
# ILLINOIS INSTITUTE OF TECHNOLOGY

CSP 571 Data Preparation and Analysis

Summer – 2024 Project

# Analysis on Online Shoppers Purchasing Intention

**Divya Sai Sree Chintala**
Computer Science Department
Illinois Institute of Technology
dchintala@hawk.iit.edu
A20561001

**Sneha Joshi**
Computer Science Department
Illinois Institute of Technology
sjoshi32@hawk.iit.edu
A20540613

**Under Guidance**

**of**

## Professor Jawahar Panchal

# Table of Contents

# 1. Abstract:

In recent years, with the rapid development of online shopping sites, more and more consumers are accustomed to online shopping. In fact, shopping sites invest a lot of money and manpower to improve shopping applications to discover different items of interest to consumers. By more accurately predicting consumers' preferences for products, shopping websites can more accurately push the products they are interested into consumers. This not only improves the experience of consumers in the online shopping process and the efficiency of shopping, but also improves the competitiveness and performance of shopping websites. This paper discusses the influence of consumer behavior on online shopping desire from four aspects: cultural factors, social factors, personal factors and psychological factors. The influence of consumer acceptance on online shopping intention is discussed. Further, this paper introduces several common methods for predicting user purchase behavior, including collaborative filtering algorithm and hybrid recommendation method. For each method, the paper also provides advantages and disadvantages in their use, which will provide reference for further research related to this topic.

*Keywords: Online shoppers, statistical models, predictions, Exploratory Data Analysis (EDA)*

# 2. Introduction:

All over the world, the use of e-commerce or online stores has increased rapidly in recent years. The growth of the internet and the rise in online shopping have led many retail companies to develop their own e-commerce websites. Every action a customer takes on these websites can be monitored, recorded, and studied to better understand how customers behave and how it relates to their purchases. Ecommerce is another word for online stores, which is a way of shopping used to process buying and selling transactions, where sellers and buyers don't have to worry about going to the store to see, buy and sell what they are looking for and want. It's just that they can view the goods online, place the desired order, then transfer the money, and then the goods will be sent to the buyer's house through a courier service, without the need to go out to the store.

In society today, a vast majority of people have embraced online shopping this shift in shopping experience even being much more accelerated with the coming of the pandemic. Shoppers now see the act of shopping online as one which saves their time and offers more convenience [1]. Ever since the coming of the internet, online shopping has been a popular way of everyday people making their purchases. A lot of individuals look for other alternative ways to shop and online shopping has easily become the perfect fit for that. According to Cho and Sagynov [3], online shopping refers to the action of buying products as well as services from stores and merchants who sell across the internet. A critical way of describing it is also a form of involving electronic commerce in the process of trade which gives consumers an opportunity to directly buy products and services from a particular seller through the internet with the help of a web browser or even a mobile app.

## 2.2 Research Goal:

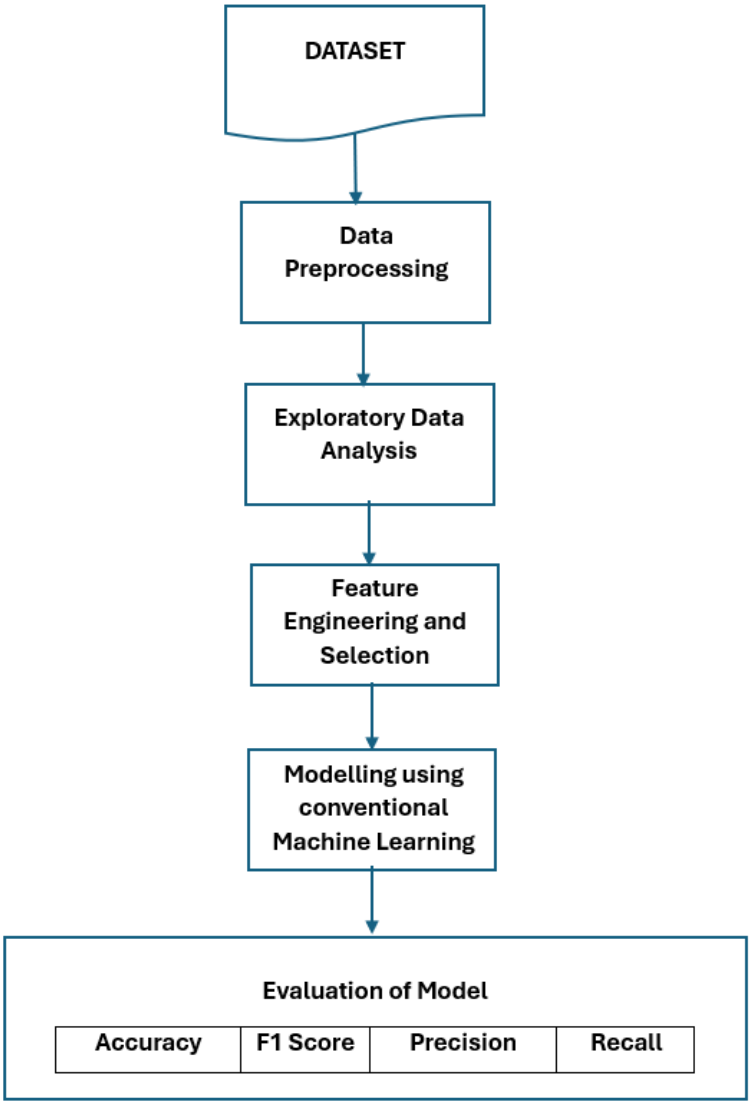
Our objective is to analyze trends in online shoppers' purchasing intention dataset using exploratory data analysis techniques and build machine learning models to predict the purchasing intentions of visitors to a store's website. We plan to approach this as both a clustering and classification problem, grouping similar customers based on their purchasing behavior and predicting whether a new customer is likely to make a purchase based on their browsing and purchasing behavior. By applying these techniques, we hope to gain insights into customer behavior and optimize marketing strategies to increase sales.

## 2.3 Specific Questions: What this project seeks to address

- What are the key factors that influence a customer's decision to make a purchase on an e-commerce website?
- What is the impact of session information and browsing behavior on the accuracy of purchase predictions?
- How can online shoppers be grouped based on their purchasing behavior?
- Can we accurately predict whether a new visitor to the website will make a purchase?
- What is the impact of different features of the data such as session information and browsing behavior have on the accuracy of purchase predictions?

## 2.4 Proposed Methodology

Our proposed methodology involves steps such as data collection, preprocessing, feature selection, modeling, and evaluation. The process starts with collecting data from the UCI Machine Learning Repository. The data is then transformed and preprocessed. Features are selected and conventional machine learning algorithms are tested with cross-validation to evaluate the model.
We will be implementing machine learning algorithms to predict purchase intention. The algorithms include Naive Bayes Classifier, K Nearest Neighbor, Random Forest, Support Vector Machines (SVM), and XGBoost for classification. Each of these algorithms will be trained and evaluated using different performance metrics such as accuracy, precision, recall, and F1 score. We would like to apply clustering algorithms to segment the users based on their purchasing behavior. The clustering algorithms that we use in this project include K-means clustering, DBSCAN clustering, and Hierarchical clustering. These algorithms aim to group similar users together and identify patterns in their behavior.




## 3. Data Source:

- Online Shoppers Purchasing Intention Dataset used in this project is sourced from the UC Irvine Machine Learning Repository, specifically designed to study online shoppers' purchasing intentions and it consists of feature vectors belonging to 12,330 sessions. https://doi.org/10.24432/C5F88Q.
- We are exploring additional datasets for comparative analysis or for enhancing the model's training phase, if available.

## Data set contributors:

- C. Okan Sakar, Department of Computer Engineering, Bahcesehir University, Istanbul.
- Yomi Kastro, Inveon Information Technologies Consultancy and Trade, Istanbul, Turkey.

## 3.1 Dataset Description:

The dataset consists of feature vectors belonging to 12,330 sessions, so this dataset has 12,330 observations with 18 features. The dataset consists of both numerical and categorical attributes. The *'Revenue'* attribute is used as the class label.

## Feature Description:

| Attribute | Type | Description |
|---|---|---|
| Administrative | Numerical | Page category |
| Administrative Duration | Numerical | Total time spent in this page |
| Informational | Numerical | Page category |
| Informational Duration | Numerical | Total time spent in this page |
| Product Related | Numerical | Page category |
| Product Related Duration | Continuous | Total time spent in this page |
| Bounce rate | Continuous | The bounce rate for a web page isthe percentage of visitors who navigate away from the site after viewing only that page, withoutinteracting with the page or visiting any other pages on the site. So, it's not just about the visitors who enter the site fromthat page, but rather about visitors who land on the page and thenleave without taking any further action. |
| Exit rate | Continuous | The exit rate for a web page is the percentage of visitors who leave the site after viewing that page as the last page in their session. Unlike bounce rate, which considers the visitors who leave after viewing a single page, the exit rate includesvisitors who may have viewed multiple pages on the site before leaving after viewing the specific page in question. To calculate the exit rate for a specific web page, you would divide the number of exits from that page by the |

| | | |
|---|---|---|
| | | total number of page views for that page. |
| Page value | Numerical | The Page Value feature is ametric in Google Analytics that represents the average value of a page that a user visited before completing an e-commerce transaction or a goal conversion on a website. It is calculated by dividing the total value of all transactions or goal completionsby the number of unique page views for a particular page or set of pages. |
| Special day | Numerical | The *Special Day* feature 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. Thevalue of this attribute is determined by considering thedynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentina's day, thisvalue takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8. |
| Operating system | Numerical | Operating system of the visitor. |
| Browser | Numerical | Browser of the visitor. |
| Region | Numerical | Geographic region from which the session has been started by the visitor. |
| Traffic type | Numerical | Traffic sources by which the visitor has arrived at the website. |
| Visitor type | Categorical | Visitor type as "New visitor", "Returning Visitor" and "Other" |
| Weekend | Categorical | True if it is either Saturday orSunday. Or else it is False. |
| Month of the year | Categorical | Jan, Feb…..,December |
| Revenue | Categorical | True is customer purchased anything or else it is False |

## 3.3 Data Processing:

- **Checking number of observations with NA values**: The dataset does not contain any missing values.

```
sum(is.na(data))
```

```
## [1] 0
```

- **Converting Month feature data type to factor data type :** Initially the Month feature column data type is character. "as.factor" is used to convert month column to factor data type, this is used when the column is categorical attribute with fixed number of categories, in our cases it is months of a year.

- **Fixing naming convention of month names in Month column:** First check count of number of observations for each month.

```
Aug  Dec  Feb  Jul June  Mar  May  Nov  Oct  Sep
433 1727  184  432  288 1907 3364 2998  549  448
```

Use "as.factor" to convert month column to factor data type from character data type, this is used when the column is categorical attribute with fixed number of categories, in our cases it is months of a year.

```
Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
  0  184 1907    0 3364    0  432  433  448  549 2998 1727
```

Even though the dataset includes data for the month of June, when we convert the data type to factor, we observe that the number of observations for June is zero. This is because in the original dataset, June is encoded as "June" instead of "Jun". Therefore, we need to convert "June" to "Jun" and then convert the data type back to factor.

We have observed that when using the table function on the Month column of the dataset, it shows the total number of observations for the month of Jun.

```
Aug  Dec  Feb  Jul  Jun  Mar  May  Nov  Oct  Sep
433 1727  184  432  288 1907 3364 2998  549  448
```

Now, when we analyze the data using the "str" function, we can observe that the Month column is ordered with 12 levels.

***$ Month : Factor w/ 12 levels "Jan","Feb","Mar",..: 2 2 2 2 2 2 2 2 2 2 …***

- **Transforming categorical attributes into "factor" data type and then perform one-hot encoding**: Categorical attributes are: Operating Systems, Browser, Region, Traffic Type, VisitorType.

- **Converting Revenue attribute data type to a factor**

- **Transforming Boolean attributes into "int" data type:** Boolean attributes: Weekend, Revenue.

- A copy of the dataset, one with one-hot encoding and other without encoding are to be kept seeing which data would achieve better accuracy/result.

- Creating train-test split for the dataset, one with one-hot encoding and the other without, using a 70/30 proportion for training and testing.

- **One hot encoded train and test data pre-processing**
  - Separating numerical and categorical attributes
  - Scaling the numerical attributes. Scale function is used to standardize by subtracting mean for each value and dividing by standard deviation, this bring value to have mean zero and standard deviation of one.
  - Combining categorical and scales numerical attributes.
  - There is huge imbalance in data set as Revenue=0 is the majority. Hence the algorithm tries to over fit on majority class. Hence we use "ovun.sample" to over-sample minority class. The function tries to generate synthetic data points of minority class using "SMOTE" algorithm, this creates new observation by interpolating between the given sample in feature space. Here we are trying to increase minority class observations by two times.
  - Oversampling is done only to the train set.
  - There are 77 columns in data after one hot encoding, in which last column is target variable.


- Preprocessing the training and testing data for the dataset without one-hot encoding

- There is a huge imbalance in the data set as Revenue=0 is the majority. Hence the algorithm tries to over fit on majority class. Hence, we use “ovun.sample” to over-sample minority class. The function tries to generate synthetic data points of minority class using “SMOTE” algorithm, this creates new observation by interpolating between the given sample in feature space. Here we are trying to increase minority class observations by two times. Oversampling is done only to the train set.

## 3.4 Data Stylized Facts

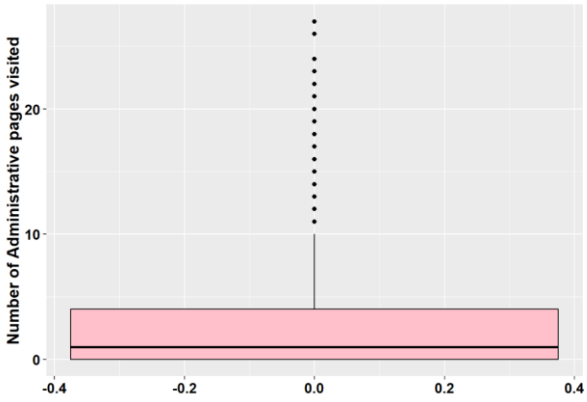
**Clustering**: We will be using K-means clustering that segments user based on browsing behavior, while hierarchical clustering reveals detailed relationships between user segments.

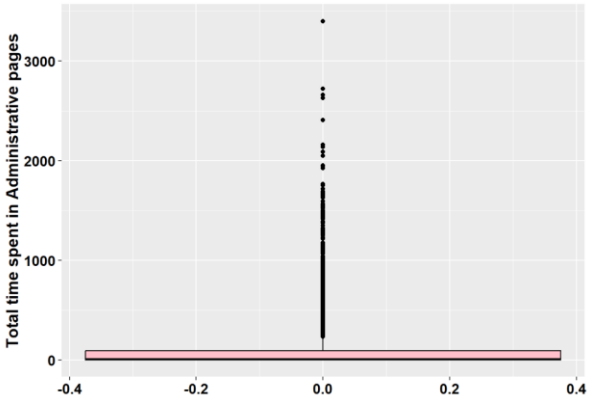
**Dimensionality Reduction**: High correlations between certain features like Product Related and Product Related Duration leads to selecting key features using Recursive Feature Elimination (RFE) to improve model performance and reduce complexity.

# 4. Exploratory Data Analysis:

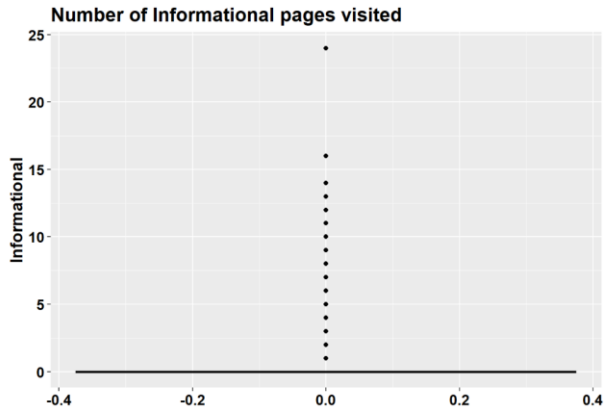
## 4.1 Exploring data distribution of different page categories and time spent in it:

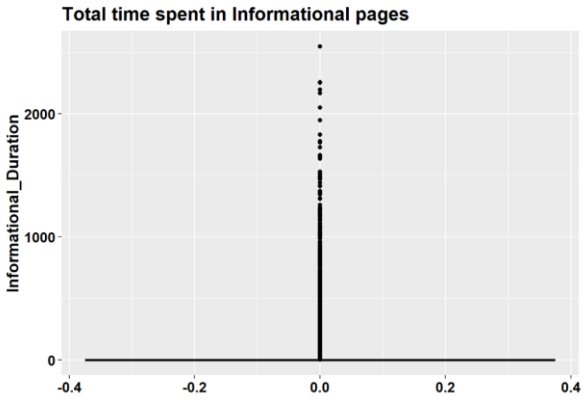
1. **Exploring data pattern of "Administrative pages" and "Administrative_Duration":** The boxplots illustrate the distribution of total time spent in administrative pages and the number of administrative pages visited by users. The presence of numerous outliers, especially in the total time spent, indicates that while most users spend minimal time and visit few administrative pages, a small number of users exhibit significantly higher engagement in these areas. The median values are low for both metrics, suggesting that most users do not extensively interact with administrative pages



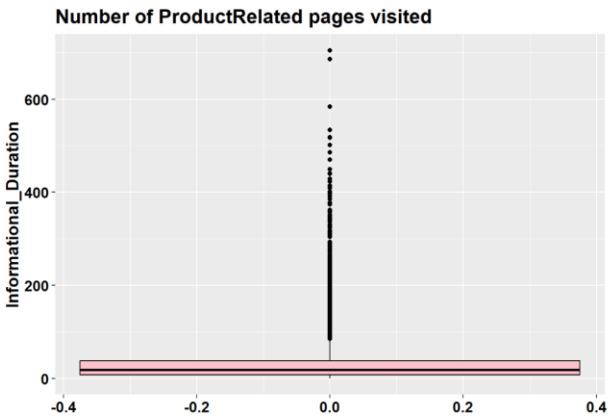



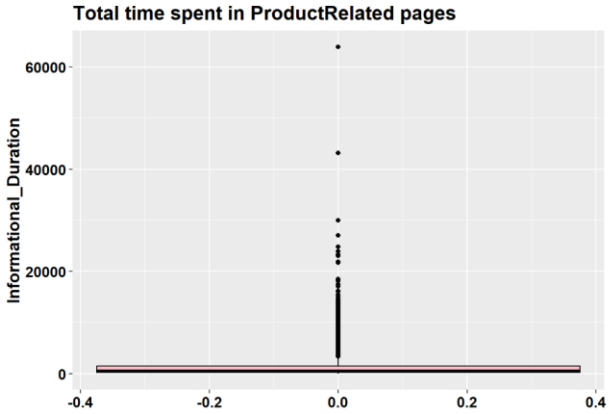

2. **Exploring data pattern of "Informational" and "Informational_Duration":** The boxplots for the number of informational pages visited and the total time spent on informational pages show that most users visit very few informational pages and spend minimal time on them. A small number of users engage significantly more with these pages. The median values for both metrics are very low, highlighting the overall limited interest in informational pages among users.



3. **Exploring data pattern of "Product Related" and "Product Related Duration":** The boxplots for the number of product-related pages visited and the total time spent on product-related pages indicate that while most users visit a moderate number of these pages and spend a significant amount of time on them, there are some extreme outliers. These outliers suggest that a few users engage much more extensively with product-related content, visiting many more pages and spending considerably more time compared to the average user. The median values are relatively higher compared to other categories, highlighting a general trend of greater interest and engagement in product-related pages.




**Analyzing summary of all above plots, we can observe that:**

- Median of Number of Administrative pages visited= 1
- Median of Number of Informational pages visited= 0
- Median of Number of Product Related pages visited= 18

- Median time spent in Administrative pages= 7.5
- Median time spent in Informational Related pages= 0
- Median time spent in Product Related pages= 598.9369

- Analyzing number of pages visit of 3 different page categories it clearly says that customers are interested more in Product related pages rather than knowing information of the product in detail.

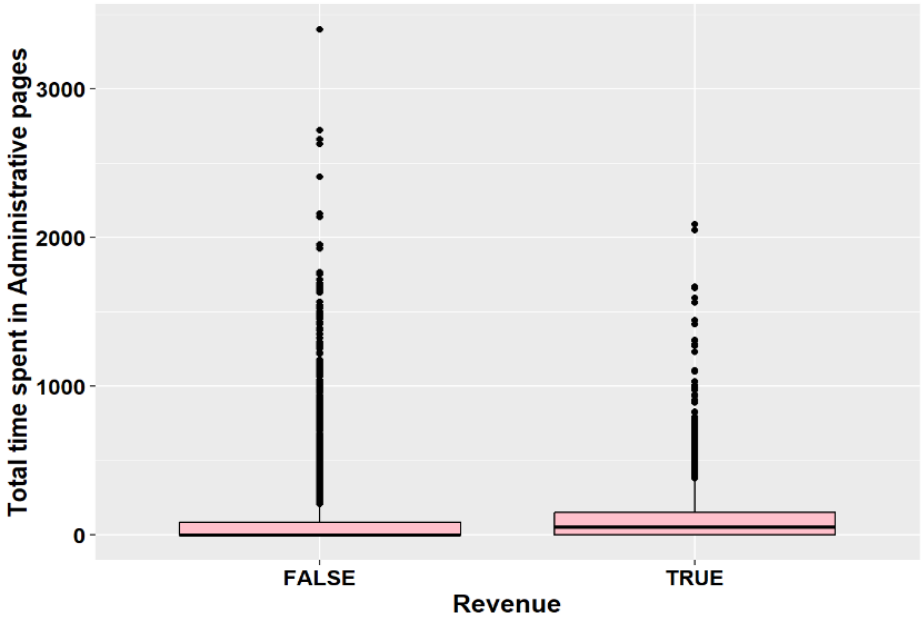
- Analyzing total time spent in 3 different page categories, it clearly says that customers spend most of the time in product related pages whereas they are not interested in spending time in information related pages.

## 4.2 Exploring the data distribution of different page categories versus the target variable Revenue, as well as the time spent on each page category versus the target variable Revenue.

#### 4.2.1.1 Exploring data pattern of "Administrative" versus "Revenue".

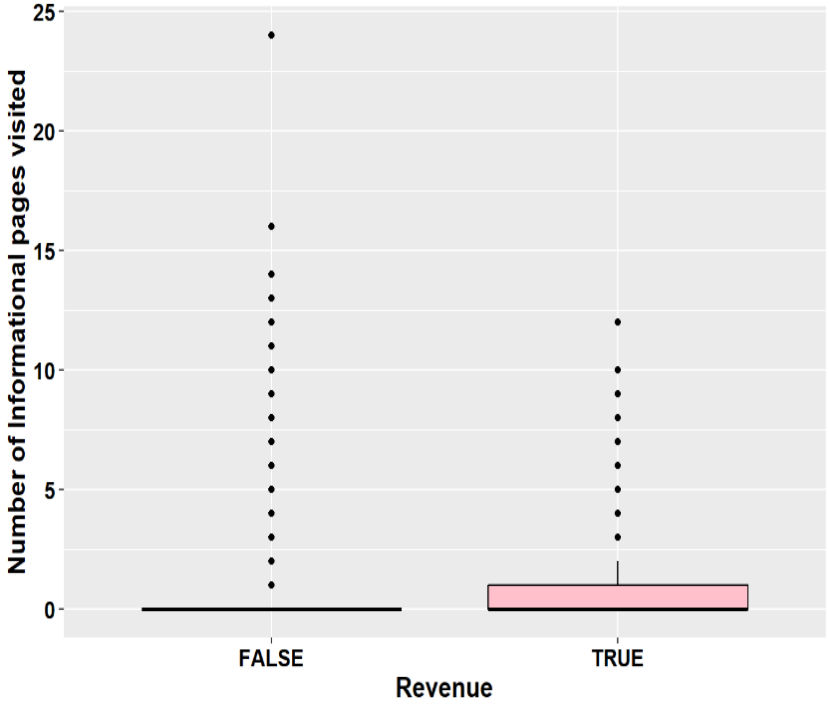

- *Median no.of administrative pages visited by a customer who end up buying = 2.*
- *Median no.of administrative pages visited by a customer who did not end up buying= 0*

#### 4.2.1.2 Exploring data pattern of "Administrative_Duration" versus "Revenue"



- *Median duration of time spent in administrative pages by a customer who end up buying is= 52.36667*
- *Median duration of time spent in administrative pages by a customer who did not end up buying is= 0*

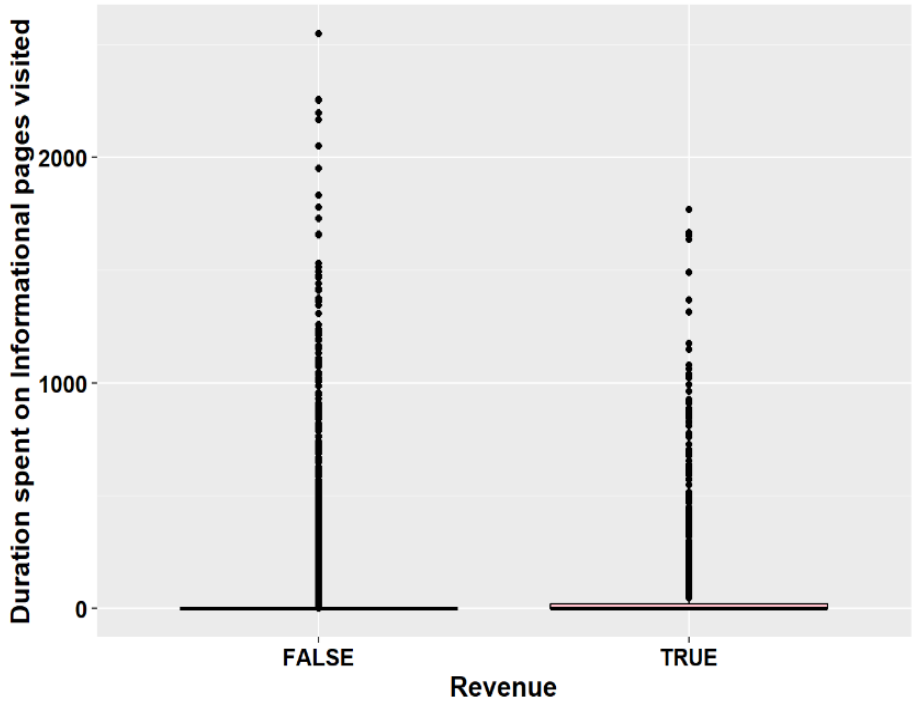
#### 4.2.2.1 Exploring data pattern of "Informational" versus "Revenue"



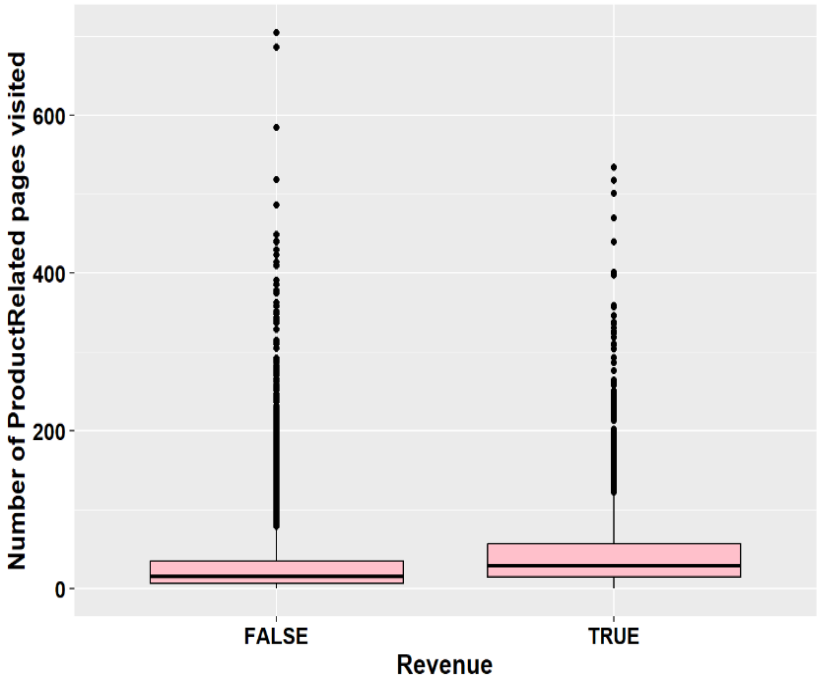

- *Median no of Informational pages visited by a customer who end up buying is= 0.*
- *Median no of Informational pages visited by a customer who did not end up buying is= 0*

#### 4.2.2.2 Exploring data pattern of "Informational_Duration" versus "Revenue"

- *Median no of duration spent on Informational pages by a customer who end up buying is= 0*
- *Median no of duration spent on Informational pages by a customer who did not end up buying is= 0*

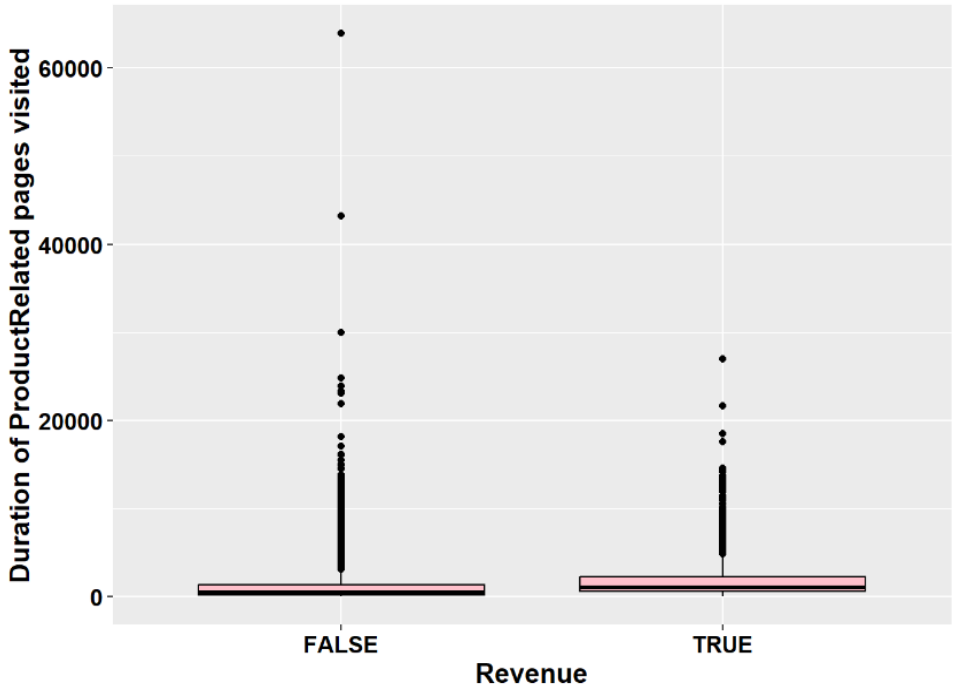



### 4.2.3 Exploring data pattern of "Product Related" versus "Revenue"




- *Median no of Product Related pages visited by customer who end up buying= 29.*
- *Median no of Product Related pages visited by a customer who did not end up buying is= 16*

#### 4.2.3.2 Exploring data pattern of “Product Related Duration” versus “Revenue”

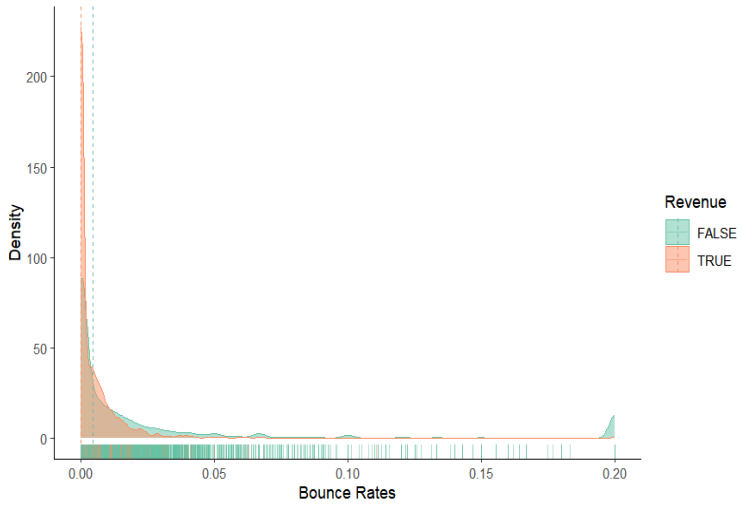


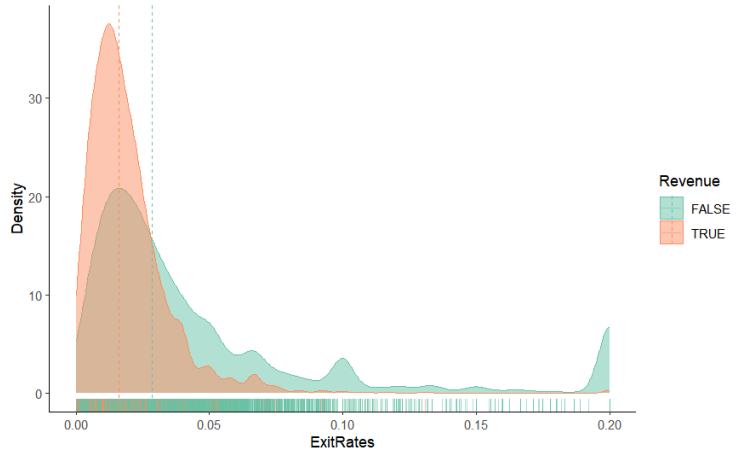

- *Median duration of Product Related pages visited by a customer who end up buying is= 1109.906*
- *Median duration of Product Related pages visited by a customer who did not end up buying is= 510.19*

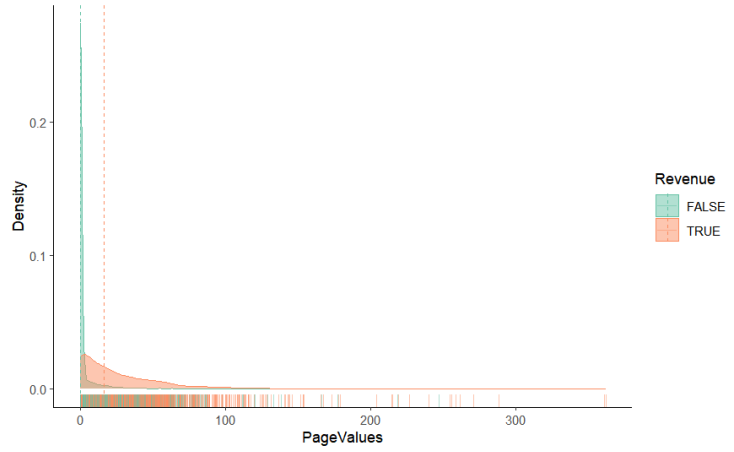
**Based on above analysis, these are the observations made:**

- People who end up buying will mostly visit administrative page and spend almost 52seconds.
- People who end up not buying will mostly not visit administrative page.
- People are least interested in visiting informational page.
- People who end up buying will mostly visit product related page and spend almost 1109 seconds.
- People who will end up buying will mostly visit product related page and spend almost 510 seconds.
- But people who end up buying will visit more product related than the ones who don't.

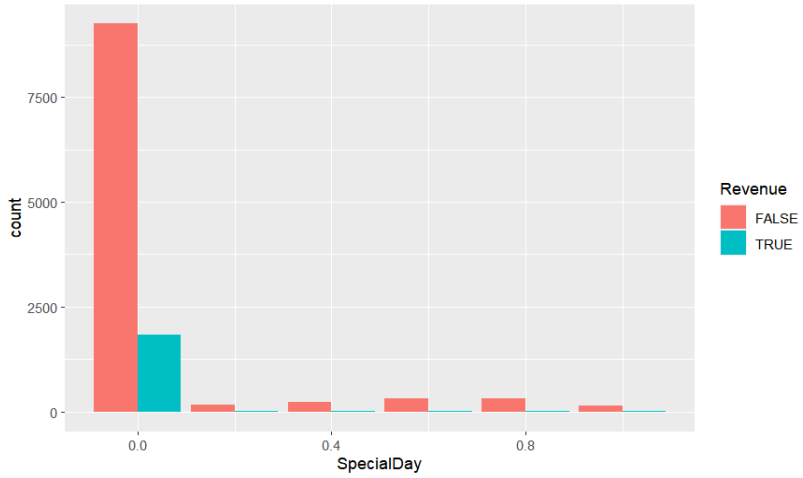
**4.3 Exploring the data distribution of “Bounce Rates”, “Exit Rates” and “Page Values” features versus the target variable Revenue respectively:** There is no noticeable disparity in Bounce Rates between customers who made a purchase and those who did not. However, customers who ended up making a purchase had lower Exit Rates on average, indicating that they were more likely to remain on the website's pages. Additionally, customers who did not make a purchase had significantly lower Page Values, suggesting that they spent less time on related pages.



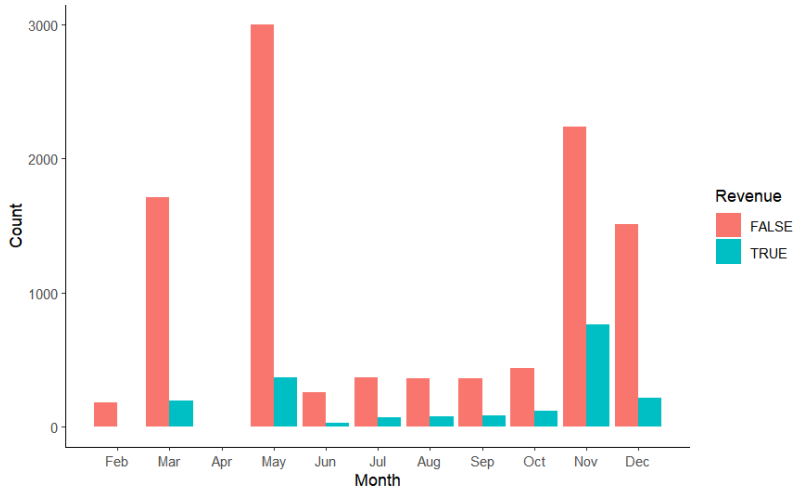




**4.4 Exploring the data distribution of "Special Day" features versus the target variable Revenue:** Here we can analyse that majority of revenue is on non special day, and impact of revenue due to special fay is very less.

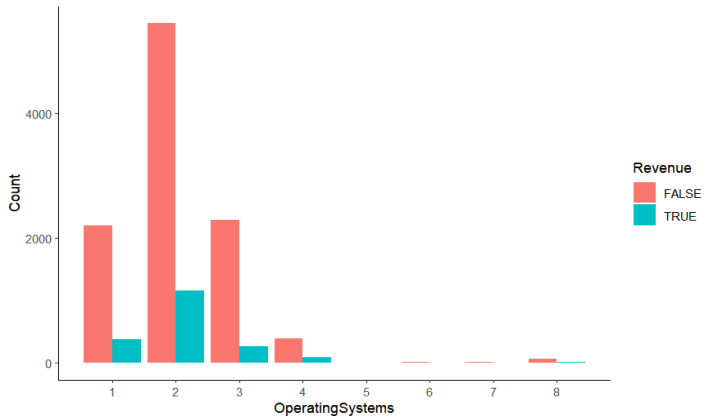



**4.5 Exploring the data distribution of "Month" features versus the target variable Revenue:**

- Most revenue is from the months of March, May, November and December
- Least revenue is from the month June, July, August, September and October
- Data is not available for January and April

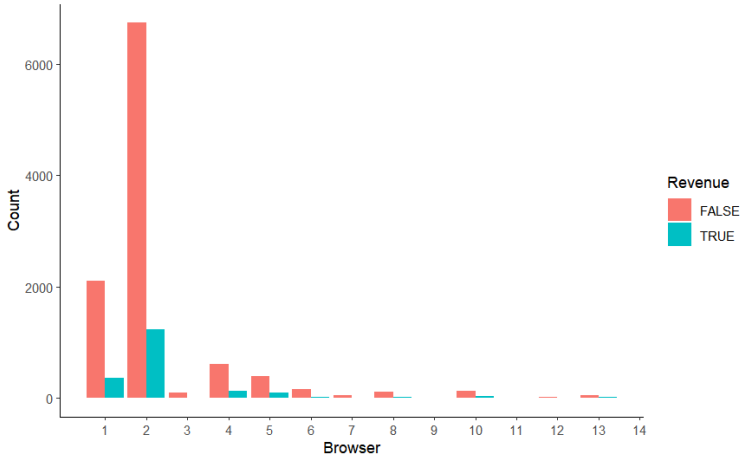



**4.6 Exploring the data distribution of "Operating Systems" features versus the target variable Revenue:** The highest revenue comes from operating system 2, and the highest non-revenue also comes from operating system 2.



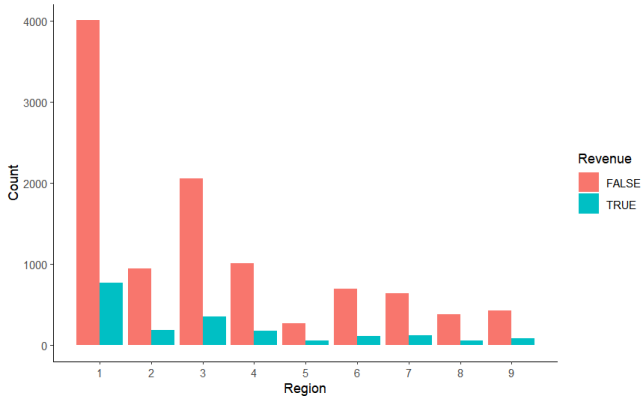

**4.7 Exploring the data distribution of "Browser" features versus the target variable**

**Revenue:** The highest revenue comes from Browser 2, and the highest non-revenue also comes from Browser 2.

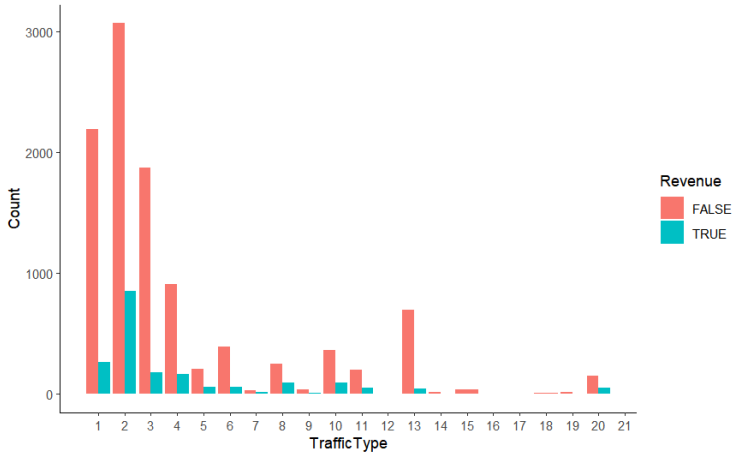



**4.8 Exploring the data distribution of "Region" features versus the target variable**

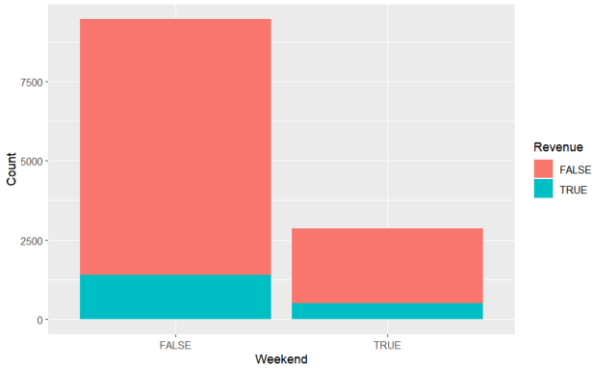
**Revenue:** The highest revenue comes from Region 1, and the highest non-revenue also comes from same.



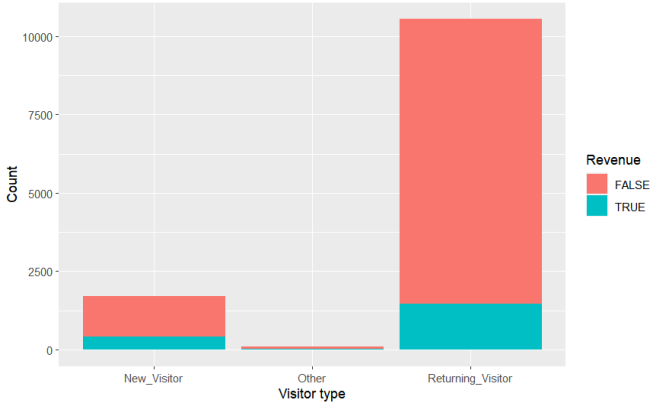
**4.9 Exploring the data distribution of "Traffic Type" features versus the target variable**
**Revenue:** The highest revenue comes from Traffic Type2, and the highest non-revenue also comes from Traffic Type 2.




**4.10 Exploring the data distribution of "Weekend" features versus the target variable**
**Revenue:** Data shows that revenue on non-weekend is almost thrice that of revenue on weekend

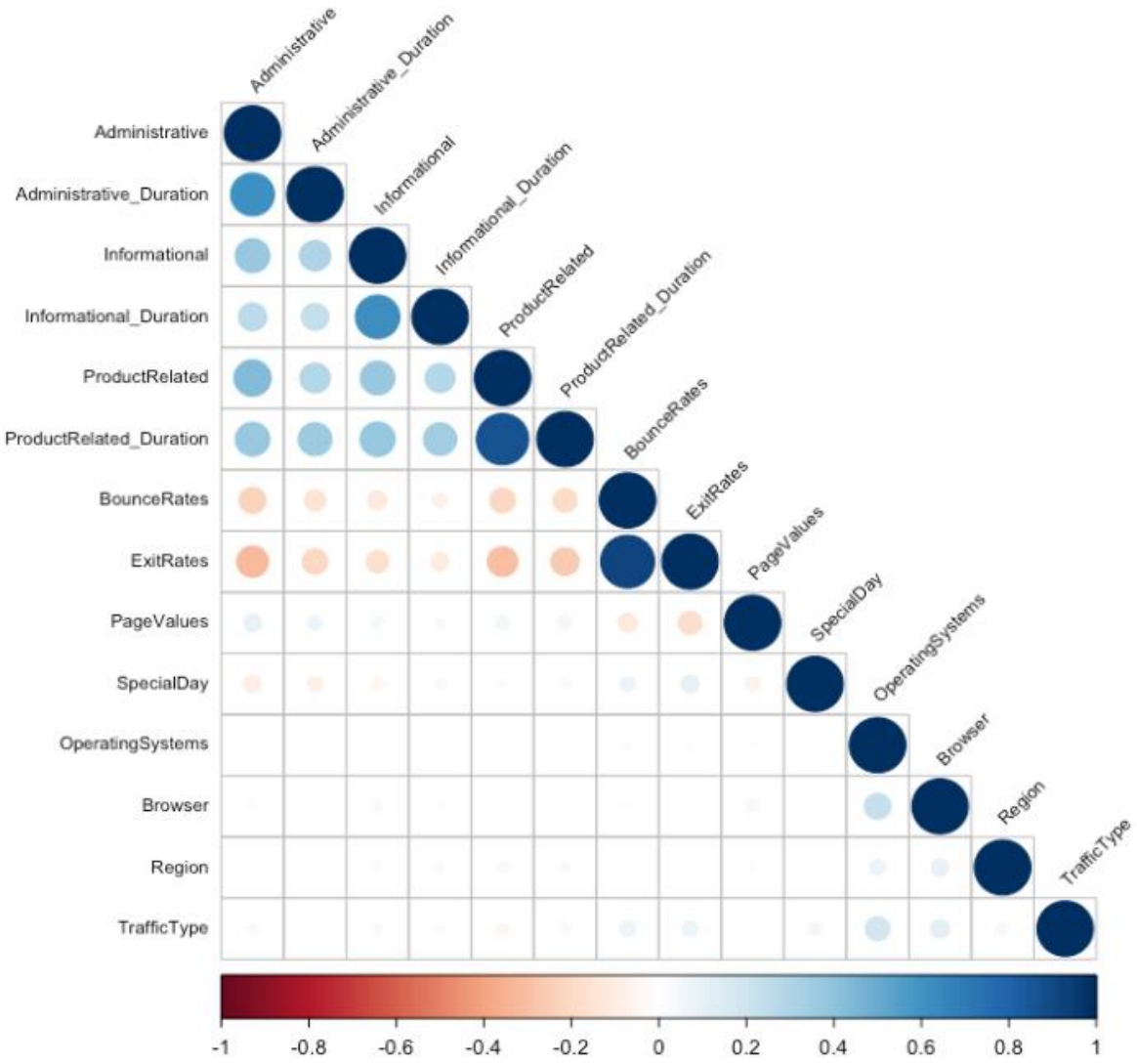



**4.11 Exploring the data distribution of "Visitor Type" features versus the target variable**
**Revenue:** Data shows that revenue from returning visitor is approximately 3.5 times as that of new visitor



| | FALSE | TRUE |
|---|---|---|
| **New_Visitor** | 1272 | 422 |
| **Other** | 69 | 16 |
| **Returning_Visitor** | 9081 | 1470 |

**4.12 Correlation plot:** Correlation plot depicts the correlation between the features " informational" and "informational duration" , "administrative" and "administrative_duration", "product related" and "product related duration". Where we can consider only one with each pair for better modelling, which will reduce the overfitting of the data.



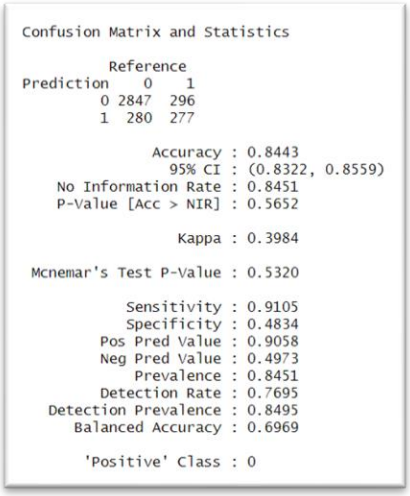
# 5. Model training

**5.1 Naive Bayes Classifier:** Naive Bayes is a probabilistic classifier that applies Bayes' theorem with the assumption of independence between each pair of features. In Naive Bayes, the algorithm calculates the probability of a data point belonging to each class, based on the values of its features. Then, the algorithm selects the class with the highest probability as the predicted class for the data point. The "naive" assumption that the features are independent can be a limitation, as this may not always be the case in real-world scenarios. Nevertheless, Naive Bayes often performs well in practice, especially when the dataset has many features and a relatively small number of training examples. Naive Bayes classifier tries to classify new observation based on conditional probability.

- We have defined a train control method with 5 fold cross validation, Average accuracy: 75.6%
- Training the model using “caret library” on data set without one hot encoding, Average accuracy: 84.5%

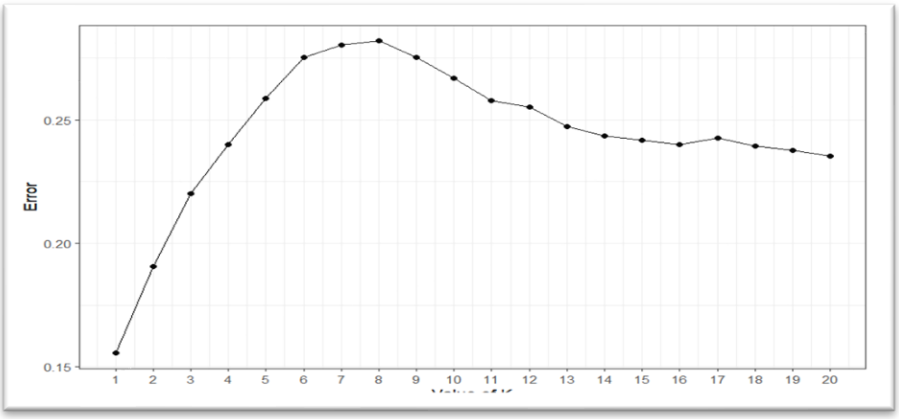
Thus, Naive Bayes classifier works better when categorical variables are one hot encoded.

**5.2 k-Nearest Neighbor:** k-Nearest Neighbor (k-NN) is a non-parametric machine learning algorithm used for classification and regression tasks. The basic idea behind k-NN is that similar data points tend to belong to the same class or have similar values. Overall, k-NN is a popular and useful algorithm for classification and regression tasks, especially for small to medium-sized datasets where it can achieve high accuracy with low computational cost. Training the model on one hot encoded data set using “knn” function of “class” library.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2847  296
         1  280  277

               Accuracy : 0.8443
                 95% CI : (0.8322, 0.8559)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : 0.5652

                  Kappa : 0.3984

 Mcnemar's Test P-Value : 0.5320

            Sensitivity : 0.9105
            Specificity : 0.4834
         Pos Pred Value : 0.9058
         Neg Pred Value : 0.4973
             Prevalence : 0.8451
         Detection Rate : 0.7695
   Detection Prevalence : 0.8495
      Balanced Accuracy : 0.6969

       'Positive' Class : 0
```

- Visualizing KNN with different k values(number of nearest neighbor), Let’s choose k range from 1 to 20.




Here we observe that for k=1 model has lowest error. Thus, Let us train knn with k=1.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2847  296
         1  280  277

               Accuracy : 0.8443
                 95% CI : (0.8322, 0.8559)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : 0.5652

                  Kappa : 0.3984

 Mcnemar's Test P-Value : 0.5320

            Sensitivity : 0.9105
            Specificity : 0.4834
         Pos Pred Value : 0.9058
         Neg Pred Value : 0.4973
             Prevalence : 0.8451
         Detection Rate : 0.7695
   Detection Prevalence : 0.8495
      Balanced Accuracy : 0.6969

       'Positive' Class : 0
```

**5.3 Random Forest:** Random Forest is a popular machine learning algorithm used for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to improve the accuracy and reduce the over fitting of the model. Random forests consider a random subset of predictor variables at each split to decrease correlation between trees and improve model performance. The parameter "mtry" determines the number of variables to consider at each split. Typically, a popular approach is to set "mtry" equal to the square root of the number of predictors. However, the random Forest package offers a more advanced method for selecting "mtry" based on the number of predictors. This method uses the formula "m = ceiling(log2(n))" where "m" is the rounded-up base 2 logarithm of the number of predictors in the dataset "n". This formula has been found to perform well for a diverse range of datasets. ntree=100, collection 100 trees to be constructed.

- Model is trained on data set without one-hot encoding.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2955  189
         1  172  384

               Accuracy : 0.9024
                 95% CI : (0.8924, 0.9118)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6227

 Mcnemar's Test P-Value : 0.3997

            Sensitivity : 0.9450
            Specificity : 0.6702
         Pos Pred Value : 0.9399
         Neg Pred Value : 0.6906
             Prevalence : 0.8451
         Detection Rate : 0.7986
   Detection Prevalence : 0.8497
      Balanced Accuracy : 0.8076

       'Positive' Class : 0

Thus accuracy of random forest is 89.81%
```

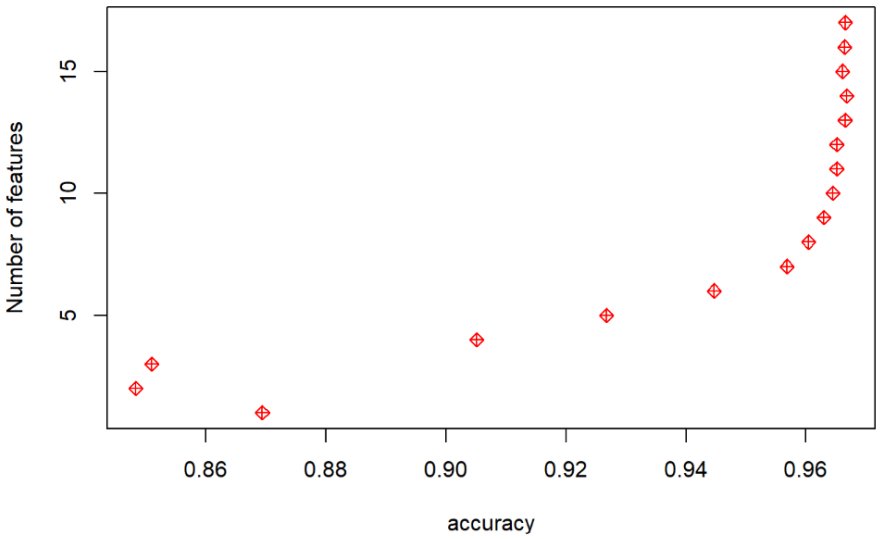
**5.3.1 Recursive Feature Elimination:** Output of RFE provides valuable information about the importance of different features and how they affect the performance of the model. This information can be used to improve the accuracy and interpretation ability of the final model.

- Recursive Feature Elimination is used to build Random Forest model with all possible subsets of features.
- "rfeControl" function is used with 10 fold cross validation.
- "rfe" function is used for training.

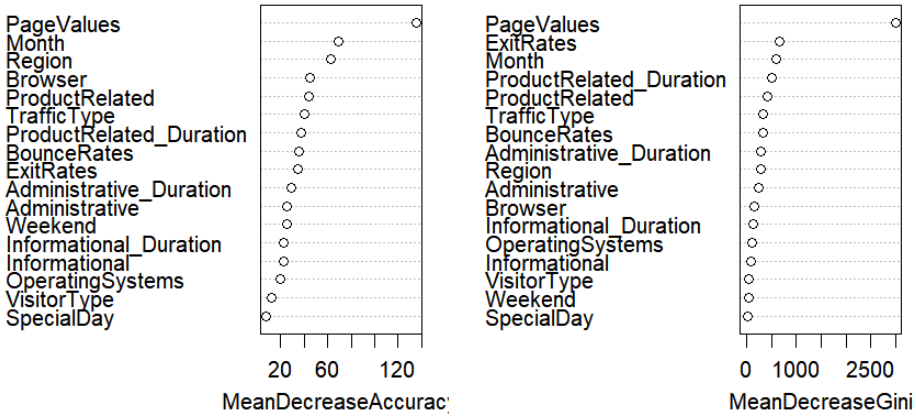
| Variables | Accuracy | Kappa | AccuracySD | KappaSD |
|---|---|---|---|---|
| 1 | 0.869431951 | 0.738863401 | 0.007049609 | 0.014092158 |
| 2 | 0.848321599 | 0.6966427 | 0.006334549 | 0.012659938 |
| 3 | 0.850994757 | 0.701988995 | 0.006064325 | 0.012121113 |
| 4 | 0.905140123 | 0.810278194 | 0.013344707 | 0.026692774 |
| 5 | 0.926730159 | 0.853460616 | 0.019189933 | 0.038381349 |
| 6 | 0.944686211 | 0.889372426 | 0.01243196 | 0.024866753 |
| 7 | 0.956819031 | 0.913638362 | 0.006278806 | 0.012557185 |
| 8 | 0.960383398 | 0.920766867 | 0.005161325 | 0.010322675 |
| 9 | 0.962919476 | 0.925838934 | 0.005074801 | 0.010149926 |
| 10 | 0.964427311 | 0.928854527 | 0.004939367 | 0.009879258 |
| 11 | 0.965112524 | 0.93022482 | 0.005336765 | 0.010674711 |
| 12 | 0.965112618 | 0.930224951 | 0.005394665 | 0.010790778 |
| 13 | 0.966483514 | 0.932966754 | 0.00542278 | 0.010846924 |
| 14 | 0.966757628 | 0.933515008 | 0.005766589 | 0.011534296 |

| | | | | |
|---|---|---|---|---|
| 15 | 0.96607218 | 0.932144105 | 0.00573966 | 0.011480498 |
| 16 | 0.966415115 | 0.932830057 | 0.005339922 | 0.010680789 |
| 17 | 0.966552148 | 0.933103963 | 0.005851419 | 0.011704343 |

- **Number of features vs Accuracy plot:** We can see accuracy increased drastically initially as more feature variables are added later it became almost same.




- **Variable importance plot:** The function "varImpPlot" uses two metrics, MeanDecreaseAccuracy (MDA) and MeanDecreaseGini (MDG), to measure the importance of features. MDA measures the decrease in accuracy when a single variable is excluded or changed, while MDG measures the decrease in node impurity.

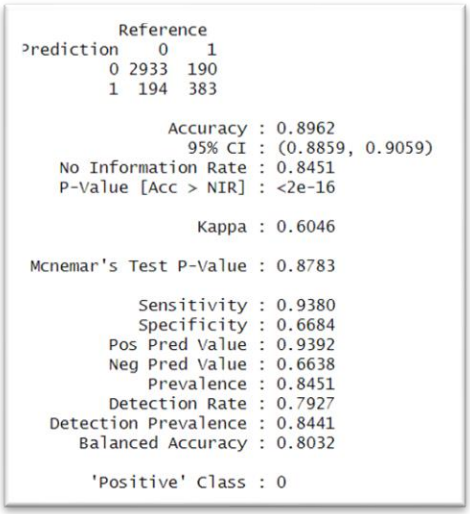



This shows that feature variables like Page values is most important one. Other than this few of the other important feature variables are Month, ExitRates, ProductRelated_Duration. In both plots we see Visitor Type and Special Day is least significant.

**Feature variable importance table:** This table is computed based on "information" function using trained random forest trained model which was shown initially shown above.

| Feature variable importance table | MeanDecreaseAccuracy |
|---|---|
| Administrative | 25.740664 |
| Administrative_Duration | 29.428259 |
| Informational | 22.772219 |
| Informational_Duration | 22.896964 |
| ProductRelated | 44.383527 |
| ProductRelated_Duration | 37.415946 |
| BounceRates | 35.813305 |
| ExitRates | 34.568105 |
| PageValues | 135.122725 |
| SpecialDay | 8.007717 |
| Month | 68.915250 |
| OperatingSystems | 19.990634 |
| Browser | 45.293001 |
| Region | 62.972472 |
| TrafficType | 39.930089 |
| VisitorType | 12.686405 |
| Weekend | 25.540213 |

**Top 10 features are:** PageValues, Month, Region, Browser, ProductRelated, TrafficType, ProductRelated_Duration, BounceRates, ExitRates and Administrative_Duration.



```
          Reference
Prediction    0    1
         0 2933  190
         1  194  383

               Accuracy : 0.8962
                 95% CI : (0.8859, 0.9059)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6046

 Mcnemar's Test P-Value : 0.8783

            Sensitivity : 0.9380
            Specificity : 0.6684
         Pos Pred Value : 0.9392
         Neg Pred Value : 0.6638
             Prevalence : 0.8451
         Detection Rate : 0.7927
   Detection Prevalence : 0.8441
      Balanced Accuracy : 0.8032

       'Positive' Class : 0
```

- Thus accuracy of random forest on top 10 feature variable is 89.6%
- Whereas random forest on all feature variable is 90.24%
- But such a small difference in accuracy is acceptable as model complexity has decreased.

**5.4 Support Vector Machine:** Support Vector Machine (SVM) is a popular machine learning algorithm used for both classification and regression tasks. The main idea behind SVM is to find the best possible boundary, called hyperplane, that separates the data into different classes. SVM works by mapping the input data to a higher-dimensional feature space where it becomes easier to find a hyperplane that separates the classes. In SVM, there are two types of classifiers: linear and nonlinear. In the linear case, the boundary between the classes is a straight line. In the nonlinear case, the boundary can be a complex curve or surface. SVM is also known for its ability to handle imbalanced datasets, where one class has significantly more samples than the other.

SVM has several hyperparameters that can be tuned to optimize its performance, such as the kernel type, kernel parameters, regularization parameter, and others.

- Training the model using data set with one hot encoding.
- Used "svm" function from "e1071" library.

Linear kernel and radial basis function (RBF) kernel are two commonly used kernel functions in SVM. Parameters used during training: Kernel : linear and radial

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2794  121
         1  333  452

               Accuracy : 0.8773
                 95% CI : (0.8663, 0.8877)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : 1.453e-08

                  Kappa : 0.5928

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8935
            Specificity : 0.7888
         Pos Pred Value : 0.9585
         Neg Pred Value : 0.5758
             Prevalence : 0.8451
         Detection Rate : 0.7551
   Detection Prevalence : 0.7878
      Balanced Accuracy : 0.8412

       'Positive' Class : 0
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2824  120
         1  303  453

               Accuracy : 0.8857
                 95% CI : (0.875, 0.8958)
    No Information Rate : 0.8451
    P-Value [Acc > NIR] : 8.165e-13

                  Kappa : 0.6136

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9031
            Specificity : 0.7906
         Pos Pred Value : 0.9592
         Neg Pred Value : 0.5992
             Prevalence : 0.8451
         Detection Rate : 0.7632
   Detection Prevalence : 0.7957
      Balanced Accuracy : 0.8468

       'Positive' Class : 0
```

- Accuracy of SVM using linear kernel=87.73%
- Accuracy of SVM using Radial basis kernel=88.56%

**5.5 XG Boost:** XGBoost (extreme Gradient Boosting) is a powerful machine learning algorithm used for regression, classification, and ranking problems. It is based on the gradient boosting framework and uses an ensemble of decision trees to make predictions. The choice of hyperparameters depends on the specific problem and the characteristics of the data. XGBoost has been used successfully in various applications, including predicting stock prices, image classification, and natural language processing.

- Training is done on top ten features selected during random forest training
- **Training parameters:**
  - objective = "binary:logistic"
  - eta = 0.3
  - max_depth = 6
  - eval_metric = "auc"
  - Nrounds = 100
  - Early stopping rounds = 10

AUC is a widely used evaluation metric for binary classification problems like XGBoost, and it measures the model's ability to distinguish between positive and negative examples. The higher the AUC, the better the model's performance. We have trained until test_auc hasn't improved in 10 rounds.

| | | | | |
|---|---|---|---|---|
| [1] | train | auc:0.940757 | test | auc:0.909651 |
| [2] | train | auc:0.949137 | test | auc:0.918012 |
| [3] | train | auc:0.952685 | test | auc:0.922717 |
| [4] | train | auc:0.954565 | test | auc:0.926051 |
| [5] | train | auc:0.957042 | test | auc:0.926611 |
| [6] | train | auc:0.958906 | test | auc:0.927626 |
| [7] | train | auc:0.961152 | test | auc:0.927922 |
| [8] | train | auc:0.963121 | test | auc:0.927406 |
| [9] | train | auc:0.963852 | test | auc:0.926375 |
| [10] | train | auc:0.965559 | test | auc:0.925545 |
| [11] | train | auc:0.967358 | test | auc:0.925664 |
| [12] | train | auc:0.969143 | test | auc:0.924629 |
| [13] | train | auc:0.970182 | test | auc:0.924950 |
| [14] | train | auc:0.970923 | test | auc:0.924509 |
| [15] | train | auc:0.973319 | test | auc:0.923537 |
| [16] | train | auc:0.974383 | test | auc:0.923199 |
| [17] | train | auc:0.975294 | test | auc:0.923111 |

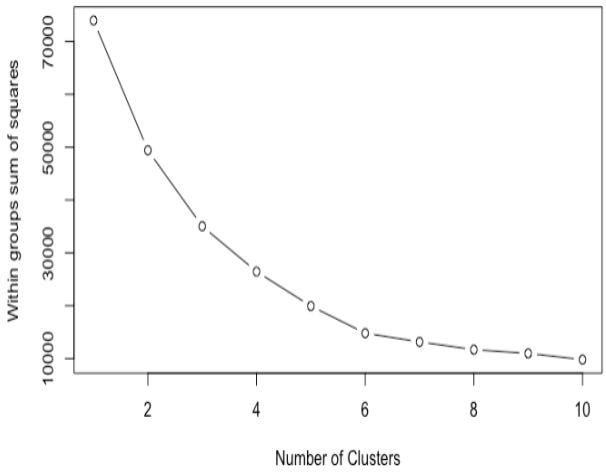
**Stopping. Best iteration:** [7] train-auc:0.961152 test-auc:0.927922

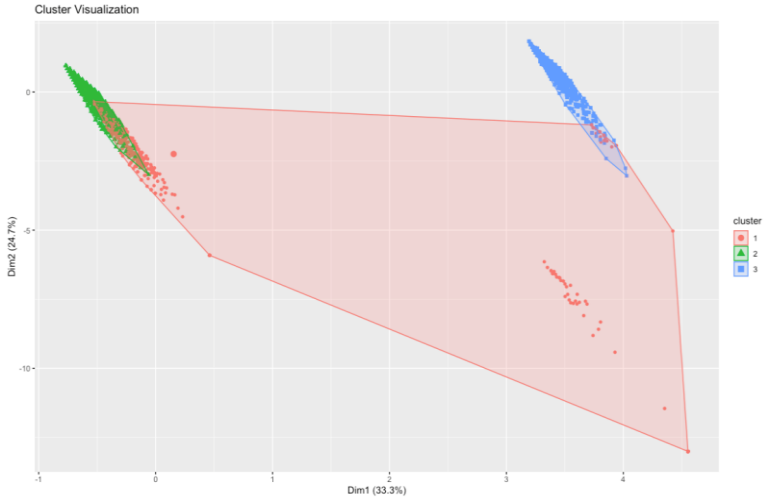
## 6. Clustering techniques implemented:

To understand the data aggregation we implemented clustering methods as below. Here we have taken into account the only the "Browser", "Region", "TrafficType", "VisitorType" features.

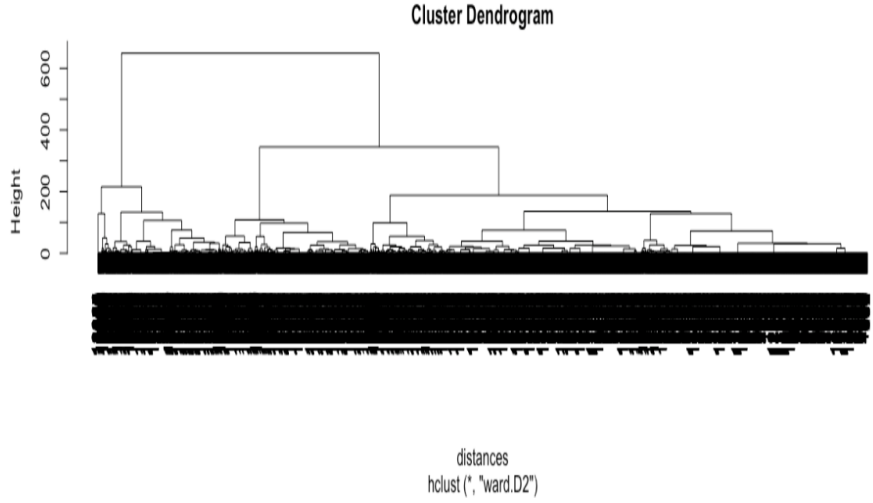
### 6.1 K-means clustering:

his is widely used unsupervised learning algorithm that partitions a dataset into K-clusters based on the similarity between data points. K-means clustering is a simple and fast algorithm which is suitable for both small and large datasets.From the elbow plot we have selected 3 clusters and looks like the cluster one intersects both the clusters.

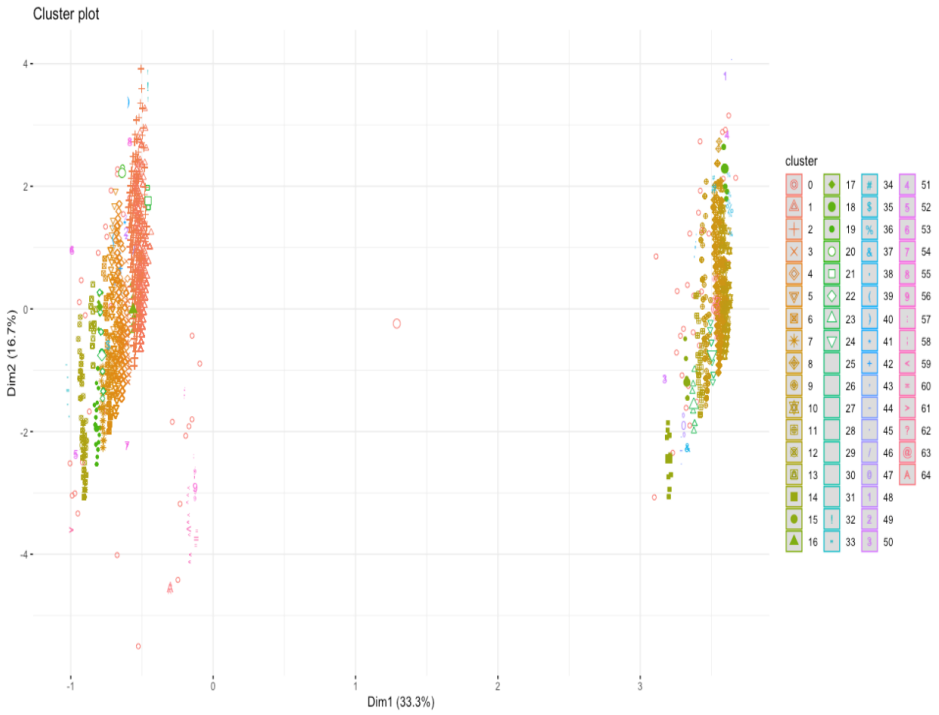



**6.2 Hierarchical clustering:** This is another unsupervised learning algorithm that clusters data points into a tree-like structure based on the similarity between them. Hierarchical clustering can be either agglomerative (bottom-up) or divisive (top-down).



**6.3 DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** As name says this is a density-based clustering algorithm that groups data points into clusters based on their density. DBSCAN is particularly useful for datasets with irregular shapes and noises. But our dataset was mostly without the noises and hence the results were as below,




## 7. Conclusion:

- Analyzing the number of page visit of 3 different page categories it clearly says that customers are interested more in Product related pages rather than knowing information of the product in detail.
- Revenue is generated by the customers who visit the product page and spend more time on it, which intuitively means whoever spends more time on administrative and informational page will only hop around rather than end up buying.
- Discounts can be given to the ones who spend more time on Product related page.
- There is no noticeable disparity in Bounce Rates between customers who made a purchase and those who did not.
- However, customers who ended up making a purchase had lower Exit Rates on average, indicating that they were more likely to remain on the website's pages.
- Additionally, customers who did not make a purchase had significantly lower Page Values, suggesting that they spent less time on related pages.

## 8. Future Work:

- We would plan to work more on the data gathering, we did look into it, but couldn't find the similar datasets.
- Work on couple more research question for example, "How does web metrics influence the revenue."
- Will explore and try to implement MLOps best practices by designing and creating a end-to-end pipelines.
- Would explore Gaussian Mixture Models for the clustering and also analyze the clusters indepth.

## 9. Bibliography:


[1]. Online Shopper Intention Analysis Using Conventional Machine Learning and Deep Neural Network Classification Algorithm [Doi:10.17933/jppi.2021.110106] https://jurnal-ppi.kominfo.go.id/index.php/jppi/article/view/341

[2]. Real-Time Prediction of Online Shoppers Purchasing Intention Using Random Forest [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7256375/ ]

[3]. Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Real- prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. Neural Comput & Applic 31, 6893–6908 (2019). https://doi.org/10.1007/s00521-018-3523-0.

[4]. Online Article: https://jurnal-ppi.kominfo.go.id/index.php/jppi/article/view/341

[5]. Data Clustering: A Review [ https://dl.acm.org/doi/pdf/10.1145/331499.331504 ]

[6]. A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised [ https://arxiv.org/pdf/1904.10604.pdf ]