Data Mining Project: Online Shopper Purchase Prediction

Team RR
Rudranil Naskar
2022MT11287
Raman Jakhar
2022MT11941

February 9, 2025

1 Non-competitive Part

1.1 Introduction

The objective of this project is to predict whether an online shopper will make a purchase based on their activity and historical user trends. The dataset consists of 10 numerical and 8 categorical attributes, with the 'Revenue' attribute serving as the class label.

1.2 Exploratory Data Analysis (EDA)

1.2.1 Dataset Overview

The dataset was loaded and examined using `.head()`, `.info()`, and `.describe()`. Figures provides a summary.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates
0	1	15.0	1	157.0	36	3010.532051	0.000000
1	0	0.0	0	0.0	57	820.363636	0.035088
2	9	228.2	1	0.0	7	186.400000	0.020000
3	3	72.6	0	0.0	17	544.100000	0.000000
4	0	0.0	4	8.0	66	1514.836310	0.022887

ExitRates	PageValues	SpecialDay	Month	OperatingSystems	Browser	Region	TrafficType	VisitorType	Weekend	Revenue
0.014620	0.0	0.0	May	2	2	3	2	Returning_Visitor	True	False
0.061651	0.0	0.0	June	3	2	3	13	Returning_Visitor	False	False
0.030000	0.0	0.0	Nov	2	2	1	20	Returning_Visitor	False	False
0.002000	0.0	0.0	Sep	2	2	9	2	New_Visitor	False	False
0.044914	0.0	0.0	Dec	2	2	6	2	Returning_Visitor	False	False

.head()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11097 entries, 0 to 11096
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           11097 non-null  int64
 1   Administrative_Duration  11097 non-null  float64
 2   Informational            11097 non-null  int64
 3   Informational_Duration   11097 non-null  float64
 4   ProductRelated           11097 non-null  int64
 5   ProductRelated_Duration  11097 non-null  float64
 6   BounceRates              11097 non-null  float64
 7   ExitRates                11097 non-null  float64
 8   PageValues               11097 non-null  float64
 9   SpecialDay               11097 non-null  float64
 10  Month                    11097 non-null  object
 11  OperatingSystems         11097 non-null  int64
 12  Browser                  11097 non-null  int64
 13  Region                   11097 non-null  int64
 14  TrafficType              11097 non-null  int64
 15  VisitorType              11097 non-null  object
 16  Weekend                  11097 non-null  bool
 17  Revenue                  11097 non-null  bool
dtypes: bool(2), float64(7), int64(7), object(2)
memory usage: 1.4+ MB
```

.info()

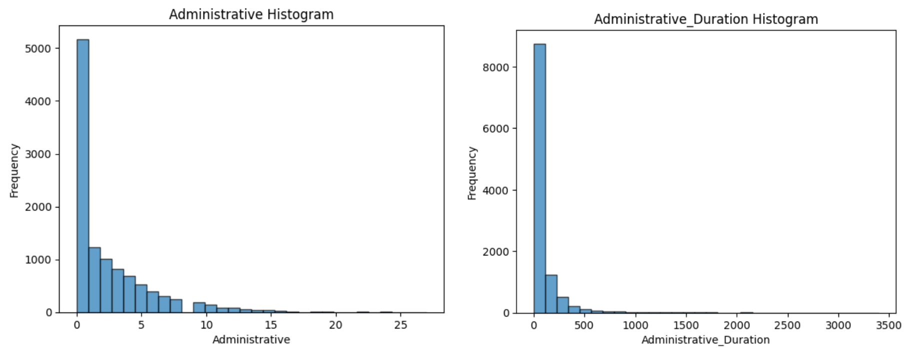
	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration
count	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000
mean	2.311886	81.118365	0.509777	34.867509	31.715419	1194.757649
std	3.317760	178.842997	1.277939	141.664660	44.192612	1908.767956
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000	187.000000
50%	1.000000	8.000000	0.000000	0.000000	18.000000	601.971429
75%	4.000000	92.300000	0.000000	0.000000	38.000000	1466.088462
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230

	BounceRates	ExitRates	PageValues	SpecialDay	OperatingSystems	Browser	Region	TrafficType
count	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000	11097.000000
mean	0.021933	0.042813	5.860658	0.061278	2.120843	2.354060	3.146796	4.072182
std	0.048070	0.048270	18.496266	0.198846	0.914069	1.718938	2.410359	4.036303
min	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000
25%	0.000000	0.014286	0.000000	0.000000	2.000000	2.000000	1.000000	2.000000
50%	0.003030	0.025000	0.000000	0.000000	2.000000	2.000000	3.000000	2.000000
75%	0.016667	0.050000	0.000000	0.000000	3.000000	2.000000	4.000000	4.000000
max	0.200000	0.200000	361.763742	1.000000	8.000000	13.000000	9.000000	20.000000

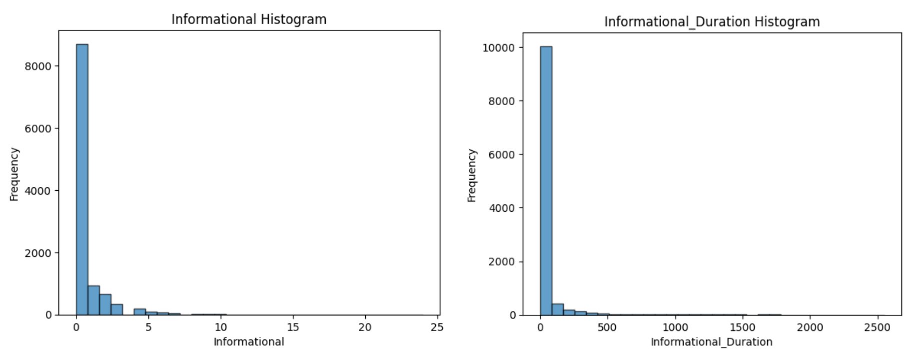
.describe()

1.2.2 Visualise the distribution of features

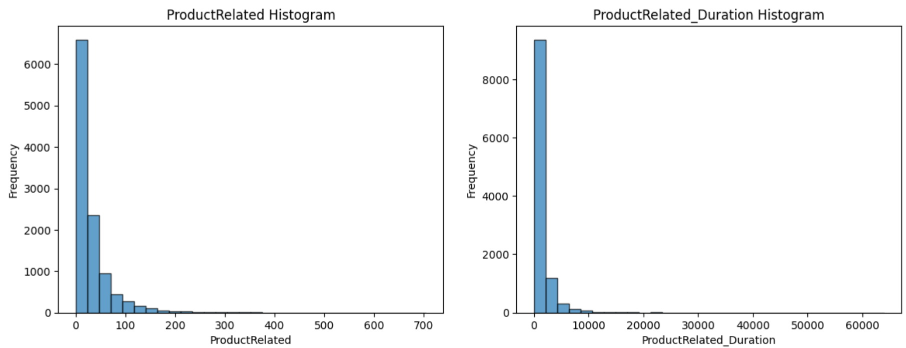
Histograms were used to visualize numerical variables, and count plots for categorical features.



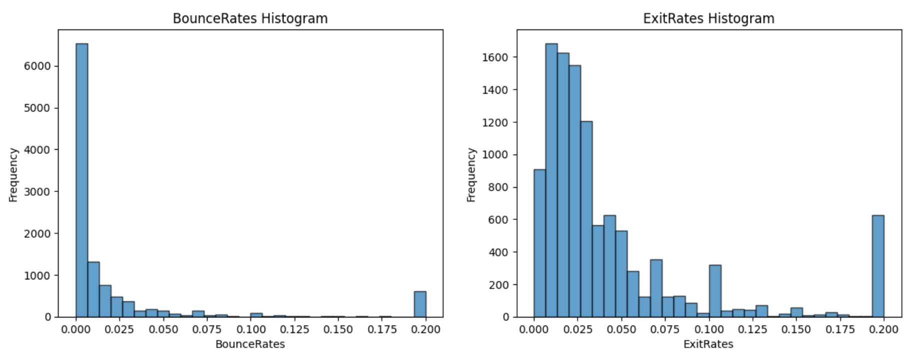
Histograms of Administrative (left) and Administrative_Duration (right)



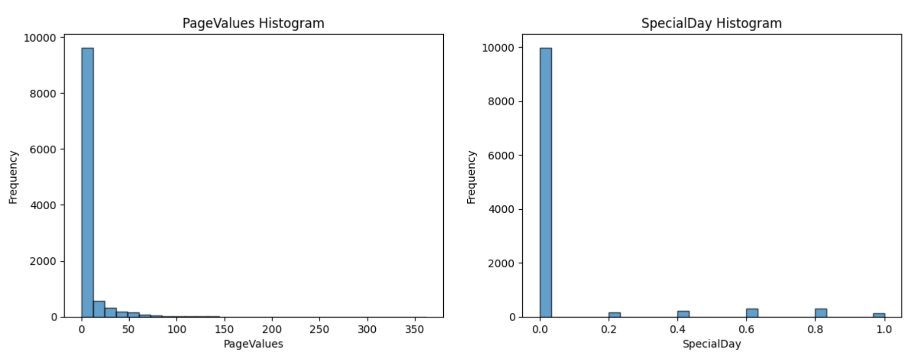
Histograms of Informational (left) and Informational_Duration (right)



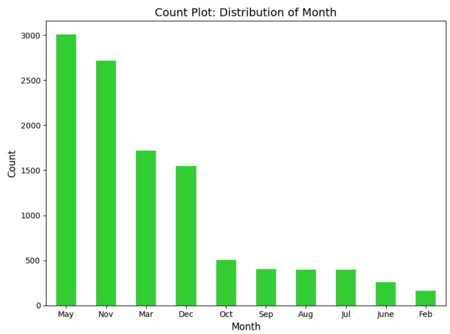
Histograms of ProductRelated (left) and ProductRelated_Duration (right)

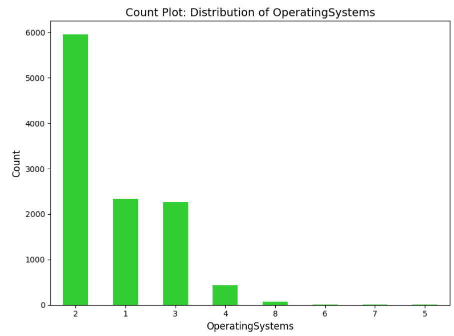


Histograms of BounceRate (left) and ExitRate (right)

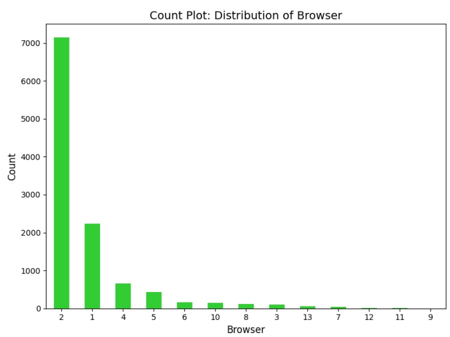


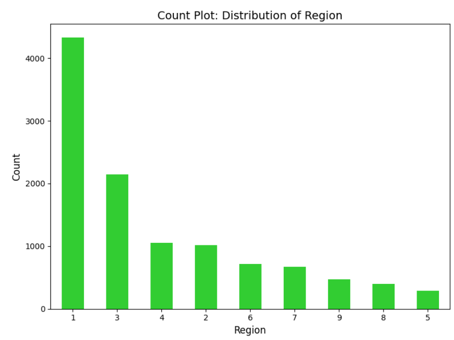
Histograms of PageValue (left) and SpecialDay (right)



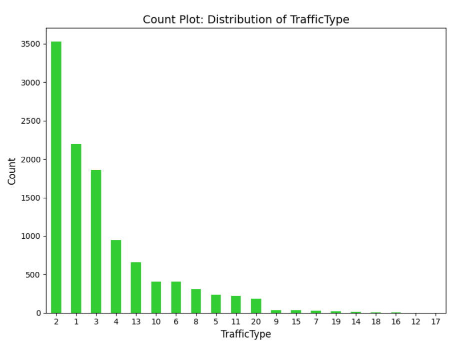


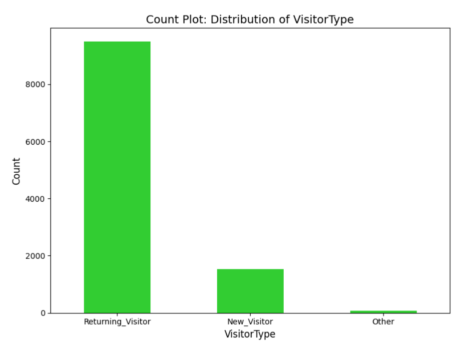
Count Plots of Months (left) and Operation Systems (right)



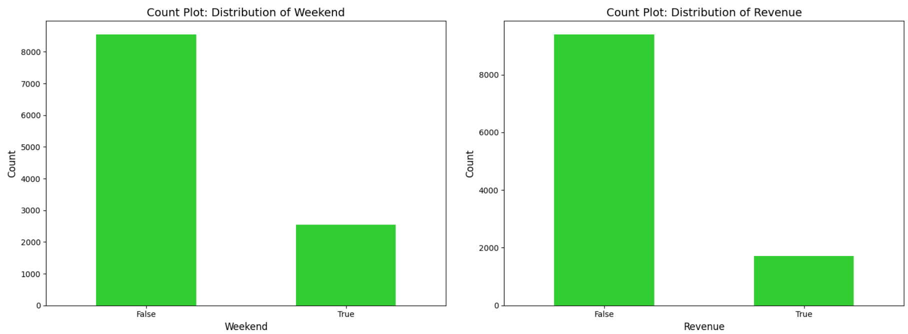


Count Plots of Browser (left) and Region (right)



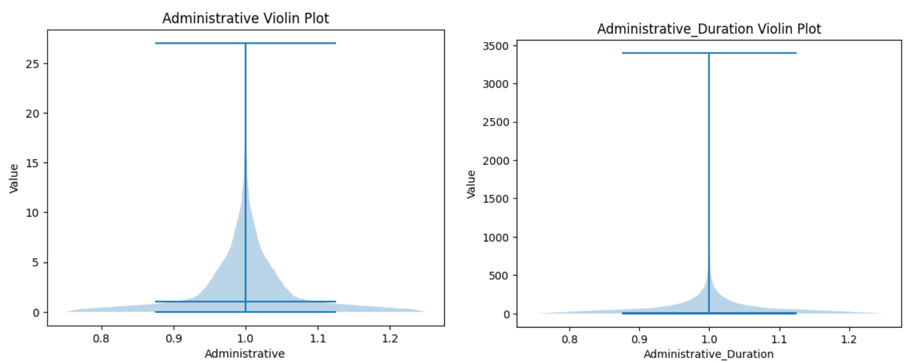


Count Plots of TrafficType (left) and VisitorType (right)

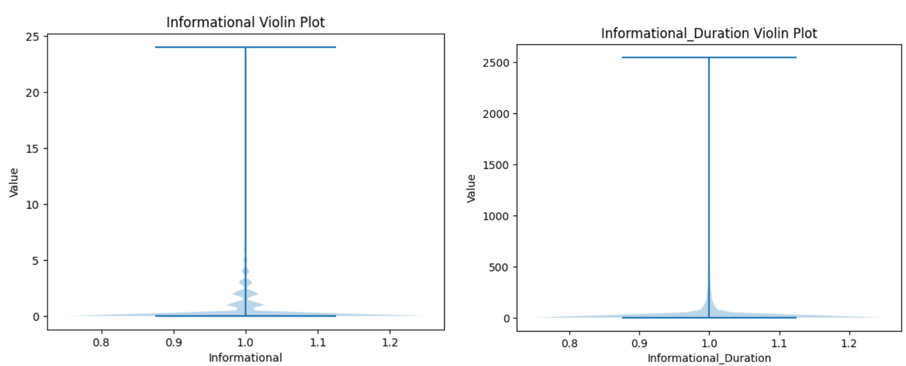


Count Plots of Weekend (left) and Revenue (right)

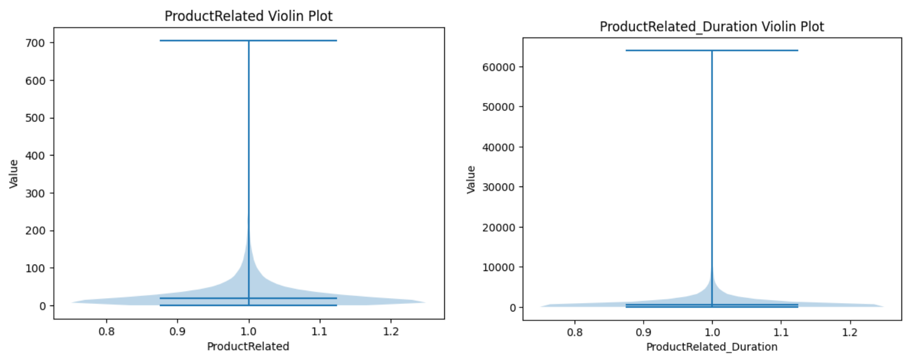
We then used violin plots and stacked bar plots for comparison of distribution of features for each target class.



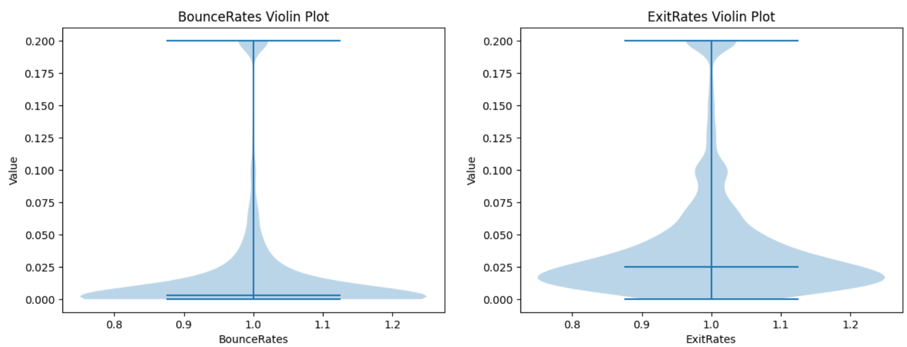
Violin Plots of Administrative (left) and Administrative_Duration (right)



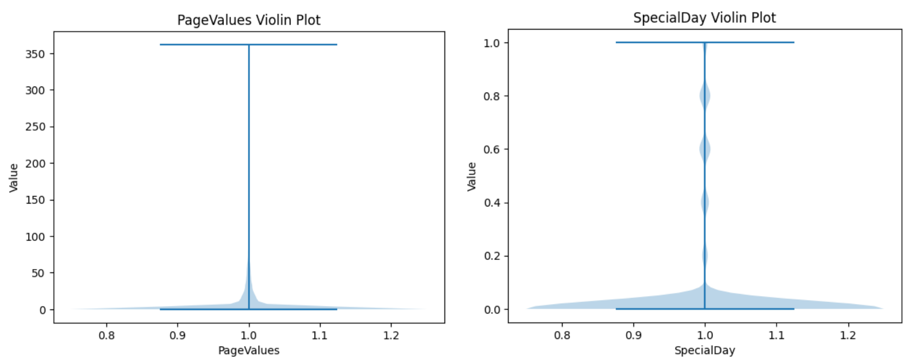
Violin Plots of Informational (left) and Informational_Duration (right)



Violin Plots of ProductRelated (left) and ProductRelated_Duration (right)

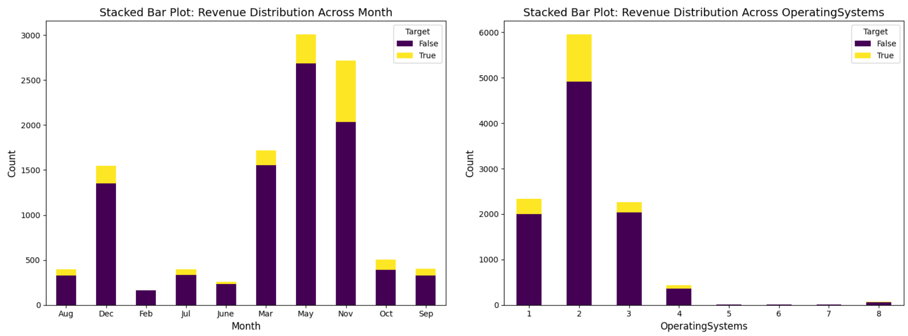


Violin Plots of BounceRate (left) and ExitRate (right)

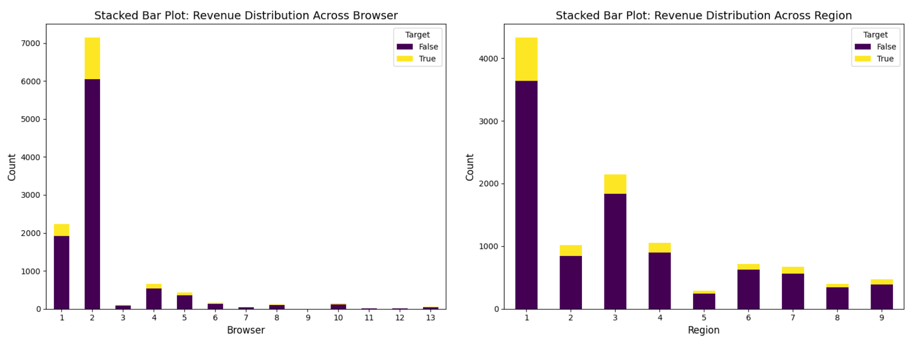


Violin Plots of PageValue (left) and SpecialDay (right)

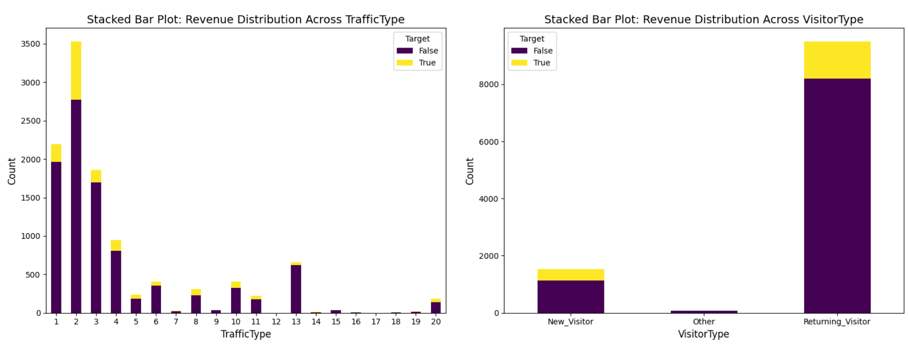
Stack Bar Plots of Categorical Features Comparison



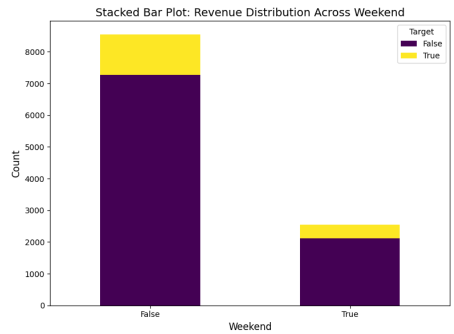
Stack Bar Plots of Months (left) and Operation Systems (right)



Stack Bar Plots of Browser (left) and Region (right)



Stack Bar Plots of TrafficType (left) and VisitorType (right)



Stack Bar Plots of Weekend

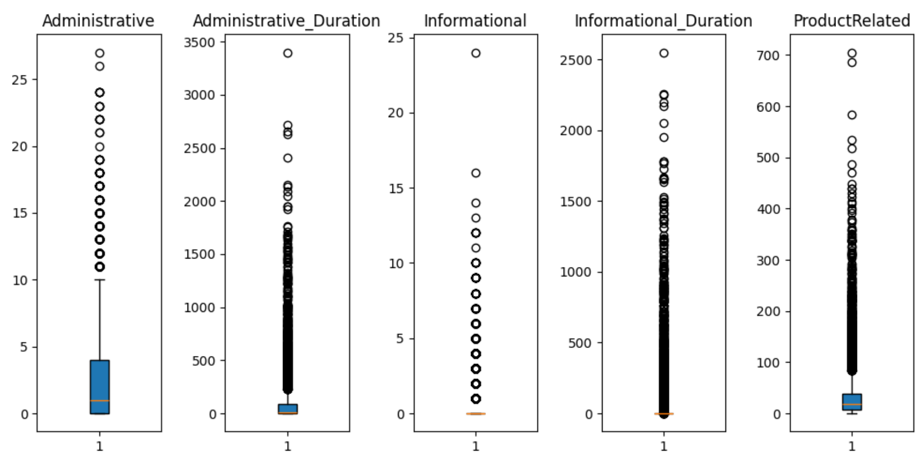
1.2.3 Takeaways from EDA

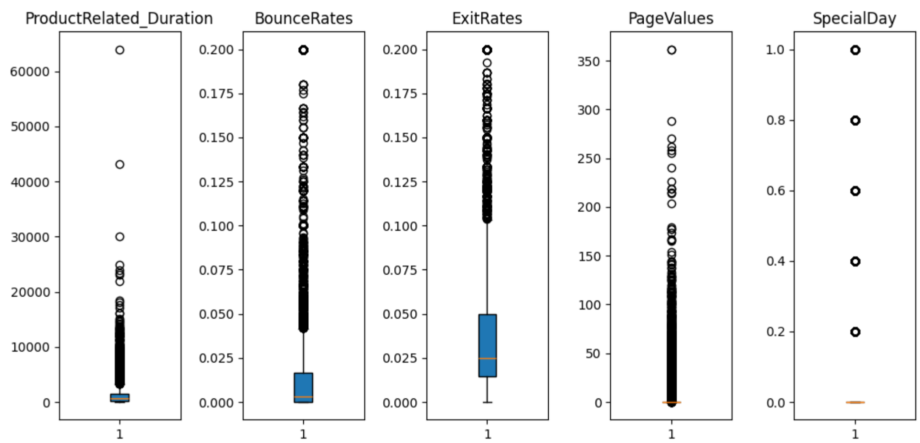
1. The data set is highly imbalanced, with very few True samples
2. All numerical features except 'Exit Rates' have a lot of 0 samples
3. Some features have a very similar plot, they might be highly correlated (more on this in competitive part)

1.3 Data Preprocessing

1.3.1 Outlier Removal

Boxplots and the interquartile range (IQR) method were used to detect and remove outliers.





Boxplots for outlier detection

Fraction of outliers in:

1. Administrative : 0.03
2. Administrative_Duration : 0.12
3. Informational : 0.28
4. Informational_Duration : 0.28
5. ProductRelated : 0.38
6. ProductRelated_Duration : 0.37
7. BounceRates : 0.49
8. ExitRates : 0.46
9. PageValues : 0.54
10. SpecialDay : 0.59

Removing more than 10 percent of the rows will result in too much data loss. Since only 'Administrative' has less than 10 percent outliers by IQR method, we only consider this feature for outlier removal.

1.3.2 Encoding and Standardization

Categorical variables were converted to numerical values using one-hot encoding. Numerical variables were standardized using z-score normalization with `StandardScaler`.

1.4 Logistic Regression Implementation

1.4.1 Data Splitting

The dataset was split into training sets (80%) and testing sets (20%) without randomization.

1.4.2 Implementation from Scratch

We implemented a logistic regression model was implemented from first principles and trained on the dataset. We used batch gradient descent for maximizing likelihood function. The model was trained on the data set using learning rate 0.0001, maximum epoch 7000 and a tolerance of 10e-8.

1.4.3 Performance Evaluation

The model's performance was evaluated using accuracy, precision, recall, and F1-score. The results are summarized in the tables below.

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression (Scratch)	0.90	0.42	0.78	0.54
Logistic Regression (Sklearn)	0.90	0.42	0.78	0.54
SVC	0.91	0.57	0.80	0.66
Decision Tree	1.00	1.00	1.00	1.00

Table 1: Performance comparison of different models on Training Dataset

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression (Scratch)	0.89	0.41	0.72	0.52
Logistic Regression (Sklearn)	0.89	0.41	0.73	0.82
SVC	0.89	0.51	0.68	0.58
Decision Tree	0.87	0.60	0.57	0.59

Table 2: Performance comparison of different models on Testing Dataset

1.5 Conclusion

Comparison of the models :

1. Our implementation of Logistic Regression (‘LogisticRegressor‘) gives the same accuracy on testing data (upto 2 decimal) places as ‘sklearn‘’s ‘LogisticRegression‘, ‘SVC‘ and ‘DecisionTree‘.

2. The Precision of ‘DecisionTree‘ is the best, better by 0.20 than our ‘LogisticRegressor‘.

3. ‘sklearn‘’s ‘LogisticRegression‘ gives the best Recall, but it is better than our model by only 0.01.

4. ‘DecisionTree‘ and ‘SVC‘ give the best F1-score, which is better than our model by 0.06.

Overall, for an Online Shopper prediction scenario, we should select the models with the best Recall (ensuring the its other metrics are not significantly bad). This is because such a model may be used to capture as many potential purchasers as possible in the industry. In such a setting missing a user who would have converted (lost revenue) is more critical than occasional false positives (FP). It is better to target a few extra users with incentives (e.g., discounts, retargeting ads) than to miss genuine buyers.

Note : We only used the default hyperparameters in all of ‘sklearn‘’s model for the above comparisons (Except ‘max_iter‘ in ‘LogiticRegression‘ to mitigate convergence issues)

2 Competitive part

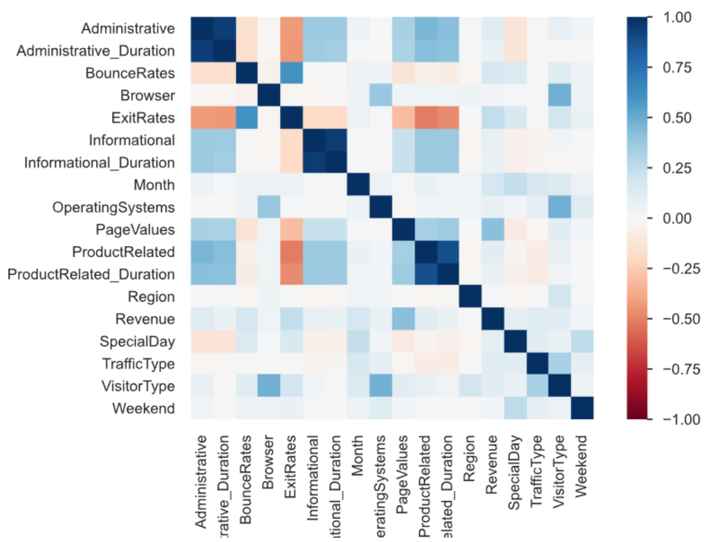
2.1 Feature Engineering and Selection

2.1.1 Duplicates

We removed 60 duplicate data points (0.5% of data set).

2.1.2 Feature selection

We found Administrative is highly correlated with Administrative_Duration, Informational is highly correlated with Informatinal_Duration and ProductRelated is highly correlated with ProductRelated_Duration.



Heatmap Representing the Correlation between different features

The features with more extreme outliers in each pair was removed (Administrative_Duration, Informational, ProductRelated_Duration.)

2.1.3 Feature Engineering

We experimented with the following engineered features:

- Administrative_timeperpage = Administrative_Duration / Administrative
- Informational_timeperpage = Informational_Duration / Informational
- ProductRelated_timeperpage = ProductRelated_Duration / ProductRelated

We did not go forward with using these in the final model.

2.2 Handling Class Imbalance

2.2.1 Oversampling

We experimented with ADASYN with strategy ranging between 0.1 to 0.6.

2.2.2 Class Weight adjustments

We balanced class weights to account for unbalanced classes.

2.3 Model Selection

We experimented with DecisionTrees, SVMs, RandomForests, XGBoost, ADABoost, ExtraTreeClassifiers, Logistic Regression, as well as combination of these using StackedClassifier. In these we experimented with feature engineering and different oversampling strategies using ADASYN.

Stratified K-Fold Cross-Validation was used for cross validation using Bayesian Optimization for Hyperparameter tuning in all models.

2.4 Final Model

Finally we used a Random Forest with the following hyperparameters:

- Criterion: entropy
- Max Depth: 20
- Max Features: sqrt
- Min samples per leaf: 1
- Min samples per split: 2
- No. of estimators: 300

We used ADASYN with strategy = 0.49.