Online Shopper Purchase Intention Prediction

Instructor: Dr. David Belanger
BIA-678-D
Team D-09

Introduction:

- The popularity of online purchasing is growing.
- An accurate analysis system of online purchase patterns allows online shopping platforms to gain a better knowledge of customer psychology and develop better business tactics to enhance sales.
- This pattern in the customer's purchasing intention can be easily predicted by analyzing the history of the customers.
- By getting valuable insights from the shopper's behavior the businesses can be benefited

Data Set:

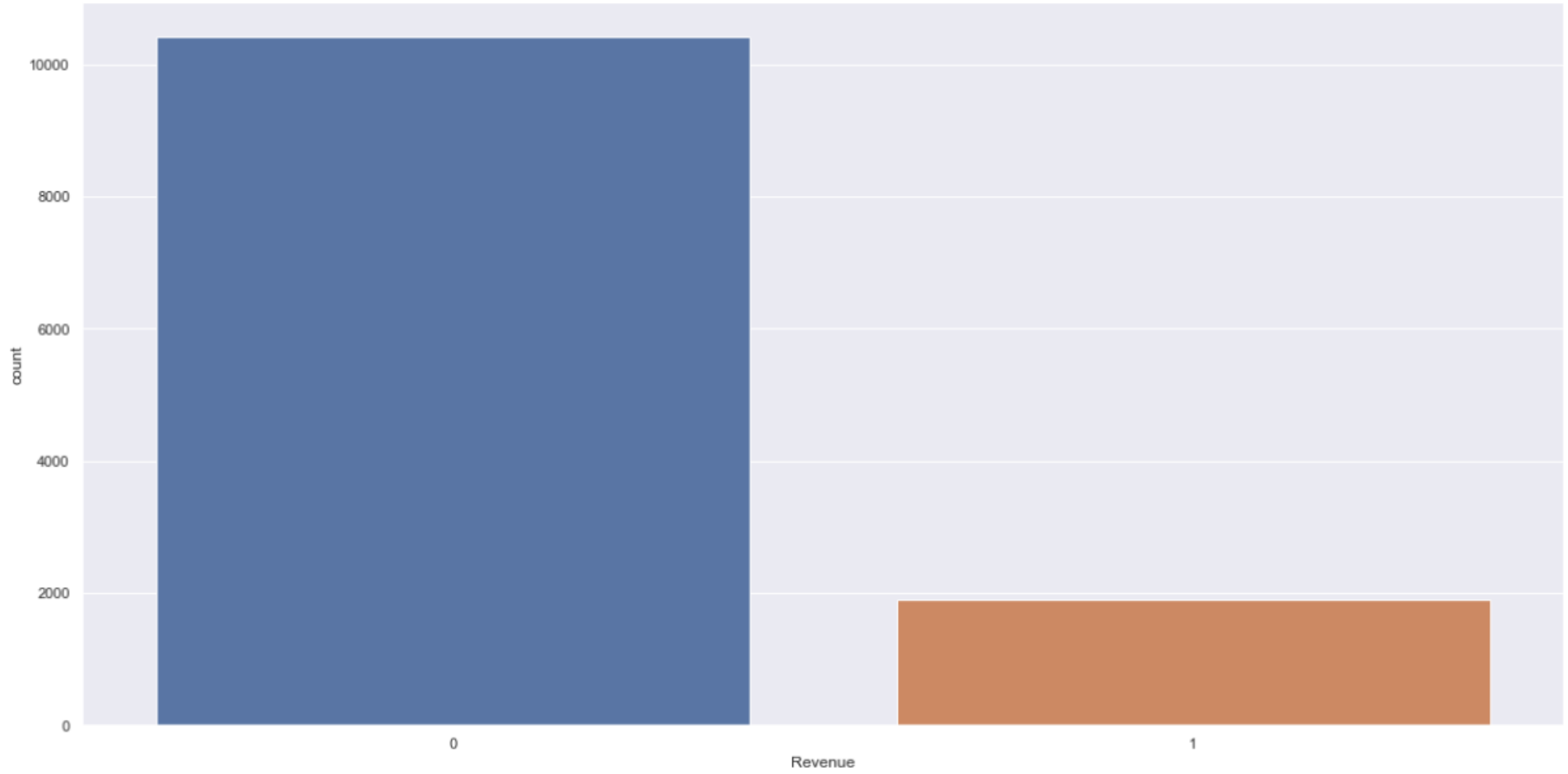
- The Dataset 'Online Shoppers Purchase Intention' is taken from UCI Machine Learning Repository.
- There are 12,330 instances in the data. These instances are sessions of online users.
- Each session has 18 features out of which 10 are numerical and 8 are categorical.
- To predict the intention of customers we take 'Revenue' as the target variable. If the output is 1 then the customer purchases, else it is not a purchase.

Numerical Features

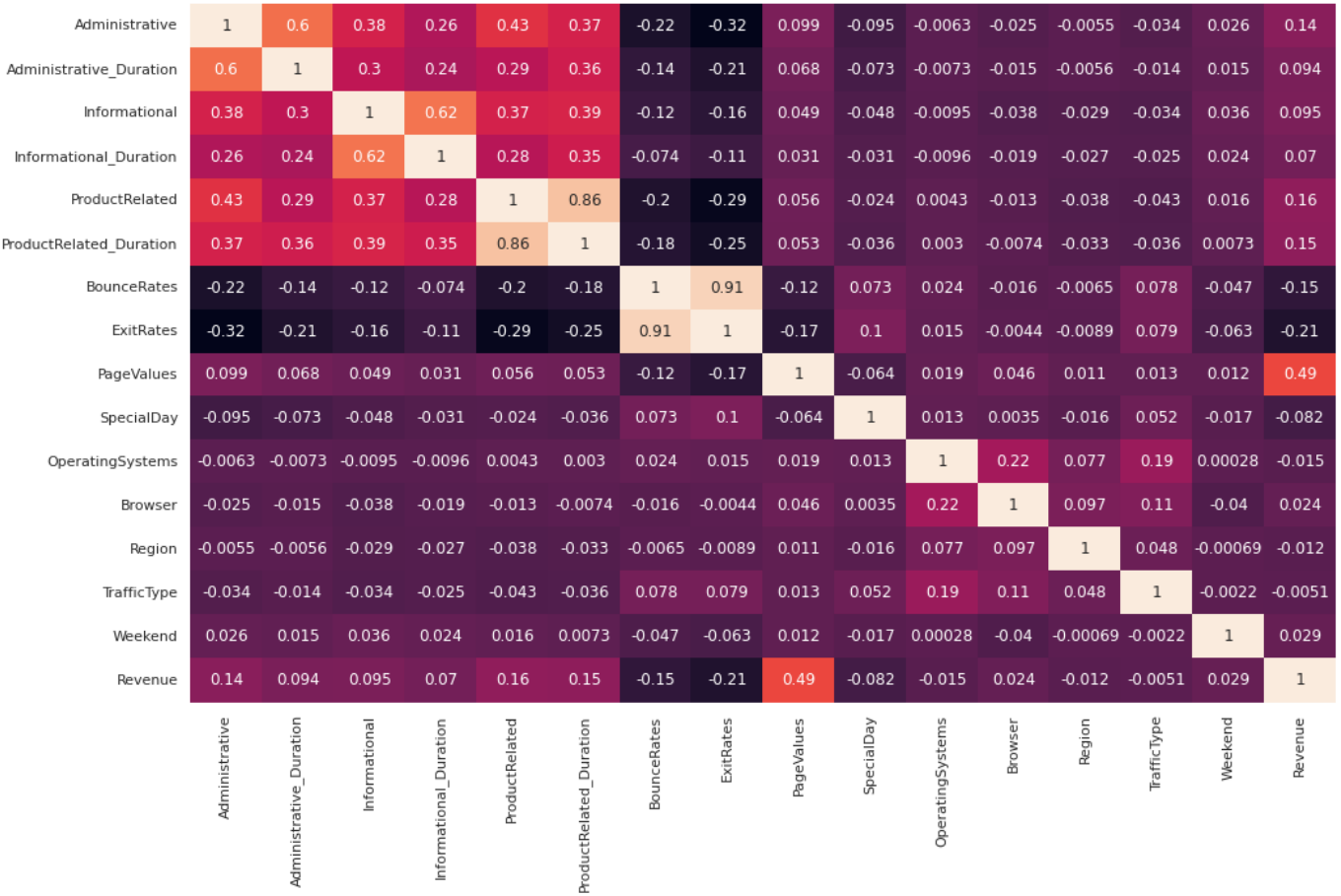
Feature name	Feature description	Min. val	Max. val	SD
Admin.	#pages visited by the visitor about account management	0	27	3.32
Ad. duration	#seconds spent by the visitor on account management related pages	0	3398	176.70
Info.	#informational pages visited by the visitor	0	24	1.26
Info. durat.	#seconds spent by the visitor on informational pages	0	2549	140.64
Prod.	#pages visited by visitor about product related pages	0	705	44.45
Prod.durat.	#seconds spent by the visitor on product related pages	0	63,973	1912.3
Bounce rate	Average bounce rate value of the pages visited by the visitor	0	0.2	0.04
Exit rate	Average exit rate value of the pages visited by the visitor	0	0.2	0.05
Page value	Average page value of the pages visited by the visitor	0	361	18.55
Special day	Closeness of the site visiting time to a special day	0	1.0	0.19

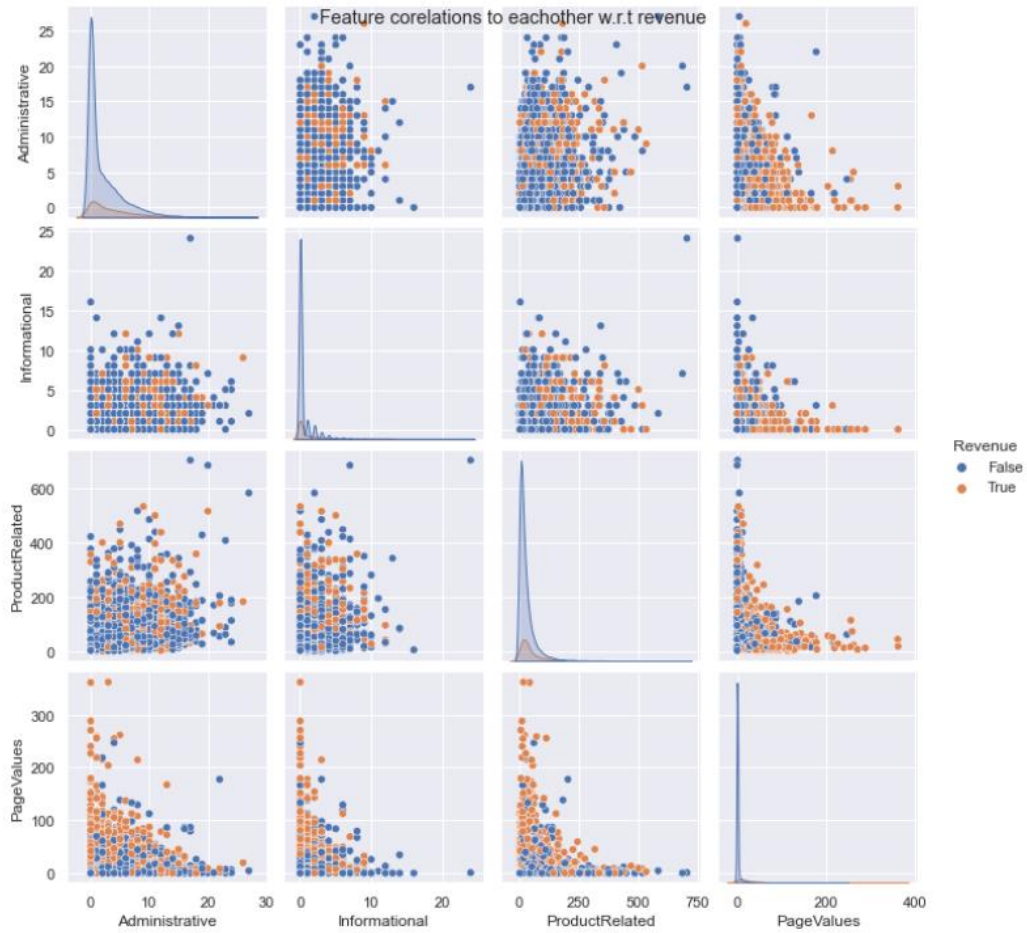
Categorical Features

Feature name	Feature description	Number of Values
OperatingSystems	Operating system of the visitor	8
Browser	Browser of the visitor	13
Region	Geographic region from which the session has been started by the visitor	9
TrafficType	Traffic source (e.g., banner, SMS, direct)	20
VisitorType	Visitor type as "New Visitor," "Returning Visitor," and "Other"	3
Weekend	Boolean value indicating whether the date of the visit is weekend	2
Month	Month value of the visit date	12
Revenue	Class label: whether the visit has been finalized with a transaction	2

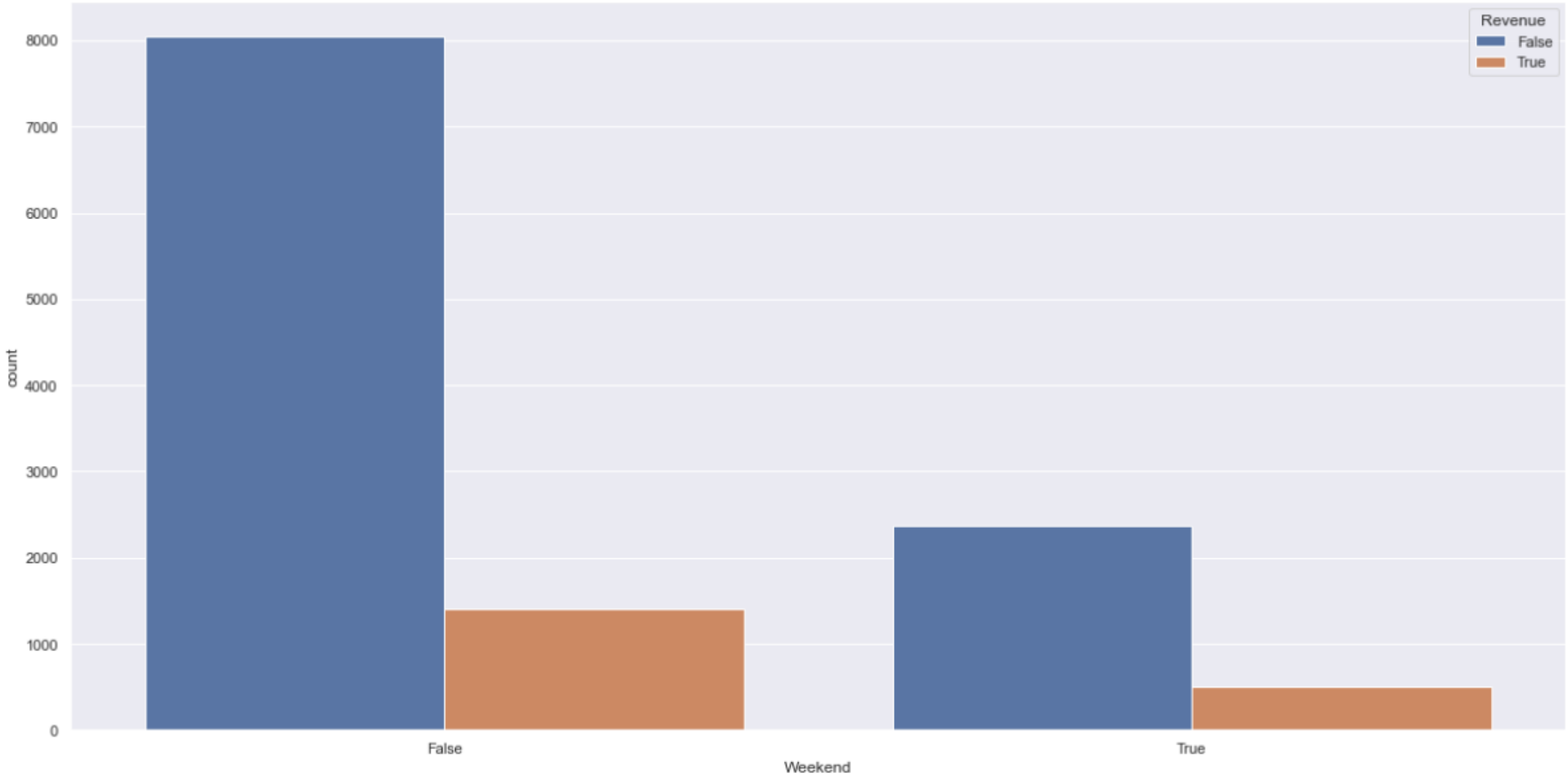


Skewed Dataset: Eighty-five percent (10,422) of the 12,330 sessions in the data set were negative class samples, which did not complete their shopping and generated no income. At the same time, the rest (1908) were positive class samples, which completed their shopping and generated money.

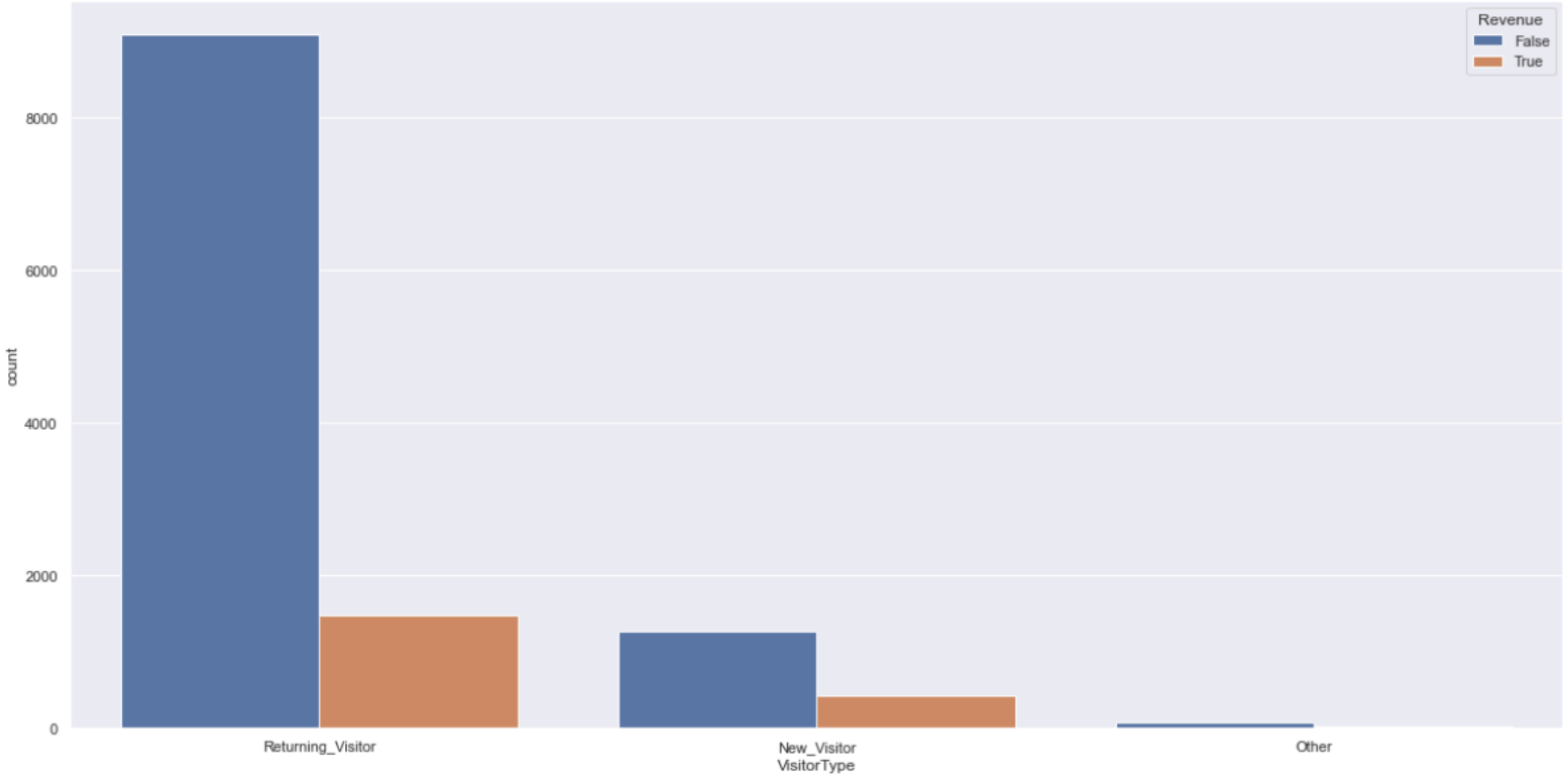




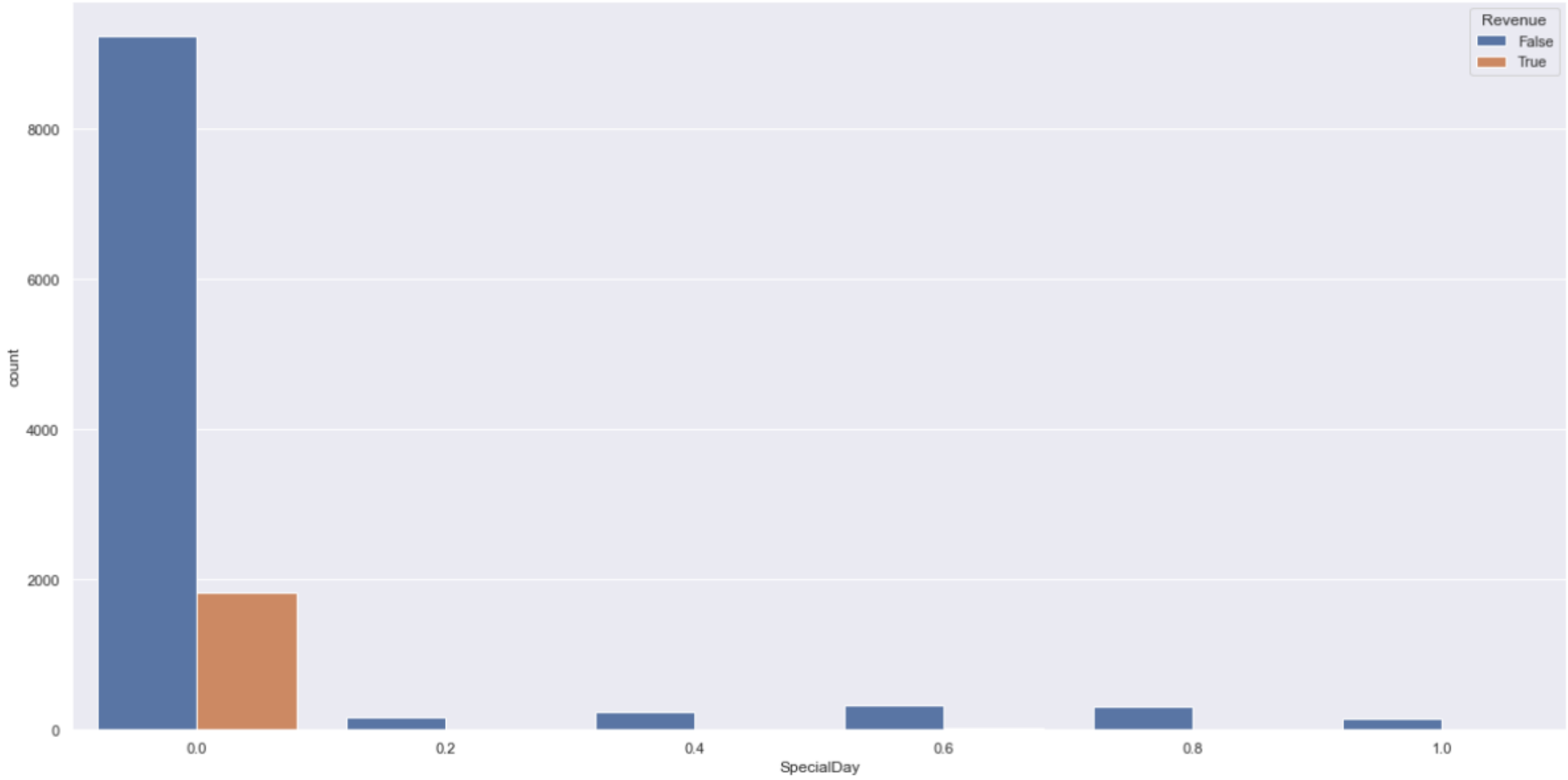
- This is the correlation plot of the whole data set.
- There are many features with a good and considerable amount of correlation with the target variable ‘Revenue’ and there are some of them highly correlated among themselves.



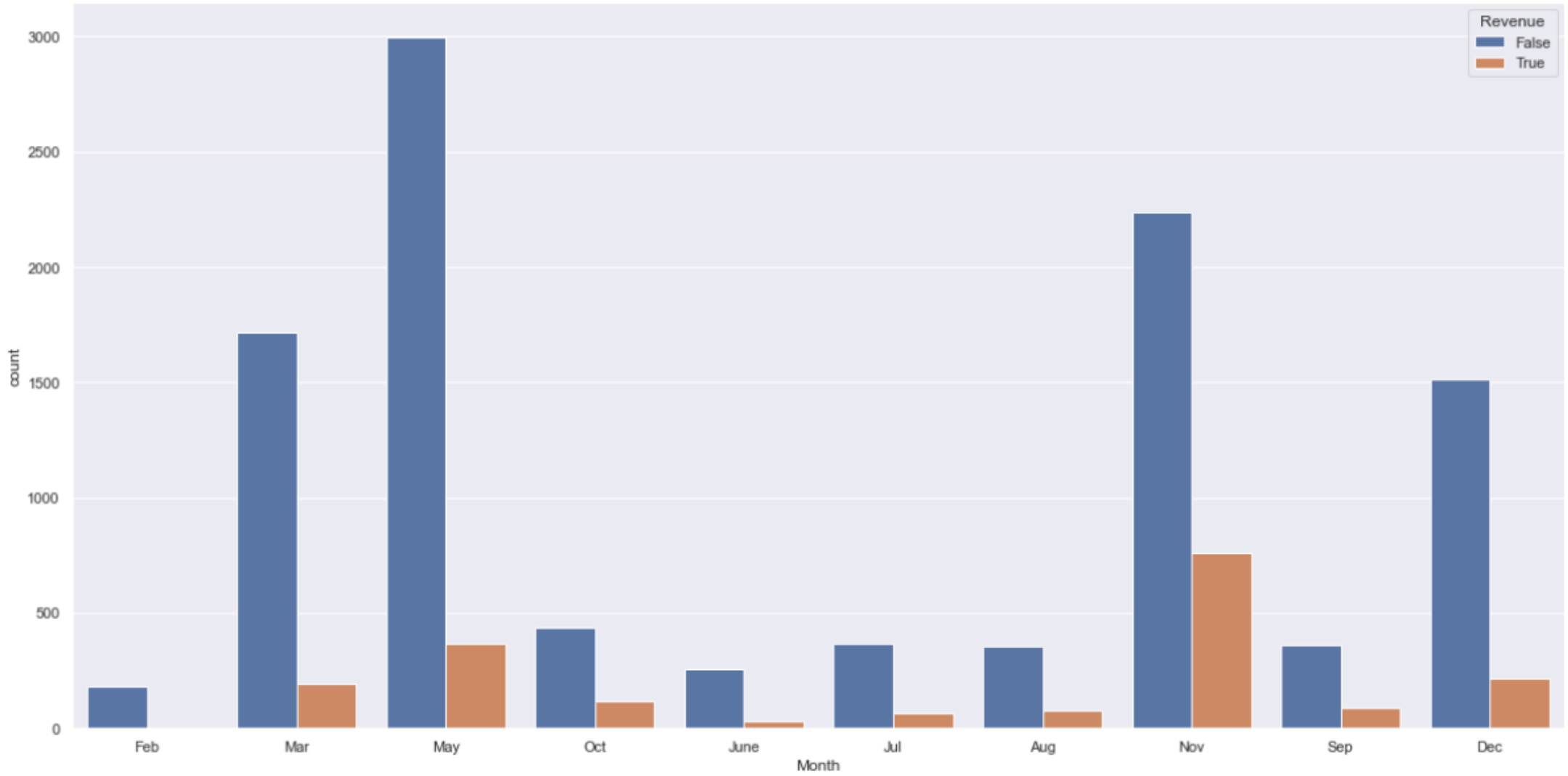
- *Customers generally indulged themselves in online shopping during the weekdays.*



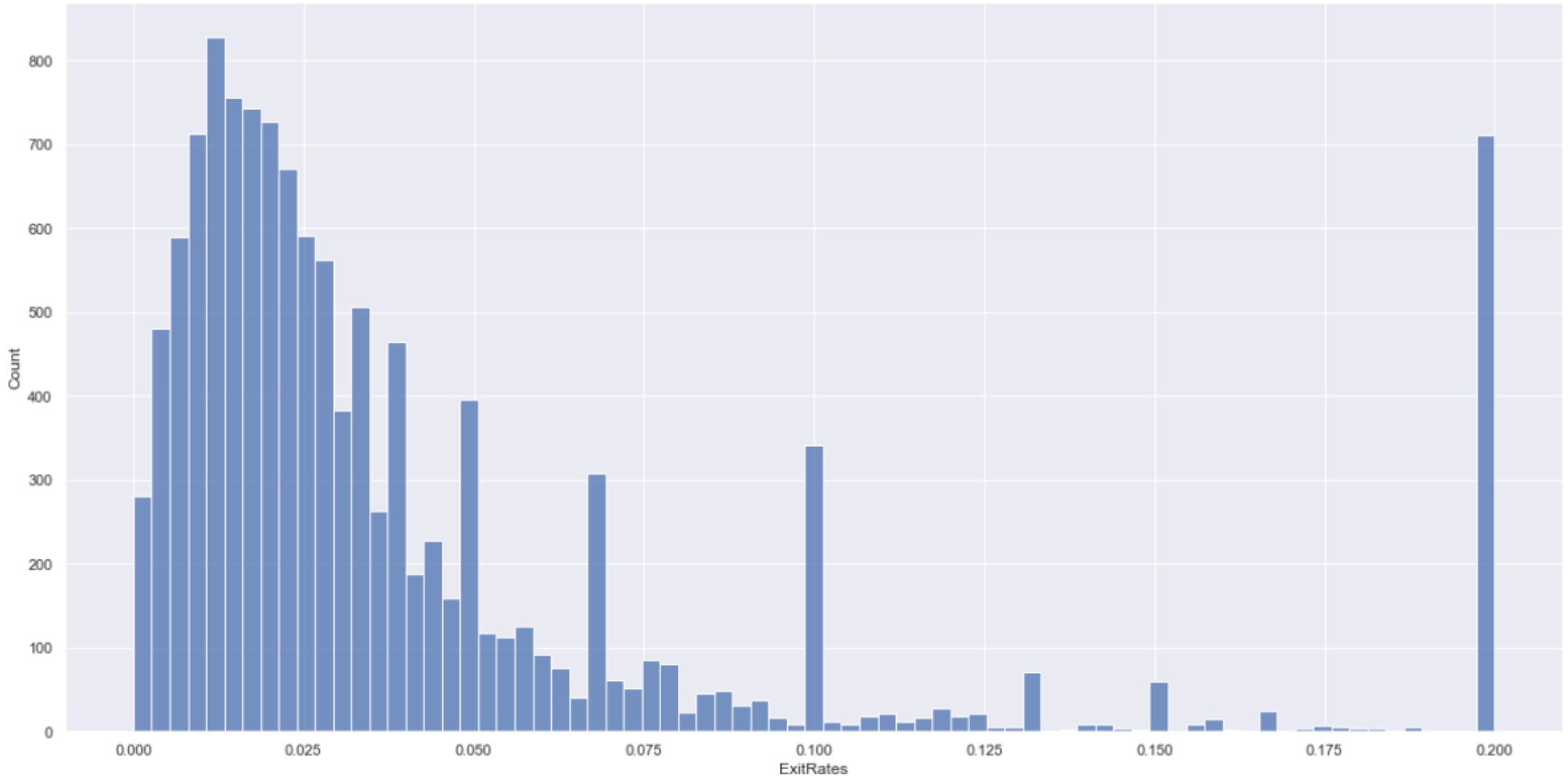
- *From this graph its clear maximum number of customers return to website even though not many of them contribute to revenue generation.*



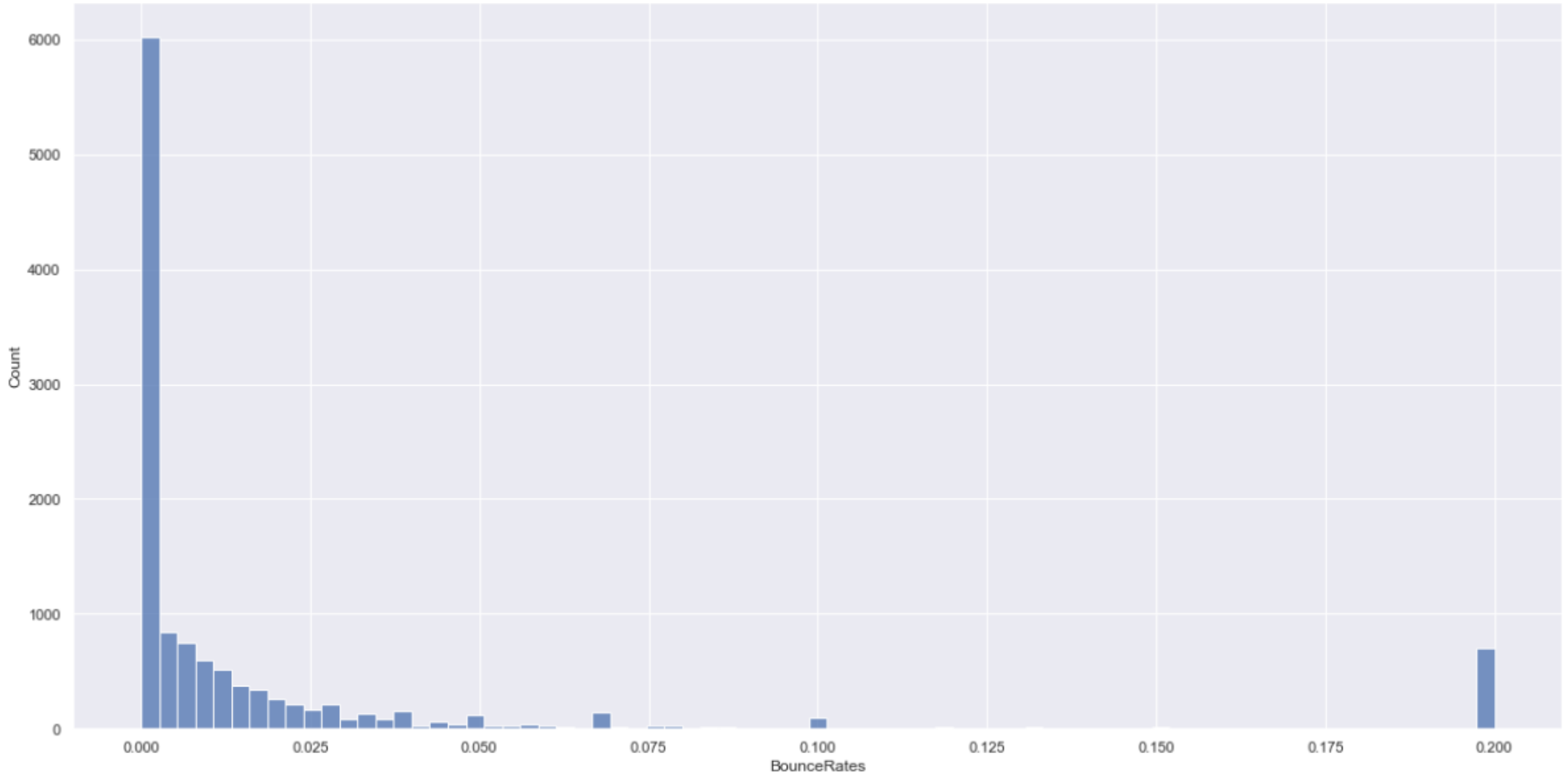
- *Majority of customers visited the website well before any special day (e.g. – Valentines day, Christmas)*



- Most of the revenue generated is during the months of May, Nov and March



This graph indicates the exit rate, lower the exit rate of a page in general the more attractive the page is to the user.



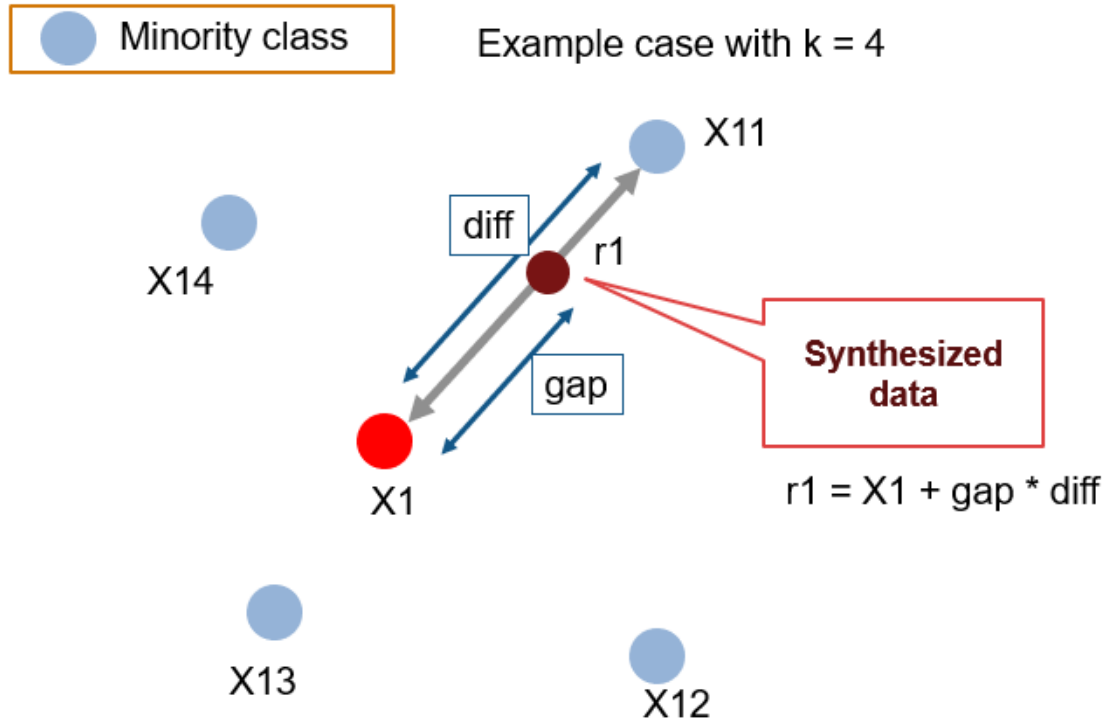
This graph indicates the bounce rate, low bounce rate show that the website visitors are not quitting after just reading one page. They are also clicking on other internal links and visiting other pages.

Implementing Oversampling methods

- From all the above results we can observe that the ‘recall’ isn’t great enough for us to consider these model as a final model.
- Since the data set is highly skewed there aren’t enough representative classes for the positive output class, Oversampling is one possible solution.
- Oversampling is replication the events of minority class.
- Potential problem: could be for this method is overfitting for noisy data, because noisy data will be replicate.
- SMOTE is one of the method with which we can perform oversampling.
- To avoid overfitting the procedure of randomized oversampling is performed (SMOTE and its alternatives) with cleaning noisy data.

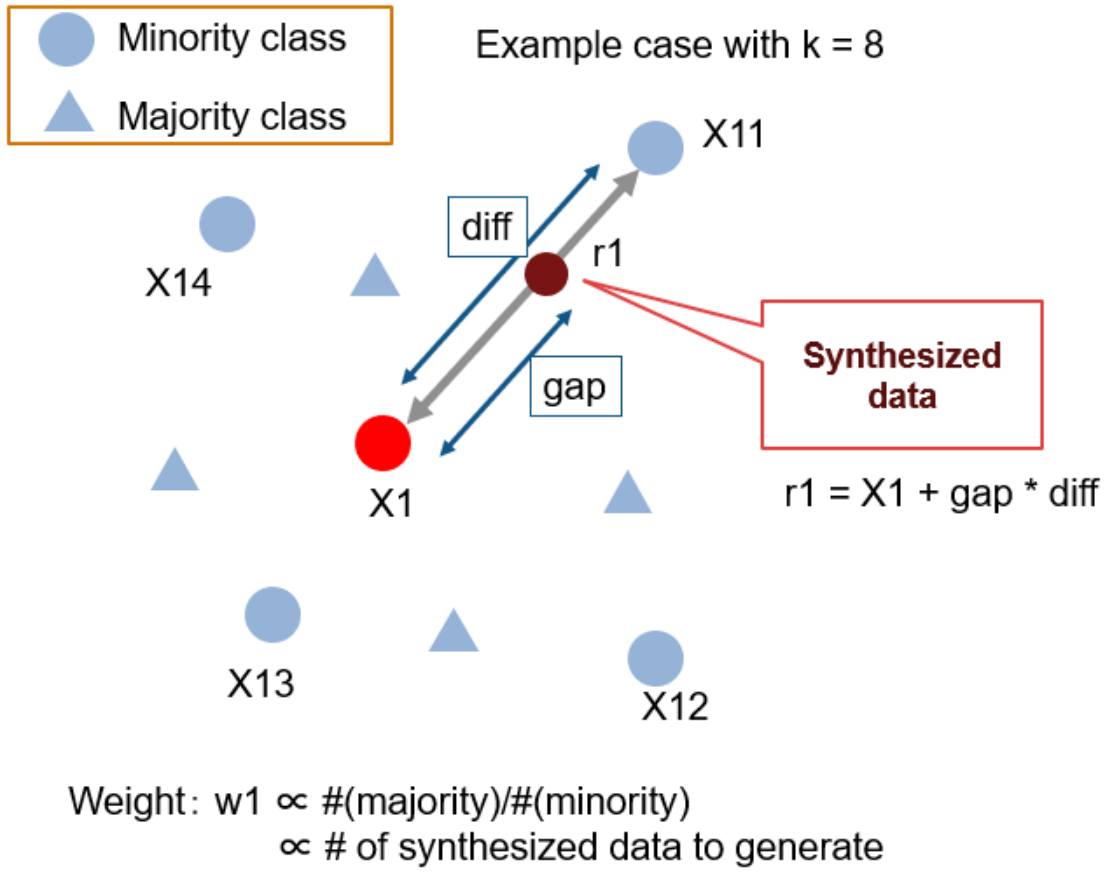
SMOTE

- SMOTE is an oversampling approach in which synthetic samples for the minority class are created.
- This approach aids in overcoming the problem of overfitting caused by random oversampling.
- It concentrates on the feature space in order to produce new examples by interpolating between positive instances that are close together.



ADASYN

- Adaptive Synthetic Sampling Approach: The ADASYN algorithm is a more generalized version of the SMOTE algorithm.
- By producing synthetic examples for the minority class, this technique also seeks to oversample it.
- However, it considers the density distribution which determines the number of synthetic instances created for difficult-to-learn data.
- As a result, it aids in adaptively adjusting judgment limits based on difficult-to-learn data.
- This is the most significant distinction from SMOTE.



SMOTE + Tomek

- Combining undersampling and oversampling approaches is required.
- The class clusters may be invading each other’s space after SMOTE’s oversampling. As a result, the model of the classifier will be overfit.
- SMOTE+TOMEK is a hybrid approach that tries to clear overlapping data points in sample space for each of the classes.
- Tomek linkages are paired samples of the opposite class that are the nearest neighbors to each other.
- As a result, the bulk of class observations from these linkages have been eliminated since it is thought that this will enhance class separation around the decision borders.
- Tomek linkages are now applied to oversampled minority class samples created by SMOTE in order to obtain better class clusters.

SMOTE + ENN

- A hybrid strategy is SMOTE + ENN, which removes a larger number of observations from the sample space.
- ENN is yet another under sampling strategy in which the majority class's nearest neighbors are approximated.
- If the nearest neighbors incorrectly label that specific instance of the majority class, it is eliminated.
- Integrating this approach with SMOTE's oversampled data aids in significant data cleaning.
- Samples from both groups are excluded due to NN's misclassification, consequently, the class distinction is more apparent and straightforward.

Implementation:

We as a team tried six models . They are:

1. Logistic Regression
2. Random Forest Classifier
3. K-NN Classifier
4. Multilayer Perceptron
5. Support Vector Classification
6. Adaboost Classifier

Logistic Regression:

	precision	recall	f1-score	support
0	0.88	0.97	0.92	2055
1	0.71	0.35	0.47	411
accuracy			0.87	2466
macro avg	0.80	0.66	0.70	2466
weighted avg	0.85	0.87	0.85	2466

Logistic Regression:

```
Test set Accuracy is 87.64190723321444 %
[[2361  260]
 [ 121  341]]
              precision    recall  f1-score   support

       False       0.95      0.90      0.93      2621
        True       0.57      0.74      0.64       462

    accuracy                           0.88      3083
   macro avg       0.76      0.82      0.78      3083
weighted avg       0.89      0.88      0.88      3083
```

Random Forest Classifier:

	precision	recall	f1-score	support
0	0.91	0.97	0.94	2055
1	0.75	0.52	0.61	411
accuracy			0.89	2466
macro avg	0.83	0.74	0.77	2466
weighted avg	0.88	0.89	0.88	2466

Random Forest Classifier:

```
Test set Accuracy is 90.78819331819656 %
[[2477  144]
 [ 140  322]]
              precision    recall  f1-score   support

       False       0.95      0.95      0.95      2621
        True       0.69      0.70      0.69       462

    accuracy                           0.91      3083
   macro avg       0.82      0.82      0.82      3083
weighted avg       0.91      0.91      0.91      3083
```

Support Vector Classifier:

	precision	recall	f1-score	support
0	0.83	1.00	0.91	2055
1	0.83	0.01	0.02	411
accuracy			0.83	2466
macro avg	0.83	0.51	0.47	2466
weighted avg	0.83	0.83	0.76	2466

Support Vector Classifier:

	precision	recall	f1-score	support
0	0.93	0.73	0.82	2055
1	0.35	0.74	0.48	411
accuracy			0.73	2466
macro avg	0.64	0.73	0.65	2466
weighted avg	0.84	0.73	0.76	2466

K Nearest Neighbor:

	precision	recall	f1-score	support
0	0.87	0.97	0.91	2055
1	0.61	0.27	0.37	411
accuracy			0.85	2466
macro avg	0.74	0.62	0.64	2466
weighted avg	0.82	0.85	0.82	2466

K Nearest Neighbor:

	precision	recall	f1-score	support
0	0.91	0.82	0.86	2055
1	0.40	0.58	0.47	411
accuracy			0.78	2466
macro avg	0.65	0.70	0.67	2466
weighted avg	0.82	0.78	0.80	2466

Multi Layer Perceptron:

```
Test set Accuracy is 88.55011352578657 %
[[2587   34]
 [ 319  143]]
              precision    recall  f1-score   support

       False       0.89      0.99      0.94      2621
        True       0.81      0.31      0.45       462

    accuracy                           0.89      3083
   macro avg       0.85      0.65      0.69      3083
weighted avg       0.88      0.89      0.86      3083
```

Multi Layer Perceptron:

```
Test set Accuracy is 88.97178073305221 %
[[2541   80]
 [ 260  202]]
              precision    recall  f1-score   support

       False       0.91      0.97      0.94      2621
        True       0.72      0.44      0.54       462

    accuracy                           0.89      3083
   macro avg       0.81      0.70      0.74      3083
weighted avg       0.88      0.89      0.88      3083
```

Adaptive Boosting:

	precision	recall	f1-score	support
0	0.91	0.95	0.93	2055
1	0.69	0.54	0.61	411
accuracy			0.88	2466
macro avg	0.80	0.75	0.77	2466
weighted avg	0.88	0.88	0.88	2466

Adaptive Boosting:

	precision	recall	f1-score	support
0	0.93	0.92	0.93	2055
1	0.62	0.65	0.64	411
accuracy			0.88	2466
macro avg	0.78	0.79	0.78	2466
weighted avg	0.88	0.88	0.88	2466

0.9 Train set & 0.1 Test set

SMOTE:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	1030
1	0.65	0.64	0.64	203
accuracy			0.88	1233
macro avg	0.79	0.78	0.79	1233
weighted avg	0.88	0.88	0.88	1233

ADASYN:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	1030
1	0.64	0.63	0.64	203
accuracy			0.88	1233
macro avg	0.79	0.78	0.78	1233
weighted avg	0.88	0.88	0.88	1233

SMOTETomek:

	precision	recall	f1-score	support
0	0.93	0.93	0.93	1030
1	0.66	0.66	0.66	203
accuracy			0.89	1233
macro avg	0.79	0.80	0.80	1233
weighted avg	0.89	0.89	0.89	1233

SMOTEENN:

	precision	recall	f1-score	support
0	0.96	0.89	0.92	1030
1	0.59	0.79	0.67	203
accuracy			0.87	1233
macro avg	0.77	0.84	0.80	1233
weighted avg	0.89	0.87	0.88	1233

0.8 Train set & 0.2 Test set

SMOTE:

	precision	recall	f1-score	support
0	0.93	0.94	0.94	2055
1	0.69	0.66	0.67	411
accuracy			0.89	2466
macro avg	0.81	0.80	0.80	2466
weighted avg	0.89	0.89	0.89	2466

ADASYN:

	precision	recall	f1-score	support
0	0.93	0.94	0.94	2055
1	0.69	0.66	0.68	411
accuracy			0.89	2466
macro avg	0.81	0.80	0.81	2466
weighted avg	0.89	0.89	0.89	2466

SMOTETomek:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	2055
1	0.69	0.68	0.69	411
accuracy			0.90	2466
macro avg	0.81	0.81	0.81	2466
weighted avg	0.90	0.90	0.90	2466

SMOTEENN:

	precision	recall	f1-score	support
0	0.96	0.90	0.93	2055
1	0.61	0.79	0.69	411
accuracy			0.88	2466
macro avg	0.78	0.84	0.81	2466
weighted avg	0.90	0.88	0.89	2466

0.7 Train set & 0.3 Test set

SMOTE:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	3124
1	0.66	0.65	0.65	575
accuracy			0.89	3699
macro avg	0.80	0.79	0.79	3699
weighted avg	0.89	0.89	0.89	3699

ADASYN:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	3124
1	0.66	0.68	0.67	575
accuracy			0.90	3699
macro avg	0.80	0.81	0.80	3699
weighted avg	0.90	0.90	0.90	3699

SMOTETomek:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	3124
1	0.66	0.67	0.66	575
accuracy			0.90	3699
macro avg	0.80	0.80	0.80	3699
weighted avg	0.90	0.90	0.90	3699

SMOTEENN:

	precision	recall	f1-score	support
0	0.96	0.90	0.93	3124
1	0.58	0.79	0.67	575
accuracy			0.88	3699
macro avg	0.77	0.84	0.80	3699
weighted avg	0.90	0.88	0.89	3699

0.6 Train set & 0.4 Test set

SMOTE:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	4170
1	0.67	0.66	0.66	762
accuracy			0.90	4932
macro avg	0.80	0.80	0.80	4932
weighted avg	0.90	0.90	0.90	4932

ADASYN:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	4170
1	0.66	0.67	0.66	762
accuracy			0.89	4932
macro avg	0.80	0.80	0.80	4932
weighted avg	0.90	0.89	0.90	4932

SMOTETomek:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	4170
1	0.67	0.69	0.68	762
accuracy			0.90	4932
macro avg	0.80	0.81	0.81	4932
weighted avg	0.90	0.90	0.90	4932

SMOTEENN:

	precision	recall	f1-score	support
0	0.96	0.90	0.93	4170
1	0.58	0.80	0.67	762
accuracy			0.88	4932
macro avg	0.77	0.85	0.80	4932
weighted avg	0.90	0.88	0.89	4932

Observations

- Out of all the models Random Forest performed best when actual data is trained and when the oversampled data is trained.
- It can be observed that when the actual data is trained the metrics for class '1'(which is of lesser observations) are not comparable to oversampled data metrics.
- Since Random Forest algorithm performed the best on the dataset we checked using different test set sizes and measure the performance.

Conclusion

Comparison of Different Models before oversampling

Model	Precision	Recall	F1-Score
Random Forest	**0.75**	**0.52**	**0.61**
Logistic Regression	0.71	0.35	0.47
Multi Layer Perceptron	0.50	0.76	0.60
K-NN Classifier	0.78	0.19	0.31
Adaptive Boost	0.69	0.54	0.61
Support vector Classifier	0.83	0.01	0.02

Comparison of Different Models after oversampling using SMOTE

Model	Precision	Recall	F1-Score
Random Forest	**0.69**	**0.67**	**0.68**
Logistic Regression	0.62	0.72	0.67
Multi Layer Perceptron	0.66	0.60	0.63
K-NN Classifier	0.40	0.58	0.47
Adaptive Boost	0.62	0.65	0.64
Support vector Classifier	0.35	0.74	0.48

Comparison of Different Models after oversampling using ADASYN

Model	Precision	Recall	F1-Score
Random Forest	**0.68**	**0.66**	**0.67**
Logistic Regression	0.60	0.72	0.65
Multi Layer Perceptron	0.76	0.32	0.45
K-NN Classifier	0.37	0.61	0.46
Adaptive Boost	0.62	0.67	0.64
Support vector Classifier	0.32	0.75	0.45

Comparison of Different Models after oversampling using SMOTE + Tomek

Model	Precision	Recall	F1-Score
Random Forest	**0.70**	**0.66**	**0.68**
Logistic Regression	0.60	0.71	0.65
Multi Layer Perceptron	0.42	0.78	0.54
K-NN Classifier	0.39	0.55	0.46
Adaptive Boost	0.60	0.72	0.65
Support vector Classifier	0.36	0.73	0.48

Comparison of Different Models after oversampling using SMOTE + ENN

Model	Precision	Recall	F1-Score
Random Forest	**0.59**	**0.80**	**0.68**
Logistic Regression	0.56	0.80	0.66
Multi Layer Perceptron	0.49	0.79	0.61
K-NN Classifier	0.35	0.69	0.46
Adaptive Boost	0.59	0.76	0.67
Support vector Classifier	0.30	0.82	0.44