Data Analytics (UE20CS312)

Online Shoppers Behaviour Prediction

V Himadhith

PES1UG20CS478

Department of CSE, PES University

Bengaluru, Karnataka

Prof Suresh Jamadagni

Introduction

Online shopping has completely revolutionized the methods available to a customer to purchase goods and services. E-Commerce Sales Are Predicted to Hit $6.5 Trillion by 2023. And in the US alone, it's expected to have 300 million[1] online shoppers in 2023, which accounts for 91% of the country's population. It is also observed that 75 percent of people shop online at least once a month.

Knowing these facts and figures, it is becoming increasingly more important that businesses can understand the rationale behind their customers' intentions whenever a purchase takes place. Studying a given user's intention can help a business better understand how to design their online store/service in a way to convert users just browsing into repeat customers.

Online shopping has been shown to provide more satisfaction to modern consumers seeking convenience and speed. On the other hand, some consumers still feel uncomfortable buying online. Lack of trust, for instance, seems to be the major reason that impedes consumers to buy online. Also, consumers may have a need to examine and feel the products and to meet friends and get some more comments about the products before purchasing. Such factors may have a negative influence on consumer decisions to shop online.

Literature review

Significant research has been done in order to decide the various models used to quantify and interpret decisions made by an user.

The models considered and the papers referred to are:

1. Logistic regression[2]
2. Decision tree classifier - Learning Decision Tree Classifiers J. R. QUINLAN University of Sydney, Sydney, Australia.
3. Random Forest - Random forest in remote sensing: A review of applications and future directions, MarianaBelgiu, LucianDrăguţ.
4. Online shopping behaviour - Factors Influencing Online Shopping Behavior: The Mediating Role of Purchase Intention -Yi Jin Lim, Abdullah Osman, Shahrul Nizam Salahuddin, Abdul Rahim Romle, Safizal Abdullah.

The Dataset

The dataset has been chosen from kaggle(UCI Machine Learning). The link for the dataset can be found here.

The dataset has a total of 18 columns which has been split into 10 numerical features and 8 categorical features.

The description of the data:

```{r}
#read the data
data <- read.csv("online_shoppers_intention.csv")

#take a look at the structure
str(data)
```

```
'data.frame':   12330 obs. of  18 variables:
 $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
 $ Administrative_Duration: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Informational_Duration : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ProductRelated         : int  1 2 1 2 10 19 1 0 2 3 ...
 $ ProductRelated_Duration: num  0 64 0 2.67 627.5 ...
 $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
 $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
 $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
 $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
 $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
 $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
 $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
 $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
 $ VisitorType            : chr  "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" ...
 $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
 $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

As stated above, our dataset has 10 numerical features and 8 categorical features. Taking a look at the dataset description:

Column Descriptions:

- *Administrative:* This is the number of pages of this type (administrative) that the user visited.
- *Administrative_Duration:* This is the amount of time spent in this category of pages.
- *Informational:* This is the number of pages of this type (informational) that the user visited.
- *Informational_Duration:* This is the amount of time spent in this category of pages.
- *ProductRelated:* This is the number of pages of this type (product related) that the user visited.
- *ProductRelated_Duration:* This is the amount of time spent in this category of pages.
- *BounceRates:* The percentage of visitors who enter the website through that page and exit without triggering any additional tasks.
- *ExitRates:* The percentage of pageviews on the website that end at that specific page.
- *PageValues:* The average value of the page averaged over the value of the target page and/or the

completion of an eCommerce transaction.[3]

- *SpecialDay:* This value represents the closeness of the browsing date to special days or holidays (eg Mother's Day or Valentine's day) in which the transaction is more likely to be finalized. More information about how this value is calculated below.
- *Month:* Contains the month the pageview occurred, in string form.
- *OperatingSystems:* An integer value representing the operating system that the user was on when viewing the page.
- *Browser:* An integer value representing the browser that the user was using to view the page.
- *Region:* An integer value representing which region the user is located in.
- *TrafficType:* An integer value representing what type of traffic the user is categorized into.[4]
- *VisitorType:* A string representing whether a visitor is New Visitor, Returning Visitor, or Other.
- *Weekend:* A boolean representing whether the session is on a weekend.
- *Revenue:* A boolean representing whether or not the user completed the purchase.

Data preprocessing

Check if the dataset has any null values.

```
```{r}
#missing value analysis

sapply(data, function(x) sum(is.na(x)))
```
```

```
         Administrative Administrative_Duration          Informational  Informational_Duration
                      0                       0                      0                       0
         ProductRelated ProductRelated_Duration          BounceRates                ExitRates
                      0                       0                      0                       0
             PageValues                 SpecialDay                Month       OperatingSystems
                      0                       0                      0                       0
                Browser                  Region            TrafficType             VisitorType
                      0                       0                      0                       0
                Weekend                 Revenue
                      0                       0
```

After using the function is.na() on each column feature it is evident that there are no missing values and no handling of na values is required.

Since our dataset has categorical data we refactor it using levels or just converting the TRUE/FALSE values to a simple 1 or 0.

```r
#fix the structure of the data

data$Revenue <- gsub(FALSE, 0, data$Revenue)
data$Revenue <- gsub(TRUE, 1, data$Revenue)
data$Weekend <- gsub(TRUE, 1, data$Weekend)
data$Weekend <- gsub(FALSE, 0, data$Weekend)

data$Month <- factor(data$Month, levels = c("Feb", "Mar", "May", "June", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"),
ordered = TRUE)
data$OperatingSystems <- factor(data$OperatingSystems)
data$Browser <- factor(data$Browser)
data$Region <- factor(data$Region)
data$TrafficType <- factor(data$TrafficType)
data$VisitorType <- factor(data$VisitorType)
data$Revenue <- factor(data$Revenue)
data$Weekend <- factor(data$Weekend)
str(data)
```

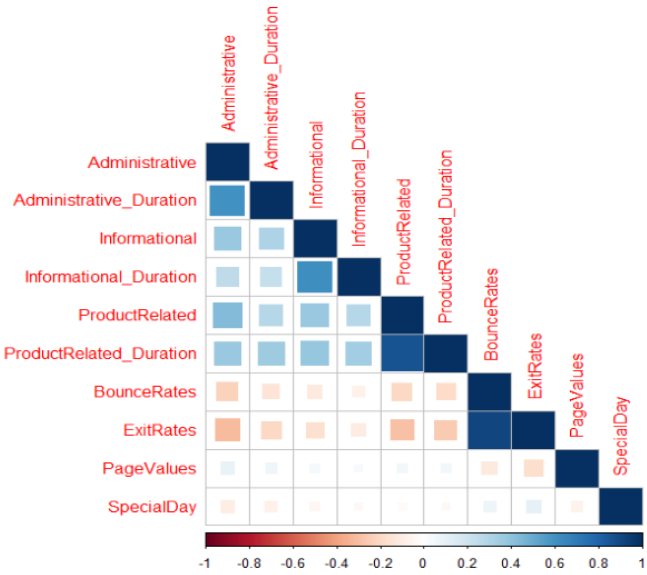
```
'data.frame':   12330 obs. of  18 variables:
 $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
 $ Administrative_Duration: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Informational_Duration : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ProductRelated         : int  1 2 1 2 10 19 1 0 2 3 ...
 $ ProductRelated_Duration: num  0 64 0 2.67 627.5 ...
 $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
 $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
 $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
 $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
 $ Month                  : Ord.factor w/ 10 levels "Feb"<"Mar"<"May"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ OperatingSystems       : Factor w/ 8 levels "1","2","3","4",..: 1 2 4 3 3 2 2 1 2 2 ...
 $ Browser                : Factor w/ 13 levels "1","2","3","4",..: 1 2 1 2 3 2 4 2 2 4 ...
 $ Region                 : Factor w/ 9 levels "1","2","3","4",..: 1 1 9 2 1 1 3 1 2 1 ...
 $ TrafficType            : Factor w/ 20 levels "1","2","3","4",..: 1 2 3 4 4 3 3 5 3 2 ...
 $ VisitorType            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Weekend                : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 2 1 1 ...
 $ Revenue                : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Moving along to the Bounce rates and exit rates. How impactful are these 2.

Let us now look into the impact of Bounce rates and exit rates.

Exit rate is the percentage of users who leave a page, whereas bounce rate is the entire percentage of a single interaction session. Thus, the former is determined by dividing the sum of one-page visits by the sum of visits to the entrance, and the latter is determined by dividing the sum of all page exits by all page visits. Bounce rates measure the proportion of visitors that participated in that single session, whereas exit rates measure the overall percentage of visitors who were in the previous session. This is a key distinction between these closely related metrics. Therefore, prior activity is not taken into account for calculating bounce rate.Hence all bounces logically define exits but conversely it is not true.

A high bounce rate could be an indication of problems with customer satisfaction due to one or more factors, such as the website's unpleasant user interface, incredibly sluggish throughput, or other technical issues. A high exit rate may indicate underperforming funnel segments and indicate opportunities for optimization because it suggests that if customers are departing, no one is ultimately making a purchase. BigCommerce 2 states that a bounce rate of between 30% and 55% is acceptable. According to our data, bounce rates are often lower than 10%. A bounce rate of less than 5%, according to UpSide Business 3, is cause for alarm and may indicate that the Google Analytics code was injected more than once.

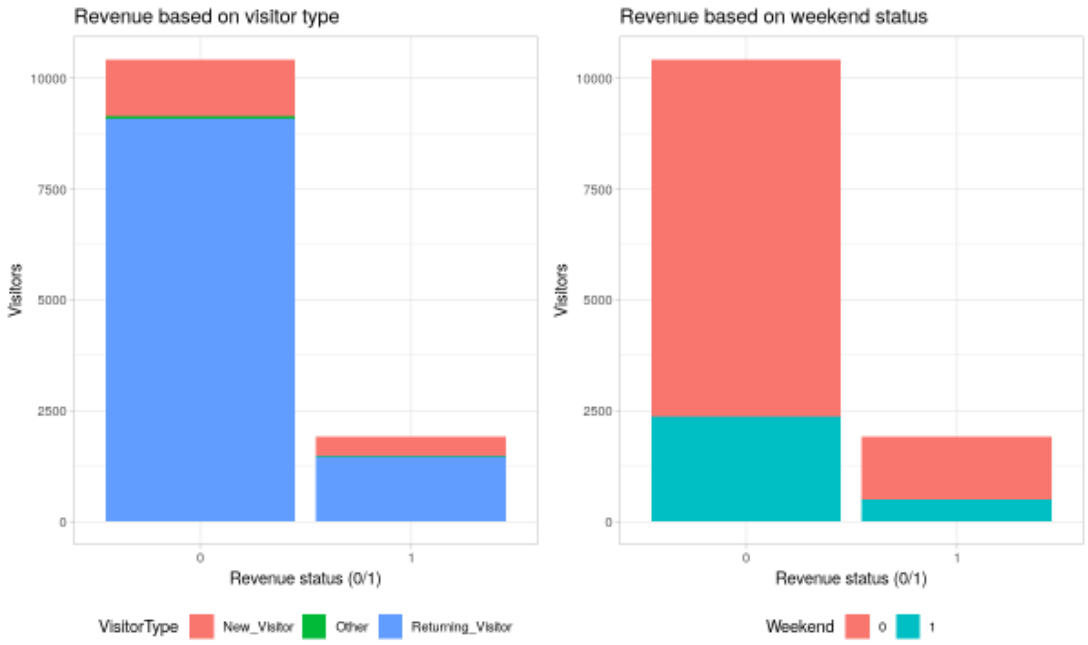


Most of the data points that we have seem to be <10% bounce rate and the rest scattered across the board. As already mentioned this number is close to the concerned range. To improve this we can do the following:

1. *Proposal 1:* An add-to-cart button that stands out, a user interface that is friendly, brief descriptions and icons when appropriate, colour impactfulness, and a smooth purchasing process

are all examples of landing page optimization. Another crucial factor is to watch out for giving the impression that the price is low before the item is added to the cart because shipping costs can have a big impact on conversion rates. Therefore, it is always preferable to present the actual cost up front.

2. *Proposal 2:* In order to give the letter a personalized touch, categorize email retargeting based on funneling as described with the previous dataset. Personalization encourages widespread loyalty, which improves retention.
3. *Proposal 3:* Introducing pop-ups offering qualitative discounts or personalized queries when a customer bounces multiple times and/or tries to leave the website.



The key inferences that can immediately be made from these graphs is that buyers of online stores tend to purchase far more on a weekend compared to the weekdays. This can be noted as a sort of a "weekend syndrome" where there is a huge rise in business.

Moreover, on the given data it is observed that the majority of purchases are done by repeat customers in comparison to first time buyers. This shows the significance of a company improving their customer retention rate and not only focusing on increasing their pool of new and first time customers.

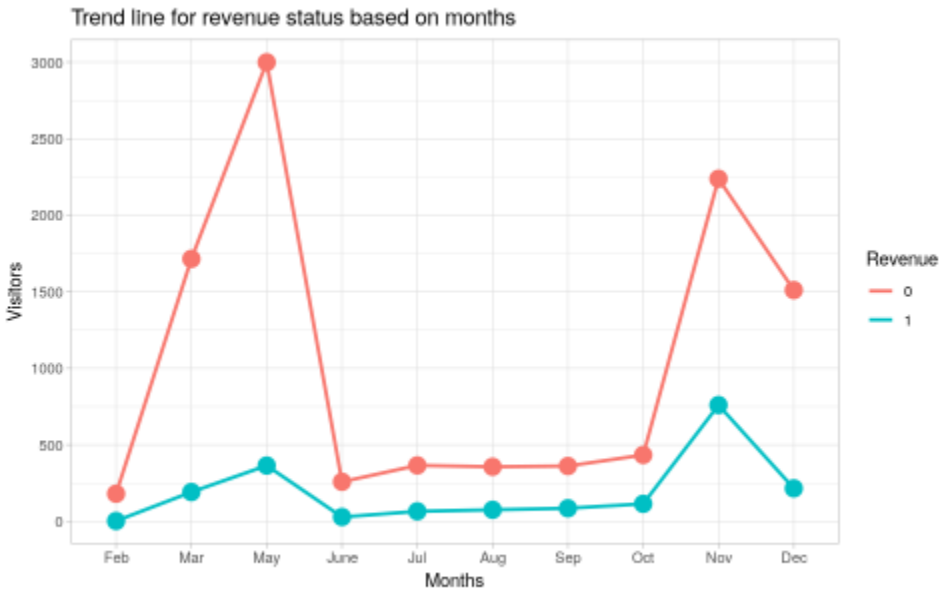
Now few more proposals can be made from these inferences.

4. *Proposal 4:* Improve the engagement of loyal and repeat customers by offering discounts/coupons/benefits for inviting new users (for instance family, friends or any

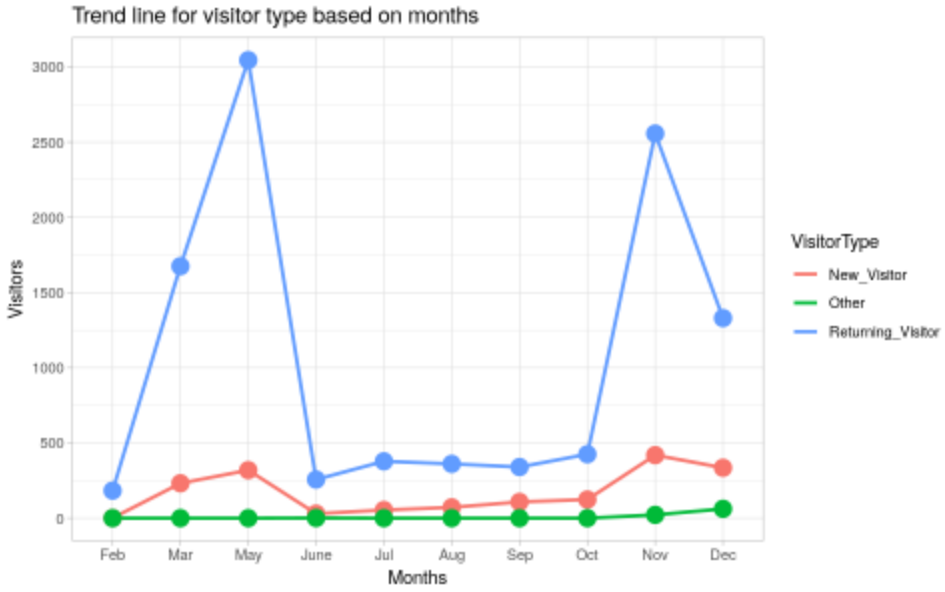
acquaintances). Allow new users obtained through this scheme to also benefit from it.

5. *Proposal 5:* Use the so-called "weekend syndrome" to an advantage to scheme limited time events / offers / campaigns centered around the weekends to further engage customers during the period where most sales occur.

The next 2 graphs depict the effect of seasons on visitors to the site.



Although it is clearly observed that there are more visitors on the weekend (marked by 0 - red line), it can also be noted that there is also a very significant dependence on the season. On the graph, March, May, November and December are months where the number of visitors is considerably higher than in other months. This can be explained with the timing of festivals and public holidays. For instance, June to October the amount of visitors are stagnating but there is a huge rise when Black Friday approaches.

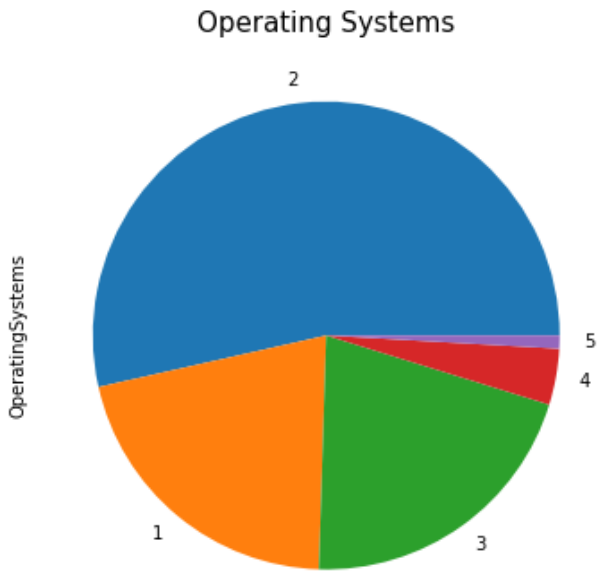


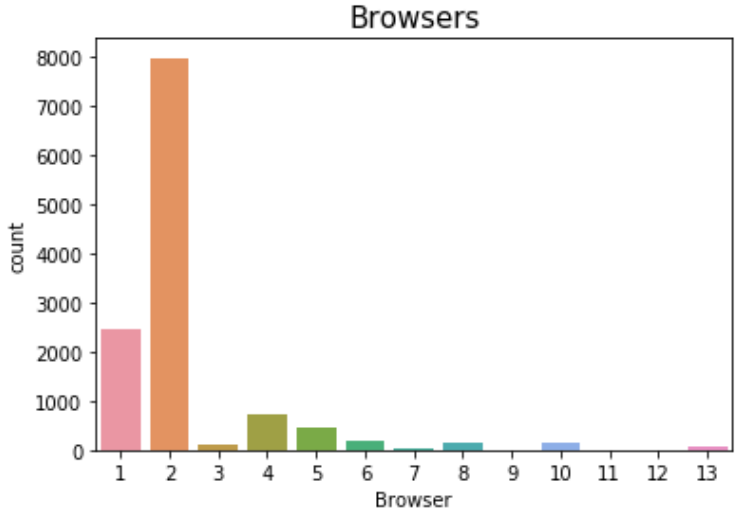
This graph depicts the type of visitors over the span of different seasons. Once again it is visibly evident that returning visitors are the clear majority, further emphasizing the importance for a business to retain their existing clientele. However, it can also be noted that there is a jump in new visitors also. In these cases, where there are new visitors, but not motivated enough to complete a purchase, there lies an opportunity to help grow the business.

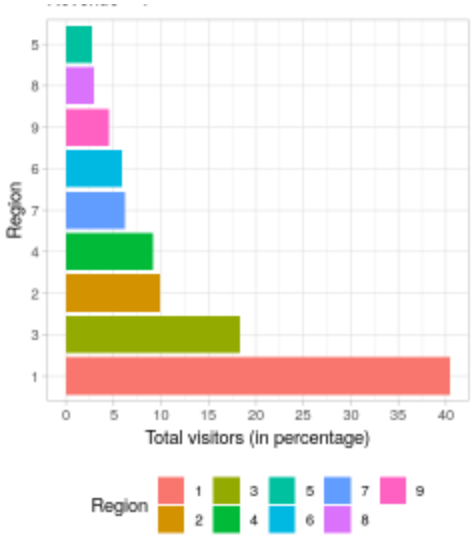
From the above two graphs the next proposal can be made.

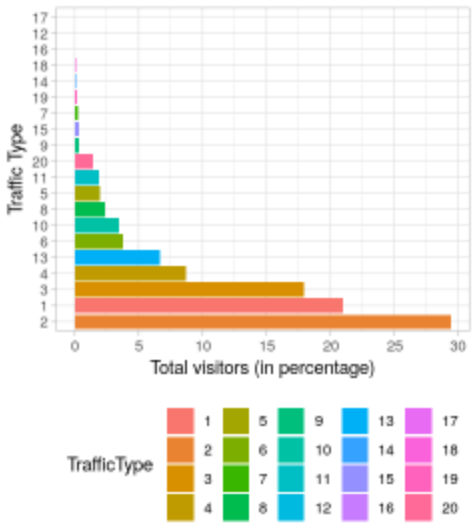
6. *Proposal 6:* Introduce seasonal campaigns/promotions with lucrative offers to entice and engage not only the loyal/repeat customers but also the new visitors.

The following graphs show the relation of revenue growth and other revenue drivers, namely, Operating systems, browser, region and traffic type sources.


Operating Systems
2
OperatingSystems
5
4
1
3


Browsers
8000
7000
6000
5000
4000
3000
2000
1000
0
count
1
2
3
4
5
6
7
8
9
10
11
12
13
Browser





These various graphs show various drivers that are pushing a given company's business. For instance '2' has the biggest share in operating systems implying that it could have an user-friendly interface to use or '1' having highest share in the region graph implies that the diversity of location of its users is wide-spread and can be attributed to its campaigning reaching wider audiences.

From these graphs the next set of proposals can be made.

7. *Proposal 7:* Businesses should ensure smooth technical operations along with creating easy to understand interfaces that can be supported by multiple system types and internet browsers.
8. *Proposal 8:* Adapting model by customizing it to fit cultural and social drivers of a given region. This can be further stacked on with personalizing ads and A/B testing to ensure the reach and conversion/retention rate is about the same among the various regions.
9. *Proposal 9:* Ensuring optimizations of SEO from various different search engines. Also collect data on region or age specific testing within google ads, facebook ads, etc.

Building a Model

No further preprocessing of data is necessary after the initial data preprocessing.

We are predicting the feature **revenue** against the other feature sets like browsers, etc.

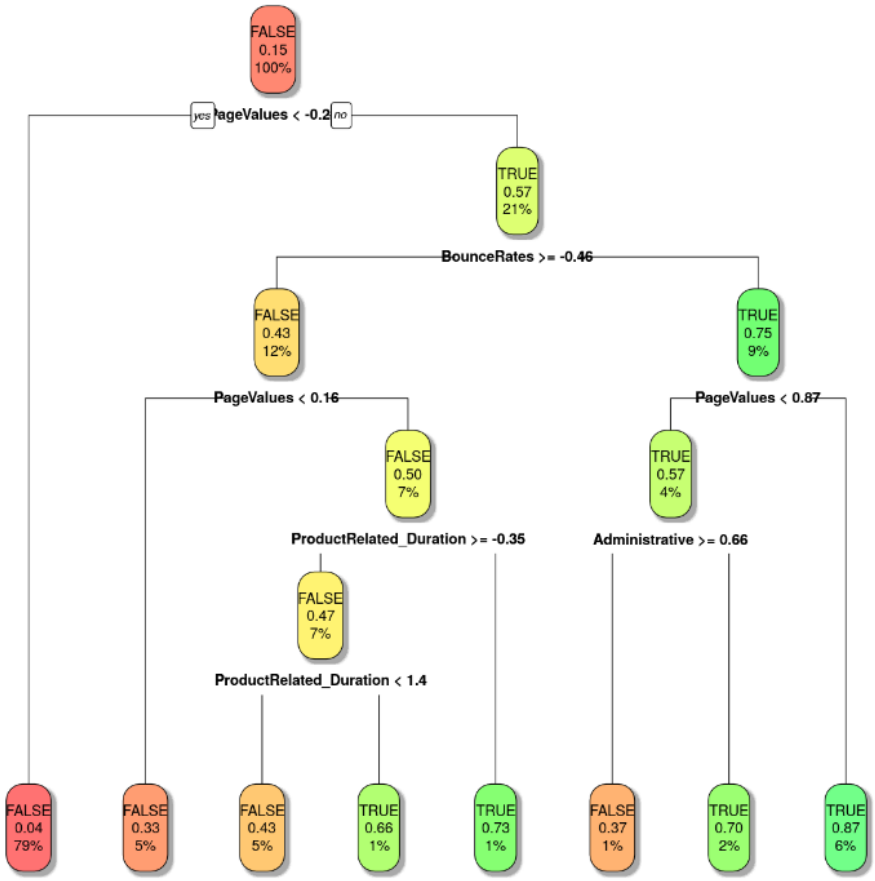
Decision Tree

I picked this model in particular as the first model because decision trees being a good classifier given their easy optimization techniques and feature importance clarifications.

For the decision tree model passing the scaled up value seemed to yield the most efficient result possible

and for that I scaled the data using the scale() function.

	model4_decision.variable.importance
	<dbl>
PageValues	1018.06683
BounceRates	103.43653
Administrative	60.56971
ProductRelated	52.19655
ExitRates	36.51023
VisitorType	28.91900
VisitorType_Num	28.91900
ProductRelated_Duration	26.33959
Administrative_Duration	19.14802
Informational_Duration	13.57494
Informational	11.44872



From the above figure which points to the feature importance table and the decision tree based on the

table, we can see that the PageValues remained at the top, followed by ProductRelated, Administrative, ProductRelated_Duration, BounceRates and Month_num.

This is inline with my proposals from previous inferences.

Predicting and analyzing the model based on metrics

```
Model1: Decision Tree Classifier
Fitness level
[1] 0.8730223

Evaluation on test set

prediction FALSE TRUE
     FALSE  1838   67
     TRUE    246  314
Accuracy =  0.218145
 Error Rate =  0.03172833
 True Positive Rate (Recall) =  0.5607143
 False Positive Rate =  0.0351706
 True Negative Rate (Specificity) =  0.9648294
 Precision =  0.824147
 F1Score =  0.6673751
```

The model was split 10:90 for test and train.The fitness level of the model was at 87.30% however the F1 Score was at 66.73% suggesting an overfit model. The intention was to capture those attributes that contribute the most to revenue growth so as to implement the above mentioned recommendations in a prioritized manner, to further improve on KPI (Key performance indicators).

10. *Proposal 10:* Due to the large impact on PageValue, it is likely that buyers would examine a wide range of products and suggestions. Therefore, considerable enhancements to bundle packages and recommendation engines would increase conversions. More revenue-generating products should be added to e-commerce that take advantage of the long tail.

Now, onto the next model to check the performance metrics and whether or not the accuracy improves.

We will use a logistic regression model to make our predictions.

In the data preprocessing done above we just need to make one change i.e instead of reordering adding one hot encoding to the column features Month and VisitorType.

Logistic Regression

After one hot encoding the data looks something like this

y	Month	OperatingSystems	Browser	Region	TrafficType	Weekend	Revenue	V_New_Visitor	V_Other	V_Returning_Visitor
	1	1	1	1	1	0	0	0	0	1
	1	2	2	1	2	0	0	0	0	1
	1	4	1	9	3	0	0	0	0	1
	1	3	2	2	4	0	0	0	0	1
	1	3	3	1	4	1	0	0	0	1
	...	...	...	...	...	...	...	...	...	...
	11	4	6	1	1	1	0	0	0	1
	10	3	2	1	8	1	0	0	0	1
	10	3	2	1	13	1	0	0	0	1
	10	2	2	3	11	0	0	0	0	1
	10	3	2	1	2	1	0	1	0	0

Notice the column month has numerical value 1 being February and 11 being December. It is similar to how levels work.

Onto to the slice and dice for training and testing data

I opted for the rule of thumb value of **30:70**

For this particular model there were iterations performed on the same model with different C(Inverse of regularization strength) values. The lower the C value the stronger the regularization.

First I made a sequence of C values to be fed into the model and iterated through each and every value of C. The predictions made by the model were as follows.

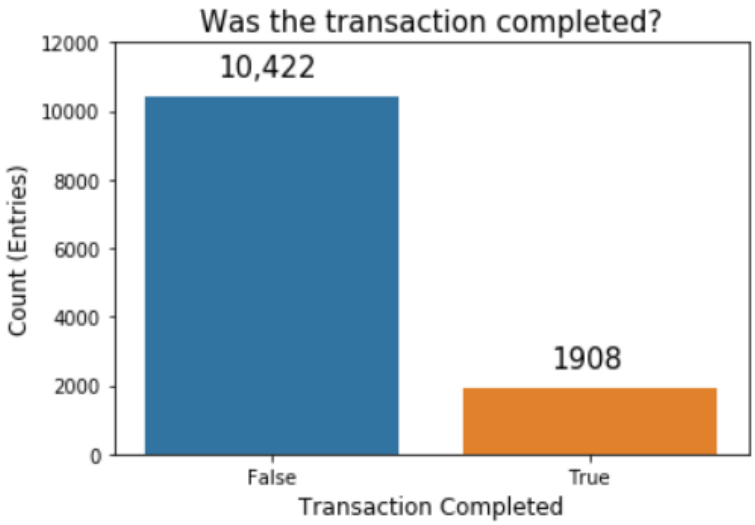
```
Model Accuracy (C=0.01): 0.8825439783491205
 Model Accuracy (C=0.1): 0.8838971583220568
 Model Accuracy (C=1.0): 0.8844384303112314
Model Accuracy (C=10.0): 0.8847090663058187
Model Accuracy (C=100.0): 0.8847090663058187
Model Accuracy (C=1000.0): 0.8847090663058187
Model Accuracy (C=10000.0): 0.8847090663058187
```

As we see the model starts performing better in the beginning on increasing the C value but slowly reaches its saturation after the 4th iteration and there is no further improvement in the model.

In conclusion the logistic regression was a better model compared to the decision tree as we see that the data here does not overfit. But this is not the best we can do so let's push further.

Now that we have a clear understanding of the dataset and how different models behave we can complete this analysis using a higher order model such as a random forest classifier.

Random forest classifier



From the graph we see that most of the shoppers are window shoppers in our case windows shoppers. This makes sense, as a majority of normal online shopping ends without a purchase. I have already mentioned my proposals to improve these numbers in the study before.

Preprocessing remains the same as for the other models but here we drop all the columns which are not statistically significant.

The column features which I dropped were based on these findings

'Region': We leave regionality out because the regionality may be slightly tied to purchase likelihood, but we want to train our model on a smaller set of features if possible.

'TrafficType': We leave this column out because Traffic sources are not quite useful for classifying if a user will make a purchase. It usually aids website owners in gauging traffic sources and can assist with determining where they should invest in advertisement.

'Weekend':There is a weak correlation between days of the week and online shopping.[5] This asserts that Sundays and Mondays have the highest traffic for eCommerce, but only by 16% of weekly revenue, and mostly on Monday, which is not classified as a weekend.

We also do not consider browser or OS since the majority of the count is taken up by #2.

Dropping the columns 'Month','Browser','OperatingSystems','Region','TrafficType','Weekend'

The training data size is 80% of the dataset. Predictions made on revenue based on the other columns.

Random Forest Classifier

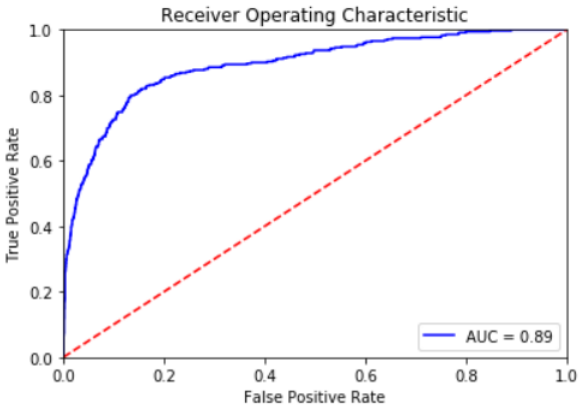
```
# Fit Random Forest Classifier to our Training Data
rfc = RandomForestClassifier(max_depth=5, random_state=2, n_estimators=750)
rfc.fit(X_train, y_train)

# make prediction using our test data and model
y_pred_rfc = rfc.predict(X_test)
y_prob_rfc = rfc.predict_proba(X_test)[:, 1]

# Comparing our prediciton to response values
print('Random Forest Classifier model accuracy(in %):', round(metrics.accuracy_score(y_test, y_p
red_rfc)*100,2))
```

```
Random Forest Classifier model accuracy(in %): 90.23
```

```
The area under the ROC curve is: 0.89
```



On setting the Maximum depth of the tree to 5 and number of trees in the forest to 750 we find a rather impressive accuracy of 90.23 which is significantly higher compared to the other models we have tackled with so far.

This may be bittersweet because on one hand the accuracy is the best of any model, but at the same time we are not considering the fact that most of the data in the revenue is heavily skewed.

But there is a silver lining to this, we can use the same model and stratify training data to handle this skewness.

We use the stratified shuffle split package included in the Sci-kit learn library to achieve this.

```
X_train_stratified, X_test_stratified, y_train_stratified, y_test_stratified = train_test_split
(X, y,stratify=y, test_size=.2, random_state=2, shuffle=True)
```

Keeping all the other values from the other constant and retraining the model on the stratified data.

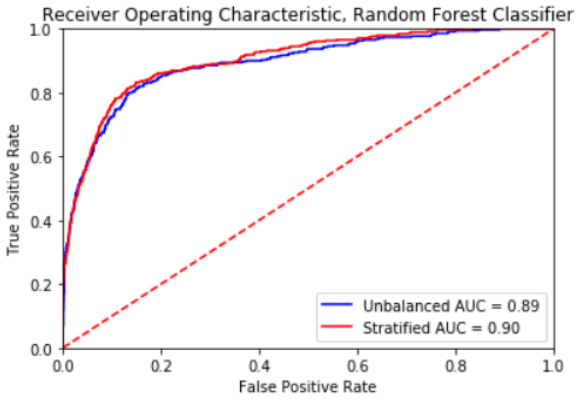
```
# Fit Random Forest Classifier to our Training Data
rfc_stratified = RandomForestClassifier(max_depth=5, random_state=2, n_estimators=750)
rfc_stratified.fit(X_train_stratified, y_train_stratified)

# make prediction using our test data and model
y_pred_rfc_stratified = rfc_stratified.predict(X_test_stratified)
y_prob_rfc_stratified = rfc_stratified.predict_proba(X_test_stratified)[:, 1]

# Comparing our prediciton to response values
print('Stratified Random Forest Classifier model accuracy(in %):', round(metrics.accuracy_score
(y_test_stratified, y_pred_rfc_stratified)*100,2))
```

```
Stratified Random Forest Classifier model accuracy(in %): 89.5
```

```
The area under the ROC curve for unbalanced data is: 0.89
The area under the ROC curve for stratified data is: 0.9
```

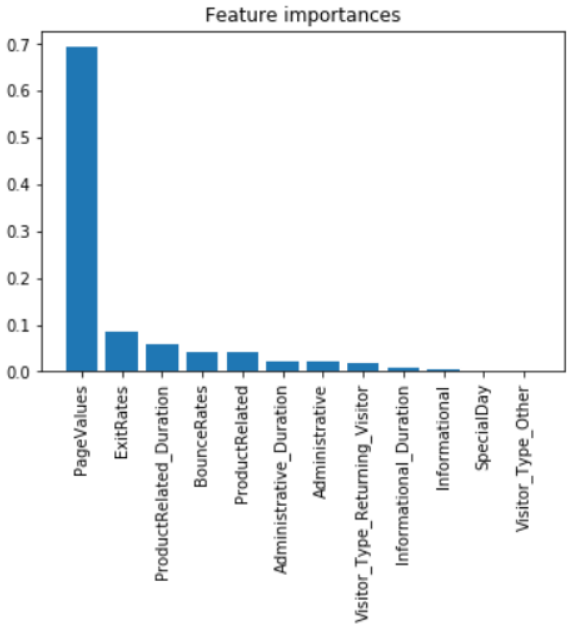


Accuracy metric after predicting based on the model has dropped while the AUC for stratified data has improved by a small margin.

We have to also account for feature importance like we did in the decision tree model. I have done the same for the stratified dataset and the output is as follows:

	Importance
PageValues	0.693368
ExitRates	0.086168
ProductRelated_Duration	0.058875
BounceRates	0.042850
ProductRelated	0.040776
Administrative_Duration	0.022842
Administrative	0.020969
Visitor_Type_Returning_Visitor	0.017604
Informational_Duration	0.008162
Informational	0.005109
SpecialDay	0.003008
Visitor_Type_Other	0.000269

Like last time PageValues tops the table again and is the most impactful feature followed by ExitRates.



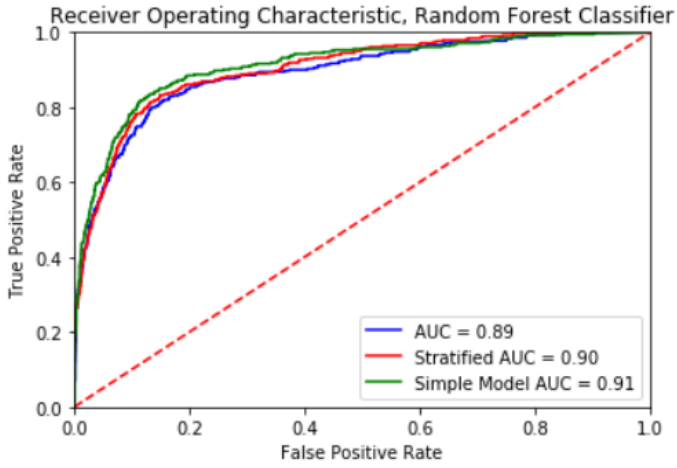
Training the model by keeping only the top 5 feature values and dropping the rest.

```
clf = RandomForestClassifier(max_depth=5, random_state=2, n_estimators=750)

scores = cross_val_score(clf, X_simp, y_simp, cv=cv)
print("Average Accuracy of Classifier over 10-folds: %0.2f (+/- %0.2f)" % (scores.mean(), score
s.std() * 2))
```

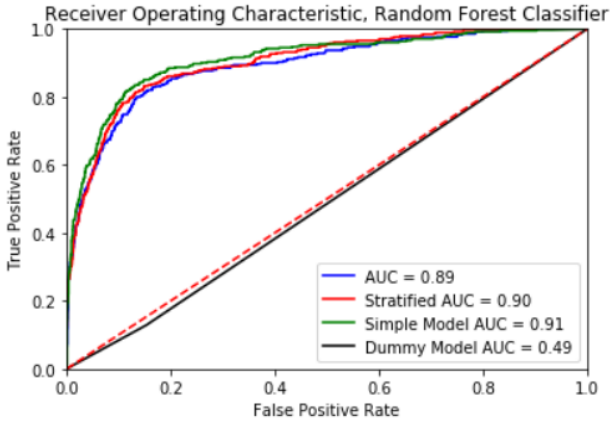
```
Average Accuracy of Classifier over 10-folds: 0.90 (+/- 0.01)
```

Let's see how the model holds up against the rest of the models that we have designed.



All three models seem to have similar performance.

To conclude that our trained models are actually better and that it cannot be matched by random guessing lets do a dummy classifier and compare the results.



Just as expected, our models are on average 90% accurate whereas the model which was made to

randomly guess was right about 50% of the time, as it is making guesses based on the distribution of a stratified dataset. If we were to deploy a model into the real world that would have to be our model which works on stratified data along with taking the feature importance values, which also gives the best AUC value.

Conclusion:

Through the course of this paper, the significance of understanding the behavior of online shoppers has been explored and was found to have a very large role in determining whether a given individual will make a purchase at a given time.

After applying transformations and tests on the data, a list of 10 proposals have been inferred in order to improve the odds of enticing an user to make a purchase in the store.

To help predict this behavior, several models were considered, namely, logistic regression, decision trees, and random forest classifier. After implementing all of said models and calculating accuracy metrics on each of them, it was found that random forest classifiers performed the best and accounted for all the significant values.