TEL AVIV אוניברסיטת
UNIVERSITY תל אביב

Introduction to Machine Learning

Final Project

Professor: Mr. Dor Bank
Professor Assistance: Mr. Ilan Vasilevsky

Submitted by:

Kirill Bulgakov
Shachar Schneider

Introduction:

In the following report, we will introduce our final project in the course 'Introduction to Machine Learning' that we took part in the past semester. This report will present the process we have been through to create a model to predict rather a certain session in an
E-Commerce website will end up as a purchase or not.

We hope you enjoy the project!

Part 1- Exploring the data:

At first, we have loaded the data and had a quick look at it to get a basic perception on what we are dealing with. We noticed there are 10,479 observations, and 23 columns ('features') that includes a 'Purchase' column values '0' (not purchased) or '1' (purchased) that we will call 'label' from now on. Our first glimpse at the data gave us a lot more information. We noticed several features that have numeric values but represented by strings (such as '627.5 minutes'), a Boolean feature ('Weekend'), and a feature that seems to be mostly NaN values ('D') that led us to check and find out that this feature has 99% NaN values and to discard it. Our next step towards better understanding the data, was to identify what are our categorial and numeric features.

Preliminary analysis of the numerical features: (Figure 1)
We had several insights from the histogram:
1. Feature 'B' distributed approximately normally- this may help us later to identify outliers.
2. Features like 'page values', 'admin page duration', 'num of admin pages' and more, have a high concentration of values in a certain range, and then some extreme outliers that may confuse our model. We will use these insights later.

Advanced analyzing the categorial features:
Earlier in the process we identified the categorial features as such: 'C', 'A', 'internet browser', 'user type', 'Region', 'device' and 'Month'. Before handling them, we wished to examine them more thoroughly to have a better understanding on the way they appear in the data.
'C': (Figure 2)
We noticed that the feature distributes approximately uniformly. This led us to a conclusion that this feature isn't contributing much, and we decided to drop it later from our dataset.
'A': (Figure 3)
Before checking this feature, we removed the 'c_' from all values and grouped together all values that start with '20_'. After this minor processing and a quick look at the plot we couldn’t determine rather this feature has an individual impact on the labels or not, so we decided to dig in! (Figure 3.1)
We had two insights from our investigation on the feature:
First, some features appear a very small amount of times (14, 16, 17, 18, 19). We will delete these rows later. **Second,** We will divide this feature into 3 categories with respect to purchase percentages: 0-10% as 'weak', 10-20% as 'average', 20% and above as 'strong'.
'Internet browser': (Figure 4)
Before plotting this feature, we saw that there are 4 primary categories, that each of them has many sub-category (e.g., 'browser_10_v10', 'browser_10_v11' etc.), we grouped them all according to the primary category. Similar to feature 'A', we had a hard time to determine the impact of those categories on the label and calculated each browser purchase percentage (Figure 4.1).
We found out that each category has roughly the same purchase percentage rate and determined that 'Internet browser' has no significant impact on the label. We will drop this feature later.

<u>'User type':</u> (Figure 5)
Based on the plot, it looks like 'user type' has an impact on the label, but the meaning of 'Other' in user type is not so clear to us. we decided to transform these values to 'Returning Visitor', being the vast majority from the dataset.
<u>'Region':</u> (Figure 6)
We didn't get any special insight from the 'Region' plot. We will turn them all into dummy variables.
<u>'device':</u> (Figure 7)
We see that this feature has 8 categories in total. Closer look at the amount of appearance for each feature shows that devices 5-8 has very small presence, and that if we group them together with device 4, the feature will distribute approximately normally. We will handle it when the time is right.
<u>'Month':</u> (Figure 8)
Two main conclusions that we extracted from the 'month' plot.
First, there are no documentation for sessions in January & April. **Second**, it seems that there are some months that have a higher purchase rate than others. This led us to further investigate this feature, and after computing the total purchase percentage per month we classified the months into two categories: "strong months" as '1', "average months" as '0', with a threshold of 15% (the rounded-up average purchase per month). (Figure 8.1)

<u>Part 2- Handling the data:</u>

Now that we have a better understanding on our data, we would like to implement our findings and 'clean' the data a little bit. In this section, we focused on three main subjects:
1. Remove outliers values.
2. Grouping together some values in specific features & remove some of the irrelevant features.
3. Filling in missing values.

<u>Outliers:</u>

<u>For feature 'B'</u>, we used the common 1.5 IQR rule. (Figure 9)
<u>The 'Region' feature</u>, although it now distributes approximately normally, we already dealt with the outliers by grouping them together, so we won't lose the rest of the data in these rows.
<u>'Num of product pages', 'page value', 'User type, 'Exit rate', 'Info page duration', 'Product page duration', 'Admin page duration', 'Num of admin pages', 'Num of info pages':</u>
For those features outliers we selected thresholds manually. We tested few thresholds until we found a unique threshold that will remove no more than 1-2% of the data.
(Threshold: 200, 85, 0.001, 550, 750, 1000, 14, 7 in order of appearance) (Figure 9.1).

<u>Grouping & Dropping:</u>

In this section, we implemented our conclusions from **Part 1** regarding the categorial and numerical features. We dropped features 'C', 'D', 'id', 'internet browser', and handled features 'Month' & 'device' as mentioned before.

<u>Filling in missing values:</u> (Figure 10)

An extremely important part in any ML project is dealing with missing values. This part is important because without it, we might lose a significant part of our data for our model to train on.
We started by looking at the number of missing values in each feature that we had left, and then dealt with each feature the way we thought is best, as follows:
1. 'B' that seems to be distributing normally, with the **mean** value of the feature.
2. 'Num of info pages', 'info page duration', 'num of product pages' and 'exit rate'- with the **mean**

value, based on the histogram we presented before.
3. 'device', 'Region', 'closeness to holiday', 'num of admin pages', 'admin page duration', 'product page duration' and 'page values' with **median** value.
4. 'User_type' Nans as 'Returning Visitor', 'Weekend' to 0 **(vast majority)**.
5. 'Months' as "average months" (0).
6. 'A' with respect to the percentage of appearance of each category.
After this treatment for the features missing values, we had left with two features that are not dealt with yet (Figure 10.1).

Part 3- Correlation Matrix, Dummy Variables & Normalization:

Correlation Matrix: (Figure 11)

Few insights:
1. 'Exit rates' and 'bounce rates' have very high correlation (91%) - we will remove bounce rates and will remain with exit rates.
2.'Total duration' & 'num of product pages' has 82% correlation. Because 'total duration' has almost 50% Nan values, we will remove it end stay with 'num of product pages'.
3. We can see that page value may be a good predictor to the label. We'll keep an eye on this one.

Dummy Variables:

Dummies are categorial variables that we believe that have an impact on the label. We transformed features 'A', 'device' & 'Region' into dummies, and 'User type' to '0' ('Returning visitor') and '1' ('New visitor') as mentioned earlier.

Normalization:

Right before we start selecting our features and building our models, we would like to normalize our data. Normalizing the data is important so that we can have the same scale for all our features without distorting differences in the ranges of values or losing information. In addition, we make sure that all our features are now numerical type.

Part 4- Feature Selection:

After cleaning the data, filling missing values, dropping irrelevant features, and creating dummy variables, we had 8 more features then we started with! This could be harmful to our model by overfitting it to the train data, so we used "Forward selection" method to reduce the number of features that our model will use. We used 'Mallow's CP' measurement to choose the optimal number of features to train our model, 10! (Figure 12).
The selected features that we implemented on LinearRegression model were: 'info_page_duration', 'num_of_product_pages', 'product_page_duration', 'ExitRates', 'PageValues', 'Month', 'user_type', 'Weekend', 'A_s', 'device_3.0'.

Part 5- Modeling & Model Evaluation:

Model evaluation function:

We created a model evaluation function that receives a data frame and a model, runs a K-Fold cross validation (n splits=5), and evaluates the model by plotting a ROC_AUC curve of all the K-Folds. Each K-Fold the model is calling .fit with the train data and makes the predictions on the validation set.

Models' basic description

In each model, first we ran a gscv (GridSearchCV) on a random 'chunk' of train data (80%) to find the best hyperparameters to use. Each model was sent to the model evaluation function with the chosen hyperparameters.

Logistic Regression (Figure 13)

The params we chose to send to the gscv and their values are:

1. **Solver: saga** – is the algorithm that is used in the optimization problem and may affect the output significantly. **Penalty: l1** – the penalty calculation method may affect on the results of the model. **C: 0.01** – the penalty used in the regularization, influences the fit or overfit potential of the model.
2. The rest of the params were set as the default as we believed they are less significant than the one's we chose: (*dual=False*, tol=*0.0001*, fit*_intercept=True*, intercept*_scaling=1*, class*_weight=None*, *random_state=None*, *max_iter=100*, *multi_class='auto'*, *verbose=0*, *warm_start=False*, *n_jobs=None*, *l1_ratio=None*)

Result: we got a mean AUC result of 90%.

KNN (Figure 14)

The params we chose to send to the gscv and their values are:

1. **n_neighbors: 9** – number of neighbors to use to decide the label of the prediction. **Weights: distance** – determines the influence of the neighbors in the neighborhood by their distance. **Metric: manhattan** – the metric function which is used for the distance calculation.
2. The rest of the params were set as the default as we believed they are less significant than the one's we chose: (*algorithm='auto'*, *leaf_size=30*, *p=2*, *metric_params=None*, *n_jobs=None*)

Result: we got a mean AUC result of 86%. This model performance was less than the rest, and this fact aligns with the fact that we know the KNN is less suitable for binary problems.

SVM (Figure 15).

The params we chose to send to the gscv and their values are:

1. **Kernel: rbf** – the kernel type transformation that is used in the algorithm. **Gamma: auto** – is kernel coefficient and uses 1/n_features. **C: 1.0** – the penalty used in the regularization, has effect on the fit or overfit potential of the model. **Probability: True** – we set it to true so we can send the model to the evaluation func.
2. The rest of the params were set as the default as we believed they are less significant than the one's we chose. The parameters: degree = 3coef0 = 0.0, shrinking=True, tol=0.001, cache_size = 200, class_weight = None, verbose = False, max_iter=- 1, decision_function_shape='ovr', break_ties = False, random_state=None)

Result: we got a mean AUC result of 90%.

Random Forest (Figure 16)

The RandomForestClassifier has a lot of parameters to control, and we could not send all the params for gscv, so we chose 4 which we believe have impact. The params we chose to send to the gscv and their values are:

1. **N_estimators: 200** – as bigger the number the better. Determines the number of trees in the forest. **Max_features: auto** – set to be sqrt (number of features). The number of features to consider in each tree split. **Min_smaples_split: 10** – the minimum number of samples required to split an internal node – means the split will be based on 10 different data samples. **Max_depth: 10** – the maximum depth of the tree.
2. The rest of the params were set as the default: criterion='gini', min samples leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)

Result: we got a mean AUC result of 92%. This is the best result we got, and we decided to make our predictions on the test data set with this model.

Overfitting

To check if our model is overfit, we compared the results of the mean AUC score of the RF model when predicting probability with validation data (92%) with the results of predicting probability with the train data (98%) (with which the model was trained). We looked at the gap between the scores and found that it is 6% which is not a big gap, so we concluded our model is not suffering of overfitting.

Confusion matrix analysis (Figure 17)

Accuracy = 0.94 - that means we are correct in 94% precents of the cases when we need to determine whether a costumer has purchased a product or not.

Precision = 0.74 - that means that in 74% of the cases we determined a costumer has purchased a product he actually did.

Sensitivity = 0.85 - that means that from the costumers who really purchased a product we predicted 85% correctly.

Specificity = 0.95 - that means that from the costumers who didn't buy the product we predicted 95% correctly.

Part 6- Conclusion:

Executive summary & summary

In this project we got a set of data regarding user's sessions in of E-Commerce website. The data provided us with different information about the user's sessions such as the time users spent in product info page, user’s browser, month of the session etc. and of course whether a product was bought in the session. Based on the train data set we received, the problem we were asked to predict is:

Will a session end up as a purchase or not?

To answer this question, we did the following steps:

1. Explored the data: explored its different features, their distribution, and the connection between features to making a purchase or not.
2. Data processing: based on the data exploration we removed features, created dummy variables from categorial features and removed outliers. Before started modelling we also normalized all the features to be on a scale between 0 to 1.
3. Feature selection: by the end of the data processing, we ended up with 30 features. To decrease the problem dimensionality, we made a forward feature selection process on the data and as a result managed to lower the dimensionality to 10 features.
4. Modeling: we processed the data on different models: logistic regression, KNN, SVM and random forest classifier. The model that we got the best results from was the random forest with 92% of AUC. We decided to use this model for making the predictions on the test data set.

   In addition, we analyzed a confusion matrix to get a better understanding on our model performance.

Finally, we ran all our process on the test dataset and made predictions on purchase using our selected random forest model.

Appendix #1: Work distribution

In general, most of our progress was made when we got together and worked side by side. We consulted a lot with each other, shared our wisdom, and helped find solutions to each other problems.

Shared responsibility:

- Data exploration
- Visualizations
- Overfitting analysis
- Feature selection

Kirill:

- Modelling
- Model evaluation
- Confusion Matrix
- Pipeline

Shachar:

- Deep feature analysis
- Outliers
- Filling NA's
- Final Report

Appendix #2: Plots & Graphs

Figure 1: Numerical features preliminary analysis

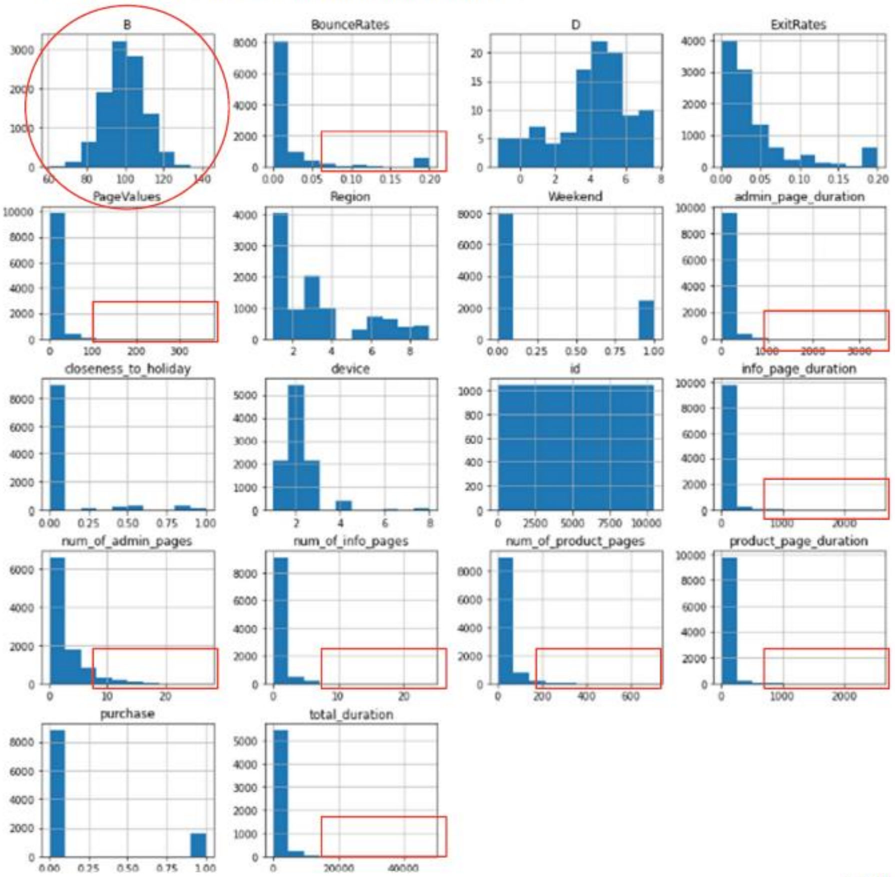


Figure 2: 'C'

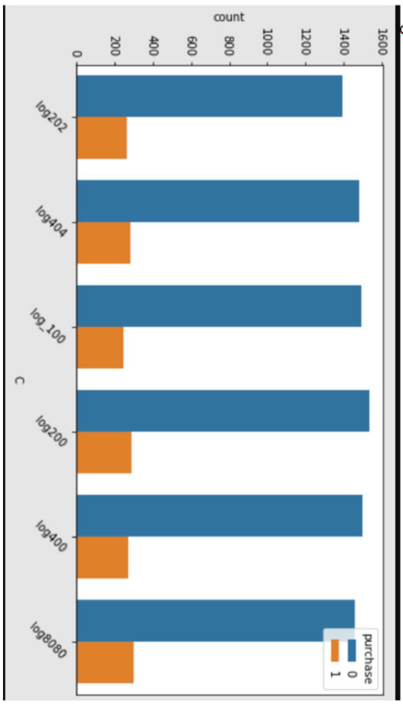
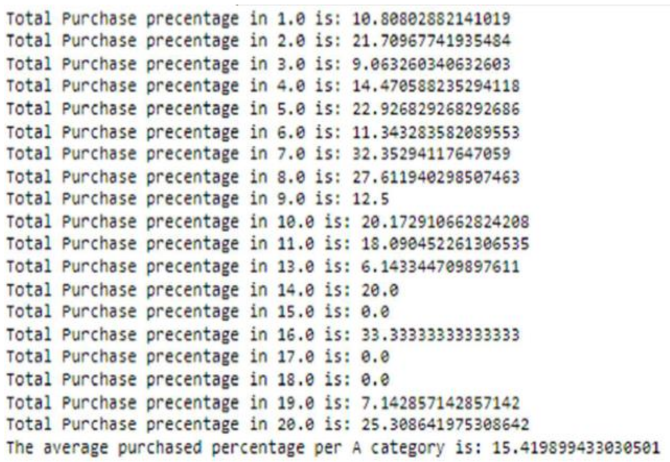
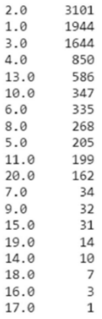


Figure 3.1:



```
Total Purchase precentage in 1.0 is: 10.80802882141019
Total Purchase precentage in 2.0 is: 21.70967741935484
Total Purchase precentage in 3.0 is: 9.063260340632603
Total Purchase precentage in 4.0 is: 14.470588235294118
Total Purchase precentage in 5.0 is: 22.926829268292686
Total Purchase precentage in 6.0 is: 11.343283582089553
Total Purchase precentage in 7.0 is: 32.35294117647059
Total Purchase precentage in 8.0 is: 27.611940298507463
Total Purchase precentage in 9.0 is: 12.5
Total Purchase precentage in 10.0 is: 20.172910662824208
Total Purchase precentage in 11.0 is: 18.090452261306535
Total Purchase precentage in 13.0 is: 6.143344709897611
Total Purchase precentage in 14.0 is: 20.0
Total Purchase precentage in 15.0 is: 0.0
Total Purchase precentage in 16.0 is: 33.33333333333333
Total Purchase precentage in 17.0 is: 0.0
Total Purchase precentage in 18.0 is: 0.0
Total Purchase precentage in 19.0 is: 7.142857142857142
Total Purchase precentage in 20.0 is: 25.308641975308642
The average purchased percentage per A category is: 15.419899433030501
```

Figure 3: 'A'



```
2.0       3101
1.0       1944
3.0       1644
4.0        850
13.0       586
10.0       347
6.0        335
8.0        268
5.0        205
11.0       199
20.0       162
7.0         34
9.0         32
15.0        31
19.0        14
14.0        10
18.0         7
16.0         3
17.0         1
```

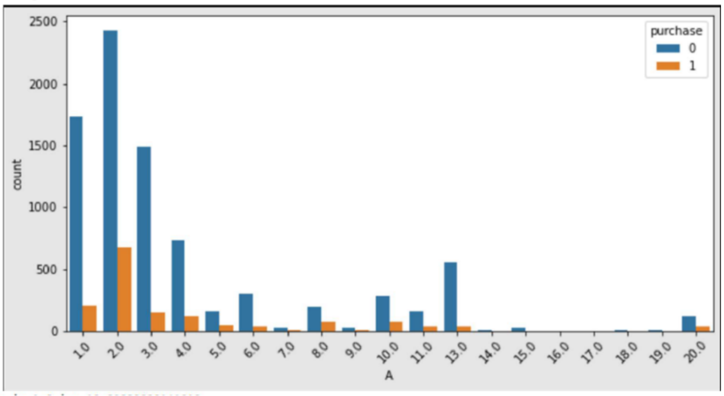


Figure 4: 'internet browser '

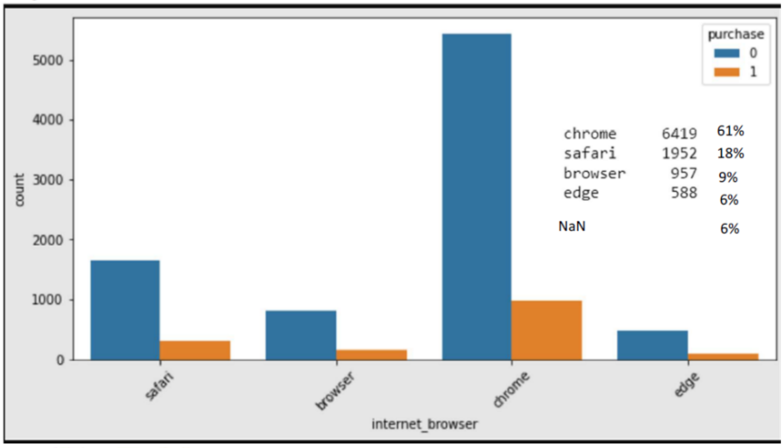
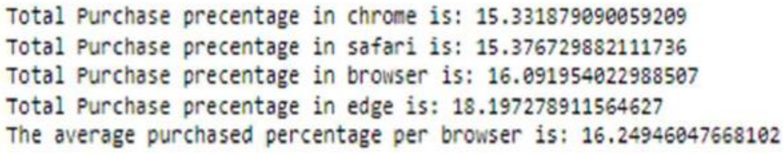


Figure 4.1



```
Total Purchase precentage in chrome is: 15.331879090059209
Total Purchase precentage in safari is: 15.376729882111736
Total Purchase precentage in browser is: 16.091954022988507
Total Purchase precentage in edge is: 18.197278911564627
The average purchased percentage per browser is: 16.24946047668102
```

Figure 5: 'user type'

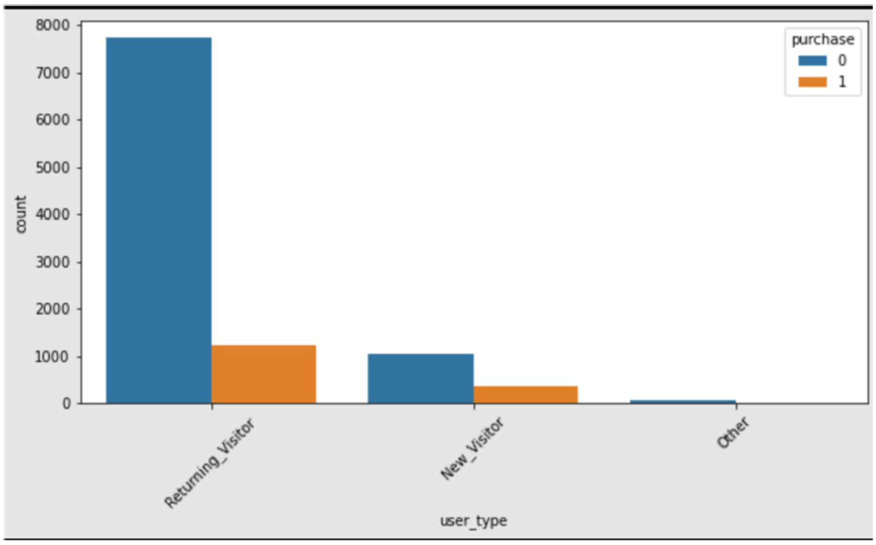


Figure 6: 'Region'

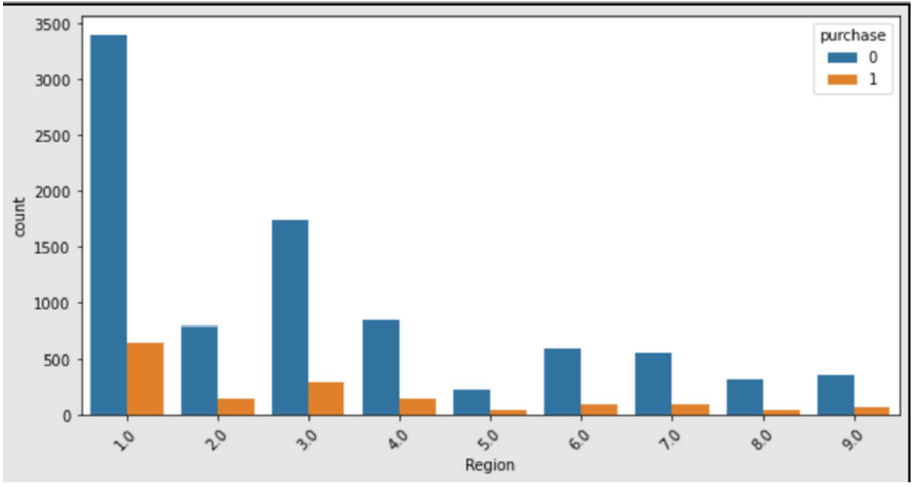


Figure 8: 'Month'

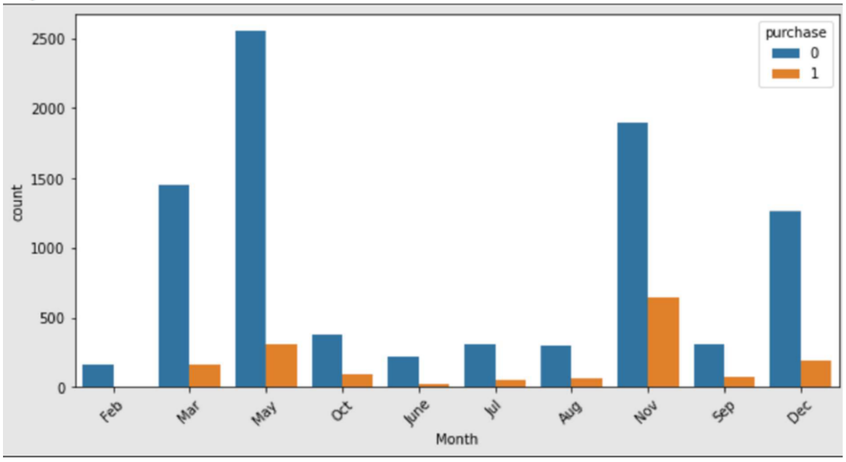
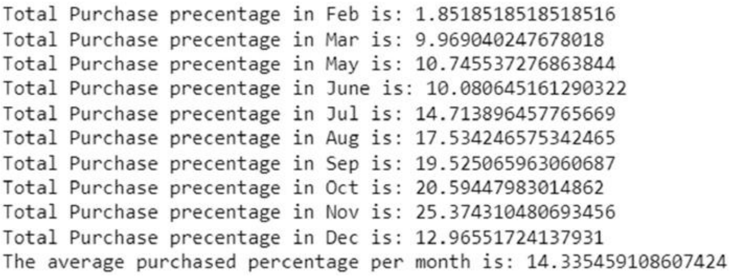
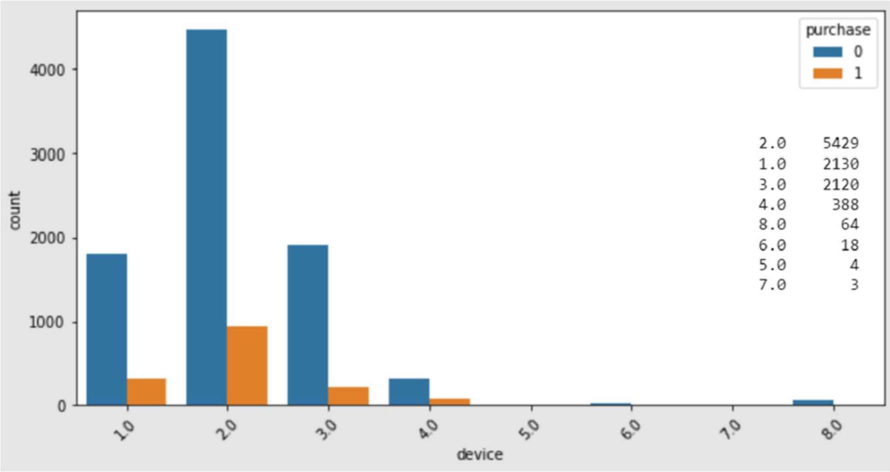


Figure 8.1:



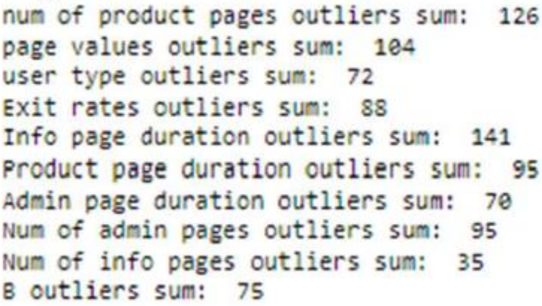
```
Total Purchase precentage in Feb is: 1.8518518518518516
Total Purchase precentage in Mar is: 9.969040247678018
Total Purchase precentage in May is: 10.745537276863844
Total Purchase precentage in June is: 10.080645161290322
Total Purchase precentage in Jul is: 14.713896457765669
Total Purchase precentage in Aug is: 17.534246575342465
Total Purchase precentage in Sep is: 19.525065963060687
Total Purchase precentage in Oct is: 20.59447983014862
Total Purchase precentage in Nov is: 25.374310480693456
Total Purchase precentage in Dec is: 12.96551724137931
The average purchased percentage per month is: 14.335459108607424
```

Figure 7: 'device'

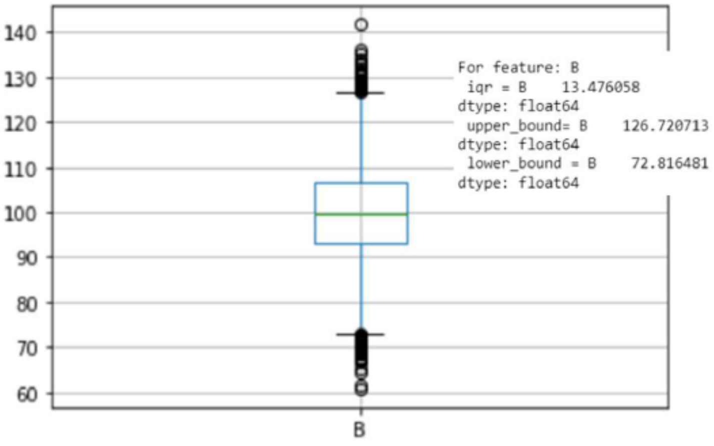


```
2.0    5429
1.0    2130
3.0    2120
4.0     388
8.0      64
6.0      18
5.0       4
7.0       3
```

Figure 9: Outliers

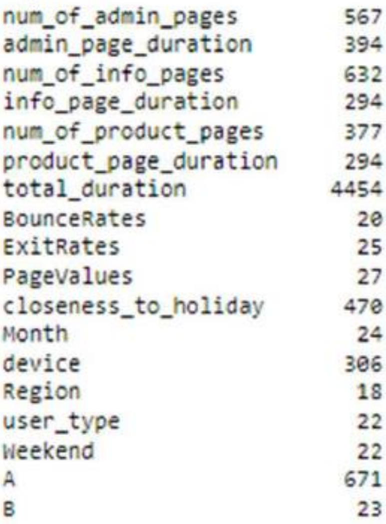


```
num of product pages outliers sum:  126
page values outliers sum:  104
user type outliers sum:  72
Exit rates outliers sum:  88
Info page duration outliers sum:  141
Product page duration outliers sum:  95
Admin page duration outliers sum:  70
Num of admin pages outliers sum:  95
Num of info pages outliers sum:  35
B outliers sum:  75
```



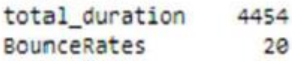
```
For feature: B
 iqr = B     13.476058
dtype: float64
 upper_bound= B     126.720713
dtype: float64
 lower_bound = B     72.816481
dtype: float64
```

Figure 10:



```
num_of_admin_pages        567
admin_page_duration       394
num_of_info_pages         632
info_page_duration        294
num_of_product_pages      377
product_page_duration     294
total_duration           4454
BounceRates                20
ExitRates                  25
PageValues                 27
closeness_to_holiday      470
Month                      24
device                    306
Region                     18
user_type                  22
Weekend                    22
A                         671
B                          23
```

Figure 10.1:



```
total_duration     4454
BounceRates          20
```

Figure 11:

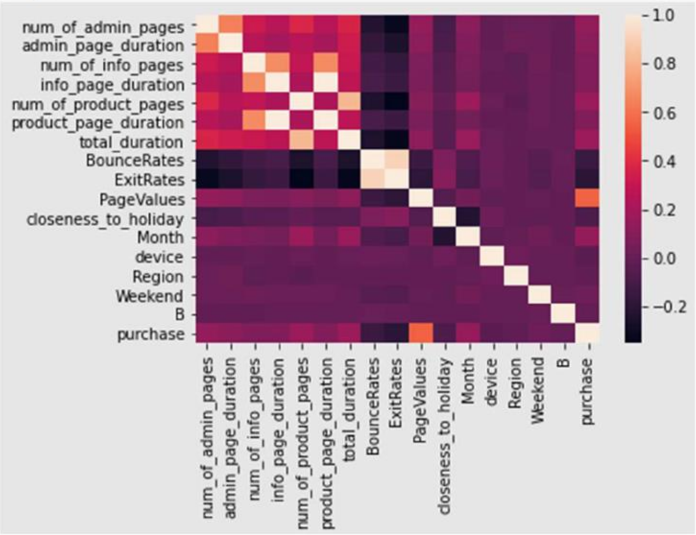
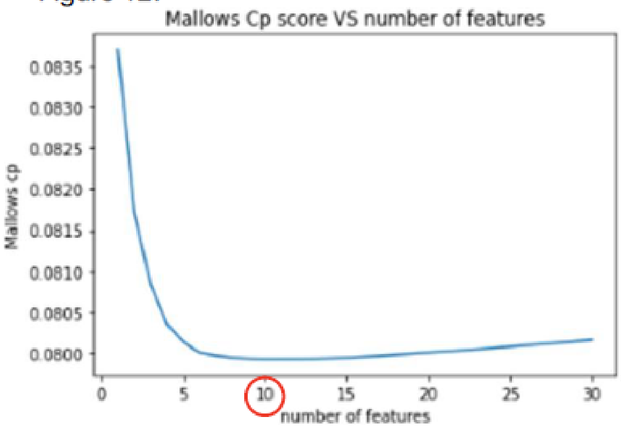
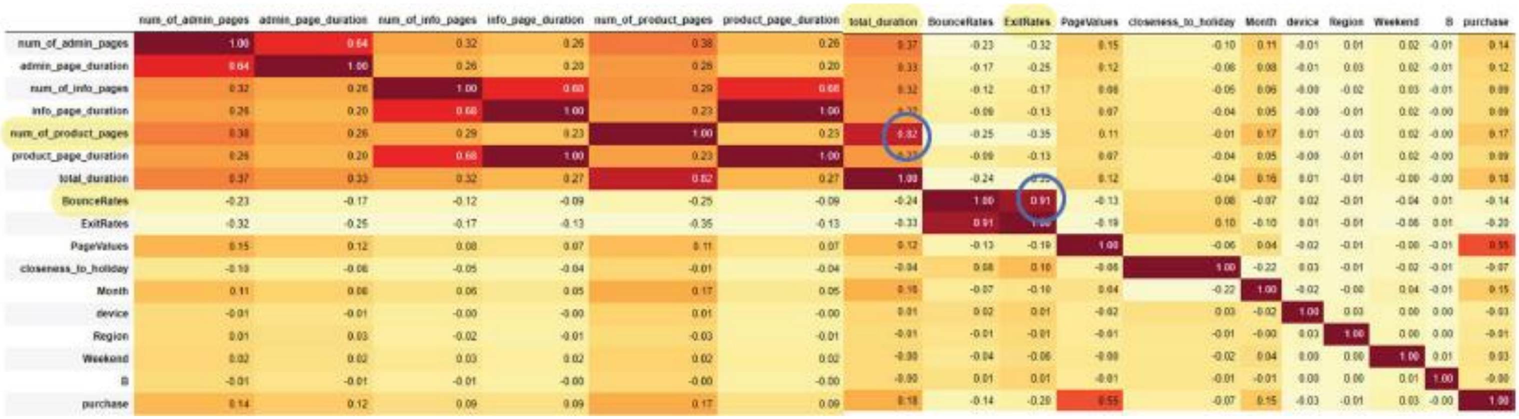


Figure 12:





	num_of_admin_pages	admin_page_duration	num_of_info_pages	info_page_duration	num_of_product_pages	product_page_duration	total_duration	BounceRates	ExitRates	PageValues	closeness_to_holiday	Month	device	Region	Weekend	B	purchase
num_of_admin_pages	1.00	0.64	0.32	0.26	0.38	0.26	0.37	-0.23	-0.32	0.15	-0.10	0.11	-0.01	0.01	0.02	-0.01	0.14
admin_page_duration	0.64	1.00	0.26	0.20	0.28	0.20	0.33	-0.17	-0.25	0.12	-0.08	0.08	-0.01	0.03	0.02	-0.01	0.12
num_of_info_pages	0.32	0.26	1.00	0.68	0.29	0.68	0.32	-0.12	-0.17	0.08	-0.05	0.06	-0.00	-0.02	0.03	-0.01	0.09
info_page_duration	0.26	0.20	0.68	1.00	0.23	1.00	[illegible]	-0.09	-0.13	0.07	-0.04	0.05	-0.00	-0.01	0.02	-0.00	0.09
num_of_product_pages	0.38	0.26	0.29	0.23	1.00	0.23	0.82	-0.25	-0.35	0.11	-0.01	0.17	0.01	-0.03	0.02	-0.00	0.17
product_page_duration	0.26	0.20	0.68	1.00	0.23	1.00	[illegible]	-0.09	-0.13	0.07	-0.04	0.05	-0.00	-0.01	0.02	-0.00	0.09
total_duration	0.37	0.33	0.32	0.27	0.82	0.27	1.00	-0.24	[illegible]	0.12	-0.04	0.16	0.01	-0.01	-0.00	-0.00	0.18
BounceRates	-0.23	-0.17	-0.12	-0.09	-0.25	-0.09	-0.24	1.00	0.91	-0.13	0.08	-0.07	0.02	-0.01	-0.04	0.01	-0.14
ExitRates	-0.32	-0.25	-0.17	-0.13	-0.35	-0.13	-0.33	0.91	1.00	-0.19	0.10	-0.10	0.01	-0.01	-0.06	0.01	-0.20
PageValues	0.15	0.12	0.08	0.07	0.11	0.07	0.12	-0.13	-0.19	1.00	-0.06	0.04	-0.02	-0.01	-0.00	-0.01	0.55
closeness_to_holiday	-0.10	-0.08	-0.05	-0.04	-0.01	-0.04	-0.04	0.08	0.10	-0.06	1.00	-0.22	0.03	-0.01	-0.02	-0.01	-0.07
Month	0.11	0.08	0.06	0.05	0.17	0.05	0.16	-0.07	-0.10	0.04	-0.22	1.00	-0.02	-0.00	0.04	-0.01	0.15
device	-0.01	-0.01	-0.00	-0.00	0.01	-0.00	0.01	0.02	0.01	-0.02	0.03	-0.02	1.00	0.03	0.00	0.00	-0.03
Region	0.01	0.03	-0.02	-0.01	-0.03	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.00	0.03	1.00	0.00	0.00	-0.01
Weekend	0.02	0.02	0.03	0.02	0.02	0.02	-0.00	-0.04	-0.06	-0.00	-0.02	0.04	0.00	0.00	1.00	0.01	0.03
B	-0.01	-0.01	-0.01	-0.00	-0.00	-0.00	-0.00	0.01	0.01	-0.01	-0.01	-0.01	0.00	0.00	0.01	1.00	-0.00
purchase	0.14	0.12	0.09	0.09	0.17	0.09	0.18	-0.14	-0.20	0.55	-0.07	0.15	-0.03	-0.01	0.03	-0.00	1.00

Figure 13:

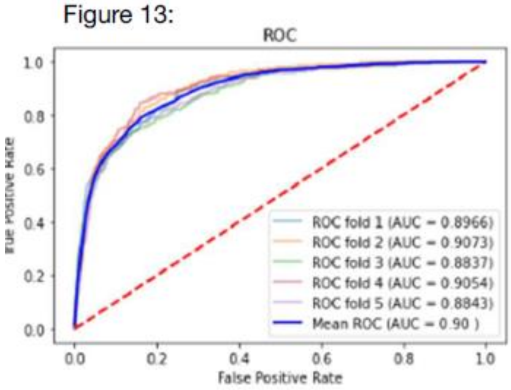


Figure 14:

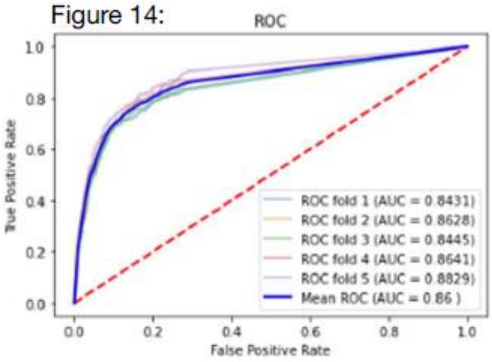


Figure 15:

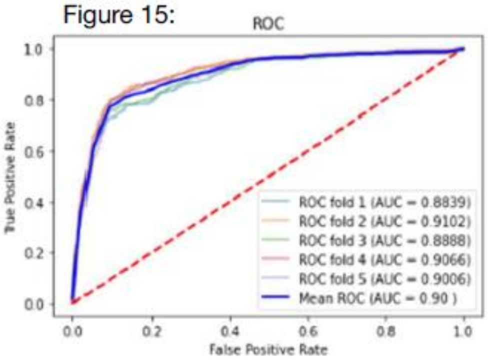


Figure 16:

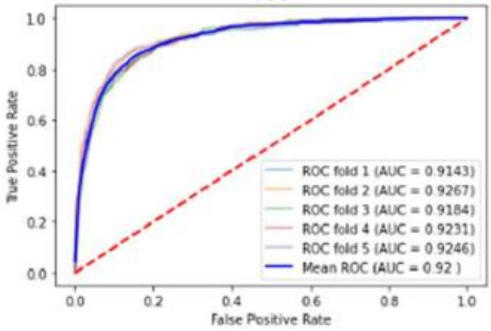
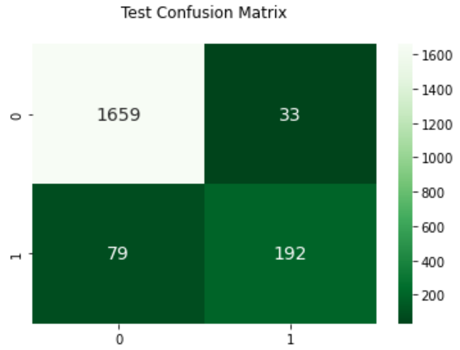


Figure 17:


Test Confusion Matrix