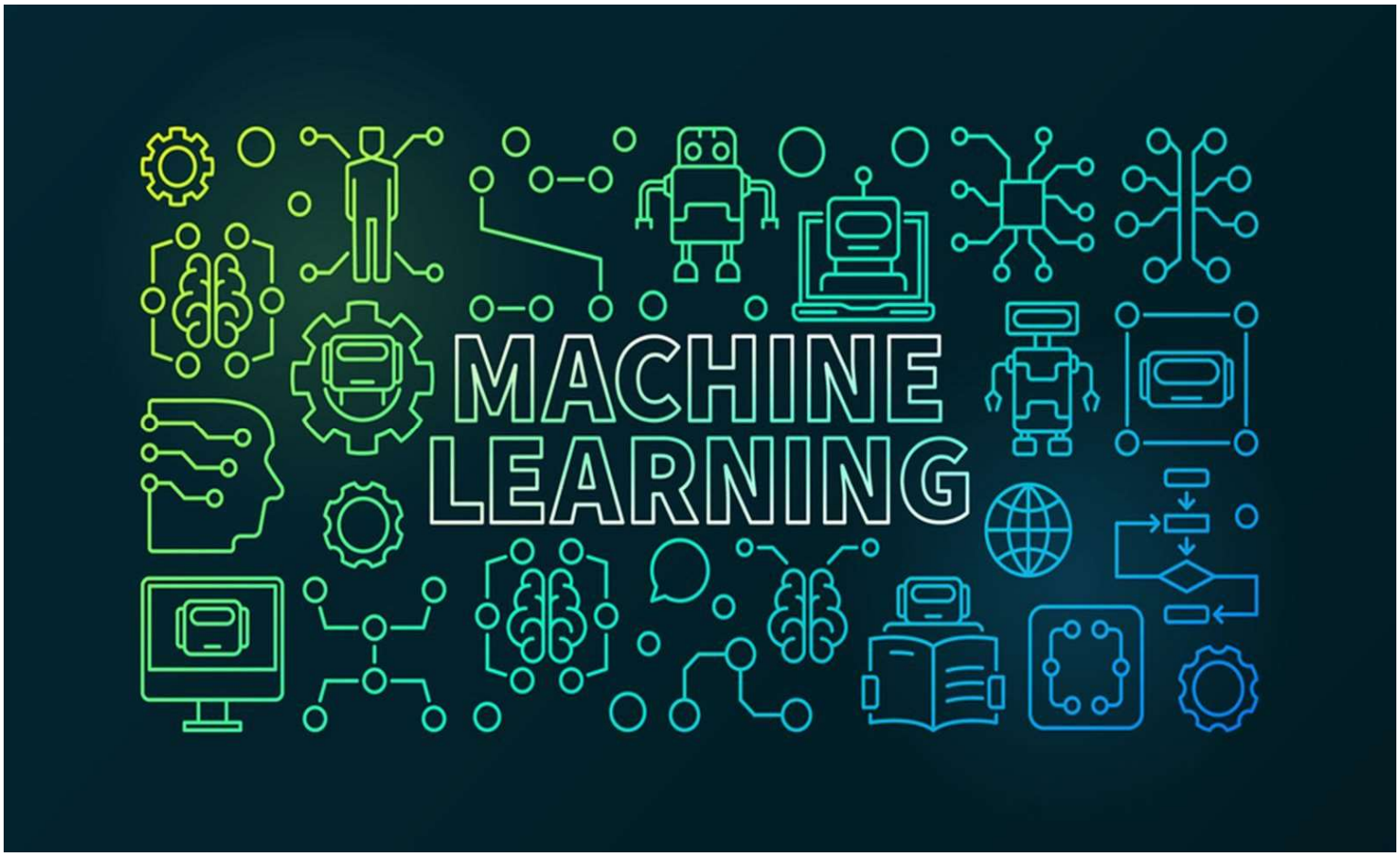


Machine Learning Project

by: Ade Dokhan

Overview:

In this project I was asked to handle a Binary Classification Problem, the goal was to build a set of different models and apply them to Online Shoppers Purchasing data that I got in the train.csv file to classify the labels into two categories - "0/1" where 1 represents people who are going to buy in the internet website and 0 represents the people who won't purchase.

I received train data with 10479 samples and 23 features along with that I received a test.csv file that contains 1851 samples and 22 features that are the same features in the train but without the labels, because the goal of the project was to classify these test samples by a model that I chose in theend. I chose 4 models: KNN, Logistic Regression, Decision Trees, and RandomForest.

Before that I started exploring the data to get statistical information and visualizations to help us more understand the data that is in our hands, after that part, I started preprocessing our train data I deleted features, converted categorical variables into dummy variables, deleted outliers, normalizedthe data and filled empty cells by zeros or mean, and in the end, I reduced the dimensionality by applying PCA on our data.

Every time I wanted to tune hyperparameters I used Grid-Search Model to get the besthyperparameters for the specific model that will give us the best AUC score.

The most important part of the work was in the 4th part-model evaluation, in this part I used Cross-validation using K-fold and I used the models that I built before, every time I did run-on different trains and validation from the data. After every run of the best model, I showed a ROC Curveand I built a Confusion matrix for two models, in the end, I chose the RandomForest model since it gave us the highest AUC Score.

First part-Data Understanding & Exploration

In this part, I separated and analyzed the data in order the understand more the Characteristics of the features and how to deal with them in the next parts. I started by loading the data into a Data Frame and I saw that it contains 23 features and 10479 samples, that each sample got a 0 or 1 label. From the features 15 of them represent number values(int or float) and the rest represents categorical values (objects), I saw that info_page_duration and product_page_duration features are categorical values but represent a number values so I deleted the minute's word at the end of each sample and converted it into number value so I could visualize it with the other numeric features, along with that I checked the number of null values in our data and I saw that the feature D got 10374 Null cells so I decided to delete it in the preprocessing part of the project.

Analyzing the data:

First, I started by plotting a Bar representing the labels and I can see that more than 80% of the samples are labeled to 0 and the rest are labeled to 1 which means that most of the people in our datadidn't buy in the website. I draw a pie chart to check whether people are likely to do their purchaseson weekend or not and I see that people tend to make purchases, not on the weekends. And in the end, I also plotted the purchases made in each month, maybe the month with the maximum purchases of 2857.

Analyzing the data of the numeric features:

First, I presented a table with these values: mean, minimum, maximum, standard error and others using describe() function. For each numeric column, I plotted a histogram to see how these features are distributed, in addition, I plotted a boxplot that gave us more information to detect outliers that are in each column. Looking at the histogram I can see that some features are not normally distributed, in the beginning, I got only 2 features to be normally distributed out of the 15 features soI decided to apply log transformation on our data, and in the end, I got 4 features that are normallydistributed(one of them is approximately normally distributed but I choose to contain it also with therest), in the end, I checked correlations between all the features using heatmap and plotted the mostcorrelated features I got a very high correlation between product_page_duration and total_duration so I decided also to remove it in the preprocessing part

Part 2-Data Preparation and Preprocessing

The preprocessing was implemented on both the train and test and it contained these parts:

-Feature Deletion: as I said in the first part I decided to delete the D column because it contained10K+ null values so it doesn't give us useful information and it's not a good idea to fill all these null values and total_duration column due to high correlation with product_page_duration feature and I chose this feature because also it contains 4k null values.

-adjusting columns to numeric and converting to dummy variables: after looking at all the columns I saw that there are a lot of features that are mixed with numbers and strings so I decided to adjust themto be numeric and to clean the strings or the chars that are in this columns, for example, info_page_duration- removed the minute's word, along with that there were features like **'internet_browser',' A',' C'** in the beginning I didn't adjust these features but after I changed all of the categorical variables to dummy variables I got a huge number of categories/features and a small number of 1's for each new category for example in **'internet_browser'** feature I got a lot of categories for the same browser (safari_15,safari_15.2,safari_15.4…) so I decided to adjust them with the rest of the features and each number in the column represents a category and for 'internet_browser' I reducedthe categories to 4-5 browsers and then I applied log transformation and converted the rest of the columns to dummy variables.

-Outliers Deletion: I removed outliers from the numeric data that is normally distributed using the IQR method and after that, I got 1587 outliers that got deleted in our train data and I plotted a boxplot for each feature before removing and after and I can see the differences.

-Filling Missing Values: First I checked the mean of each column that contains null values and if themean is high I filled the null cells with mean so it doesn't affect the whole model mean in case I filled with zeros and the rest of the columns they had a low mean so I filled with zeros.

-Feature Scaling / Normalization: as I saw in the exploration part the data isn't normalized and each column has a different range of variables so I wanted to uniform the data and minimize the biases between the variables, it was very important for us to normalize the data because I used chose to use PCA to reduce the dimensionality and PCA requires us to normalize the data to use it, I normalized the data using (Standardization scaling) to get a data with mean 0 and standard deviation 1.

-Dimensionality reduction: after all the preprocessing that I have done, I saw that I got a high dimensionality problem for data that I think it's a huge disadvantage because a big amount of data Causes a lot of noise in the data since I added a lot of features without increasing the number of

training samples which could lead to Overfitting, as well so I can't decide whether this data is noise or outliers or it's a very important data, the dimensionality is a problem also for time complexity and the complexity of working with such data. So, after applying PCA I go from 50 features (without the label) down to 23 features.

Part 3 - Modeling

To get the optimal model I divided this part into two steps: In the first step I started with tuning hyperparameters, so I did that using the "GridSearchCV" function which returns the best hyperparameters. the best hyperparameters are the ones that give us the best AUC score.

- KNN model: To get the best K neighbors, I did a Gridsearch over the range 1 -1001 in jumps of 100, then I got that the best k was K=101 so I did a grid search from the range 1-110 in jumps of 10 andthen I got K=81 so, in the end, I did the last grid search on the values[71,79,81,83,85,87,89,91,93,101] and got **K=79** and another parameter.

Parameters	Possibility	The parameters selected at the end
Weights	`["uniform","distance"]`	`['distance']`
Algorithm	`["auto", "ball_tree", "kd_tree", "brute"]`	`['auto']`
Metric	`['euclidean','manhattan']`	`['euclidean']`

Logistic Regression: Logistic regression works based on a given dataset of independent variables and since our labels are binary I chose this model, this model always tries to estimate the effect of every feature on the labels, so it was very important to check the penalty function as a part of the hyperparameters, so as the penalty function implements regularization to prevent model overfitting it increases the penalty as the model complexity increases.

Parameters	Possibility	The parameters selected at the end
Penalty_and_solver	`[("liblinear","l1"),("liblinear","l2"),("newton-cg", "l2"), ("saga", "l2"), ("lbfgs", "l2")]`	`("l2","liblinear")`
C	`powers=range(-10,0)` `c_lst = [10**p for p in powers]`	`'C' : [1e-06],`

Decision tree:

This model can be used as a prediction model, which maps observations of an item and shows conclusions about the target value of the item, providing Mapping between observations and values that are appropriate for them.

Parameters	Possibility	parameters selected at the end
Criterion	["gini", "entropy"]	["entropy"]
Max_depth	[3,10]	[3]
Min_samples_split	[2,4,6,8,12]	[8]
Max_features	[1,2,7,10,14,19,20,21,23]	[21]
splitter	['best','random']	['best']

Random Forest:

This model contains many Decision trees where every tree trains individually from different samples and features than the other ones and in every new tree the model tries to improve more and more

Parameters	Possibility	The parameters selected at the end
Criterion	['gini', 'entropy']	["entropy"]
Max_depth	[2,4,8]	[8]
Min_samples_leaf	[1, 2, 4]	[1]
Max_samples_split	[2, 4, 5]	[2]
Max_features	[1,7,15]	[15]
n_estimators	[20,40,100]	[100]

Part 4- Model Evaluation

I evaluated the models with the K-fold validation method- after running the models on the train set I evaluated their performance, I took each model with its best hyperparameters using K-fold I split the data into 75% train data and 25% validation, in this method I are splitting the data K times inevery new train and validation set. I chose random samples to minimize serial dependence. I choseK-fold hyperparameter =5 considering our data size.

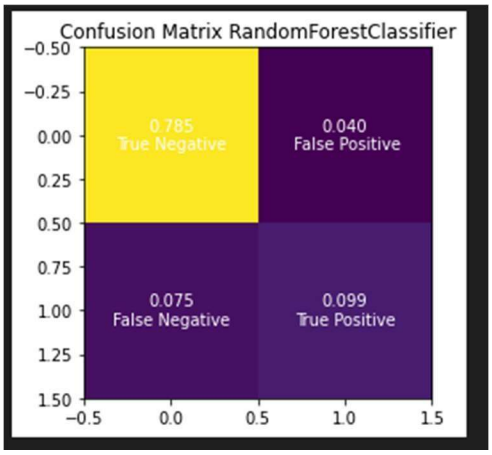
Summary Table:

The Model	Accuracy Score	AUC Score
KNN	**0.912**	**0.911**
LogisticRegression	**0.853**	**0.874**
Decision tree	**0.864**	**0.870**
Random forest Classifier	**0.918**	**0.921**

I can clearly see that the Random Forest CLassifier got the best AUC

Score.Confusion Matrix:

I chose to build a Confusion Matrix for the best model I got-RandomForest



1. in 78.5% of the cases the real label is 0 and I classify 0 (TN)
2. in 9% of the cases the real label is 1 and I classify 1 (TP)
3. in 4% of the cases the real label is 0 and I classify 1 (FP)
4. in 7.5% of the cases the real label is 1 and I classify 0 (FN)

Checking overfitting in our models:

I defined that there is overfitting if the difference between AUC Train and AUC Test is bigger than 10%And I didn't get overfitting in all 4 models.

Summary:

The goal of the project was to research the data I got and its all features to classify the samples into one out of two categories, I did a very deep analysis of the features to understand more the way theyare behaving and all of that with the help of visualizations, after seeking all that information I started preprocessing the data and adjusting features, deleting outliers and dimensionality reducing. After I got a new adjusted fresh data set I chose 4 models and started tuning hyperparameters for each model after getting the best hyperparameters I jumped on the most important part of the project, model evaluation-here I decided to evaluate the 4 models using k-fold cross validation after I evaluated and checked for overfitting, in the end, happily I had a winner classifier and the winner is **Random forest classifier!** and finally, I did a prediction on the test set Believing that I got good results and for last words, I enjoyed and learned a lot in this project and I'm diffidently gonna talk about this project in my future interviews!

נספחים:

	id	num_of_admin_pages	admin_page_duration	num_of_info_pages	info_page_duration	num_of_product_pages	product_page_duration	total_duration	I
0	0	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	NaN	
1	1	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	0.000000	
2	2	0.0	0.0	0.0	NaN	NaN	627.5 minutes	627.500000	
3	3	0.0	0.0	0.0	0.0 minutes	19.0	154.2166667 minutes	154.216667	
4	4	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	NaN	
5	5	0.0	0.0	0.0	0.0 minutes	2.0	NaN	NaN	
6	6	0.0	0.0	0.0	0.0 minutes	3.0	738.0 minutes	NaN	
7	7	0.0	0.0	0.0	0.0 minutes	3.0	395.0 minutes	395.000000	
8	8	NaN	0.0	0.0	0.0 minutes	16.0	407.75 minutes	NaN	
9	9	0.0	0.0	0.0	0.0 minutes	7.0	280.5 minutes	280.500000	

10 rows × 23 columns

uration	total_duration	BounceRates	ExitRates	...	device	internet_browser	Region	user_type	Weekend	A	B	C	D	purchase
ninutes	NaN	0.200000	0.200000	...	1.0	safari_15	1.0	Returning_Visitor	False	c_1	118.880094	log202	NaN	0
ninutes	0.000000	0.200000	0.200000	...	4.0	safari_14	9.0	Returning_Visitor	False	c_3	113.358423	log404	NaN	0
ninutes	627.500000	0.020000	0.050000	...	3.0	browser_3_v17	1.0	Returning_Visitor	True	c_4	121.507695	log202	NaN	0
ninutes	154.216667	0.015789	0.024561	...	2.0	chrome_99.1.3	1.0	Returning_Visitor	False	c_3	93.747176	log_100	NaN	0
ninutes	NaN	0.200000	0.200000	...	2.0	edge_96.0.1054.75	3.0	Returning_Visitor	False	c_3	99.545824	log202	NaN	0
NaN	NaN	0.000000	0.100000	...	2.0	NaN	2.0	Returning_Visitor	False	c_3	104.712405	log200	NaN	0
ninutes	NaN	0.000000	0.022222	...	2.0	edge_96.0.1054.72	1.0	Returning_Visitor	False	c_2	89.786568	log404	NaN	0
ninutes	395.000000	0.000000	0.066667	...	1.0	safari_15	3.0	Returning_Visitor	False	c_3	101.184534	log_100	NaN	0
ninutes	NaN	0.018750	0.025833	...	1.0	safari_15.4	4.0	Returning_Visitor	False	NaN	83.931739	log_100	NaN	0
ninutes	280.500000	0.000000	0.028571	...	1.0	safari_15.2	1.0	Returning_Visitor	False	c_3	97.899633	log200	NaN	0

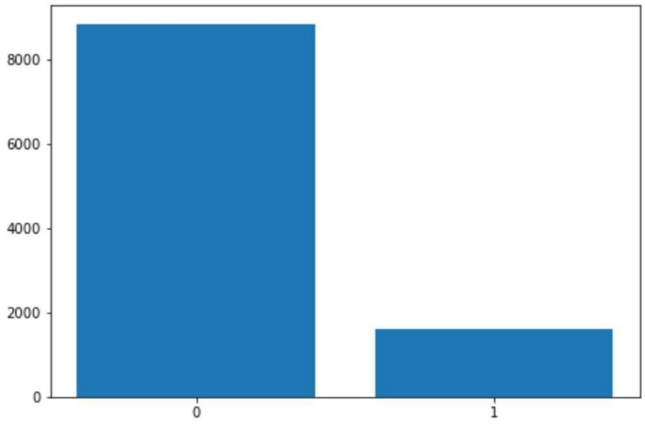
```
                id  num_of_admin_pages  admin_page_duration  \
count  10479.00000         9874.000000         10066.000000
mean    5239.00000            2.321957            80.462468
std     3025.17107            3.335331           179.217548
min        0.00000            0.000000             0.000000
25%     2619.50000            0.000000             0.000000
50%     5239.00000            1.000000             8.000000
75%     7858.50000            4.000000            93.000000
max    10478.00000           27.000000          3398.750000

       num_of_info_pages  num_of_product_pages  total_duration   BounceRates  \
count        9792.000000          10076.000000     5726.000000  10457.000000
mean            0.502655             31.870187     1302.457216      0.021854
std             1.265812             44.816259     2039.312905      0.048100
min             0.000000              0.000000        0.000000      0.000000
25%             0.000000              7.000000      221.000000      0.000000
50%             0.000000             18.000000      677.851191      0.002968
75%             0.000000             38.000000     1605.733333      0.016667
max            24.000000            705.000000    47850.920680      0.200000
```

```
 the number of the null cells in the data for each feature
id                          0
num_of_admin_pages        605
admin_page_duration       413
num_of_info_pages         687
info_page_duration        317
num_of_product_pages      403
product_page_duration     621
total_duration           4753
BounceRates                22
ExitRates                  26
PageValues                 27
closeness_to_holiday      496
Month                      25
device                    323
internet_browser          563
Region                     19
user_type                  23
Weekend                    23
A                         706
B                          23
C                          23
D                       10374
purchase                    0
dtype: int64
```

```
The Data types
id                       int64
num_of_admin_pages     float64
admin_page_duration    float64
num_of_info_pages      float64
info_page_duration      object
num_of_product_pages   float64
product_page_duration   object
total_duration         float64
BounceRates            float64
ExitRates              float64
PageValues             float64
closeness_to_holiday   float64
Month                   object
device                 float64
internet_browser        object
Region                 float64
user_type               object
Weekend                 object
A                       object
B                      float64
C                       object
D                      float64
purchase                 int64
dtype: object
```

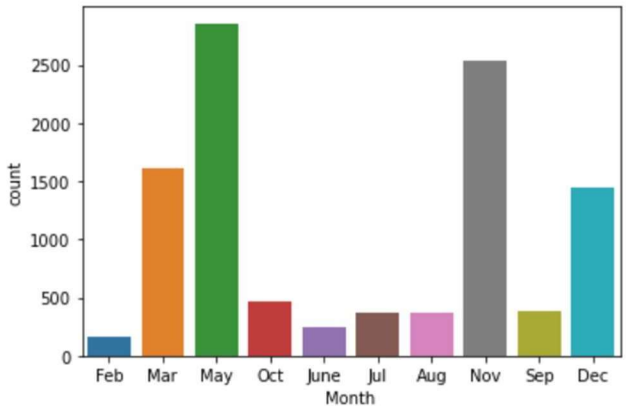
```
          ExitRates    PageValues  closeness_to_holiday        device  \
count  10453.000000  10452.000000           9983.000000  10156.000000
mean       0.042859      5.900387              0.061985      2.122981
std        0.048297     18.727496              0.199429      0.906859
min        0.000000      0.000000              0.000000      1.000000
25%        0.014283      0.000000              0.000000      2.000000
50%        0.025161      0.000000              0.000000      2.000000
75%        0.050000      0.000000              0.000000      3.000000
max        0.200000    361.763742              1.000000      8.000000

            Region             B           D      purchase
count  10460.000000  10456.000000  105.000000  10479.000000
mean       3.155641     99.769628    4.060440      0.154690
std        2.404672     10.050268    2.189818      0.361626
min        1.000000     60.733919   -1.235571      0.000000
25%        1.000000     93.030568    3.141657      0.000000
50%        3.000000     99.787249    4.432221      0.000000
75%        4.000000    106.506626    5.531061      0.000000
max        9.000000    141.735142    7.618254      1.000000
```

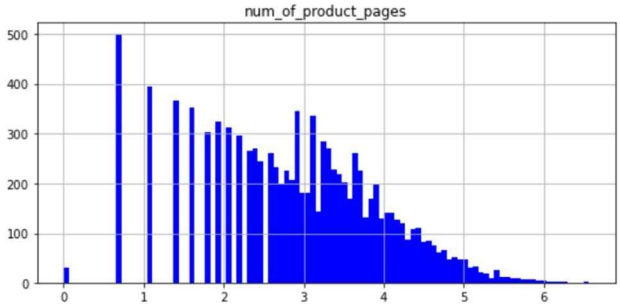
```
The Number of the purchases made in the session

<BarContainer object of 2 artists>
```

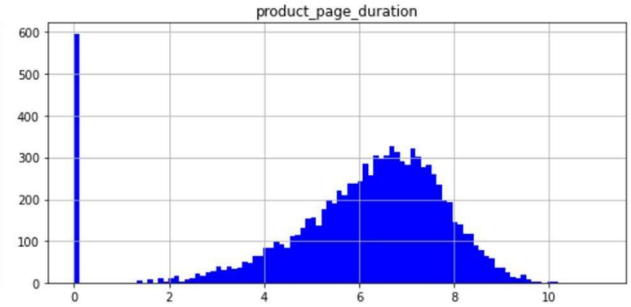


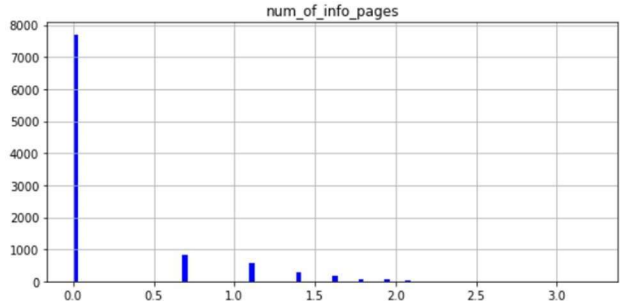

```
([<matplotlib.patches.Wedge at 0x23293a102b0>,
  <matplotlib.patches.Wedge at 0x23293a10bb0>],
 [Text(-0.8977161187881741, 0.7963075850875069, 'Not in weekend'),
  Text(0.8977160815103247, -0.7963076271126306, 'In weekend')],
 [Text(-0.5236677359597682, 0.46451275796771235, '76.90'),
  Text(0.523667714214356, -0.46451278248236777, '23.10')])
```

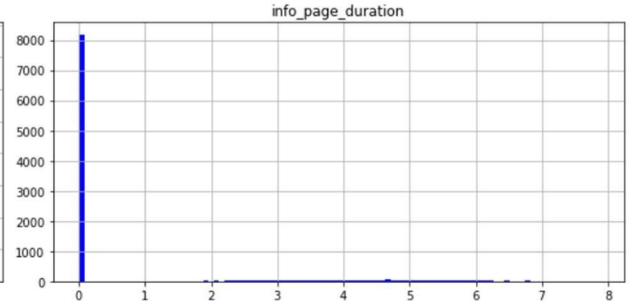

```
Number of people that made a purchase in each month:
May      2857
Nov      2539
Mar      1615
Dec      1450
Oct       471
Sep       379
Jul       367
Aug       365
June      248
Feb       163
Name: Month, dtype: int64
```

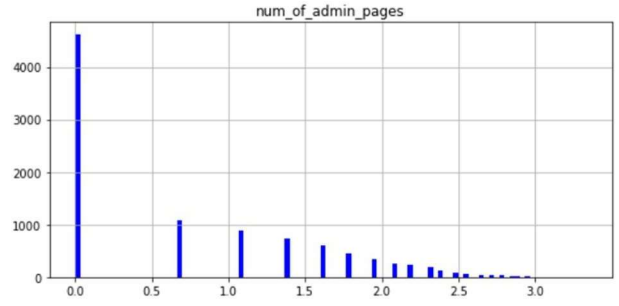


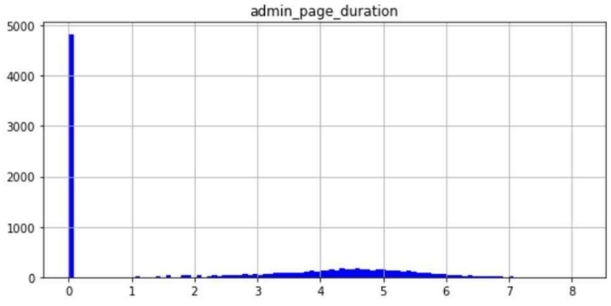

num_of_product_pages

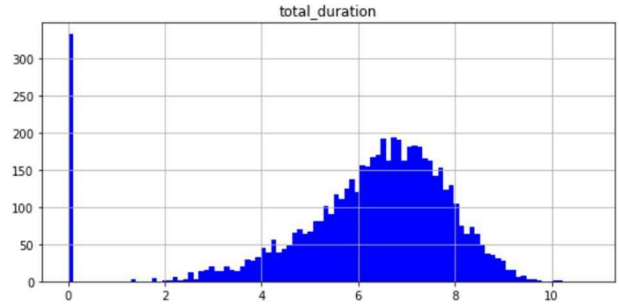

product_page_duration

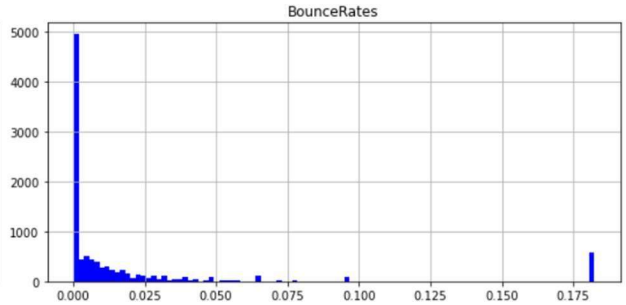

num_of_info_pages

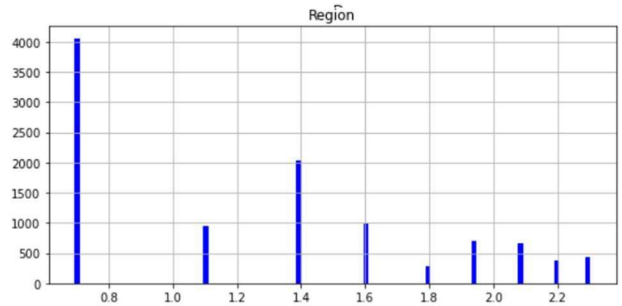

info_page_duration

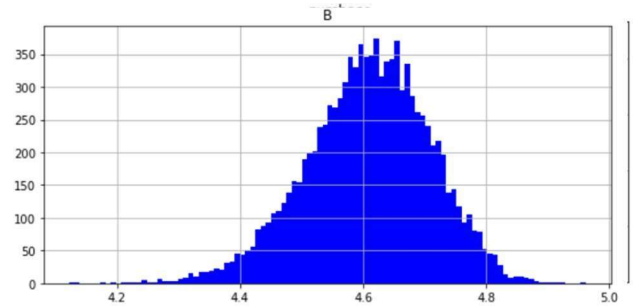

num_of_admin_pages

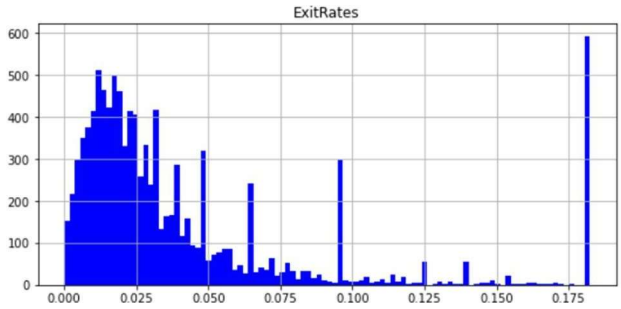

admin_page_duration

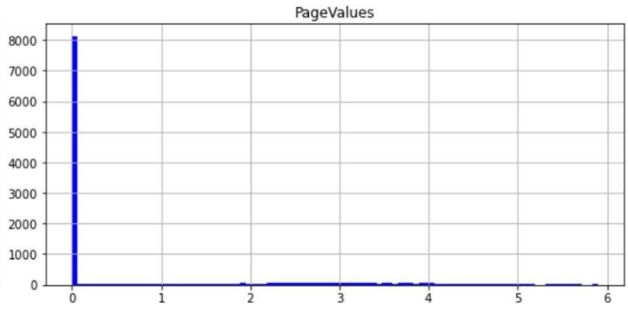

total_duration

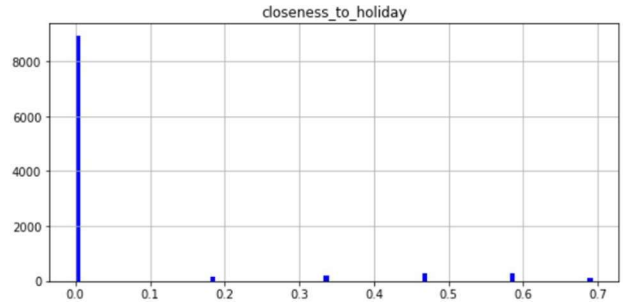

BounceRates

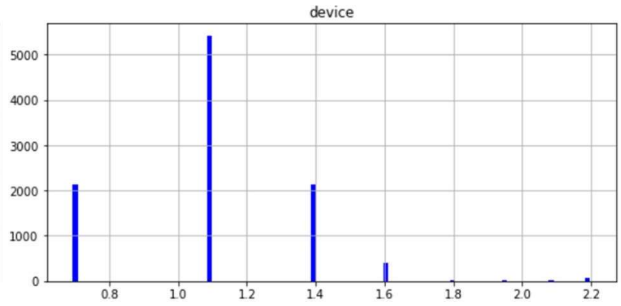

Region


B

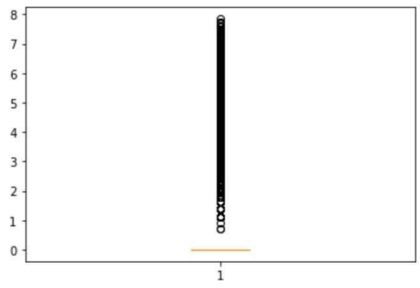




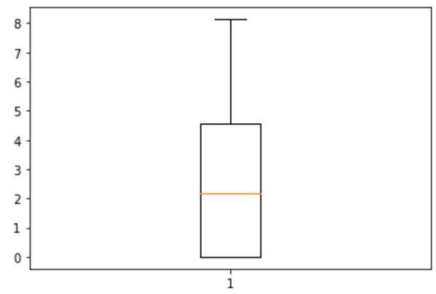




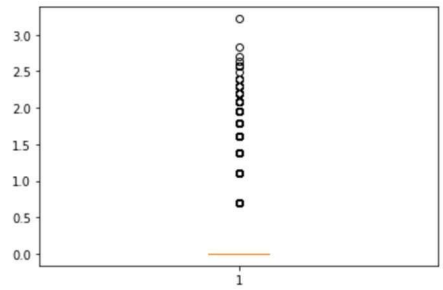
info_page_duration



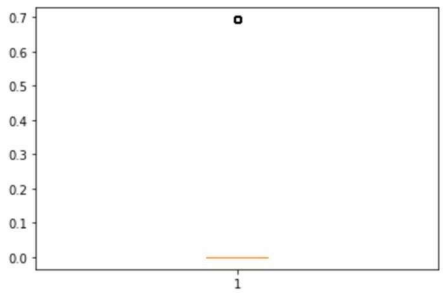
admin_page_duration



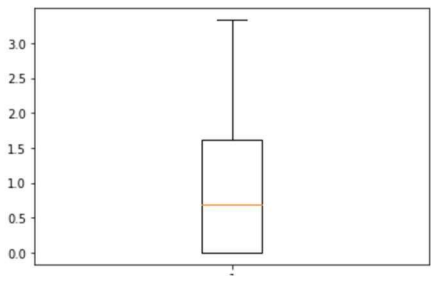
num_of_info_pages



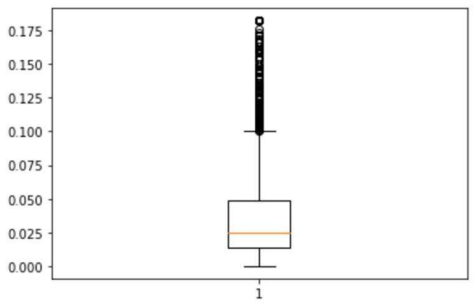
purchase



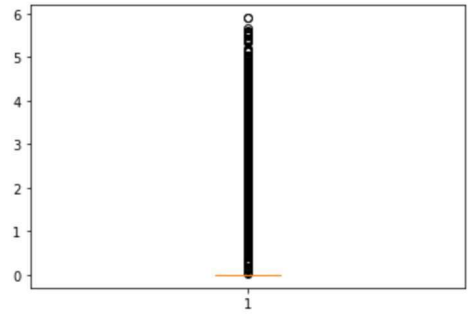
num_of_admin_pages



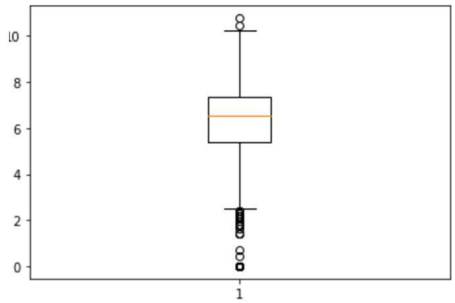
ExitRates



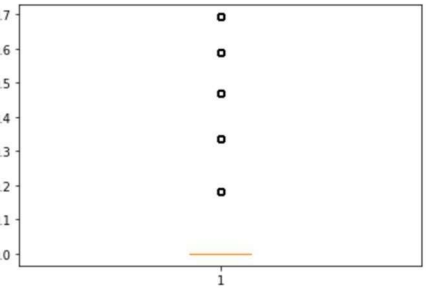
PageValues



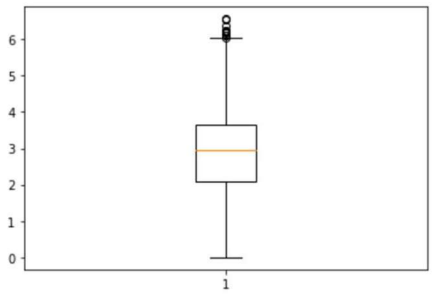
otal_duration



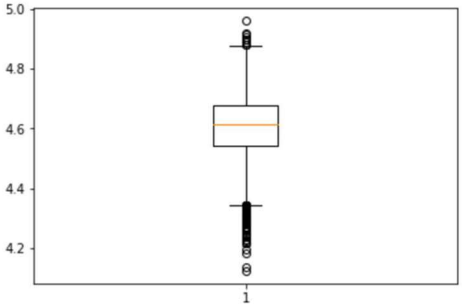
closeness_to_holiday



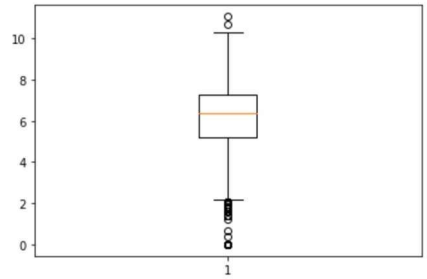
num_of_product_pages



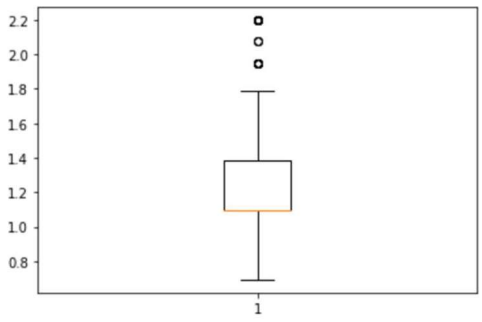
B



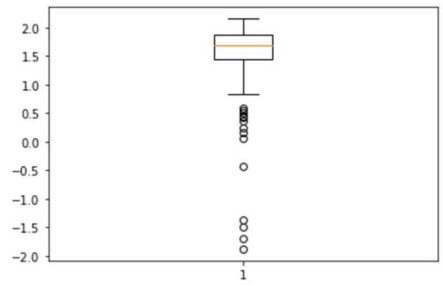
product_page_duration



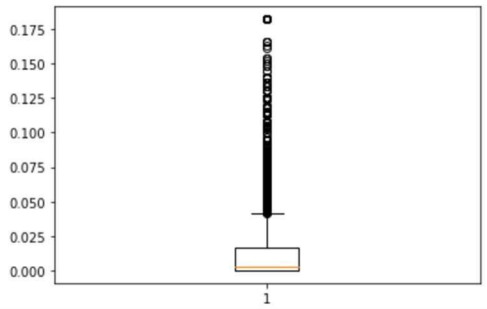
device



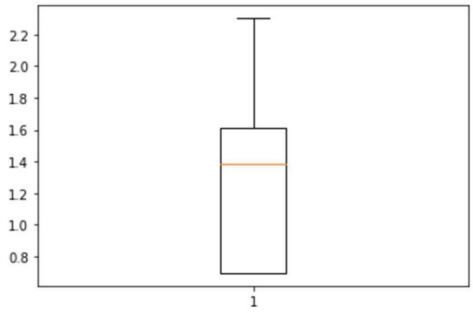
D



BounceRates

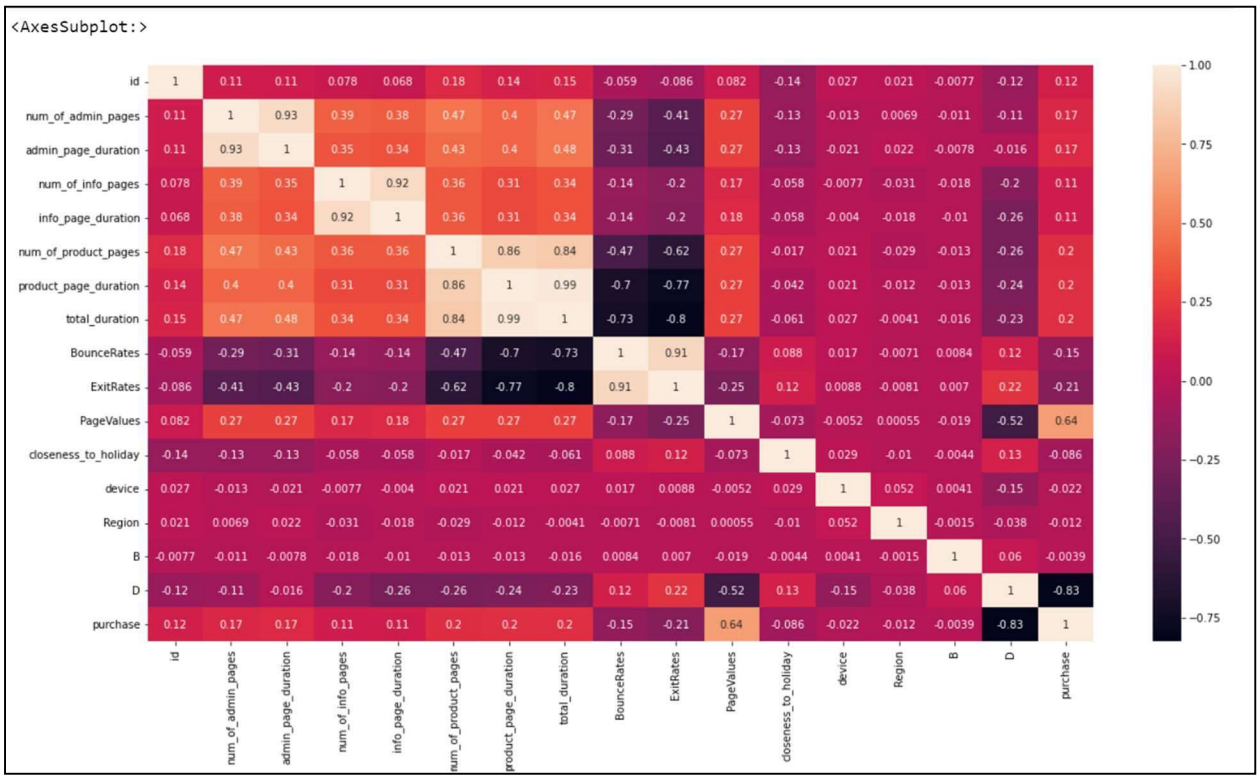


Region



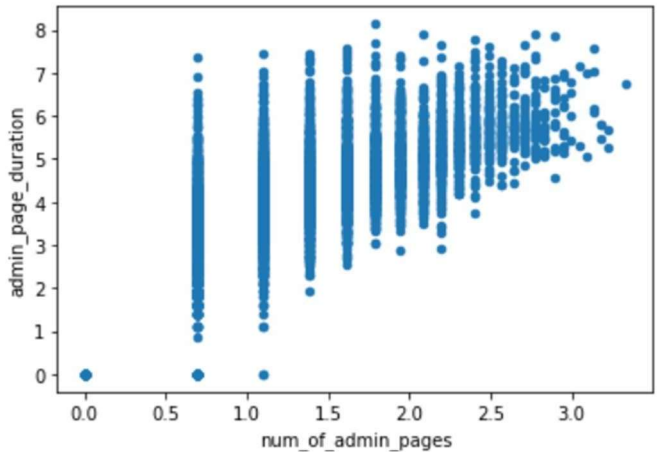
	id	num_of_admin_pages	admin_page_duration	num_of_info_pages	info_page_duration	num_of_product_pages	product_page_duration	BounceRates
0	0	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	0.200000
1	1	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	0.200000
2	2	0.0	0.0	0.0	NaN	NaN	627.5 minutes	0.020000
3	3	0.0	0.0	0.0	0.0 minutes	19.0	154.2166667 minutes	0.015789
4	4	0.0	0.0	0.0	0.0 minutes	1.0	0.0 minutes	0.200000
5	5	0.0	0.0	0.0	0.0 minutes	2.0	NaN	0.000000
6	6	0.0	0.0	0.0	0.0 minutes	3.0	738.0 minutes	0.000000
7	7	0.0	0.0	0.0	0.0 minutes	3.0	395.0 minutes	0.000000
8	8	NaN	0.0	0.0	0.0 minutes	16.0	407.75 minutes	0.018750
9	9	0.0	0.0	0.0	0.0 minutes	7.0	280.5 minutes	0.000000

10 rows × 21 columns

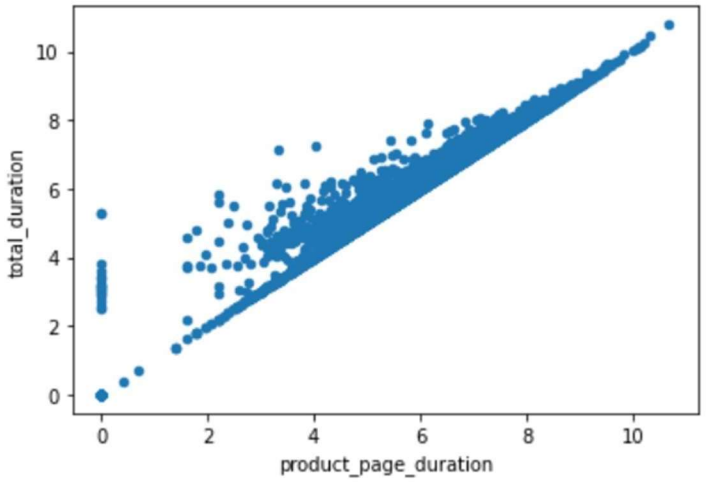


uration	BounceRates	ExitRates	PageValues	...	Month	device	internet_browser	Region	user_type	Weekend	A	B	C	purchase
ninutes	0.200000	0.200000	0.0	...	Feb	1.0	safari_15	1.0	Returning_Visitor	False	c_1	118.880094	log202	0
ninutes	0.200000	0.200000	0.0	...	Feb	4.0	safari_14	9.0	Returning_Visitor	False	c_3	113.358423	log404	0
ninutes	0.020000	0.050000	0.0	...	Feb	3.0	browser_3_v17	1.0	Returning_Visitor	True	c_4	121.507695	log202	0
ninutes	0.015789	0.024561	0.0	...	Feb	2.0	chrome_99.1.3	1.0	Returning_Visitor	False	c_3	93.747176	log_100	0
ninutes	0.200000	0.200000	0.0	...	Feb	2.0	edge_96.0.1054.75	3.0	Returning_Visitor	False	c_3	99.545824	log202	0
NaN	0.000000	0.100000	0.0	...	Feb	2.0	NaN	2.0	Returning_Visitor	False	c_3	104.712405	log200	0
ninutes	0.000000	0.022222	0.0	...	Feb	2.0	edge_96.0.1054.72	1.0	Returning_Visitor	False	c_2	89.786568	log404	0
ninutes	0.000000	0.066667	0.0	...	Feb	1.0	safari_15	3.0	Returning_Visitor	False	c_3	101.184534	log_100	0
ninutes	0.018750	0.025833	0.0	...	Feb	1.0	safari_15.4	4.0	Returning_Visitor	False	NaN	83.931739	log_100	0
ninutes	0.000000	0.028571	0.0	...	Feb	1.0	safari_15.2	1.0	Returning_Visitor	False	c_3	97.899633	log200	0

num_of_admin_pages admin_page_duration
corr 0.9256560102715061
product_page_duration total_duration
corr 0.9858872025492549
('num_of_admin_pages', 'admin_page_duration')



('product_page_duration', 'total_duration')



	id	num_of_admin_pages	admin_page_duration	num_of_info_pages	\
0	0	0.0	0.0	0.0	
1	1	0.0	0.0	0.0	
2	2	0.0	0.0	0.0	
3	3	0.0	0.0	0.0	
4	4	0.0	0.0	0.0	
...	...	...	...	...	
10474	10474	3.0	145.0	0.0	
10475	10475	0.0	0.0	0.0	
10476	10476	0.0	0.0	0.0	
10477	10477	4.0	75.0	0.0	
10478	10478	0.0	0.0	0.0	

	info_page_duration	num_of_product_pages	product_page_duration	\
0	0.0	1.0	0.000000	
1	0.0	1.0	0.000000	
2	NaN	NaN	627.500000	
3	0.0	19.0	154.216667	
4	0.0	1.0	0.000000	
...	...	...	...	
10474	0.0	53.0	1783.791667	
10475	0.0	5.0	465.750000	
10476	0.0	6.0	184.250000	
10477	0.0	15.0	346.000000	
10478	0.0	3.0	21.250000	

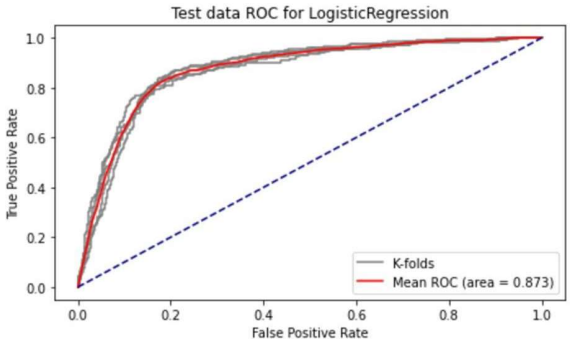
	BounceRates	ExitRates	PageValues	...	Month	device	\
0	0.200000	0.200000	0.000000	...	Feb	1.0	
1	0.200000	0.200000	0.000000	...	Feb	4.0	
2	0.020000	0.050000	0.000000	...	Feb	3.0	
3	0.015789	0.024561	0.000000	...	Feb	2.0	
4	0.200000	0.200000	0.000000	...	Feb	2.0	
...	...	...	...	...	...	...	
10474	0.007143	0.029031	12.241717	...	Dec	4.0	
10475	0.000000	0.021333	0.000000	...	Nov	3.0	
10476	0.083333	0.086667	0.000000	...	Nov	3.0	
10477	0.000000	0.021053	0.000000	...	Nov	2.0	
10478	0.000000	0.066667	0.000000	...	Nov	3.0	

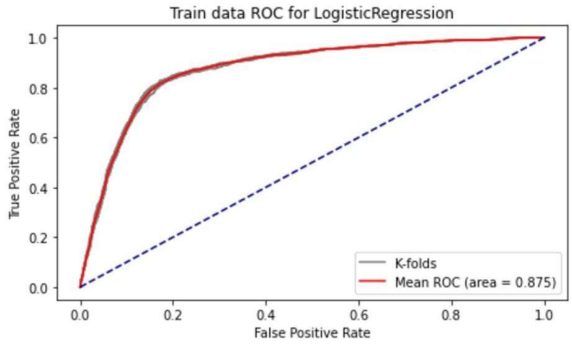
	internet_browser	Region	user_type	Weekend	A	B	\
0	safari	1.0	Returning_Visitor	False	1.0	118.880094	
1	safari	9.0	Returning_Visitor	False	3.0	113.358423	
2	browser	1.0	Returning_Visitor	True	4.0	121.507695	
3	chrome	1.0	Returning_Visitor	False	3.0	93.747176	
4	edge	3.0	Returning_Visitor	False	3.0	99.545824	
...	...	...	...	...	...	...	
10474	browser	1.0	Returning_Visitor	True	1.0	79.548460	
10475	chrome	1.0	Returning_Visitor	True	8.0	99.457255	
10476	chrome	1.0	Returning_Visitor	True	13.0	94.417344	
10477	chrome	3.0	Returning_Visitor	False	11.0	108.919057	
10478	chrome	1.0	New_Visitor	True	2.0	90.072077	

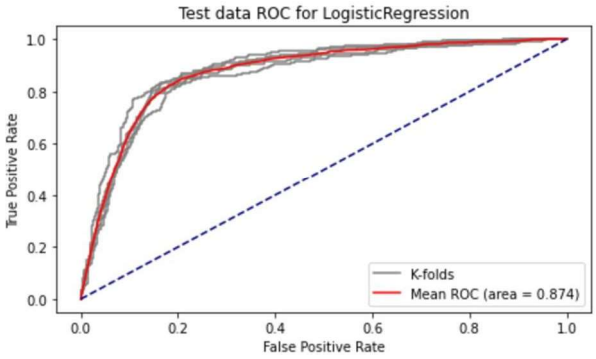
	C	purchase
0	202.0	0
1	404.0	0
2	202.0	0
3	100.0	0
4	202.0	0
...	...	...
10474	400.0	0
10475	202.0	0
10476	8080.0	0
10477	400.0	0
10478	8080.0	0

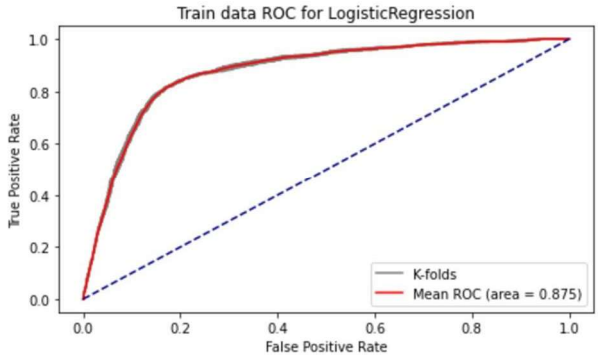
[10479 rows x 21 columns]

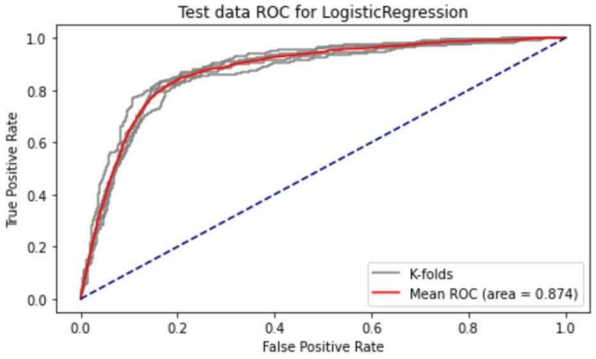
```
(10479, 51)
(1851, 50)
(8892, 51)
Index(['num_of_admin_pages', 'admin_page_duration', 'num_of_info_pages',
       'info_page_duration', 'num_of_product_pages', 'product_page_duration',
       'BounceRates', 'ExitRates', 'PageValues', 'closeness_to_holiday', 'A',
       'B', 'C', 'purchase'],
      dtype='object')
```

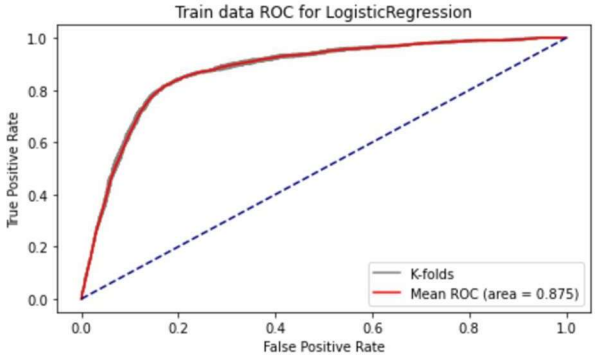


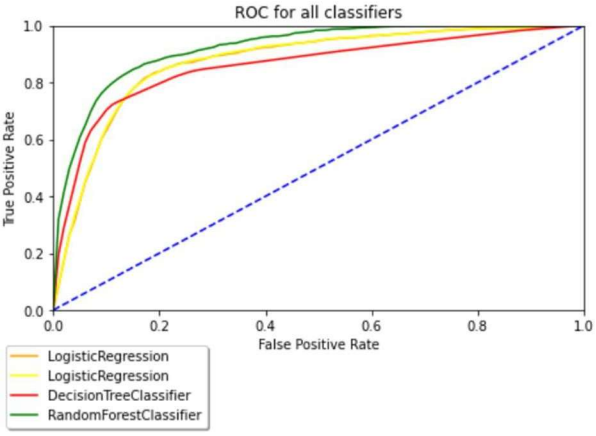


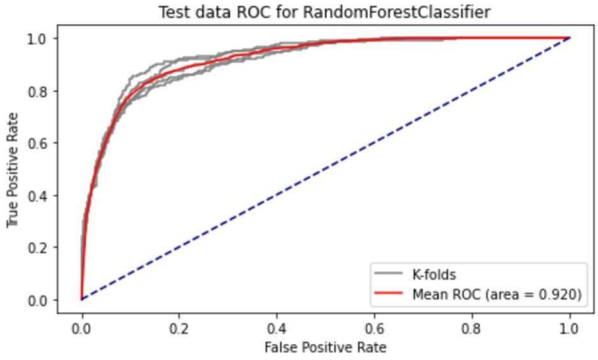

Test data ROC for LogisticRegression
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.874)

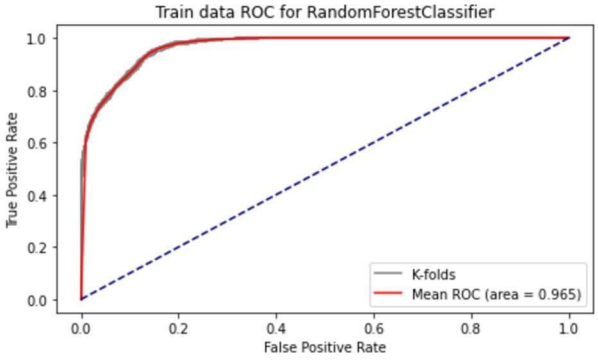

Train data ROC for LogisticRegression
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.875)

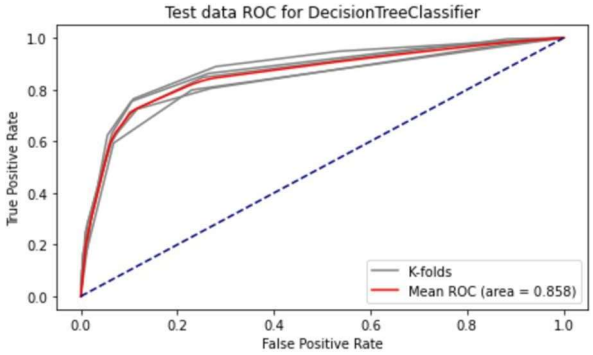

Test data ROC for LogisticRegression
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.874)

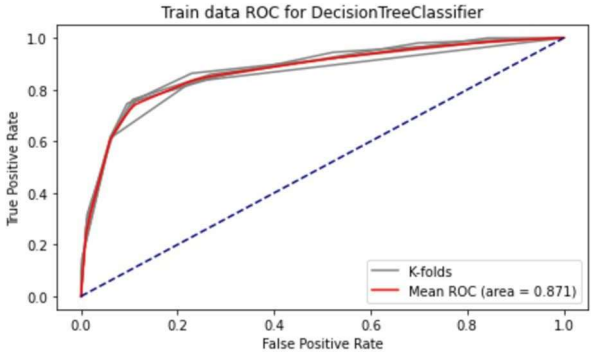

Train data ROC for LogisticRegression
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.875)

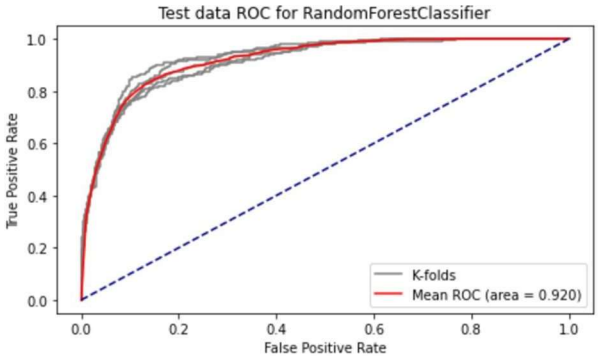

ROC for all classifiers
True Positive Rate
False Positive Rate
LogisticRegression
LogisticRegression
DecisionTreeClassifier
RandomForestClassifier

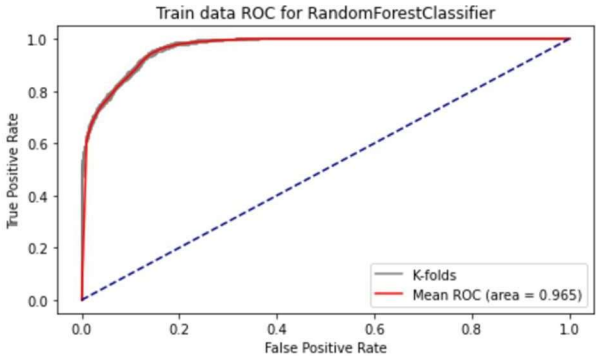

Test data ROC for RandomForestClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.920)

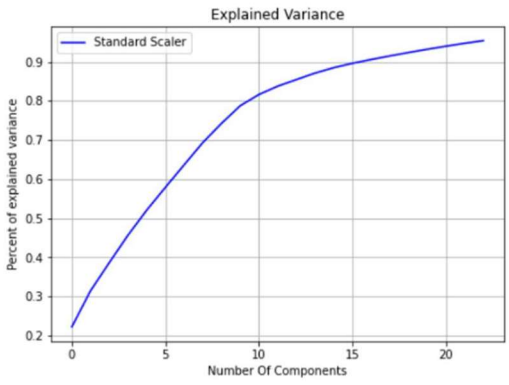

Train data ROC for RandomForestClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.965)

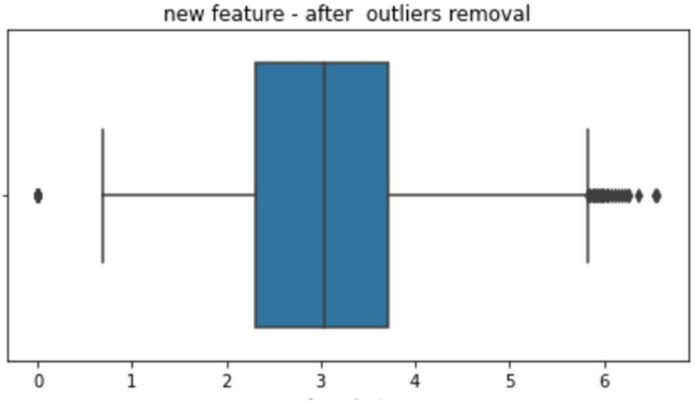

Test data ROC for DecisionTreeClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.858)

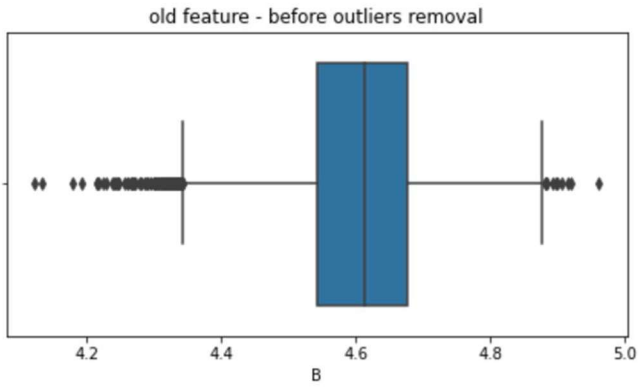

Train data ROC for DecisionTreeClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.871)

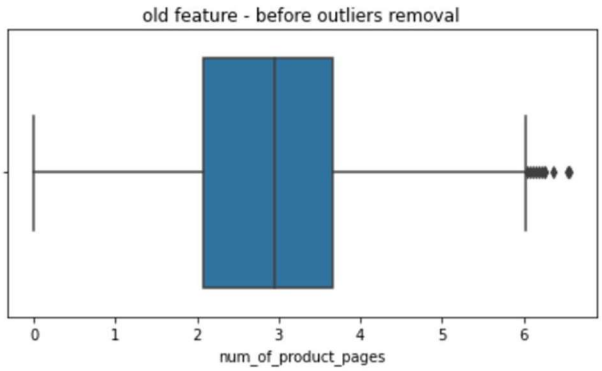

Test data ROC for RandomForestClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.920)

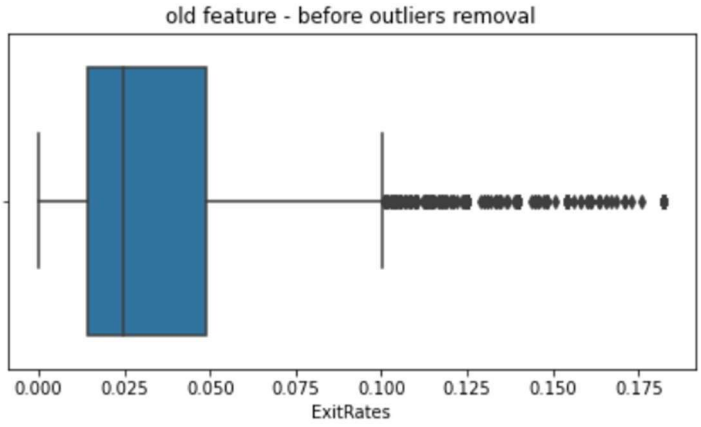

Train data ROC for RandomForestClassifier
True Positive Rate
False Positive Rate
K-folds
Mean ROC (area = 0.965)

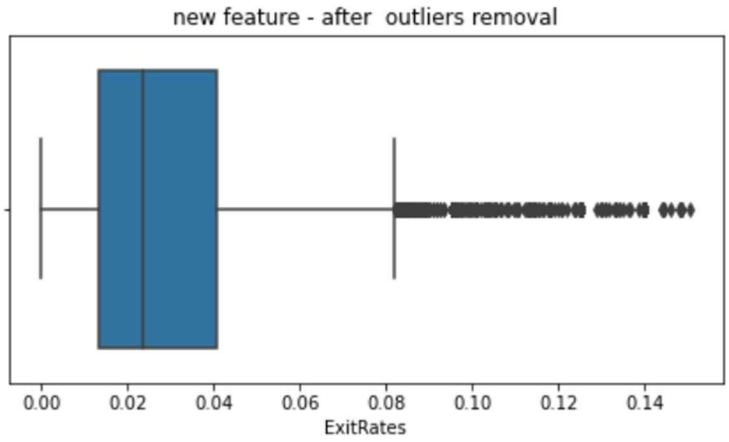

Explained Variance
Standard Scaler
Percent of explained variance
Number Of Components

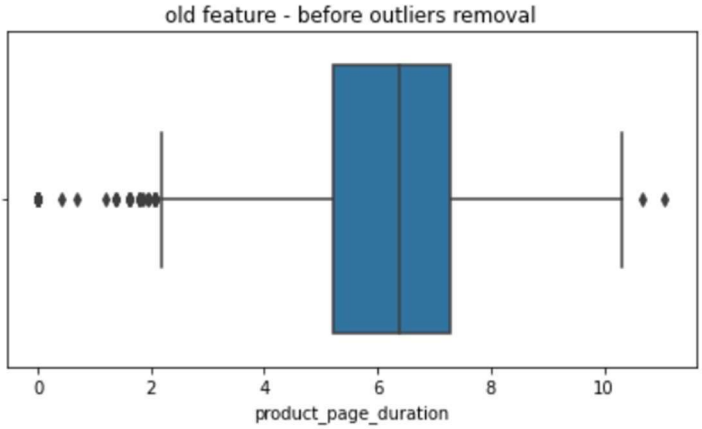

new feature - after outliers removal

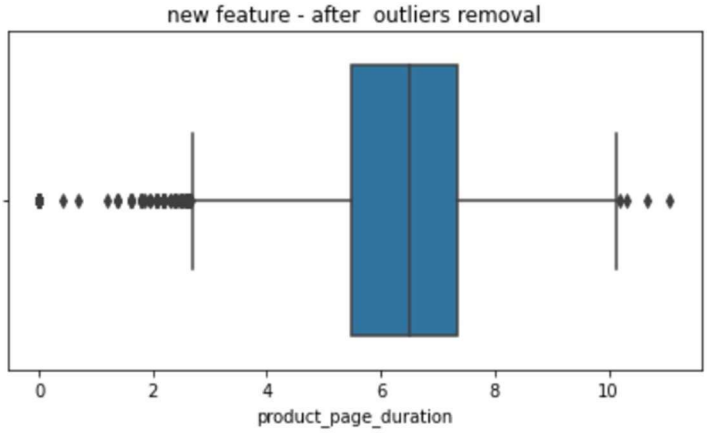

old feature - before outliers removal
B

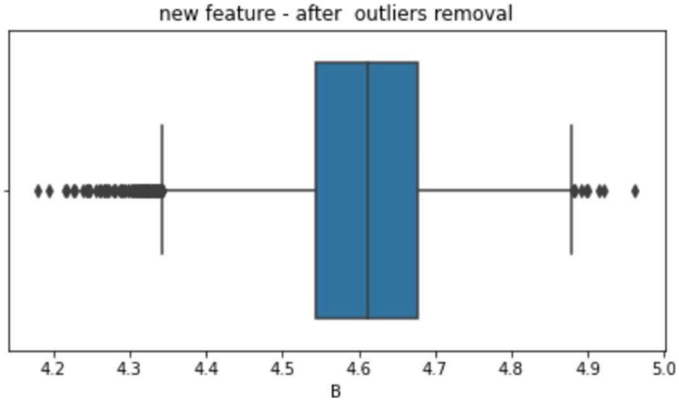

old feature - before outliers removal
num_of_product_pages

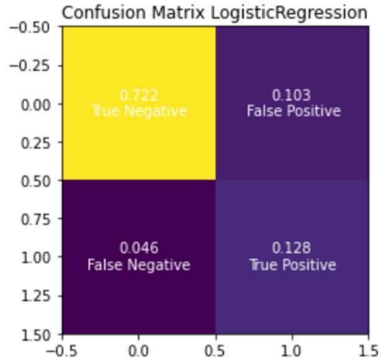




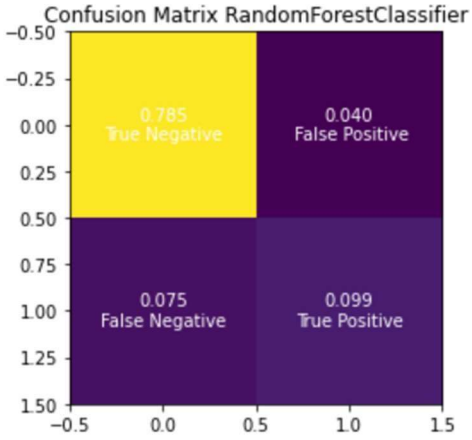




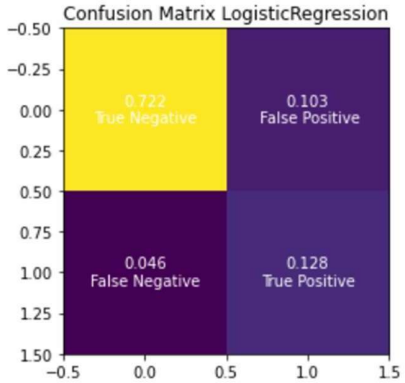





```
The weighted accuracy for LogisticRegression is 0.8505617977528089

0.8505617977528089
```



The weighted accuracy for RandomForestClassifier is 0.8842696629213483



The weighted accuracy for LogisticRegression is 0.8505617977528089

0.8505617977528089