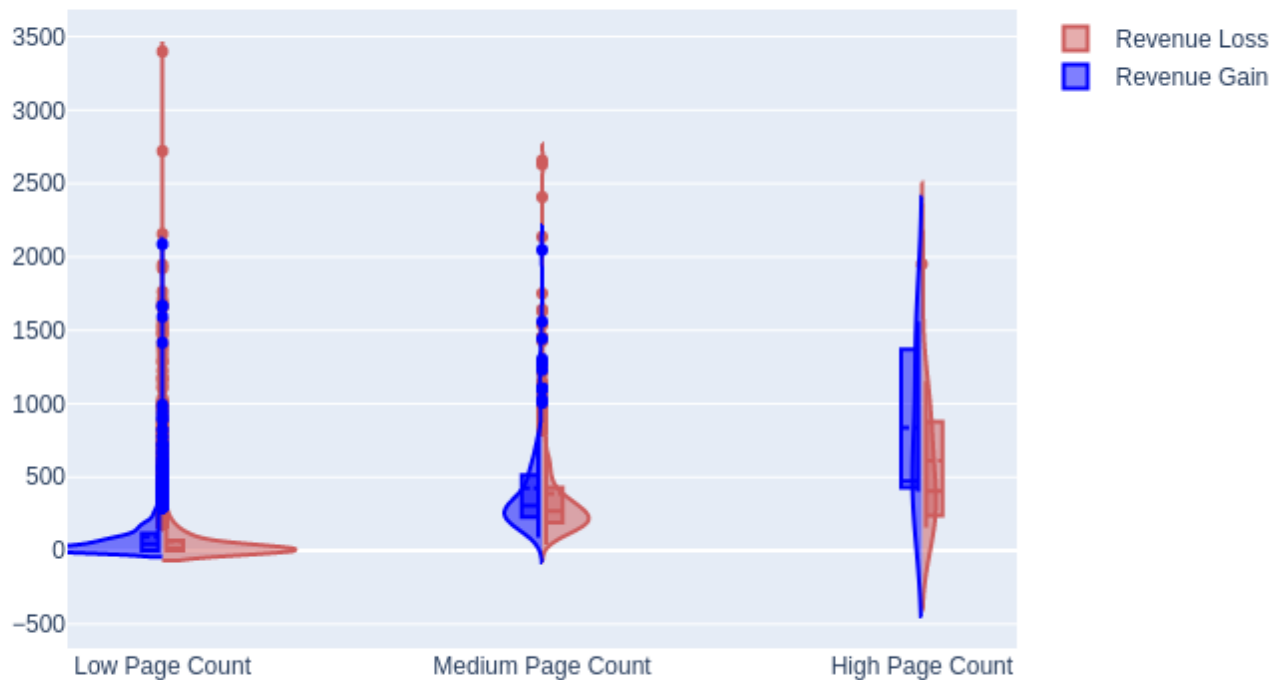
Box plot of Administrative Duration



Observation of Administrative Duration as Box plot

- The Administrative duration is right skewed.
- People don't usually spend a lot of time editing their profile on an E-Store website.
- However, there are still some outliers.
- This might be due to people leaving the pages open and forget about the tab.

Comparing Administrative Duration with Respect to Revenue based on Administrative Page



Observation of Administrative Duration against Revenue with Administrative Page Category

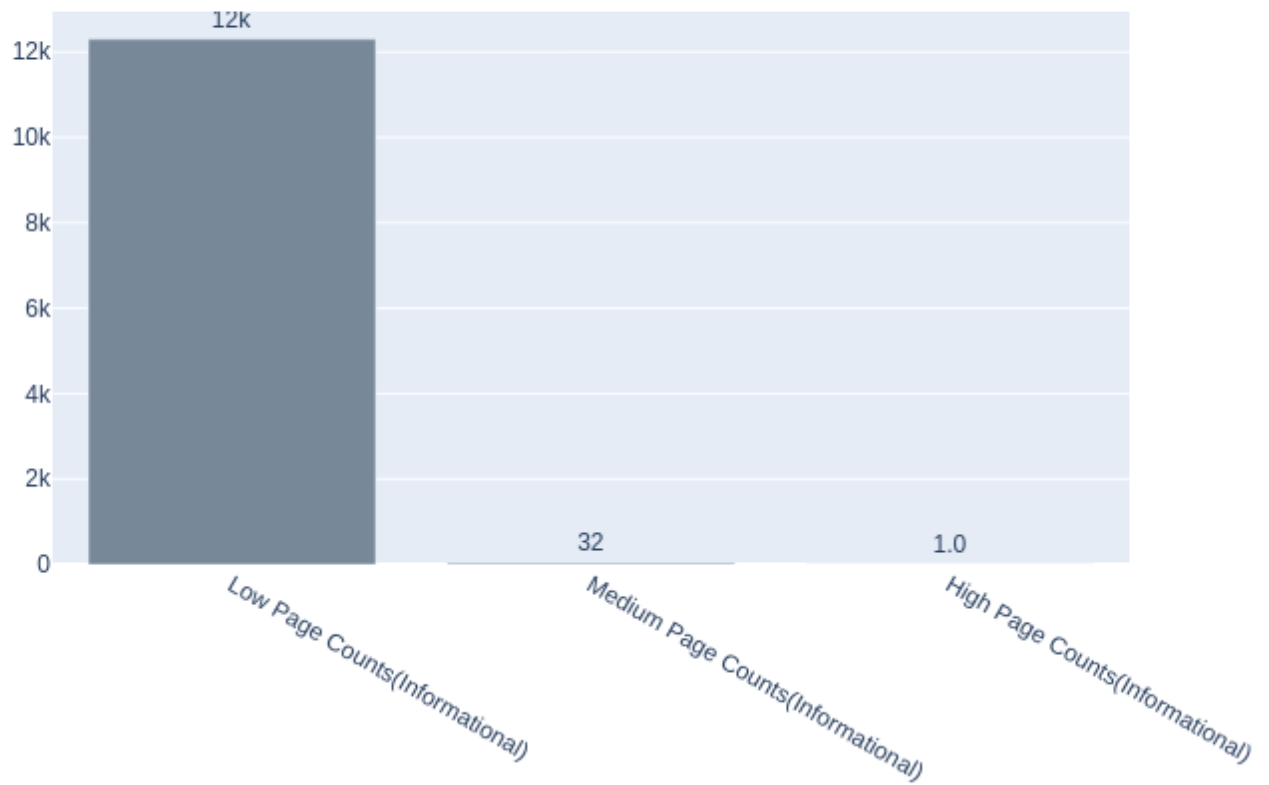
- The duration spent by the visitor in administrative page based on Administrative Page categories, it can be found that there is a difference between the values and this can be attributed to a change in the revenue.
- Also we can notice a trend here, the average time spent by the visitor who generate revenue is higher compared to visitors who did not generate revenue, in all the categories.
- This is especially distinct as in Medium and High Page Count Categories of Administrative Page Values.

Informational Feature

Description and Summary Statistics

- This feature represents the number of pages visited by the user related to Website, Communication and Information.
- Summary Statistics of the 'Informational Column are,
 - Min value: 0
 - Max value: 24
 - Mean: 0.5035
 - Median: 0.0
 - Standard Deviation: 3.32
- We know that the pages cannot be partial.
- It will always be a discrete value.
- We also know that the Informational values ranges from 0-24.
- We can split the data into 3 categories, i.e.,
 - Low Page Views (Informational).
 - Medium Page Views (Informational).
 - High Page Views (Informational).

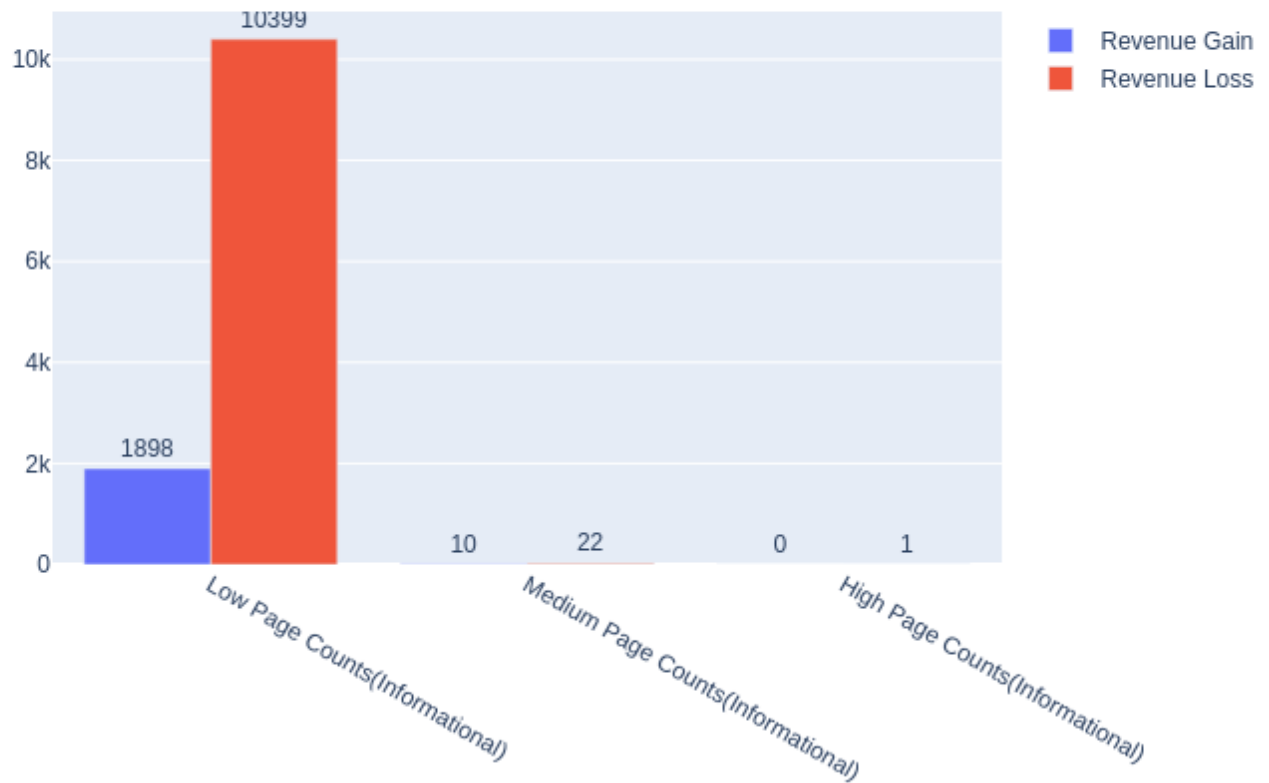
Informational Page Counts



Observation of Informational Page Category Counts

- The number of observations in each category are,
 - Low Administrative Page counts: 12,297
 - Medium Administrative Page counts: 32
 - High Administrative Page counts: 1

Comparison of Revenue with Informational Page visits by the user



Observation of Informational Pages against Revenue

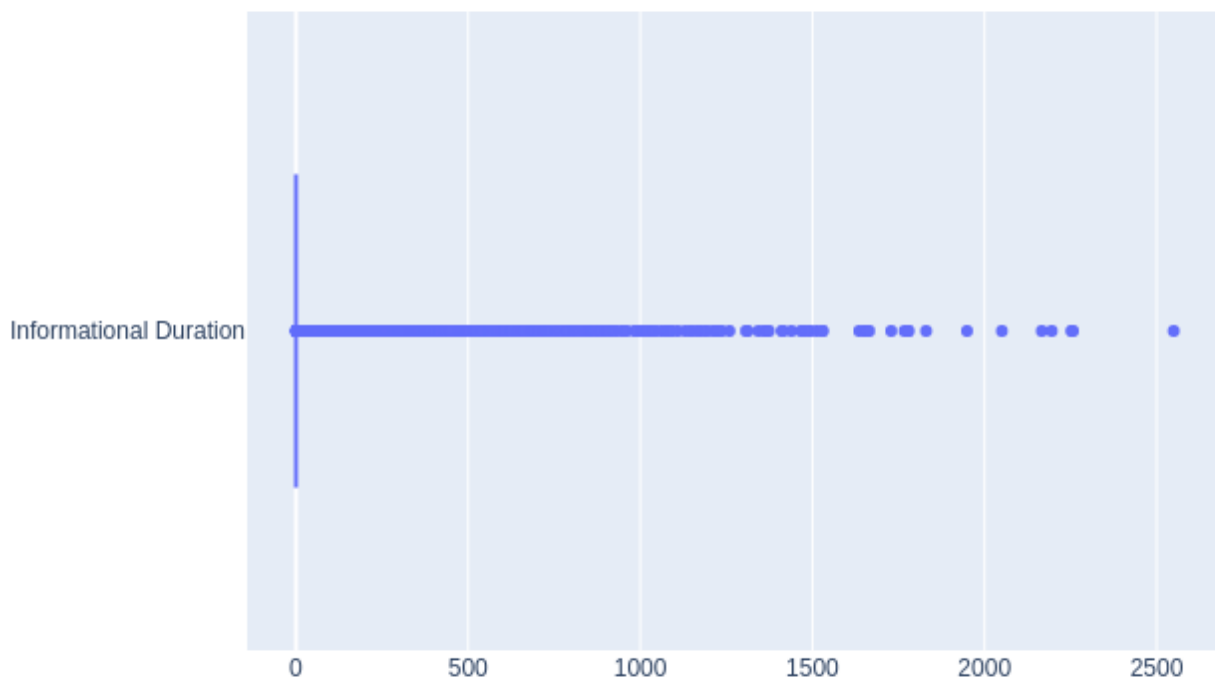
- The percent of visitors converted into revenue gain actually increases as the number of pages visited by the user increases as well.
- There is still an outlier at very high informational page view with revenue as loss.
- This might be a shopper collecting info regarding a product rather than buying it.

Informational Duration

Description and Summary Statistics

- The time spent by the user in pages containing information about the E-Commerce website and details regarding communication in seconds.
- Summary Statistics of the "Information Duration" are,
 - Min: 0
 - Max: 2549.37
 - Mean: 34.47
 - Median: 0
 - Standard Deviation: 140.79

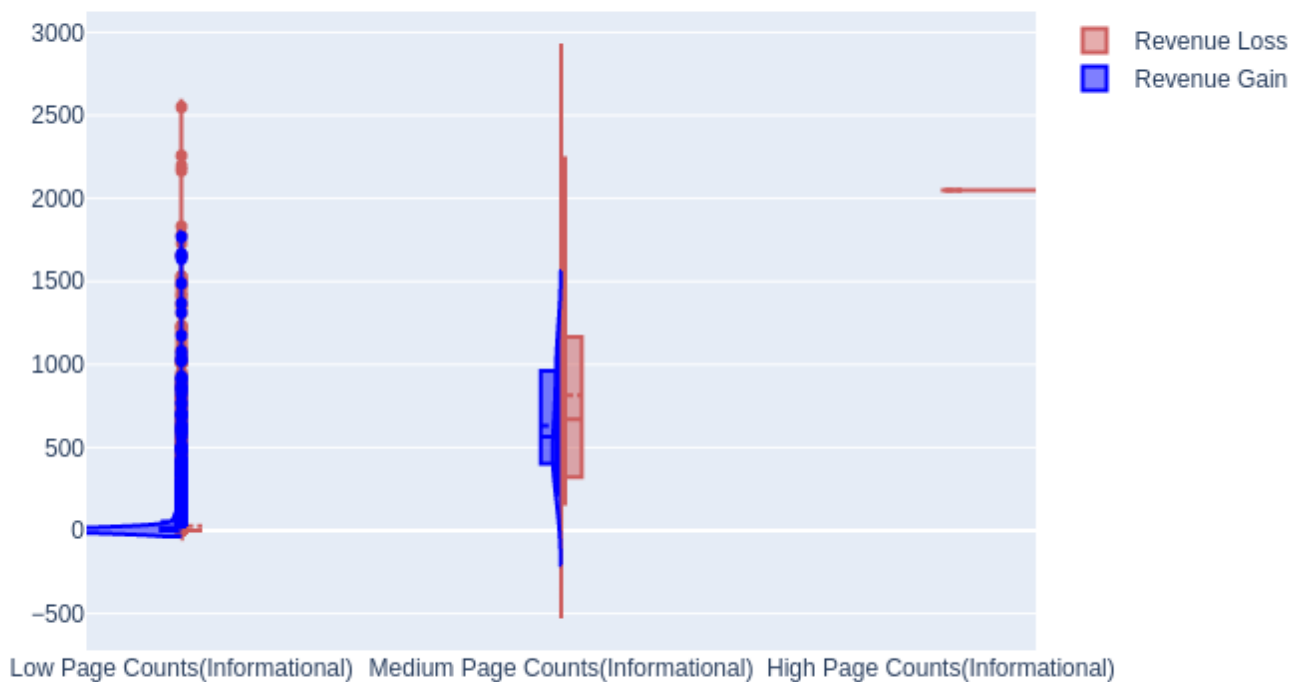
Box plot of Informational Duration



Observation of Informational Duration as Box plot

- The data is highly skewed towards the right.
- The 1st, 2nd and 3rd quantiles of the box plot is primarily the value 0, since it occurs in a huge number of times.
- This shows the people are primarily not interested in the pages related to E Commerce website information and communication.

Comparing Informational Duration with Respect to Revenue based on Informational Page



Observation of Informational Duration against Revenue based in Informational Page Categories

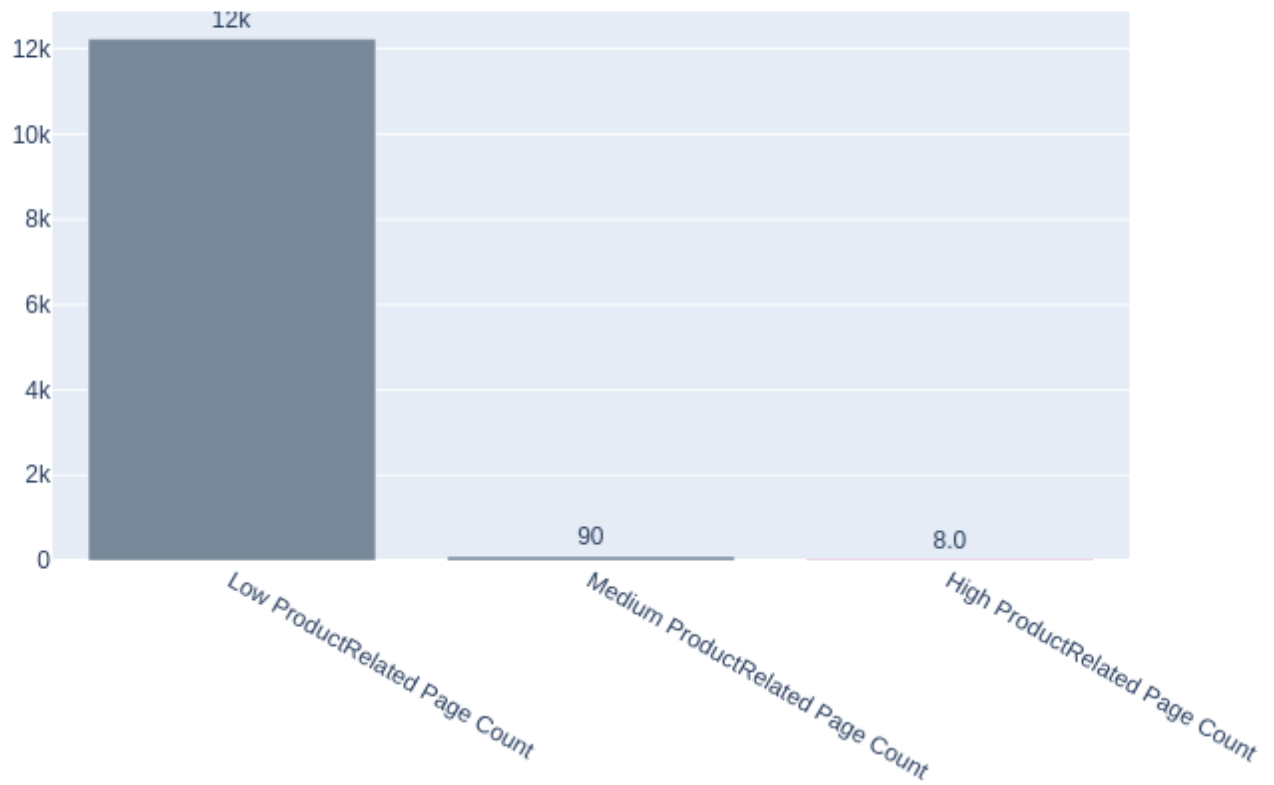
- We can be sure that an E-Commerce website would not have very high pages on its site, describing itself.
- Since there are only a very few number of visitors who keeps visiting the informational pages, we can understand that the User Interface should be pretty friendly.
- Also the time spend by the users on low page counts when it is too high it suggest that we could make the instructions a bit more cleared on the working of the website.

Product Related Feature

Description and Summary Statistics

- This refers to the number of pages visited by the user, which are related to a product.
- Summary Statistics of the "Product Related" column are,
 - Min: 0
 - Max: 705
 - Mean: 31.73
 - Median: 18
 - Standard Deviation: 44.475
- Splitting them into 3 Categories,
 - Low Product Related Pages
 - Medium Product Related Pages
 - High Product Related Pages

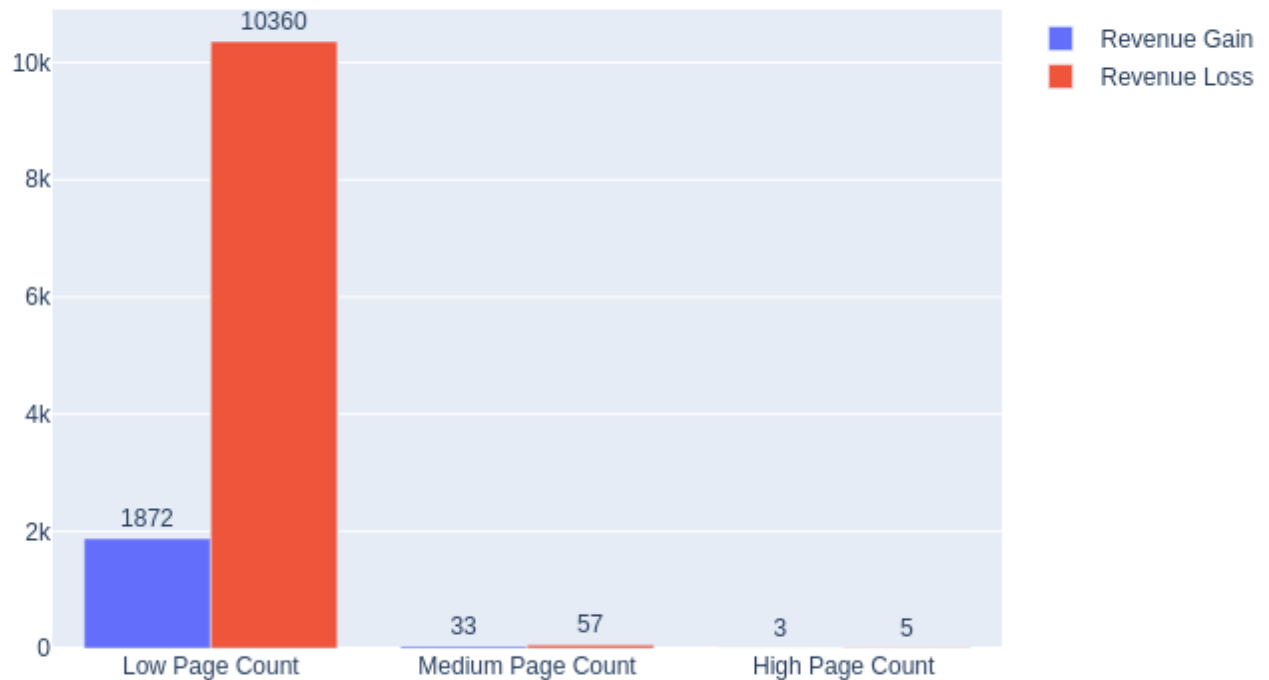
Product Related Page Counts



Observation on the Product Related Page Counts

- It follows the normal distribution where most people visit only a few pages related to the product
- Only a very small percentage of the total visitors visit very high pages related to products.
- This might be because visitors usually have an idea what they are looking for and they might decide on whether to buy the product or not and looking for cheaper alternatives.

Comparison of Revenue with Product Related Page visits by the user



Observation on Revenue with Product Related Pages

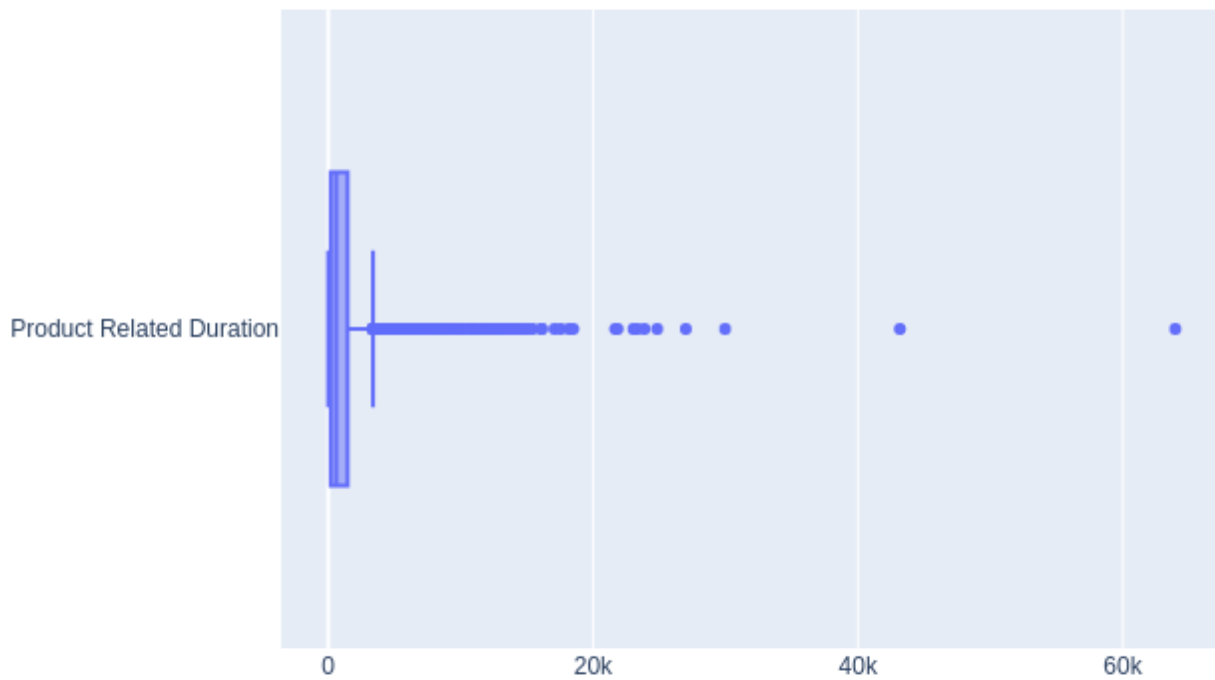
- The percentage of the users looking at product pages, when high the users buying the product is also very high in comparison.
- This shows the importance of displaying products properly.
- These people with high page counts might be randomly browsing and might have found an item in a cheap cost and that might have led to the revenue generation.

Product Related Duration

Description and Summary statistics

- This features has values that tell us the time spent by the user in product related pages, in seconds.
- The Summary Statistics of “Product Related Duration” Column are ,
 - Min: 0
 - Max: 63973.522
 - Mean: 1194.746220
 - Median: 598.93
 - Standard Deviation: 1913.669

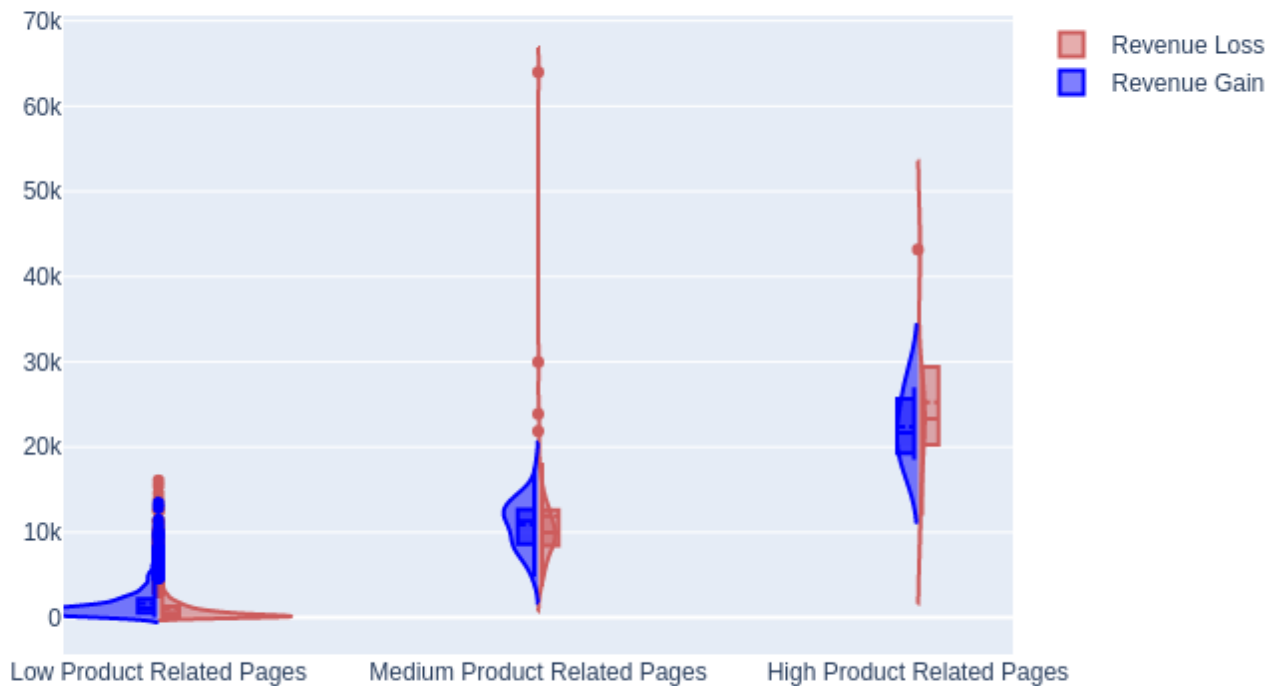
Box plot of Product Related Duration



Observation of Product Related Duration Box plot

- We can see that the data is right skewed.
- This tell that most of the time spent by the users are usually in normal range, since the outlier are in an abnormal range on 60,000+ seconds.

Comparing Product Related Duration with Respect to Revenue based on Product Related



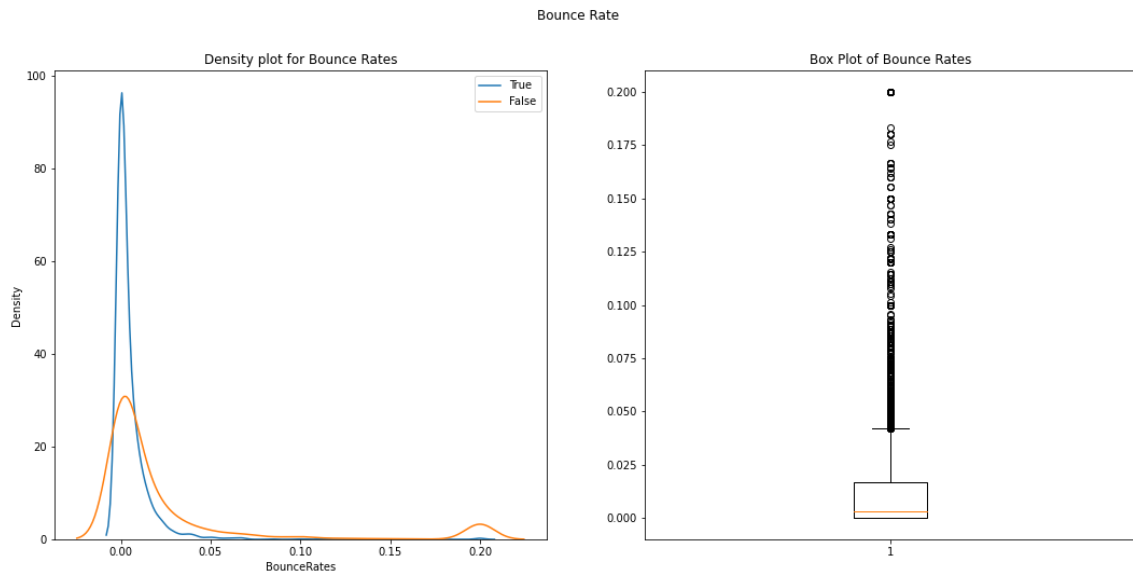
Observation of Product Related Duration with Respect to Revenue based on Product Related Pages

- We can see that in low and medium Product related pages, when the duration spent by the user in average when higher, leads to Revenue Gain.
- This is not the case in High Product related pages. The Duration on the average when lower, the user generate revenue.
- This shows the importance of maintaining consistent and relevant product information across all the products and price comparisons between the products.
- We can understand that when the user actually wants to purchase a product the user spend an increased amount of time on the product related pages to check for details and comparing products.
- This behaviour could not be extrapolated to the higher product related pages. They shows the exact opposite behaviour, even though they have a higher product related duration spent by the users. This leads me to believe that these users are modern day window shopper in an E-Commerce platform.

Bounce Rate Feature

Description and Summary Statistics

- It represents the percentage of visitors who enter the site and then leave ("bounce") rather than continuing to view other pages within the same site. Bounce rate is calculated by counting the number of single page visits and dividing that by the total visits. It is then represented as a percentage of total visits.
- Summary Statistics of Bounce Rate:
 - Min : 0.0000
 - Max : 0.200
 - Mean : 0.0221
 - Median : 0.00312
 - Standard Deviation : 0.048488

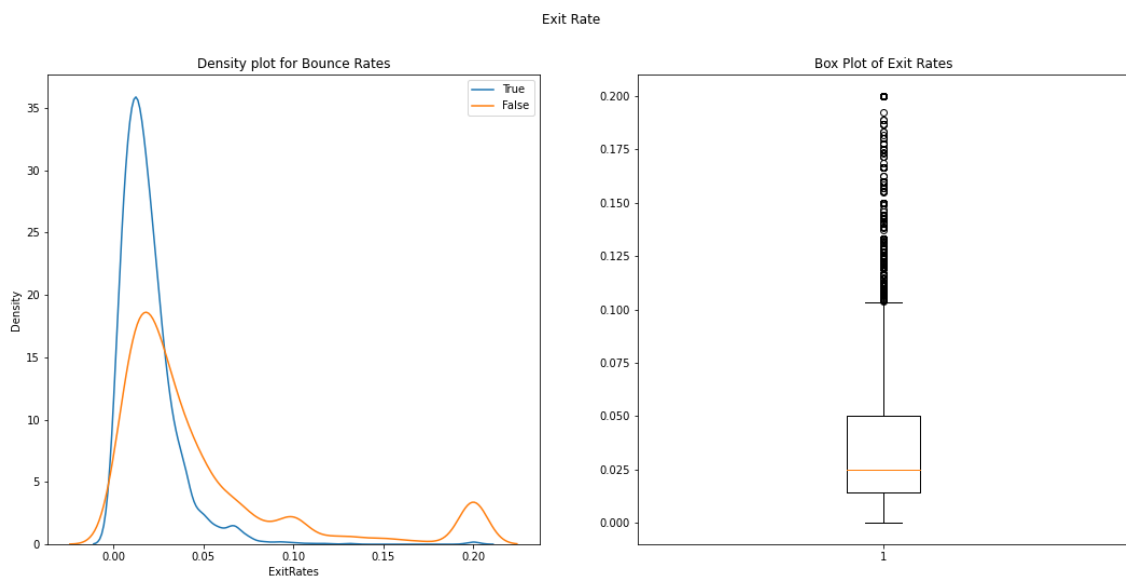


We can see the highest concentration of the users within the range of values from 0.00 to 0.1 for the bounce rates. We can also see there is a second high concentration of users at 0.2, which means that there are a high percentage of users who has the max value of bounce rates.

Exit Rate Feature

Description and Summary Statistics

- Exit rate as a term used in web site traffic analysis is the percentage of visitors to a page on the website from which they exit the website to a different website. The visitors just exited from that specific page.
- Summary Statistics of Exit Rate
 - Min : 0.000
 - Max : 0.200
 - Mean : 0.043
 - Median : 0.025
 - Standard Deviation : 0.048

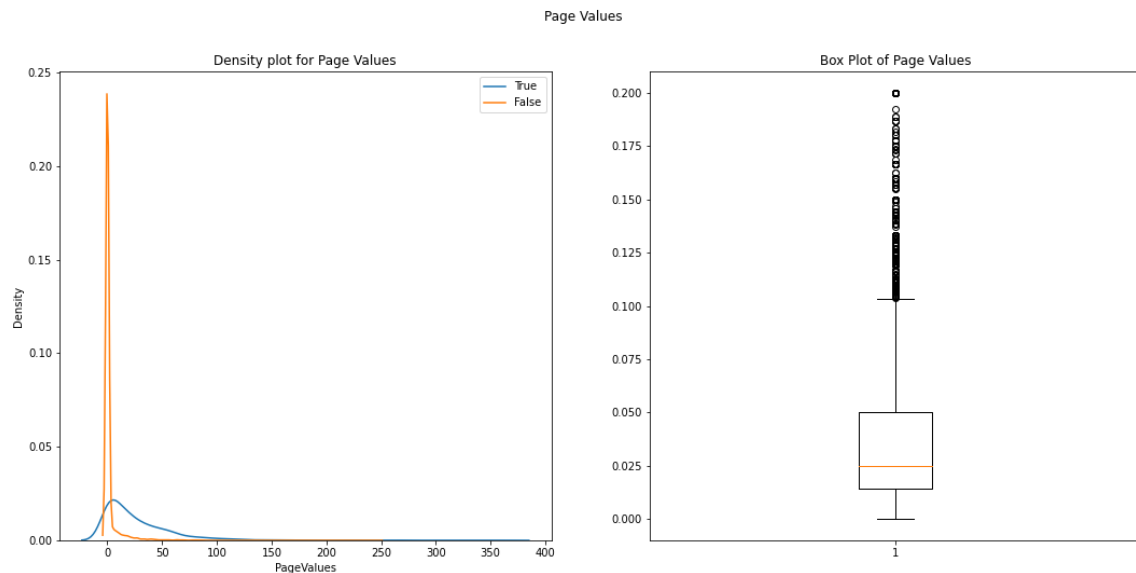


We can see that the exit rates for the users are highly condensed at the 0.000 to 0.05 values. The higher the values the chances of user churning becomes higher.

Page Values Feature

Description and Summary Statistics

- It is the average page value of the pages visited by the visitor.
- Summary Statistics of Page Values
 - Min : 0
 - Max : 361
 - Mean : 5.889
 - Median : 0
 - Standard Deviation : 18.56

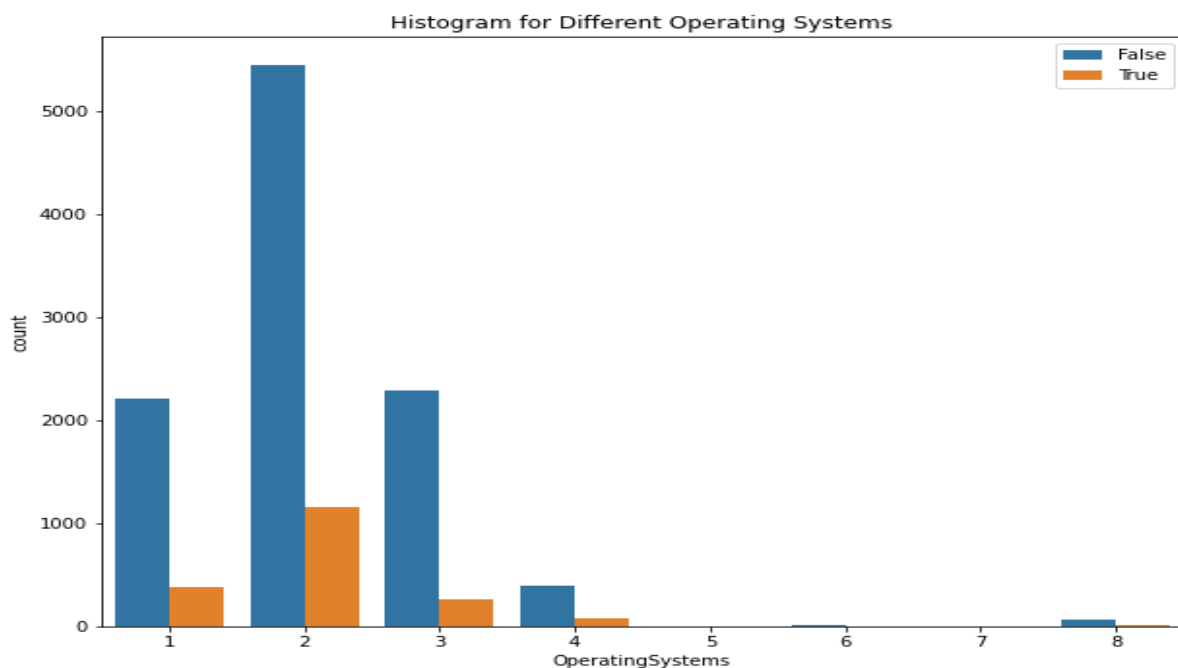


The number of users who do not generate revenue usually have very low page values compared to the users who generate revenue.

Operating Systems Features

Description and Summary Statistics

- The number of pages visited by each user in average for a session.

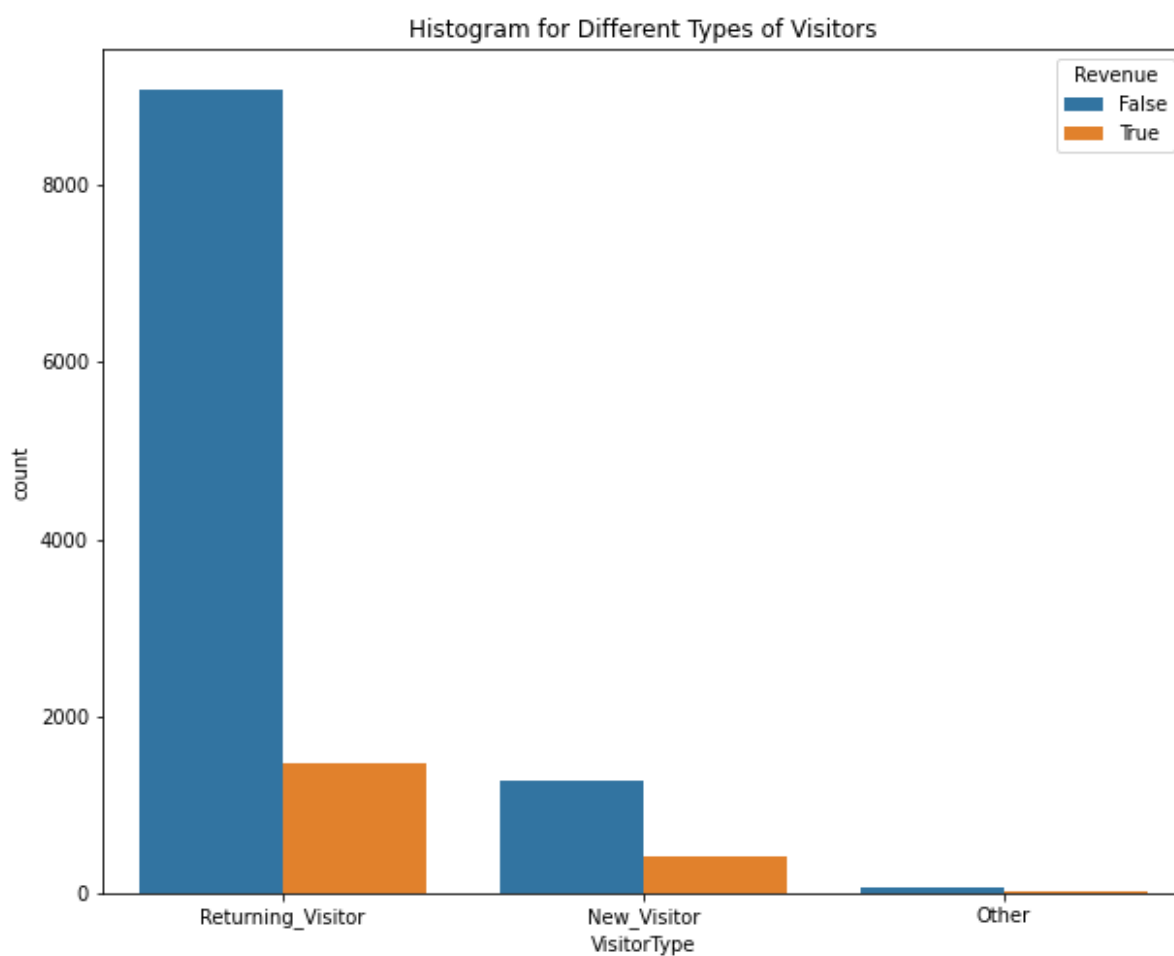


We can see that most people who purchase actually use Operating system type 2 followed by type 1 and 3.

Visitor Type

Description and Summary Statistics

- We have three types of visitors.
 - Returning Visitors
 - New Visitors
 - Others
- Summary Statistics of Visitor Types are,
 - Returning Visitors : 85.5%
 - New Visitors : 13.7%
 - Other : 0.68%

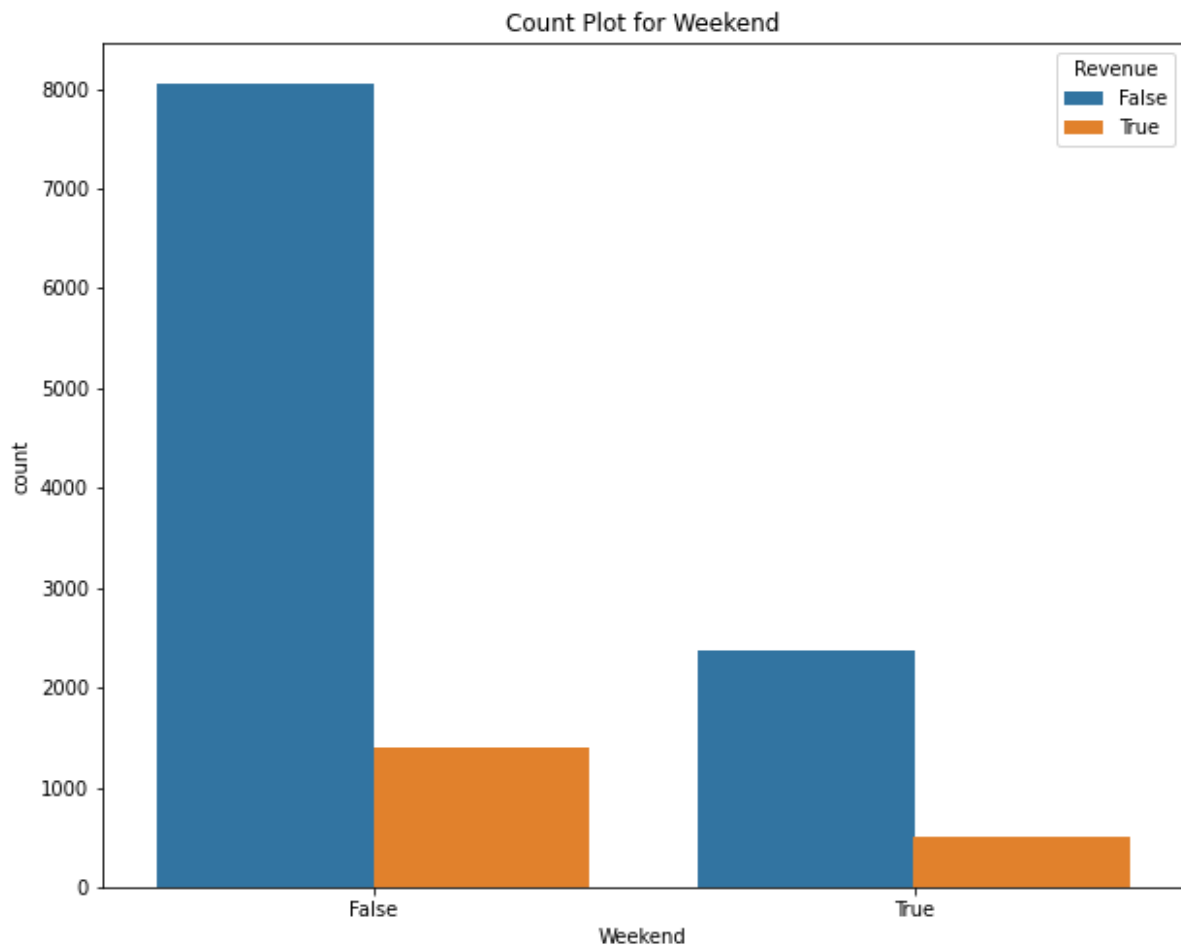


For the each visitor type, the revenue generated by the new visitor is higher (24%) of the total new visitors, when compared with other types (Returning Visitors: 13%). We don't have enough data for the other category to predict the characteristics of the visitor type.

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

Weekend

Description and Summary Statistics



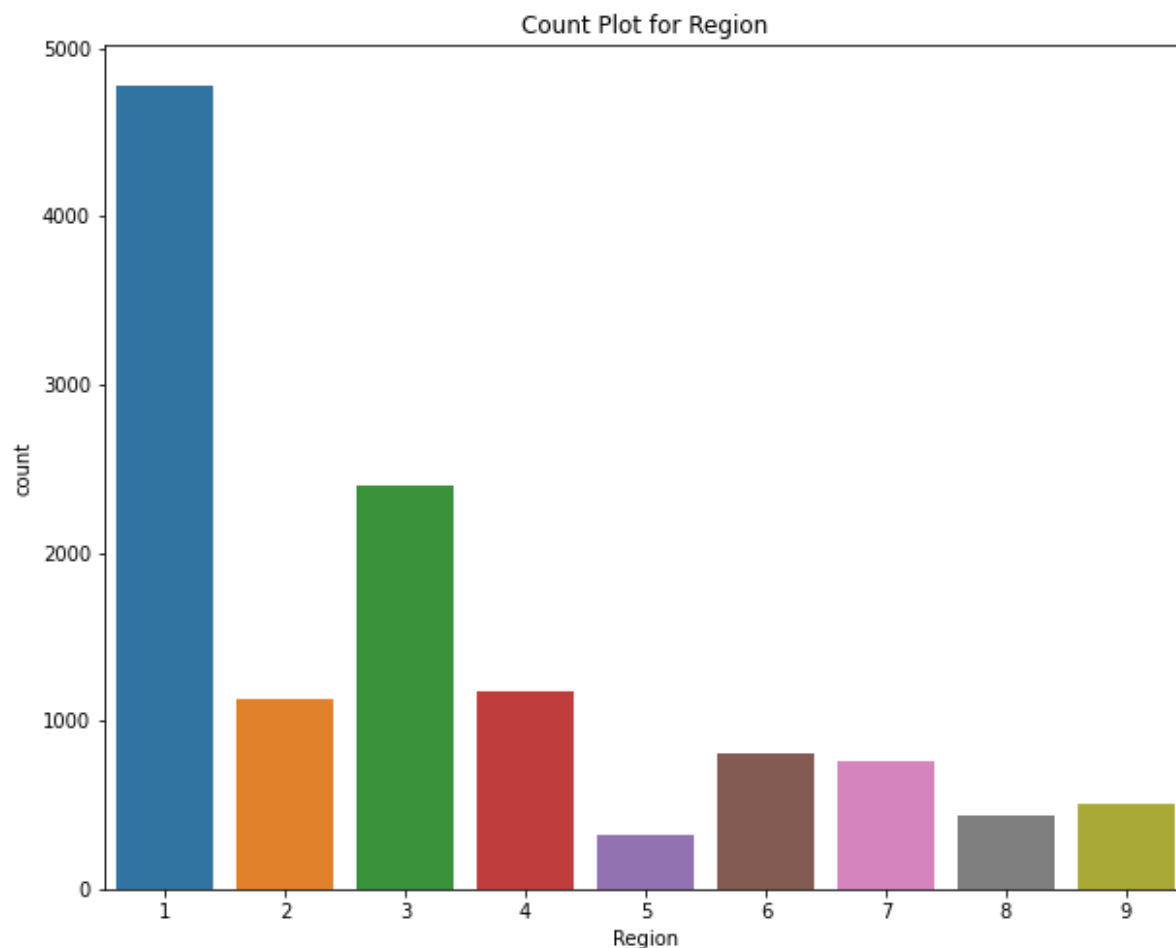
We can see that the total count of customers are lower in weekends and greater during the weekdays. This shows that the weekend doesn't affect the buying pattern with respect to the online markets. However, this may be an important factor in real life markets.

| Weekend | False | True | All |
|--------------|-------|------|-------|
| Revenue | | | |
| False | 8053 | 2369 | 10422 |
| True | 1409 | 499 | 1908 |
| All | 9462 | 2868 | 12330 |

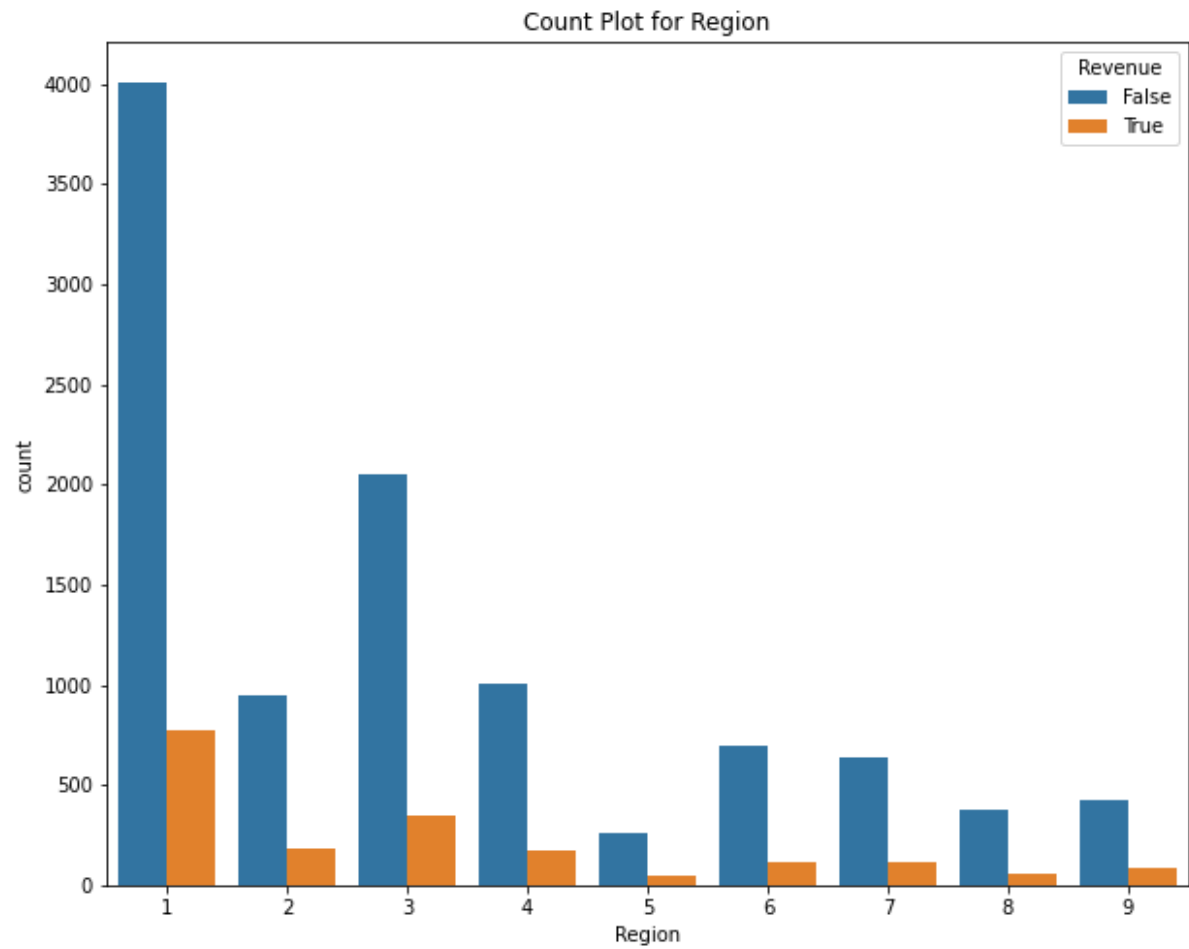
Upon further analysis with comparison of revenue against weekend, we can observe that the positive revenue generation is 14% on the weekdays and 17% on the weekends. This is a marginal percent and this confirms the above deduction that the weekends and weekdays don't play an important role in e-market is strengthened.

Region

Description and Summary Statistics



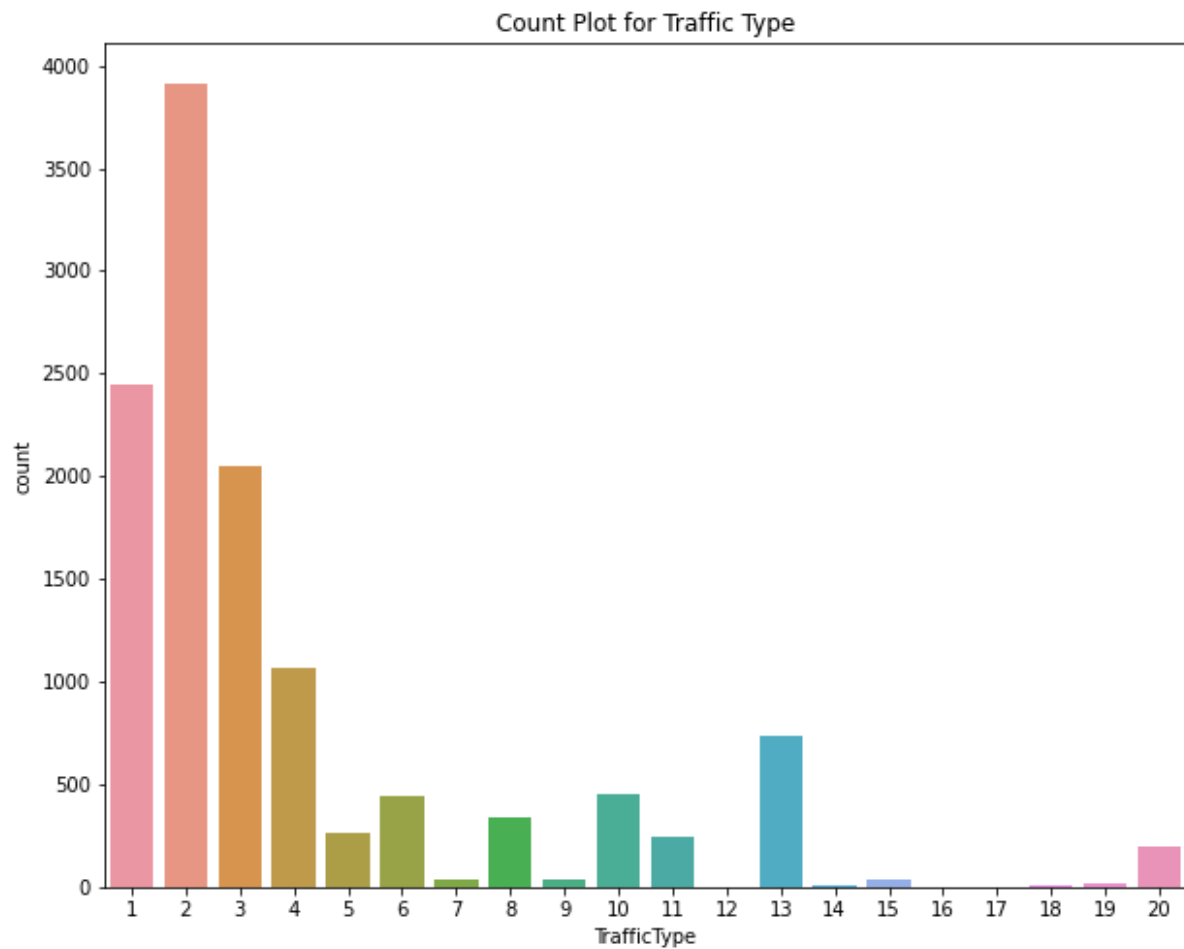
We can see that the number of customer is high in the regions 1,3,4,2 in the respective order. Some of the regions with the lowest customer base are 5, 8 and 9. We should focus more on these low customer base and try to increase the number of customers in the regions. We could increase the advertisements in the particular region, give them a bit more promotional offers.



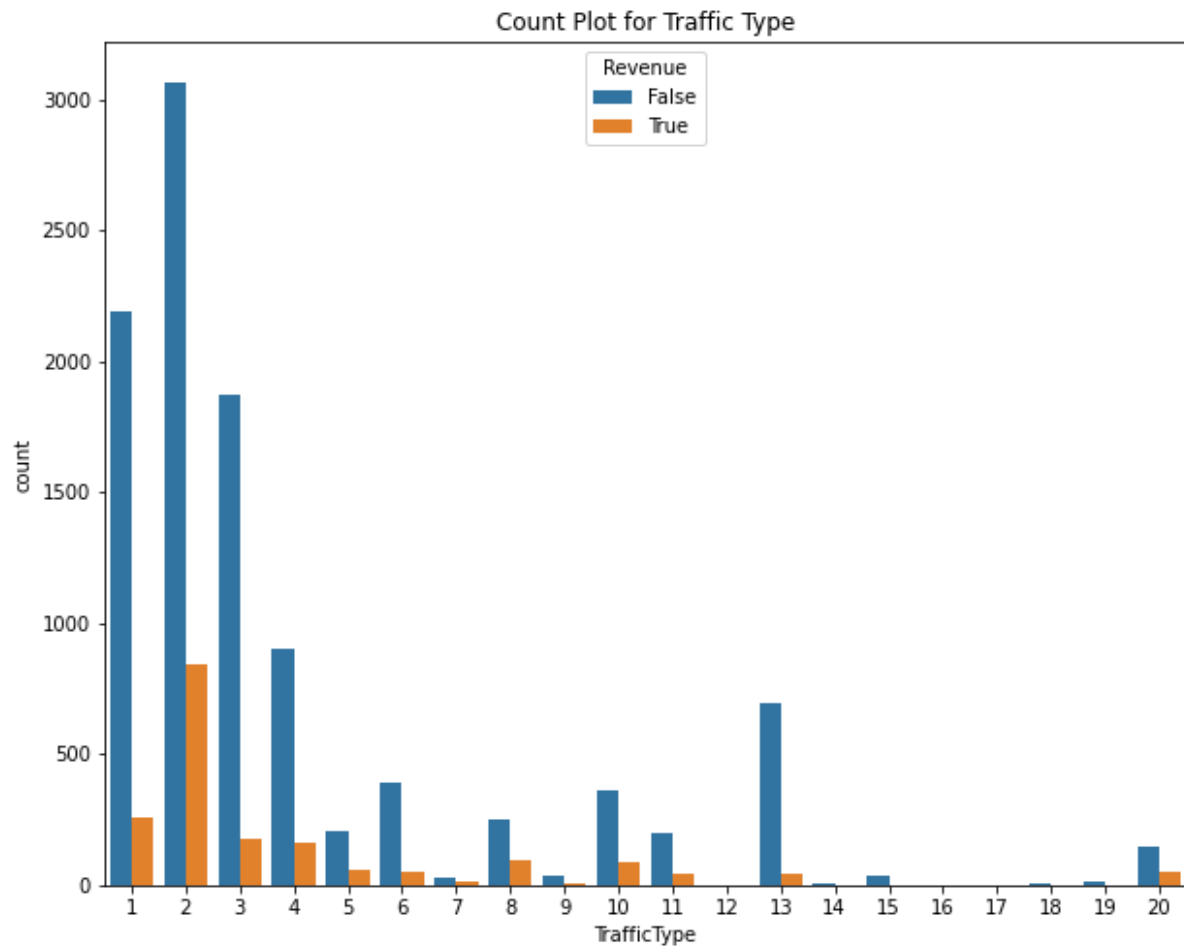
We can observe that the revenue generation is higher when we have a larger Customer base. So we should focus on increasing our customer base along with the satisfaction rate.

Traffic Types

Description and Summary Statistics



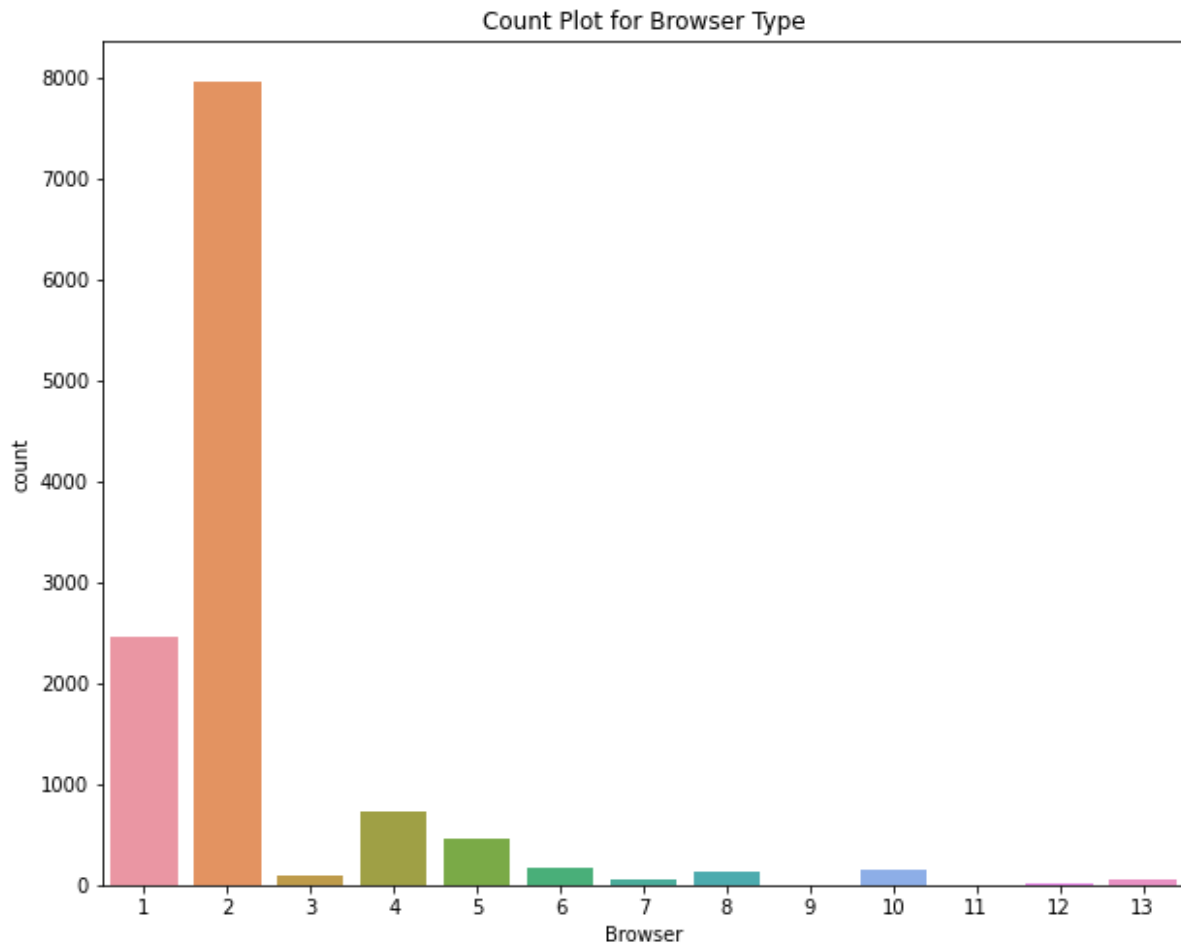
These indicates the quantity of customer traffic generated by different traffic sources. We can see that the highest traffic source is from 2,1,3,4. We also need to confirm that they also provide the highest revenue generation.



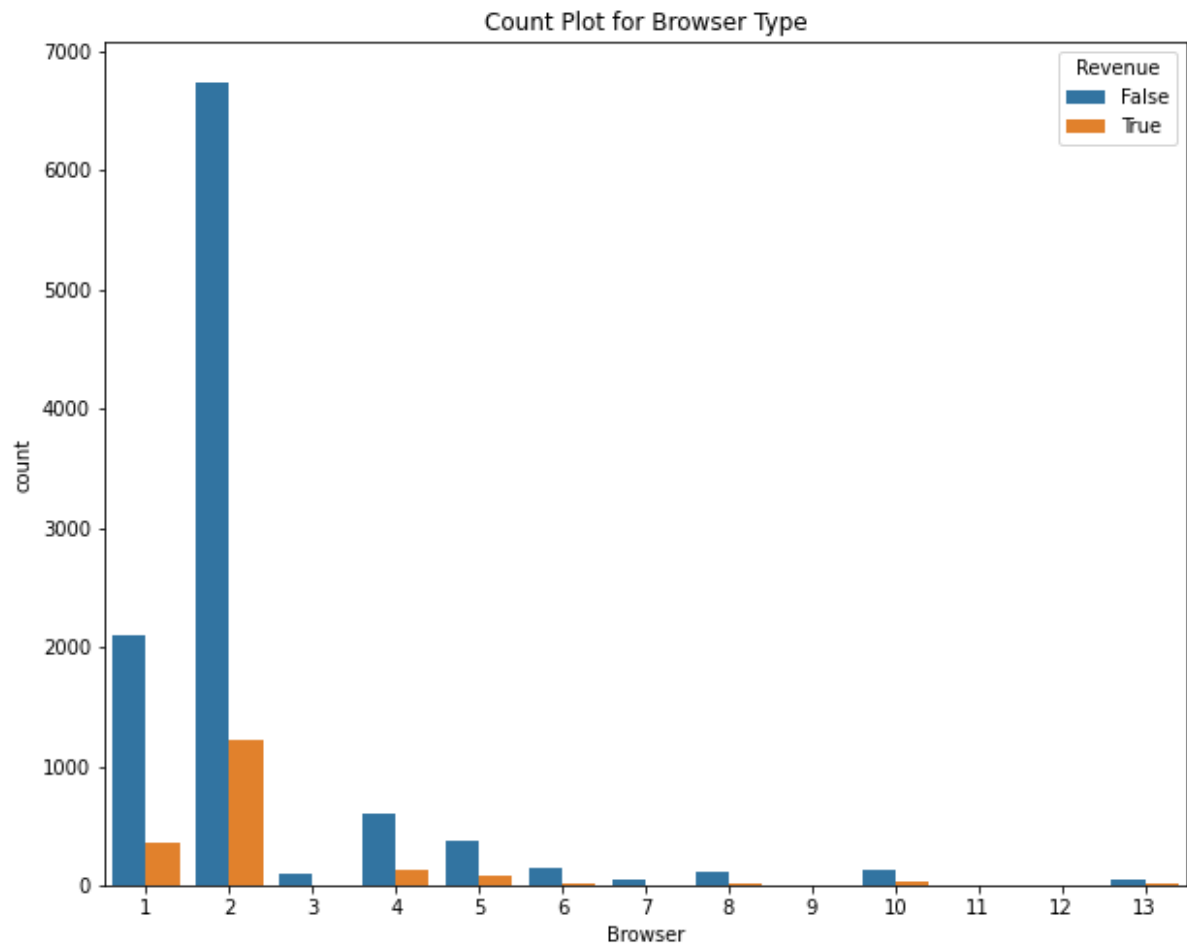
This confirms the above assumption that the high traffic generation can also increase the customer base, there by indirectly influence the revenue generation by the e-market site. However the traffic source 13 even with a high traffic relative to its other source has very less revenue generation potential. We could check the revenue generation potential, or we can use the highest traffic source types to give advertisement or promotions about the e-market site.

Browser

Description and Summary Statistics



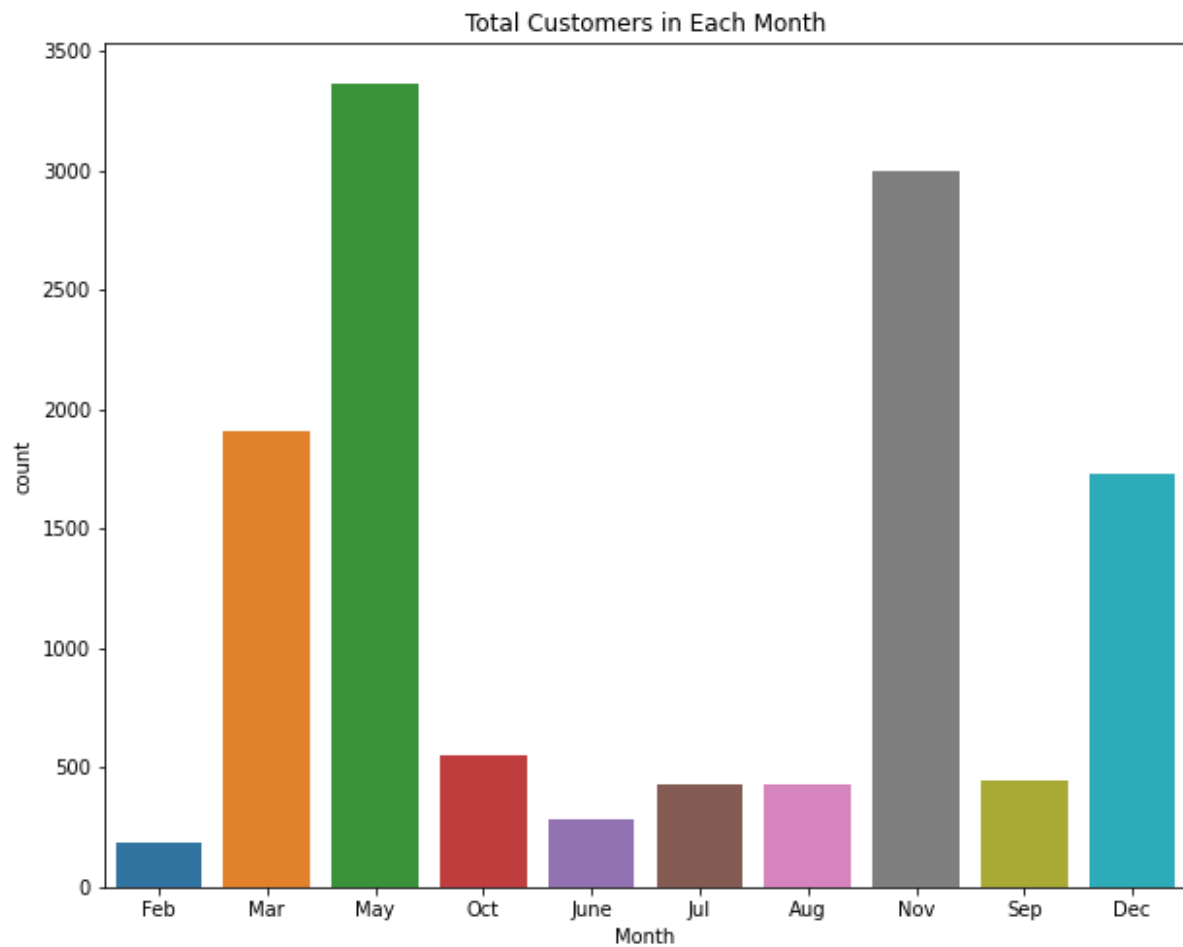
We can see the browser used by different customers. As first glance we might not take browser as a good variable to base our revenue generation on. But most people use branded phones and laptops. Some of the brands have their own in-built browsers which people prefer. These brands might be expensive, example, Apple Products. This might be a representative of their economic status. This can be forwarded to their shopping mentality of buying products. We can sell a product on month start easily rather than on a month end. So we should focus on more advertisement on specific browsers so as to increase the revenue generation on those browsers. However this should not be taken at face value and a statistical test should be conducted to confirm the hypothesis.



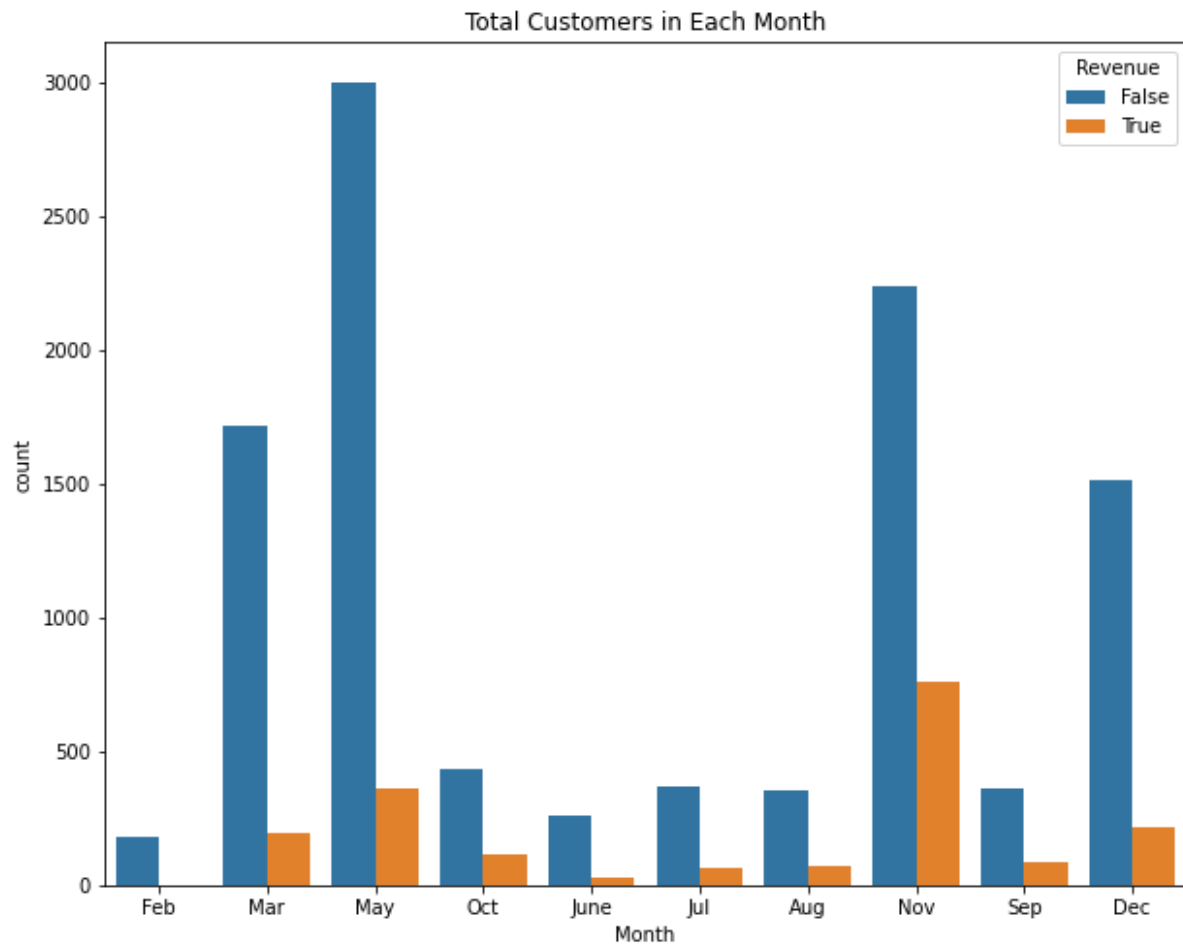
This might not be clear cut visible on the plots however there is a statistical significance of the feature with respect to the revenue.

Month

Description and Summary Statistics



Months with the highest customers are November, May, March and December. This can be attributed to holidays such as Christmas, Halloween and May has Labours day and a lot of holidays. However, we need to confirm the revenue generation against customer traffic to confirm the supposition that high customer traffic leads to higher revenue generation, even along month wise direction.



May had the highest number of customers, however the revenue generated is lower compared to the month of November. This can be attributed to the age of the e-market. The Customer traffic to revenue generation has improved over the time. If had data collection for one more year, we can check for seasonality prediction data, to check if the people follow specific buying patterns. The model could be improved with one more year of data.

Bounce Rate vs Exit Rates

These two features of the dataset are highly correlated and may act as a redundant feature for target feature. This is to be noted for future analysis.

Special Days

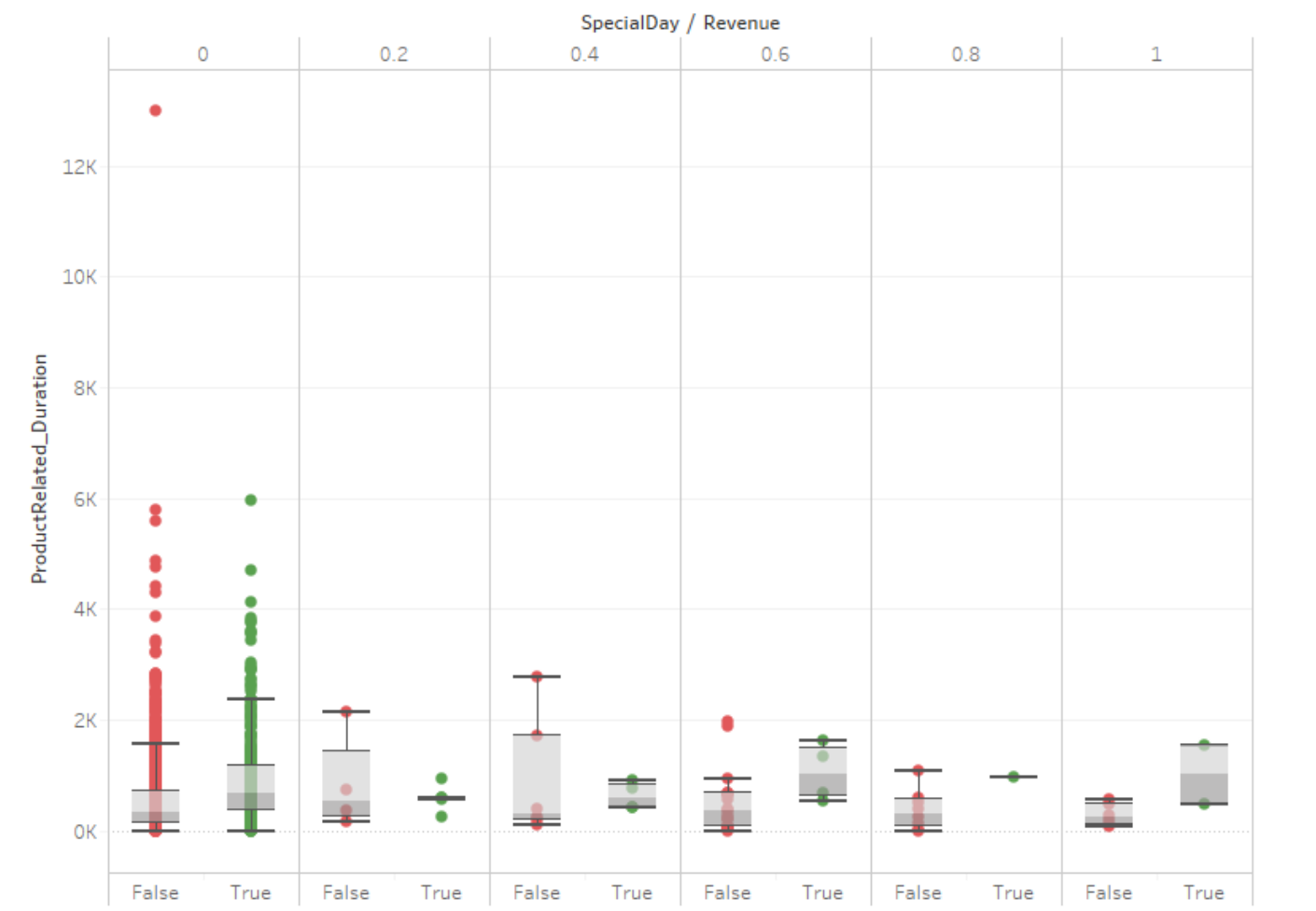
These are days that are present only in February and May. The Special days represent the closeness of the event and the buying patterns during the special event occurrence.

| Revenue | SpecialDay | VisitorType | | |
|---------|------------|-------------|-------------------|-------|
| | | New_Visitor | Returning_Visitor | Other |
| False | 0 | 1,230 | 7,949 | 69 |
| | 0.2 | 4 | 160 | |
| | 0.4 | 6 | 224 | |
| | 0.6 | 18 | 304 | |
| | 0.8 | 8 | 306 | |
| | 1 | 6 | 138 | |
| True | 0 | 406 | 1,409 | 16 |
| | 0.2 | 5 | 9 | |
| | 0.4 | 4 | 9 | |
| | 0.6 | 4 | 25 | |
| | 0.8 | 1 | 10 | |
| | 1 | 2 | 8 | |

We can clearly see that the only 23% of the total new visitors generated revenue. However, returning customers generated revenue only by 15% of the total returning customers. This might tell us that the returning customers are churning, we might need to find out the cause.

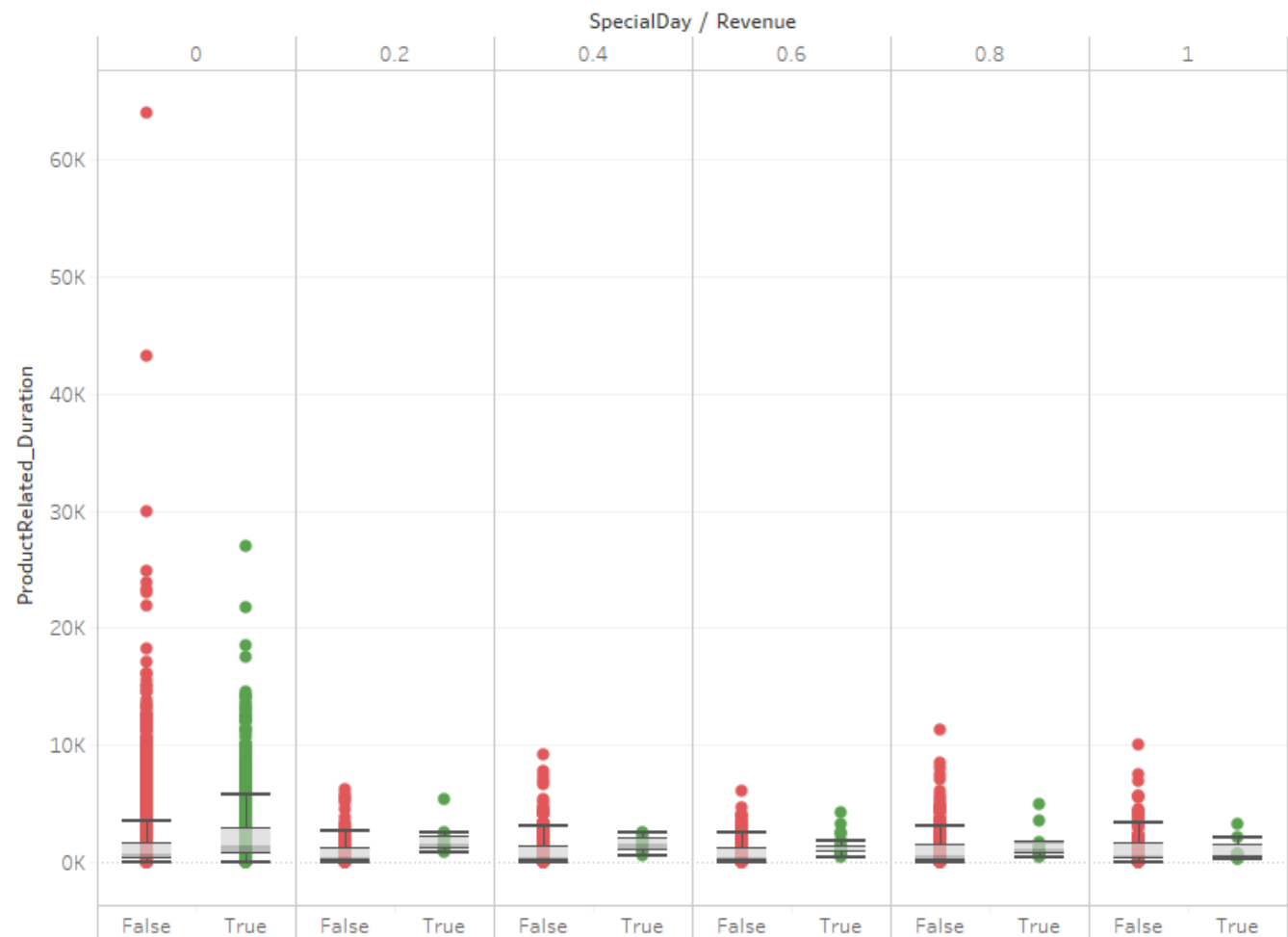
Product Related Duration of New Customers vs returning Customers.

Product Duration of New Customers



We can observe that the time spent by the new customers on product related pages is quite high for customers who generate revenue. This product duration increases when the days get closer to the special day. However the number of customers actually decreases. This means that we should focus of retaining the max customers are the first day and convert them to revenue gained.

Product Duration of Returning Customers



We can observe that the returning customers churn quite a lot compared with the new customers in terms of percentage. We can see that the generality of the customer base of the returning customers the product duration is higher in order to generate revenue.

February

Even though the present of special day, there customer revenue generation was only 1.63% of the total customers who visited the website.

| Special Day | Revenue | Customer Count |
|-------------|---------|----------------|
| 0.0 | True | 1 |
| | False | 104 |
| 0.2 | False | 15 |
| 0.4 | False | 21 |
| 0.6 | False | 19 |
| 0.8 | True | 1 |
| | False | 18 |
| 1.0 | True | 1 |
| | False | 4 |

May

May has special day, the customer revenue generation is 10.85% which is almost 10 times more when compare with February.

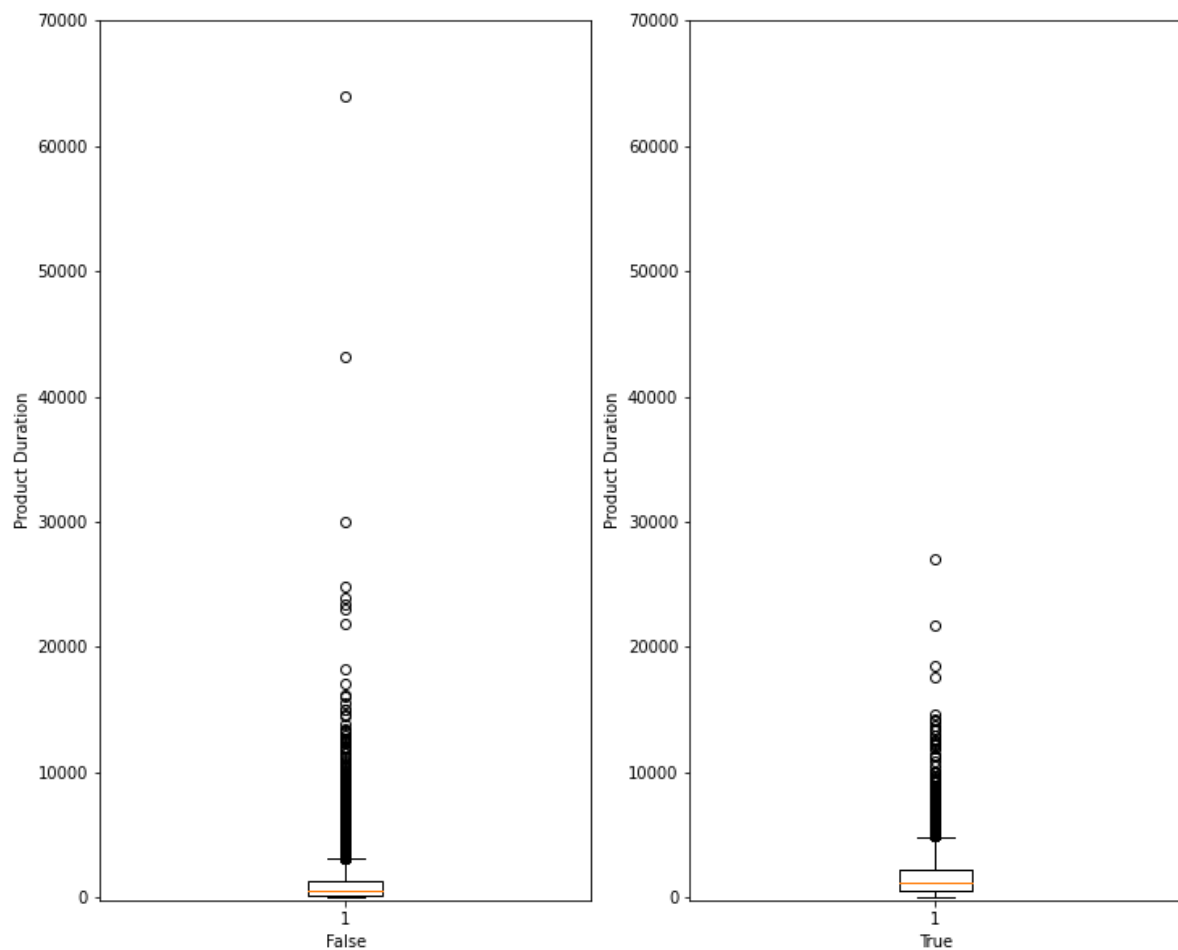
| Special Day | Revenue | Customer Count |
|-------------|---------|----------------|
| 0.0 | True | 290 |

| | | |
|-----|-------|------|
| | False | 1902 |
| 0.2 | True | 14 |
| | False | 149 |
| 0.4 | True | 13 |
| | False | 209 |
| 0.6 | True | 29 |
| | False | 303 |
| 0.8 | True | 10 |
| | False | 296 |
| 1.0 | True | 9 |
| | False | 140 |

Product Duration, Administrative Duration and Informational Duration

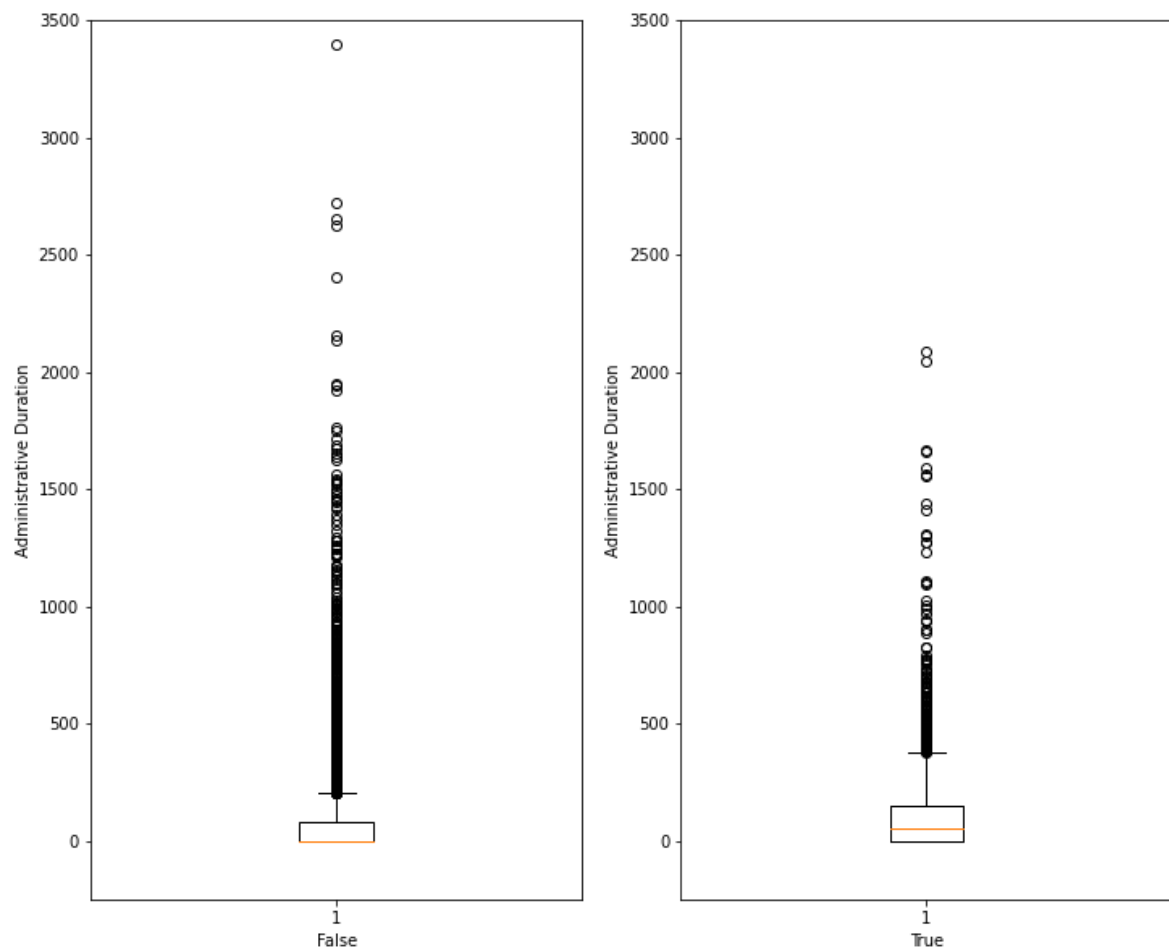
Product Duration

Boxplot Comparison of Product Duration



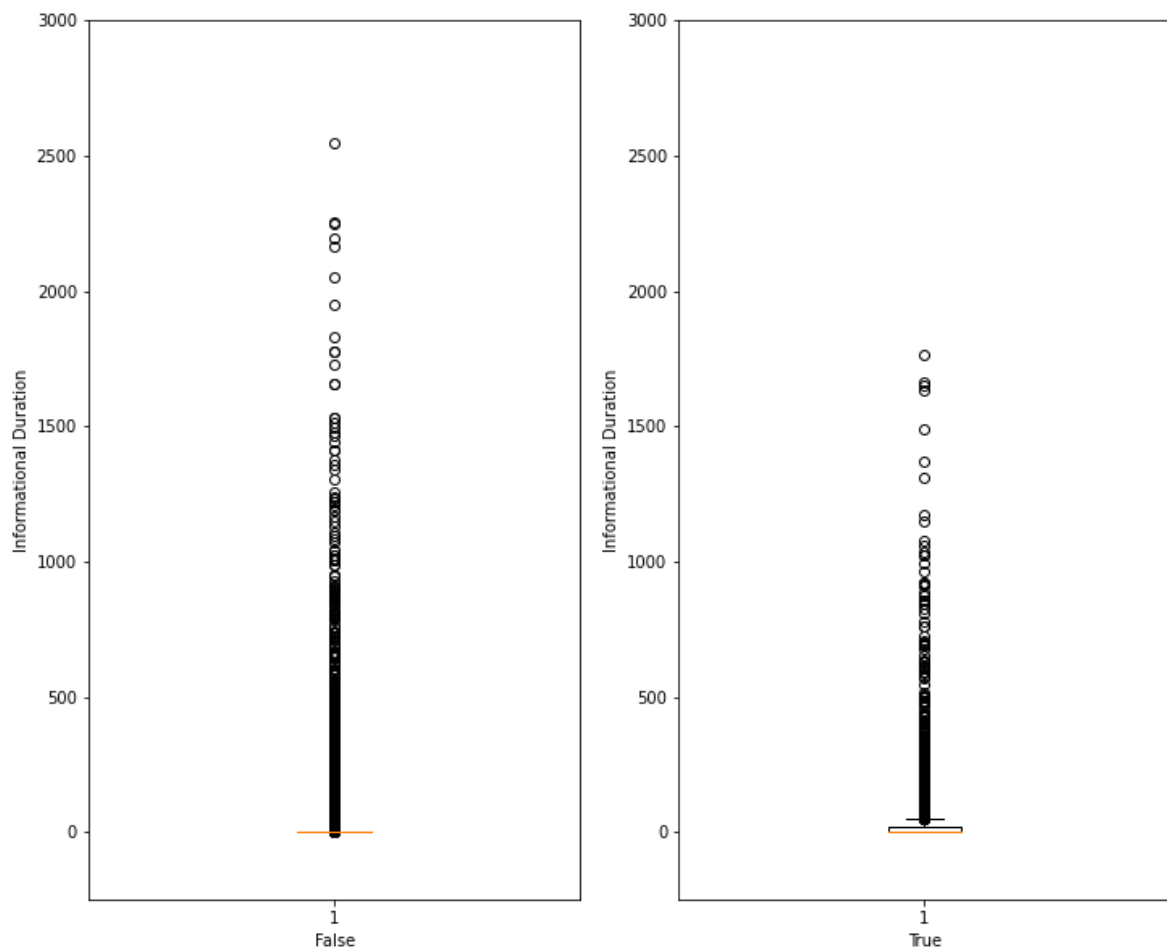
We can see a clear difference in the duration spent by the users when the revenue is true and the revenue is false. When people usually spend more time on the product related pages, the chances of converting them to positive revenue gained is higher. This is also clearly indicated on the graph, through the difference between the means when the revenue generated is true or false.

Boxplot Comparison of Administrative Duration



We can also see that there is a clear difference between the time spent by the user in administrative pages in who generate Revenue and users who don't generate revenue. A lot of users spend a lot of time in Administrative pages generate revenue. However, a lot of outliers are present in users who do not generate revenue.

Boxplot Comparison of Informational Duration



People do not generally spend time in informational pages. Hence, the people who generally spend very less time do not generate revenue. When the people do spend more on informational pages, it must probably to buy the product rather than to just skip over the products or just do window shopping. This might turn out to be an important factor along with the product duration factor.

Important Note

We can observe very high outlier values for all the three columns, Product Related Duration, Administrative Duration and Informational Duration for users who don't generate revenue. So this means that we should not allow users to get high duration times. We should try to convince the customers to buy a product aka, generate revenue rather than let them take a lot more time, since it converts positive revenue to negative. This focuses more on the importance of user interface of the website.

Data Preparation

Binning Columns

Pages are considered as a whole. We have three independent features that are dedicated to the page value counts. They are Administrative Pages, Product Related Pages and Informational Pages. We can bin users who visit the pages in Low, Medium and High Count. Once, the columns are binned their data type changes to category.

Treatment of the Independent Features

There are columns with values for denoting the types. However these values do not represent a numerical advantage over the lesser valued type. So it is better to apply One Hot Encoding method to those categorical columns, in order not to create issues with values. The categorical features are Administrative, Product Related, Informational, Month, Operation Systems, Browser, Traffic Type, Region, Visitor Type, Special Day and Weekend.

Pipelining the Data

We are pipelining the data through two pipelines, namely numerical pipeline and a categorical pipeline.

- Numerical Pipeline is to apply standard scaling on the numerical values.
- Categorical Pipeline is to apply One Hot Encoding Technique.

Then we use the column transformer to concatenate the data and then we will be using the prepared model to train our model.

```
cat_pipeline = Pipeline([
    ('encoding', OneHotEncoder())
])
```

```
num_pipeline = Pipeline([
    ('scaling', StandardScaler())
])
```

```
prep_data = ColumnTransformer([
    ('Categoricals', cat_pipeline, cat_cols),
    ('Numericals', num_pipeline, num_cols)
])
```

Models

Logistic Regression (Base Estimator)

The data doesn't have any null values. We are splitting the data into training set and a validation set at 20% split and stratified it against our target variable. We are applying Logistic Regression as our base model. We are applying SMOTE since our training dataset is imbalanced. Then we are hyper parameter tuning our model based on the following parameters.

```

params = [
    {
        'penalty':['l2'],
        'solver':['newton-cg', 'sag', 'lbfgs'],
        'C':np.arange(0,1.1,0.1)
    },
    {
        'penalty':['elasticnet'],
        'solver':['saga'],
        'C':np.arange(0,1.1,0.1)
    },
    {
        'penalty':['l1'],
        'solver':['liblinear'],
        'C':np.arange(0,1.1,0.1)
    },
    {
        'penalty':['none'],
        'solver':['newton-cg', 'sag', 'lbfgs', 'saga'],
    }
]

```

We use Grid Search Method for the best parameters. Applying cross validation score function to determine the mean recall score with 5 splits we get an Average Accuracy Score of 0.83 and Average Recall Score of 0.80 for the training data.

```

[87] print(f'Logistic Regression, Average Recall Score for the Cross Validated Model is {np.mean(lrrecscores)}')
     print(f'Logistic Regression, Variance of the Recall Scores of the Cross Validated Model is {np.std(lrrecscores,ddof=1)}')

```

```

↳ Logistic Regression, Average Recall Score for the Cross Validated Model is 0.8066678031300214
   Logistic Regression, Variance of the Recall Scores of the Cross Validated Model is 0.005992712222184504

```

```

[88] print(f'Logistic Regression, Average Accuracy Score for the Cross Validated Model is {np.mean(lraccscores)}')
     print(f'Logistic Regression, Variance of the Accuracy Scores of the Cross Validated Model is {np.std(lraccscores,ddof=1)}')

```

```

↳ Logistic Regression, Average Accuracy Score for the Cross Validated Model is 0.8328735452417675
   Logistic Regression, Variance of the Accuracy Scores of the Cross Validated Model is 0.005359732448113184

```

The Classification Report of the Logistic Regression Model is,

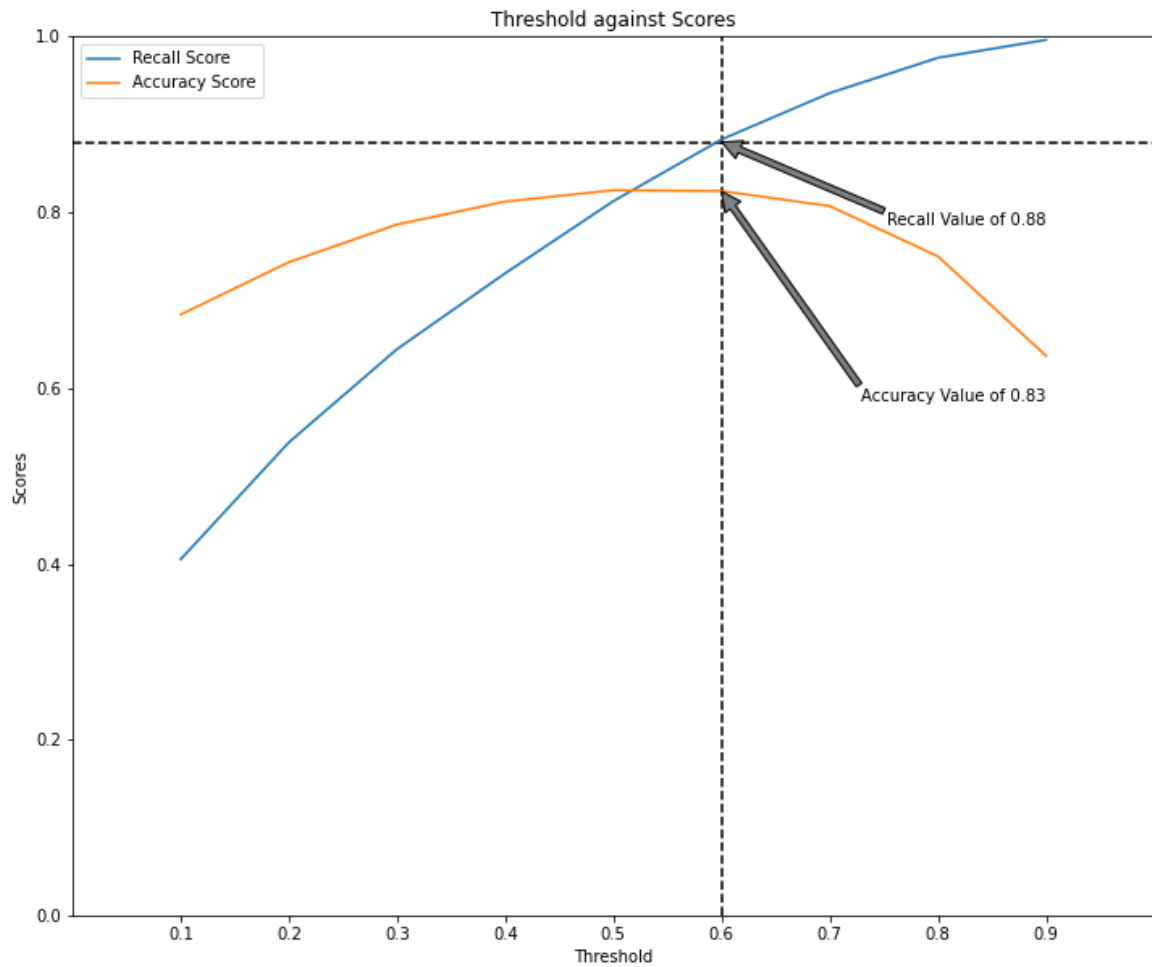
```

[55] print(classification_report(y_test,y_pred))

```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.82 | 0.84 | 0.83 | 1668 |
| True | 0.83 | 0.81 | 0.82 | 1668 |
| accuracy | | | 0.83 | 3336 |
| macro avg | 0.83 | 0.83 | 0.83 | 3336 |
| weighted avg | 0.83 | 0.83 | 0.83 | 3336 |

We are focusing on the recall score, because we want to predict the customer who can generate revenue for the e-market from the dataset. We can adjust the threshold in order to get a better recall score against losing accuracy.



So adjusting the threshold at 60% increases the recall score to 88%.

Classification Report of the adjusted threshold,

```
[79] print(classification_report(y_test,y_pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.87 | 0.77 | 0.82 | 1668 |
| True | 0.79 | 0.89 | 0.84 | 1668 |
| accuracy | | | 0.83 | 3336 |
| macro avg | 0.83 | 0.83 | 0.83 | 3336 |
| weighted avg | 0.83 | 0.83 | 0.83 | 3336 |

We have a higher recall score with overall accuracy at 60% compared with our previous model. The cross validation scoring for 5 splits is done in order to get overall accuracy and recall for a generalized model for new threshold.


```
[75] print(f'Average Recall Score for the Cross Validated Model is {np.mean(lrrecscores)}')
      print(f'Variance of the Recall Scores of the Cross Validated Model is {np.std(lrrecscores,ddof=1)}')
```

```
↳ Average Recall Score for the Cross Validated Model is 0.8843832672314458
   Variance of the Recall Scores of the Cross Validated Model is 0.006483473015985492
```

```
[76] print(f'Average Accuracy Score for the Cross Validated Model is {np.mean(lraccscores)}')
      print(f'Variance of the Accuracy Scores of the Cross Validated Model is {np.std(lraccscores,ddof=1)}')
```

```
↳ Average Accuracy Score for the Cross Validated Model is 0.8371309489140323
   Variance of the Accuracy Scores of the Cross Validated Model is 0.007129441717898257
```

After the new Threshold, we are getting same accuracy with better recall score for the training dataset.

Decision Tree Classifier

We are applying Decision Tree Classifier with hyper tuned parameters set for max recall score. The tuned parameter values are,

```
[98] gs.best_params_
```

```
↳ {'criterion': 'gini', 'max_features': None, 'splitter': 'best'}
```

```
[99] gs.best_score_
```

```
↳ 0.8981259370314841
```

The Accuracy and the Recall scores of the tuned decision tree model for the test data are, 0.89 and 0.91 respectively.

```
[103] print(f'Tuned Decision Tree Accuracy score:{accuracy_score(y_test,y_pred)}')
       print(f'Tuned Decision Tree Recall score:{recall_score(y_test,y_pred)}')
```

```
↳ Tuned Decision Tree Accuracy score:0.894484412470024
   Tuned Decision Tree Recall score:0.9124700239808153
```

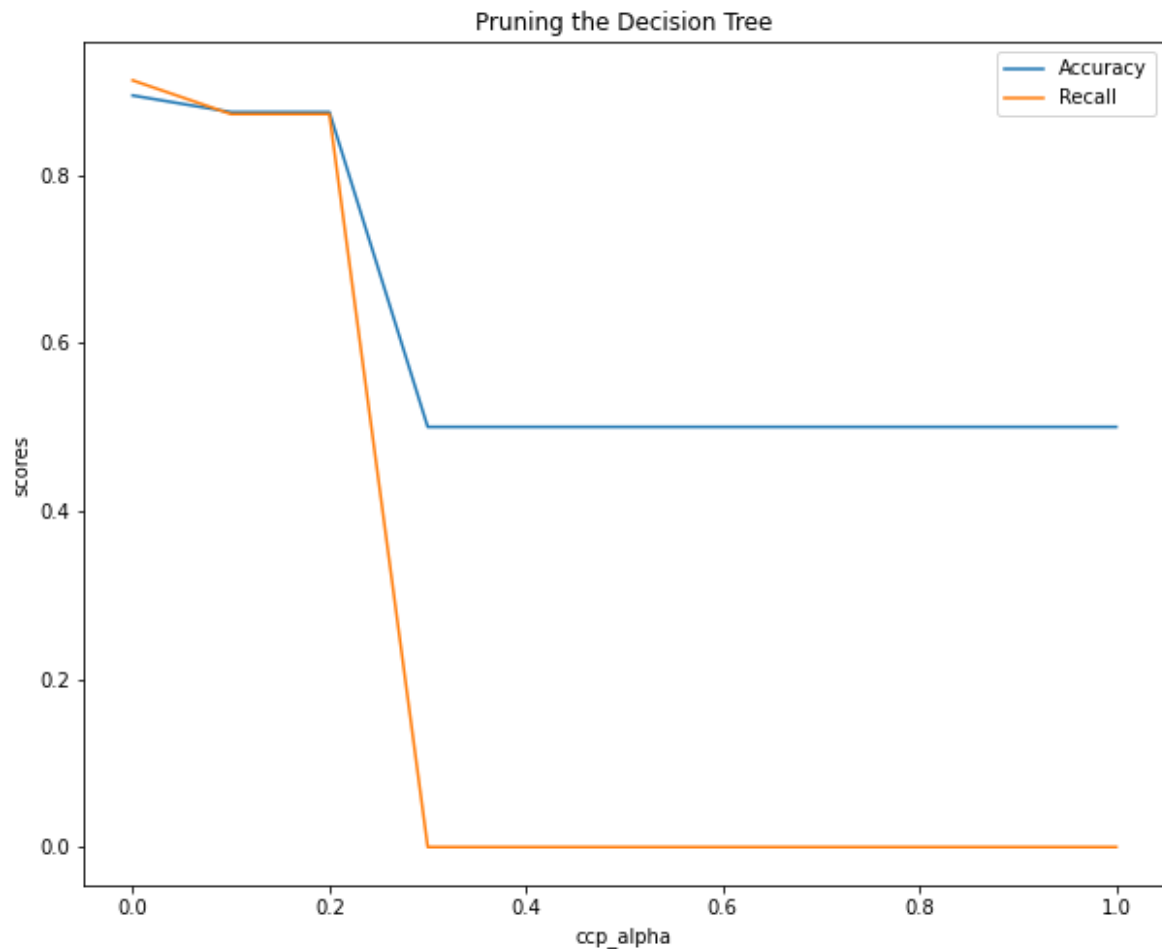
Classification Report of the Tuned Decision Tree model is,

```
▶ print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.91 | 0.88 | 0.89 | 1668 |
| True | 0.88 | 0.91 | 0.90 | 1668 |
| accuracy | | | 0.89 | 3336 |
| macro avg | 0.89 | 0.89 | 0.89 | 3336 |
| weighted avg | 0.89 | 0.89 | 0.89 | 3336 |

Pruning the Decision Tree model, to get a better accuracy and recall score.



We can see that the best scored for both accuracy and recall is at cost complexity pruning value is at 0.0. Applying cross validation scoring technique to find the average recall and accuracy score of the model.

The classification report of the Final Decision Tree model is,

```
[113] print(classification_report(y_test,y_pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.91 | 0.88 | 0.89 | 1668 |
| True | 0.88 | 0.91 | 0.90 | 1668 |
| accuracy | | | 0.89 | 3336 |
| macro avg | 0.89 | 0.89 | 0.89 | 3336 |
| weighted avg | 0.89 | 0.89 | 0.89 | 3336 |

The Accuracy and the recall score of our model with the generalized model is given in the following image.

```
[115] print(f'Decision Tree, Recall Score of the test set is {recall_score(y_test,y_pred)}')
      print(f'Decision Tree, Accuracy Score of the test set is {accuracy_score(y_test,y_pred)}')
```

```
Decision Tree, Recall Score of the test set is 0.9124700239808153
Decision Tree, Accuracy Score of the test set is 0.894484412470024
```

Applying Stratified KFold validation scoring technique to find the average recall score and the variance of the recall score of the Decision Tree model.

```
[131] print(f'The Average Accuracy Score of the Decision Tree model is {np.mean(dtaccscores)}')  
      print(f'The Variance of the Accuracy Scores of the Decision Tree model is {np.std(dtaccscores,ddof=1)}')
```

```
➤ The Average Accuracy Score of the Decision Tree model is 0.9023747299012363  
  The Variance of the Accuracy Scores of the Decision Tree model is 0.005067681740485135
```

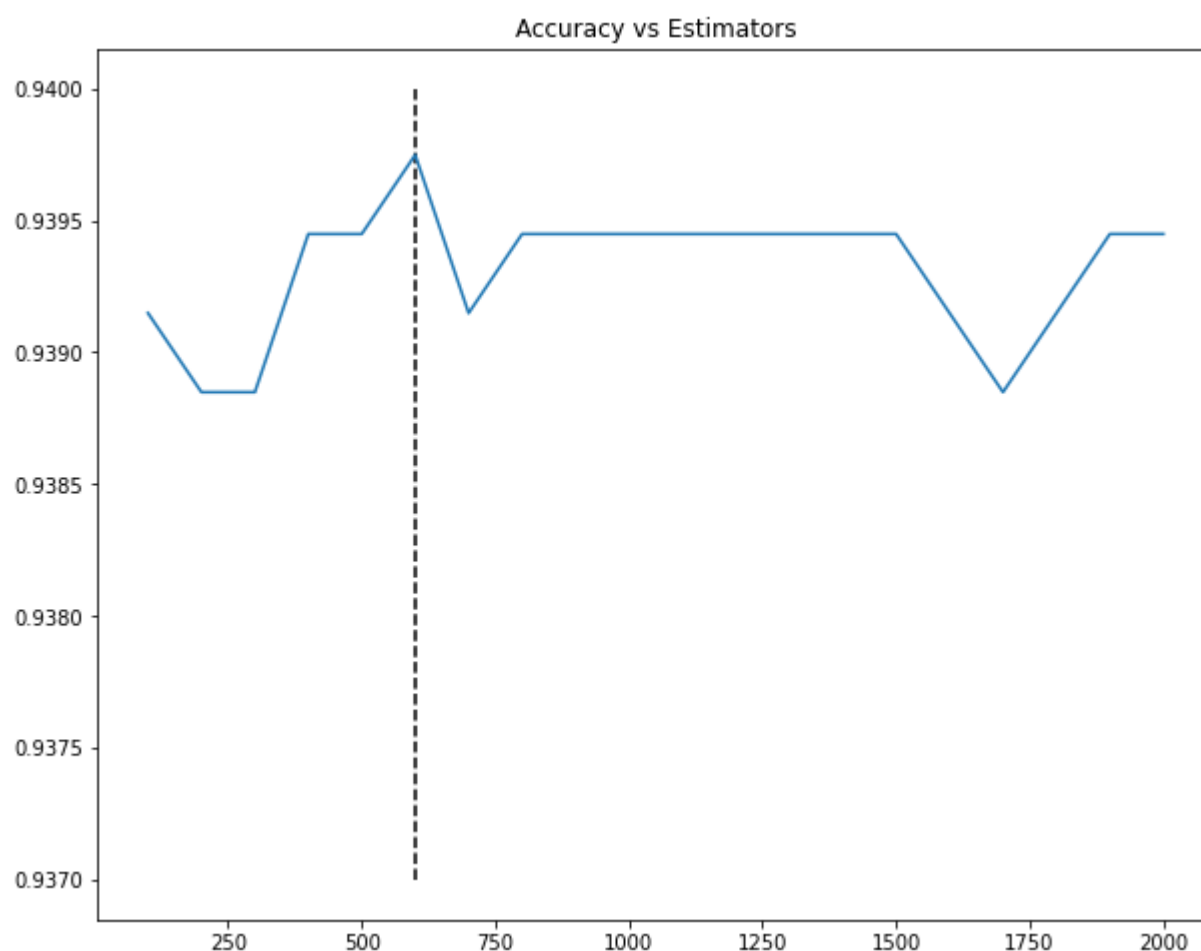
```
[123] print(f'The Average Recall Score of the Decision Tree model is {np.mean(dtreccscores)}')  
      print(f'The Variance of the Recall Scores of the Decision Tree model is {np.std(dtreccscores,ddof=1)}')
```

```
➤ The Average Recall Score of the Decision Tree model is 0.9117291649583752  
  The Variance of the Recall Scores of the Decision Tree model is 0.0024619262453125655
```

The Stratified KFold Average Recall and Accuracy scores for the Decision Tree model are, 0.91 and 0.90 respectively.

Random Forest Classifier

We are going to determine the number of estimators we need to get good accuracy on the trainings data.



We can determine to have 600 estimators for better accuracy for the trainings data. The classification report of the Random Forest Classifier for 600 estimators is,

```
[140] print(f'Random Forest Accuracy Score:{accuracy_score(y_test,y_pred)}')  
      print(f'Random Forest Recall Score:{recall_score(y_test,y_pred)}')
```

```
➤ Random Forest Accuracy Score:0.939748201438849  
  Random Forest Recall Score:0.9592326139088729
```

The Classification Report of Random Forest Tuned Model is,

```
[141] print(classification_report(y_test,y_pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.96 | 0.92 | 0.94 | 1668 |
| True | 0.92 | 0.96 | 0.94 | 1668 |
| accuracy | | | 0.94 | 3336 |
| macro avg | 0.94 | 0.94 | 0.94 | 3336 |
| weighted avg | 0.94 | 0.94 | 0.94 | 3336 |

On applying Stratified KFold Cross Validation,

```
[146] print(f'Tuned Random Forest Classifier Average Accuracy Score:{np.mean(rfaccscores)}')  
print(f'Tuned Random Forest Classifier Variance Scores:{np.std(rfaccscores,ddof=1)}')
```

```
↳ Tuned Random Forest Classifier Average Accuracy Score:0.9399739518729844  
Tuned Random Forest Classifier Variance Scores:0.005422925808685955
```

```
[147] print(f'Tuned Random Forest Classifier Average Recall Score:{np.mean(rfrecscores)}')  
print(f'Tuned Random Forest Classifier Variance Scores:{np.std(rfrecscores,ddof=1)}')
```

```
↳ Tuned Random Forest Classifier Average Recall Score:0.9565834315151358  
Tuned Random Forest Classifier Variance Scores:0.005073163648892156
```

The Average Accuracy and Recall Scores of the Tuned Random Forest model for the training data is, 0.93 and 0.95 respectively.

Adaptive Boosting Classifier

We are using Decision Tree Classifier as the base model, with its best tuned parameters. We are using max depth of 2 with 500 estimators. The Accuracy, Recall and the Classification Report of the model are,

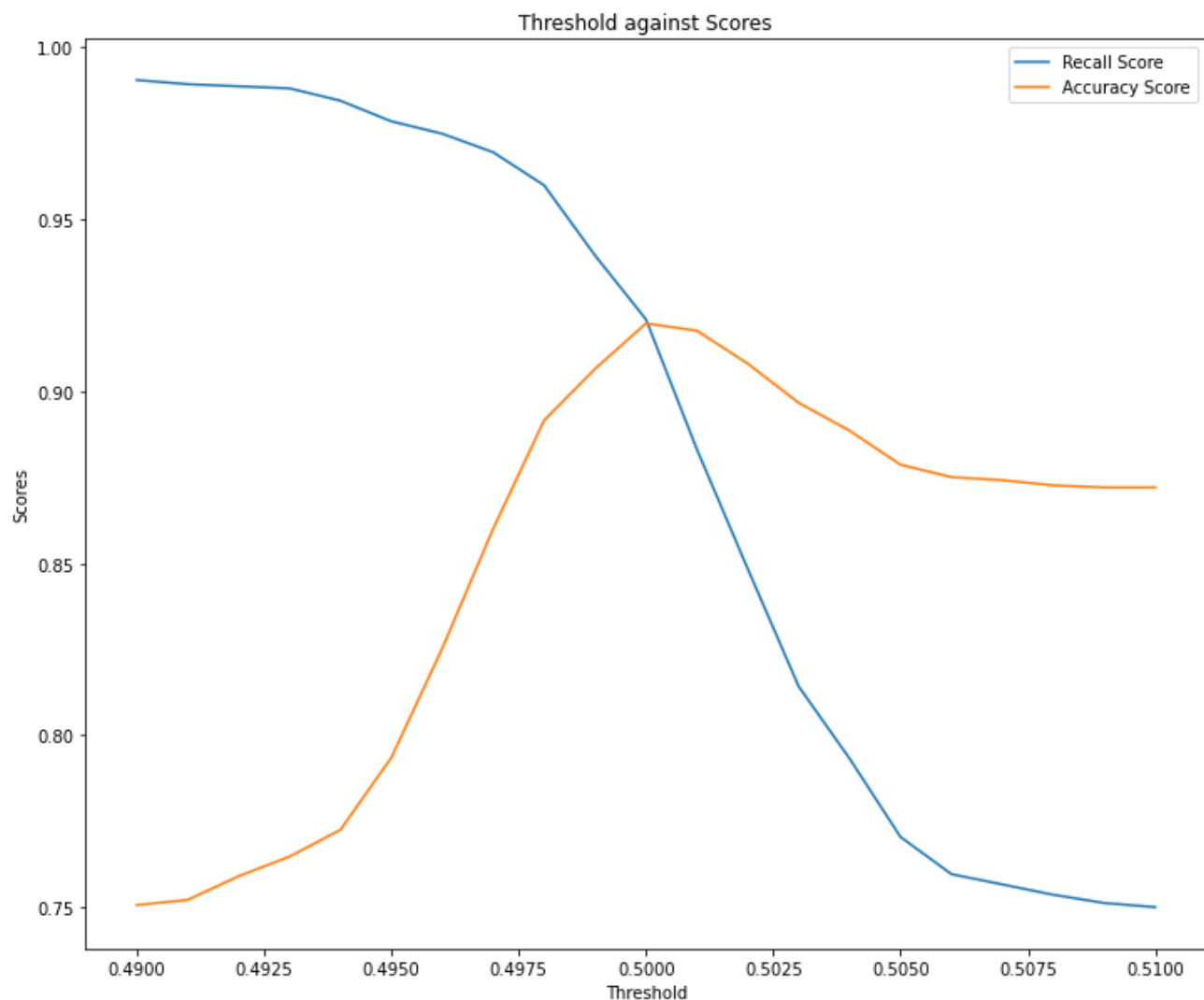
```
[151] print(f'Accuracy Score for the Adaptive Boosting Classifier is {accuracy_score(y_test,y_pred)}')  
print(f'Recall Score for the Adaptive Boosting Classifier is {recall_score(y_test,y_pred)}')
```

```
↳ Accuracy Score for the Adaptive Boosting Classifier is 0.919664268585132  
Recall Score for the Adaptive Boosting Classifier is 0.920863309352518
```

```
[152] print(classification_report(y_test,y_pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.92 | 0.92 | 0.92 | 1668 |
| True | 0.92 | 0.92 | 0.92 | 1668 |
| accuracy | | | 0.92 | 3336 |
| macro avg | 0.92 | 0.92 | 0.92 | 3336 |
| weighted avg | 0.92 | 0.92 | 0.92 | 3336 |

The Threshold against the Accuracy vs Recall Scores values are plotted.



Setting the new threshold to 0.4975 for better recall and accuracy scores.

The new Threshold Accuracy, Recall and Classification report for the test data is,

```
[153] y_pred = ab.predict_proba(X_test)[: ,1]>0.4975
```

```
[154] print(f'Accuracy Score for the Adaptive Boosting Classifier is {accuracy_score(y_test,y_pred)}')
      print(f'Recall Score for the Adaptive Boosting Classifier is {recall_score(y_test,y_pred)}')
```

```
➤ Accuracy Score for the Adaptive Boosting Classifier is 0.875599520383693
  Recall Score for the Adaptive Boosting Classifier is 0.9634292565947242
```

```
[155] print(classification_report(y_test,y_pred))
```

```
➤
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.96 | 0.79 | 0.86 | 1668 |
| True | 0.82 | 0.96 | 0.89 | 1668 |
| accuracy | | | 0.88 | 3336 |
| macro avg | 0.89 | 0.88 | 0.87 | 3336 |
| weighted avg | 0.89 | 0.88 | 0.87 | 3336 |

On Applying Stratified KFold Cross Validation Method, the average accuracy and recall scores are,

```
[160] print(f'Average Accuracy Score for New Threshold in Ada Boosting :{np.mean(abaccscores)}')
      print(f'Variance in Accuracy Scores for New Threshold in Ada Boosting : {np.std(abaccscores,ddof=1)}')
```

```
➤ Average Accuracy Score for New Threshold in Ada Boosting :0.8764617691154423
  Variance in Accuracy Scores for New Threshold in Ada Boosting : 0.003243374460687239
```

```
[161] print(f'Average Recall Score for New Threshold in Ada Boosting :{np.mean(abrecscores)}')
      print(f'Variance in Recall Scores for New Threshold in Ada Boosting : {np.std(abrecscores,ddof=1)}')
```

```
➤ Average Recall Score for New Threshold in Ada Boosting :0.9652213442203644
  Variance in Recall Scores for New Threshold in Ada Boosting : 0.002388377826381591
```

Gradient Boosting Classifier

We are using Gradient Boosting Classifier with loss as exponential and max depth 2 with 500 estimators. The recall and the accuracy score of the gradient boost classifier on the training and the test data points are,

```
[179] print(f'The Gradient Boost Classifier Accuracy Score:{accuracy_score(y_test,y_pred)}')
      print(f'The Gradient Boost Classifier Recall Score:{recall_score(y_test,y_pred)}')
```

```
➤ The Gradient Boost Classifier Accuracy Score:0.9229616306954437
  The Gradient Boost Classifier Recall Score:0.934052757793765
```

```
[180] print(classification_report(y_test,y_pred))
```

```
➤
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.93 | 0.91 | 0.92 | 1668 |
| True | 0.91 | 0.93 | 0.92 | 1668 |
| accuracy | | | 0.92 | 3336 |
| macro avg | 0.92 | 0.92 | 0.92 | 3336 |
| weighted avg | 0.92 | 0.92 | 0.92 | 3336 |

On Application of Stratified KFold Cross Validation,

```
[184] print(f'Gradient Boost Classifier Model Average Accuracy Score:{np.mean(acc)}')
      print(f'Gradient Boost Classifier Model Variance of Accuracy Score:{np.std(acc,ddof=1)}')
```

```
➤ Gradient Boost Classifier Model Average Accuracy Score:0.928400368161243
  Gradient Boost Classifier Model Variance of Accuracy Score:0.00463675752191195
```

```
[185] print(f'Gradient Boost Classifier Model Average Recall Score:{np.mean(rcs)}')
      print(f'Gradient Boost Classifier Model Variance of Recall Scores:{np.std(rcs,ddof=1)}')
```

```
➤ Gradient Boost Classifier Model Average Recall Score:0.9343967177787464
  Gradient Boost Classifier Model Variance of Recall Scores:0.0040798942809289335
```

Gradient Boost Classifier has very high Accuracy with respect to recall, compared to all the models as of now.

Support Vector Machines Classifier

We are using the kernel rbf after applying a Grid Search for the tuned parameters. The Accuracy, Recall and the Classification report of the model on the test data are,

```
[189] print(f'Support Vector Classifier Accuracy Score:{accuracy_score(y_test,y_pred)}')
      print(f'Support Vector Classifier Recall Score:{recall_score(y_test,y_pred)}')
```

```
↳ Support Vector Classifier Accuracy Score:0.8794964028776978
   Support Vector Classifier Recall Score:0.8938848920863309
```

```
[190] print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.89 | 0.87 | 0.88 | 1668 |
| True | 0.87 | 0.89 | 0.88 | 1668 |
| accuracy | | | 0.88 | 3336 |
| macro avg | 0.88 | 0.88 | 0.88 | 3336 |
| weighted avg | 0.88 | 0.88 | 0.88 | 3336 |

Stratified KFold Cross validation of the Support vector Classifier model is done and the scores are,

```
[196] print(f'Support Vector Classifier Average Accuracy Score:{np.mean(acc)}')
      print(f'Support Vector Classifier Variance of Accuracy:{np.std(acc,ddof=1)}')
```

```
↳ Support Vector Classifier Average Accuracy Score:0.8841448520344144
   Support Vector Classifier Variance of Accuracy:0.002031104135405102
```

```
[197] print(f'Support Vector Classifier Average Recall Score:{np.mean(rcs)}')
      print(f'Support Vector Classifier Variance of Recall:{np.std(rcs,ddof=1)}')
```

```
↳ Support Vector Classifier Average Recall Score:0.8759893345071992
   Support Vector Classifier Variance of Recall:0.004825879581167832
```

K Neighbours Classifier

The tuned KN Model Accuracy, Recall and Classification Report of the trained and the test data, are

```
[205] print(f'KNeighbors Accuracy Score:{accuracy_score(y_test,y_pred)}')
      print(f'KNeighbors Recall Score:{recall_score(y_test,y_pred)}')
```

```
↳ KNeighbors Accuracy Score:0.8872901678657075
   KNeighbors Recall Score:0.9904076738609112
```

```
[206] print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.99 | 0.78 | 0.87 | 1668 |
| True | 0.82 | 0.99 | 0.90 | 1668 |
| accuracy | | | 0.89 | 3336 |
| macro avg | 0.90 | 0.89 | 0.89 | 3336 |
| weighted avg | 0.90 | 0.89 | 0.89 | 3336 |

Stratified KFold Cross validation scores are


```
[208] print(f'KNeighbours Average Accuracy Score:{np.mean(acc)}')
      print(f'KNeighbours Variance of Accuracy:{np.std(acc,ddof=1)}')
```

```
↳ KNeighbours Average Accuracy Score:0.896737728258173
   KNeighbours Variance of Accuracy:0.003167308773548507
```

```
[209] print(f'KNeighbours Average Recall Score:{np.mean(rcs)}')
      print(f'KNeighbours Variance of Recall:{np.std(rcs,ddof=1)}')
```

```
↳ KNeighbours Average Recall Score:0.9884860437984345
   KNeighbours Variance of Recall:0.0051151777186626566
```

Bagging Classifier

The tuned Bagging Classifier scores are,

```
[214] print(f'Bagging Classifier Accuracy Score:{accuracy_score(y_test,y_pred)}')
      print(f'Bagging Classifier Recall Score:{recall_score(y_test,y_pred)}')
```

```
↳ Bagging Classifier Accuracy Score:0.9256594724220624
   Bagging Classifier Recall Score:0.9502398081534772
```

```
[215] print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.95 | 0.90 | 0.92 | 1668 |
| True | 0.91 | 0.95 | 0.93 | 1668 |
| accuracy | | | 0.93 | 3336 |
| macro avg | 0.93 | 0.93 | 0.93 | 3336 |
| weighted avg | 0.93 | 0.93 | 0.93 | 3336 |

Cross validation Score of the tuned Bagging Classifier model are,

```
[217] print(f'Bagging Classifier Average Accuracy Score:{np.mean(acc)}')
      print(f'Bagging Classifier Variance of Accuracy:{np.std(acc,ddof=1)}')
```

```
↳ KNeighbours Average Accuracy Score:0.930679030988103
   KNeighbours Variance of Accuracy:0.005738595371688532
```

```
[219] print(f'Bagging Classifier Average Recall Score:{np.mean(rcs)}')
      print(f'Bagging Classifier Variance of Recall:{np.std(rcs,ddof=1)}')
```

```
↳ Bagging Classifier Average Recall Score:0.9469881563255693
   Bagging Classifier Variance of Recall:0.006700996038818823
```

Gaussian Naïve Bayes

The Base Model Scores are,


```
[224] print(f'The GNB Classifier Accuracy Score:{accuracy_score(y_test,y_pred)}')
      print(f'The GNB Classifier Recall Score:{recall_score(y_test,y_pred)}')
```

```
↳ The GNB Classifier Accuracy Score:0.6019184652278178
   The GNB Classifier Recall Score:0.959832134292566
```

```
[225] print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.86 | 0.24 | 0.38 | 1668 |
| True | 0.56 | 0.96 | 0.71 | 1668 |
| accuracy | | | 0.60 | 3336 |
| macro avg | 0.71 | 0.60 | 0.54 | 3336 |
| weighted avg | 0.71 | 0.60 | 0.54 | 3336 |

Cross Validation Scores

```
[229] print(f'GNB Classifier Model Average Accuracy Score:{np.mean(acc)}')
      print(f'GNB Classifier Model Variance of Accuracy Score:{np.std(acc,ddof=1)}')
```

```
↳ GNB Classifier Model Average Accuracy Score:0.6012832612470744
   GNB Classifier Model Variance of Accuracy Score:0.006166211705098695
```

```
[230] print(f'GNB Classifier Model Average Recall Score:{np.mean(rcs)}')
      print(f'GNB Classifier Model Variance of Recall Score:{np.std(rcs,ddof=1)}')
```

```
↳ GNB Classifier Model Average Recall Score:0.9588608896925649
   GNB Classifier Model Variance of Recall Score:0.009816645145161326
```

XGBoost Random Forest Classifier

XGB Classifiers Tuned Model scores are,

```
[234] print(f'Accuracy Score for the XG Boosting Classifier is {accuracy_score(y_test,y_pred)}')
      print(f'Recall Score for the XG Boosting Classifier is {recall_score(y_test,y_pred)}')
```

```
↳ Accuracy Score for the XG Boosting Classifier is 0.8192446043165468
   Recall Score for the XG Boosting Classifier is 0.9610311750599521
```

```
[235] print(classification_report(y_test,y_pred))
```

```
↳
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| False | 0.95 | 0.68 | 0.79 | 1668 |
| True | 0.75 | 0.96 | 0.84 | 1668 |
| accuracy | | | 0.82 | 3336 |
| macro avg | 0.85 | 0.82 | 0.82 | 3336 |
| weighted avg | 0.85 | 0.82 | 0.82 | 3336 |

Cross Validated Scores

```
[239] print(f'XGB Classifier Average Accuracy Score:{np.mean(acc)}')
      print(f'XGB Classifier Variance of Accuracy:{np.std(acc,ddof=1)}')
```

```
↳ XGB Classifier Average Accuracy Score:0.8336531554366701
   XGB Classifier Variance of Accuracy:0.005694696061272896
```

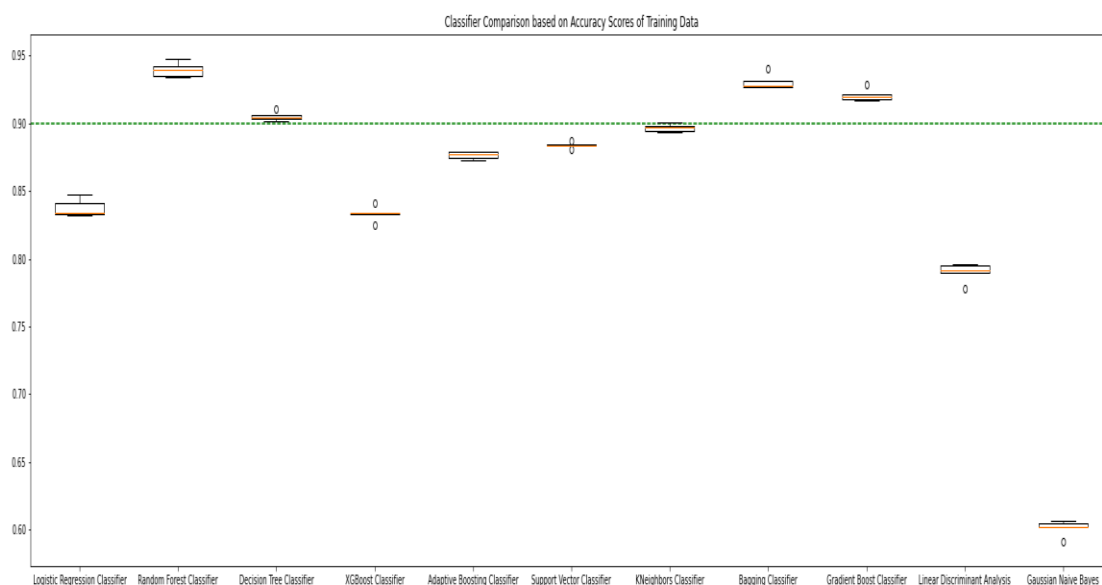
```
[241] print(f'XGB Classifier Average Accuracy Score:{np.mean(rcs)}')
      print(f'XGB Classifier Variance of Accuracy:{np.std(rcs,ddof=1)}')
```

```
↳ XGB Classifier Average Accuracy Score:0.968218298786286
   XGB Classifier Variance of Accuracy:0.00305108673739877
```

Model Selection

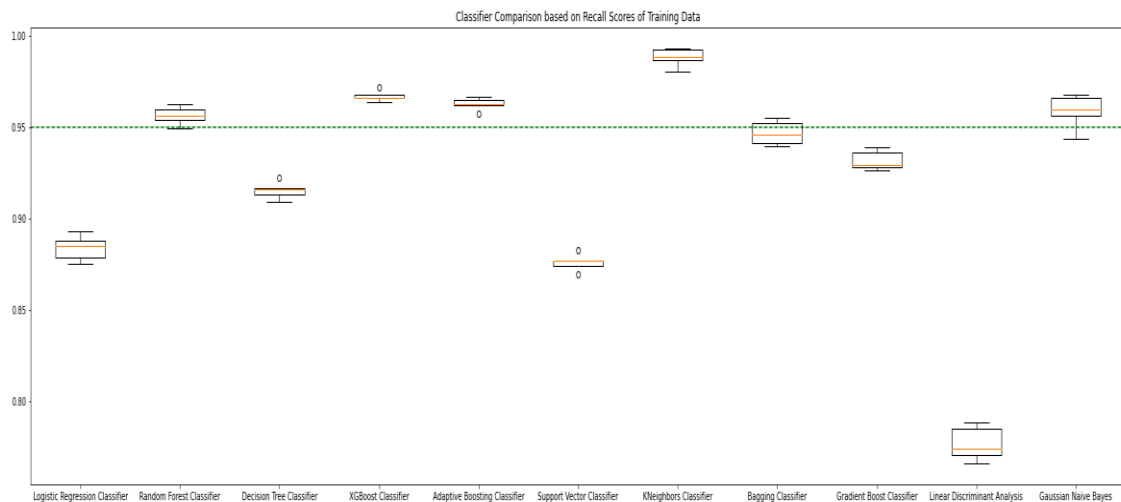
Comparison of Cross Validated Scores of all the Models

Accuracy Scores



Based on the accuracy score the Random Forest Classifier proves to be a better model closely followed by Bagging and Gradient Boosting.

Recall Scores



Based on the recall scores K-Neighbours is the best performer with Random Forest, XGBoost, Adaptive Boost, Bagging and Gaussian Naïve Bayes also above the 95% mark.

Applying Validation Set to Classifiers

```
[19] print('Scores on Validation Set')
pd.DataFrame(scores).T
```

➤ Scores on Validation Set

| | Accuracy Score | Recall Score |
|---|----------------|--------------|
| Logistic Regression Classifier | 0.845093 | 0.753927 |
| Random Forest Classifier | 0.900243 | 0.725131 |
| Decision Tree Classifier | 0.856853 | 0.625654 |
| XGBoost Random Forest Classifier | 0.725466 | 0.942408 |
| Adaptive Boosting Classifier | 0.801298 | 0.803665 |
| Support Vector Classifier | 0.868613 | 0.743455 |
| KNeighbors Classifier | 0.784266 | 0.704188 |
| Bagging Classifier | 0.889294 | 0.691099 |
| Gradient Boost Classifier | 0.888483 | 0.751309 |
| Linear Discriminant Analysis | 0.786294 | 0.738220 |
| Gaussian Naive Bayes | 0.348743 | 0.926702 |

Conclusion

The Accuracy Score of a model shows how many correct predictions were made of the total predictions. Hence it is required that the accuracy of the model be high along with the dataset being balanced. On the other hand Recall is the number of correct positive predictions divided by the number of all relevant predictions. Recall score tells us how much of the Customers who generate Revenue are predicted correctly out of the total Customer who Generated Revenue. Hence a high recall score would mean we can predict the number of customers generate revenue correctly.

Accuracy can be sacrificed for a better recall score and the reasoning for that is, loss potential customers would be high if our recall score is low and there by our model predicting erroneously. Even if accuracy is lower with a higher recall score model, it could be used persuade customers by taking actions to make them buy, by other factors such as Product Related Page Duration times, reducing bounce and exit rates, etc.

Looking at Cross Validated Model's Accuracy and Recall scores, the best fit model is the Random Forest Classifier with 94% accuracy and 96% recall, in terms of the customer making a purchase. The recall score for the customer not making a purchase is about 92%. This model provides the company the chance to better predict the customer intentions based on their habits on the website and the knowledge on how to target and push the right customers to increase the overall chances of converting a visit into a purchase. With more and more customers moving towards online purchasing this knowledge will be the boon necessary for a company to survive and grow in the online market.

However, when the models were evaluate with the validation set, XGBoost Random Forest Classifier performs better in terms of Recall and Accuracy score, with 94% Recall Score and 72% Accuracy Rate. It would be best to choose this model on the basis that accuracy can be sacrificed for a better recall score, with the aim of serving a larger customer base and trying to convert them into generating revenue rather than, aiming at a selected few customers with a higher accuracy.

The Final Model Selected would be XGB Random Forest Classifier for predicting and Customers Intention to Purchase from the particular dataset that we have collected.