Machine Learning Project

Data assumptions

1. Our provided dataset examples are dispersed independently and identically (IID) according to an unknown probability distribution.
2. Each training data point is made up of a set of vectors and a class label for each vector.
3. Our selection was based on our dataset from a single e-commerce store. We did not presume domain information about this e-commerce store, product, website, or other unrelated parts. taken into account.
4. Our selection was considered for our study conclusions (there is no bias selection, thus we will rely on our insight).

Exploratory Data Analysis

The dataset consists of 21 features,[1] of which is divided into 13 numerical and 8 categorical attributes. The 'purchase' attribute is used as the class label. The exploration part required extensive and varied work on the data while utilizing methods and plots to examine the columns. Below is a list related to the main insight we have been drawing out:

1. Our dataset does not contain duplicated rows.
2. The ID column is an observation index column and is not effective for the dataset, hence we removed it.
3. Technical indicators in several columns are related to user logistic information, and we wondered if they had any effect on purchases—such as devices and internet browsers.
4. We noticed a clear implication that we will handle working and training the data from our very first look at the data values, data description, and statistical values. Our high-level overview leads us to how to build our data preparation part.

Correlation and covariance We observed a few interesting insights based on the correlation matrices[2] and plots on our data set. 'PageValue' is an important attribute that, indicating the highest positive association with our label of 0.486, suggests that more product pages increase the possibility of selling a product. The 'D' indicator has a negative relationship with the purchase label of -0.75. For 'ExitRates' and 'bounceRates' have a strong correlation with each other of 0.91, we could use this correlation when dealing with dimension reduction. The number of product pages and total duration has a high correlation of 0.87. This makes sense because we can expect that the more pages of a product a user views, the more time a user spends in the session. purchase

In t

[1] See table 1
[2] see plot A3

he histoplot, pairplot[3], and destiny sections, we reviewed a graphical representation of how the features data points into specific ranges, their destinies, and data point correlation. We discovered that 'BounceRates' accounts for more than half of all '0' observations. as well as the number of admin pages grouped with about 40% doesn't have any admin info pages. 'B' indicator, which we do e-commerce the feature spec, but it looks like a normal distribution with a mean of about 100.

Our 'ExitRate', a Google Analytics metric, has a maximum of 0.2 percent, indicating the number of individuals that departed the website during a certain session. The purchase has a 0.2 negative connection with this indicator. From a domain standpoint, we can see how this influences the purchase because the contrast of the consequence of leaving the website reduces the purchase of a product greatly. Despite the similarities in the characteristics of 'Exit Rate' and 'Bounce Rate,' the count and destiny plots[4] display different observation scales and different outcomes, supporting our decision not to move them. We used a scatter plot[5] for each of the two feature combinations in our pairplot, with the data points colored by the label. 'Orange' is purchased, whereas 'blue' is not. This graph helps us analyze precise relationships between characteristics as well as provides a high-level summary of the impact on purchases.

Data Preparation First, we read the data and save it in 'y' the last column of purchase, ensuring that we have our labels and don't lose them during the X process. Before beginning training, we choose, clean, create, and format the final data set.

Missing values We first reviewed how many null values we have for each column with count and plot[6], as well as how much it constitutes from the column values. We examine each feature's attributes to determine how to fill[7] in the gaps in the null results. Our assumptions about missing values are that we are filled with column mean, float values in the duration columns, and continuous values. We filled in the median with the categorical values like months and regions because they are decimals. For binary values, we checked the median and filed zero because we think filing one can affect the prediction. For example, filling 'user_type' is safer as zero to not affect the prediction (1 for returning). The categorical values were filled with Midian. For the x test, we implemented the same assumptions, without checking the set.

We removed the string 'minutes' from objects in duration columns. In columns A, C we removed the strings 'c_' and 'log' from the objects turning them into numeric. Since those columns are categorical, we turned them into dummy columns, so there is no correlation

[3] see plot A4
[4] see plot A5
[5] see plot A6
[6] see plot A1
[7] see table 3

between higher value and purchase. Also, we found variable 'total_duration' has a very high percentage of nulls- 99% and 45%, so our assumptions are:

1. Our first guideline was to fill out website pages by the mean because we assumed that the average could be a good indication of those with fields.
2. For the columns that related to the session duration, due to the observation, we decided to fill them out with 0 because the duration has a very wide range. We chose to not use the meaning but to relate them as 0 values as an assumption that we don't know how long the user stays in the session.
3. Based on the high percentage, we should consider removing the total duration column. In addition, this column has many outliers. This is what we eventually chose.
4. 'D' has 99% of null values. Therefore, since we don't know the meaning of the column and in order to not affect the prediction we removed 'D'.

We implemented the same changes on the test set.

Categorical Analysis[8] for columns 'A', 'C', 'Months' we split the objects and filled the median, and turned them into dummy columns. For 'Weekend' and 'user_type', we turned into binary and filled 0. For the "internet_browser", we filtered the values by type of browser (chrome, safari, browser, edge), and saw that 'chrome' had the most observations and the median. in order to minimize the number of columns, we left the browser variable with 1 if chrome and else 0. For 'A' dummy columns we found that there are columns in the test that are not in the train and vice versa[9], so we removed them, to keep the same dimensions.

Outliers The main technique to find outliers was to look at an outlier plot and look for examples that were out of the ordinary. Total duration[10] has the most unusual data observations in terms of outliers. To handle this column outliers, we presume that the user session was accidentally opened. Our assumption was that a user could only stay for 6 hours. We eliminate the values from this range using this method. For admin page duration - we detected anomalies and took a closer look to examine. Due to similar conditions as the total duration column, we make the same assumption for admin page duration.

Standardization We scaled our prepped data with a standard scaler, we tried to use a minimax scalar but got lower accuracy. We think it is because we have a lot of float values that are smaller than 1.

Dimensional reduction We start with 21 columns in the train. After the preparation, we were left with 53 columns. We use PCA to reduce dimensions to the number of dimensions that explain 95% of the variance[11]. Reduced dimensions might help prevent overfitting, PCA reduces the dimension by the number of components, 42 in our data.

---

[8] see table 7
[9] see table 8
[10] see plot A2
[11] see plot B1

Train & Validation Set The validation set size is 20% of the training set, and the random state is 42. The first split is to implement PCA, the second one is on the reduced data.

Our Models

In all our models, we searched for the hyperparameters with GridSearchCV. We created functions that plot the AUC-roc curve, and confusion matrix in each model.

Logistic Regression We use the values 'C': 2.0, 'random_state': 1, 'solver': 'newton-cg' from gridsearch, we added n_jos=-1 to avoid warnings in the pipeline. It provides good performance with approximately 88.8% accuracy, the AUC is 88% also. These two values are quite comparable, we got 88% with 5 folds, and 88% in cross-validation. So, there is no question of overfitting. We plotted the confusion matrix[12] and got 151 true positive values. We created a function that checks how many true positive values we will get if we remove one column with low feature importance(coefficient)[13]. The conclusion was, that removing columns with low coefficients does not improve the value of tp samples, and still, TP counted 151 values if column 25 was removed. Then we tried the other way around, and created a new np array with columns that look to have high feature importance in the plot. To our surprise, we got a lower true positive rate of 130 (lower). Based on that analysis, fewer columns tend to overfit more in our data. Therefore, we didn't do any feature selection.

KNearest Neighbors We used gridsearchCV and found that the best parameters are the default ones of an algorithm: 'auto', n_neighbors:3, weights: 'uniform'. We also plotted N neighbors by accuracy score[14] and saw that more neighbors improve the score by 0.01. We got 90 true positive[15] values , accuracy 84%, 5 folds 84%,cross-validation 84% and AUC of 68%. We can presume that the model is overfitted due to the density of our points. we can see that in the "Classification by 3 Nearest Neighbors"[16], how close the points are to each other, and in the code in the 3D model.

SVM-support vector machine the hyper parameters are C:10, kernel: 'linear'. The accuracy is 89%, for 5 folds we got 89%, we did it with 5 folds and not the function because of the long run time. We got an AUC of 69% and 155 true positive[17]. We definitely see an over-fitted model, probably due to a large number of dimensions. We checked the coefficients and tried to remove the lowest ones, and got the same tp rate if not less.

[12] see B2
[13] see B3
[14] see B4
[15] see B5
[16] see B6
[17] see B7

Random Forest, we used RandomizedSearch[18], and we got an 86% accuracy score. For the true positives, we got 43 values [19]and 85%. In 13 folds, we got 86%. This model recalls the lowest number of true positives, and random forest might not fit this type of data well.

Model Evaluation We have drawn our implicit by the common methods as a confusion matrix, AUC roc curve[20], Kfolds, cross-validation, Classification report, R^2, and MSE. We found that all models except knn, have higher AUC -roc values then 80% [21]. Also, the classification report showed that overall, the logistic regression model has the best parameters[22] of precision, recall, and f1 out of our models. Our models have similar mse[23], and the logistic regression one is the second best. Therefore, we chose logistic regression to predict. We build a pipeline function at the end of the file.

A summary - executive report

In recent years, the number of online e-commerce sites has exploded, and our research focuses on predicting future sales. Our purpose was to investigate ways to predict purchases to improve profits based on our research of e-commerce sites' users' data.
Ml models have the power to help find what is affecting people's decision-making while shopping, and spending more.

Hypothetically, if we had the predictions to check our model's accuracy on the test, using the technique of classifying clients, like purchases or non-purchases, based on a collection of features from an e-commerce website, we would focus more on feature importance and feature selection, in order to find what fine-tune improvements can be made in different aspects of the e-commerce shop.

Making the best forecast based on the parameters appears to be similar to using a logistic regression model to estimate the likelihood of an E-commerce purchase. Hence, using logistic regression might be a good fit since the nature of the model is to estimate probabilities.

We trained our model, based on more than 10,000 results, and received a trained model that can predict results 88% of the time on the available data, with 76% precision. During the process, we took into account every step that could help us avoid overfitting, increase accuracy, and perform well based on the quality check indicators in the machine learning industry, such as AUC & ROC.

We found it interesting that a simple logistic regression using the hyper-parameters random state = 1, C = 10, and a fair number of features and dimensions, yielded the best results for our analysis.

[18] see table for parameters
[19] see plot B8
[20] under each model in the code
[21]see plot C1
[22]see plot C2
[23]see ploc C3

This paper describes the many analyses performed on the purchase dataset, including the use of a logistic regression model to estimate the likelihood of an E-commerce user making a purchase. This model may be applied to a variety of e-commerce datasets and used as a tool to assist in making better decisions and boosting sales.

Appendix

Contribution

Ofir Haim - Outliers, part 1 of exploration, sns plots, plots in exploration part, NA values exploration, categorical values, random forest model, SVM model, pipeline, summarized the data to the report , comments and notes in the code, removing columns, randomizedsearchCV,executive report.

Veronica Iovchik - split objects in data exploration, correlation matrix, turning plots into plotly, dimensionality reduction, standardization, grid search, logistic regression model, knn model, graphs in models, model evaluation functions and graphs, pipeline(we both worked on it), suggesting turning to dummy variables, exploring model improvement.

Tables

1. column type

Numeric columns	Object
num_of_admin_pages	info_page_duration
admin page duration	info_page_duration
num_of_product_pages	internet browser
total duration	Month
Bounce Rates	Weekend
Exit_Rates	user_type
Page Values	A
closeness_to_holiday	

device	
B	
D	
purchase - (Label)	

2 categorical

Split the object into numbers	String to binary	String to number
info_page_duration product_page_duration A C	user_type Weekend internet browser	Month

3

How we filled the values

mean	median	Filled with 0
num_of_info_pages num_of_admin_pages num_of_product_pages BounceRates B ExitRates PageValues	Region Month A C	admin_page_duration total_duration device closeness_to_holiday internet_browser

9 Columns we removed

data train	data test
‘A_16’,’A_17’	‘A_12’

10 Best RandomizedSearch parameters

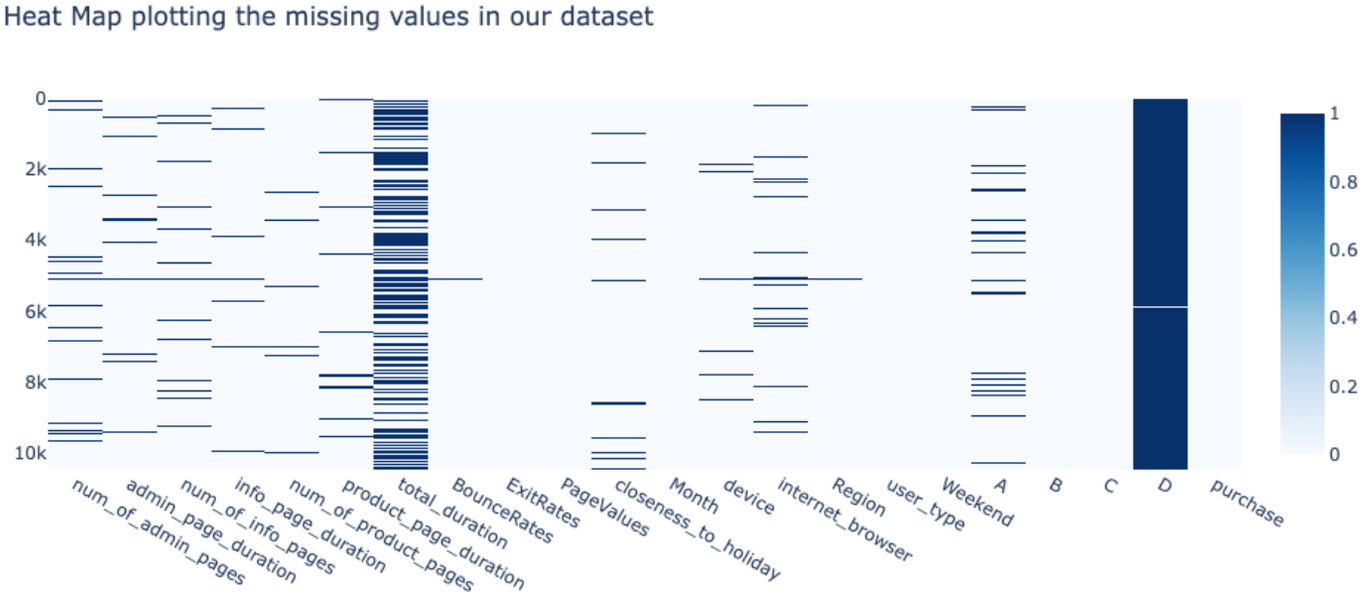
paramet	value

‘n_estimators’	150
‘min_weight_fraction_lead’	0.01
‘min_sample_split’	0.209
‘max_features’	‘auto’
‘criterion’	‘gini

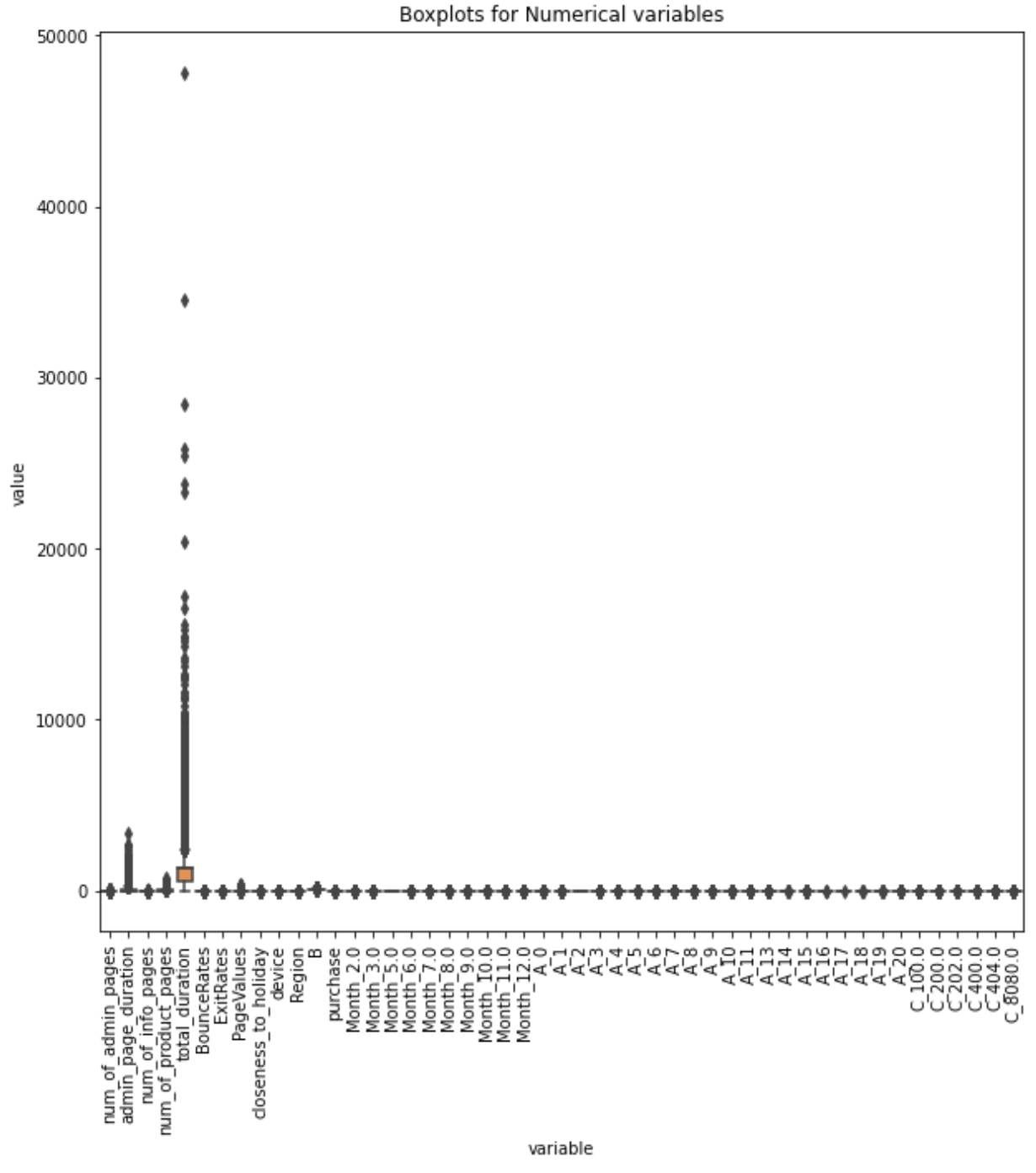
Plots

Note: some plots are made with plotly, you can see the numaric value of the column in the code.

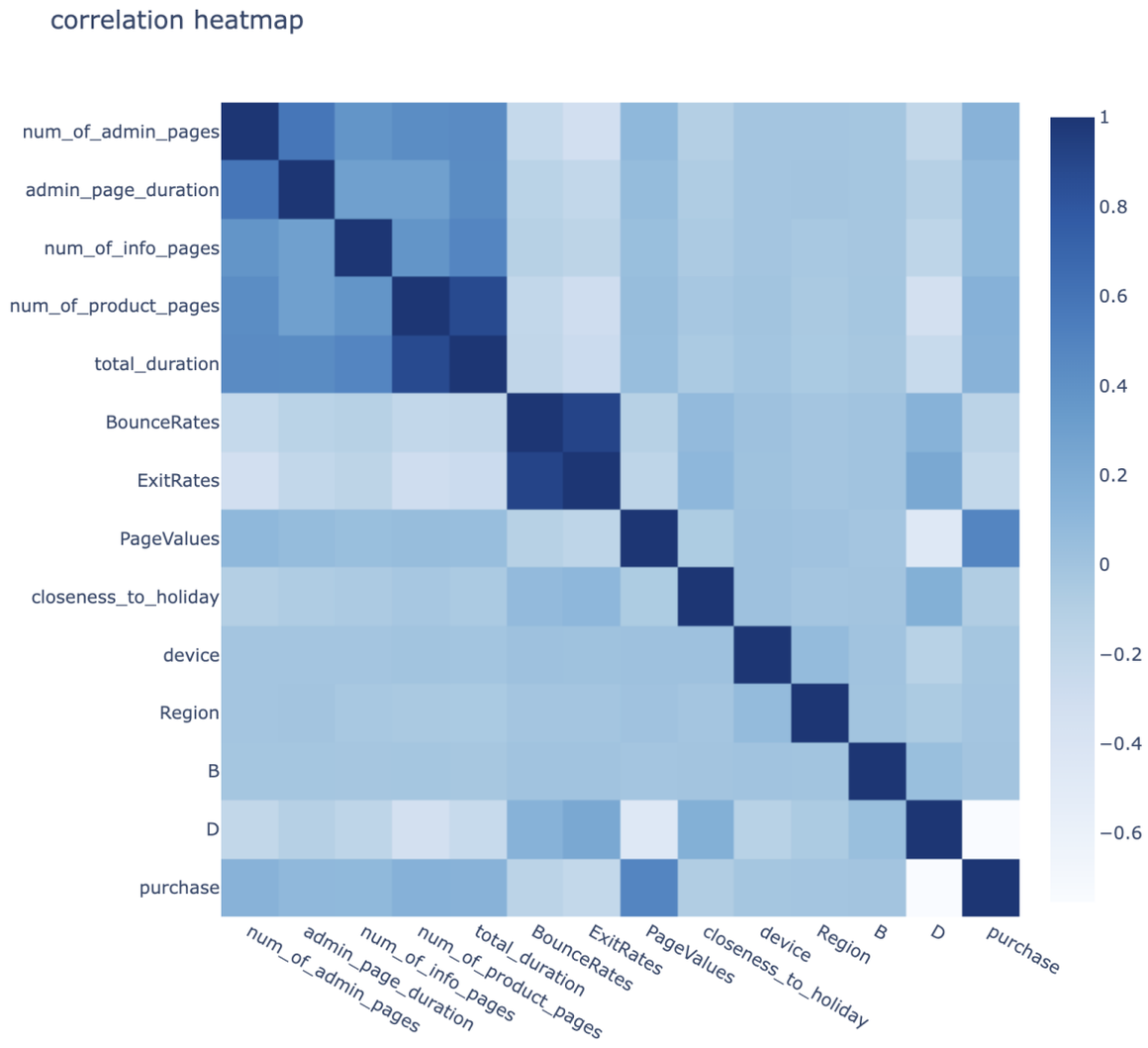
A1 heat map plotting the missing values in our dataset



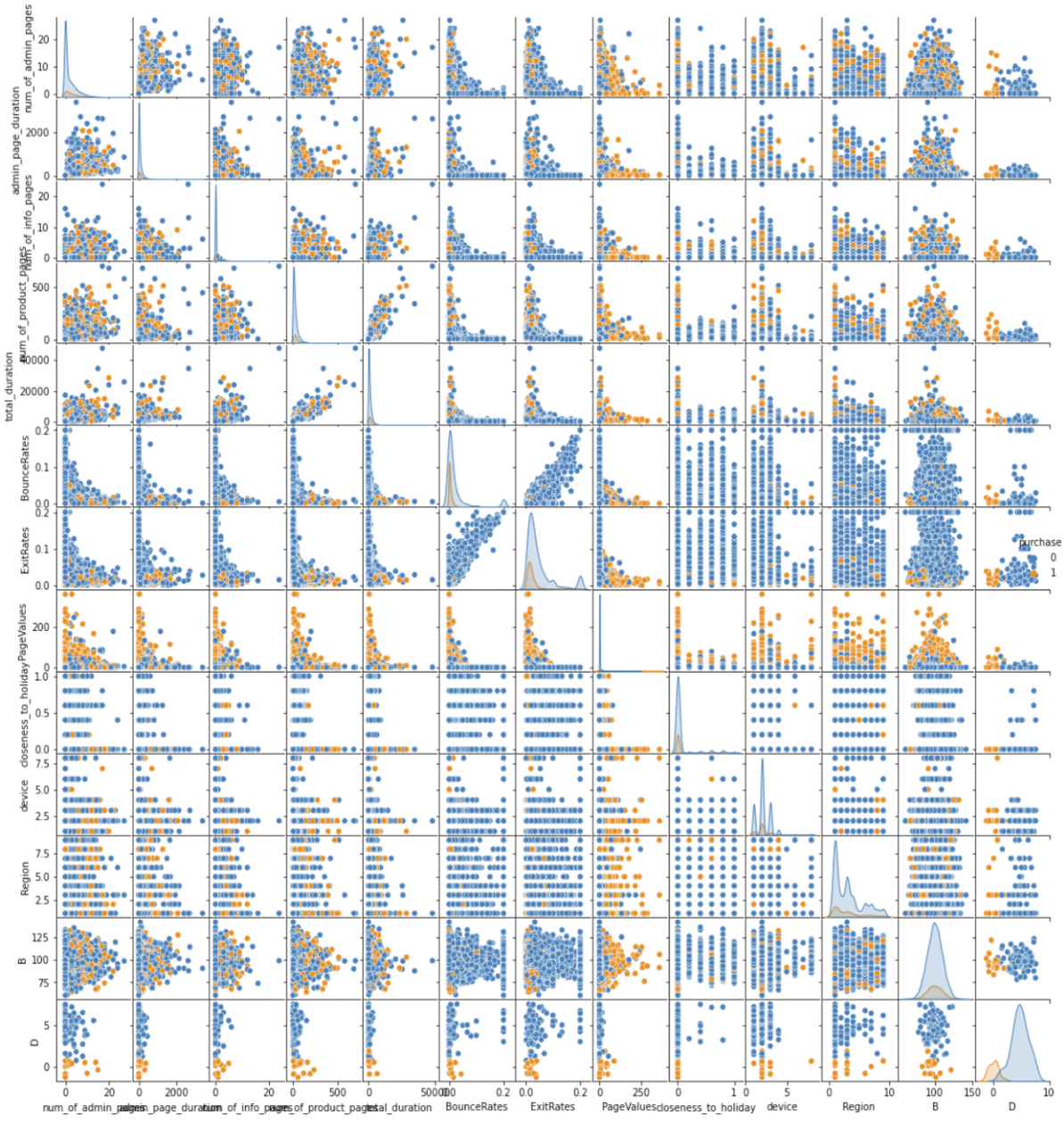
A2 Boxplots for Numerical Variables



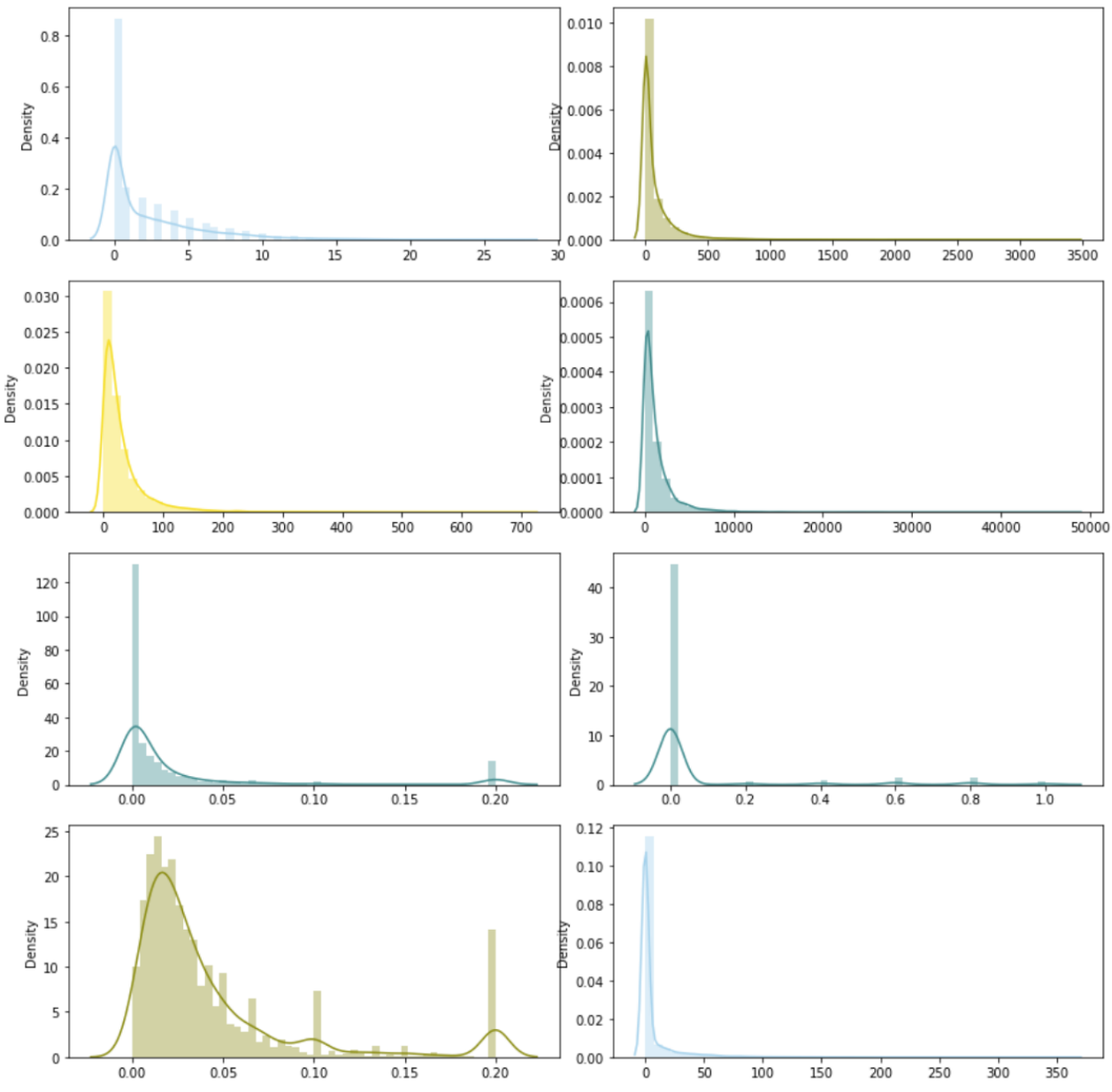
A3 correlation heatmap



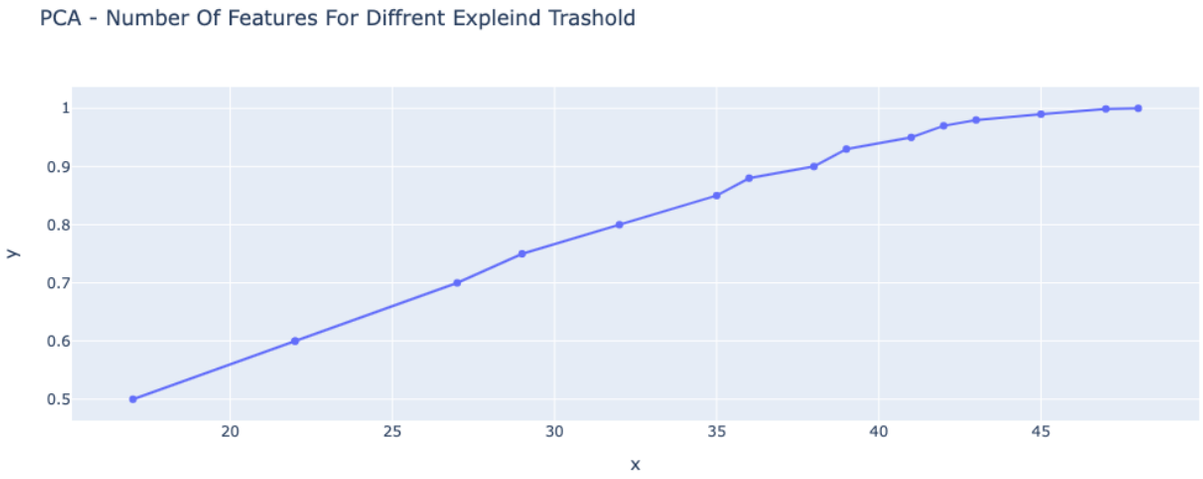
A4 pairplot



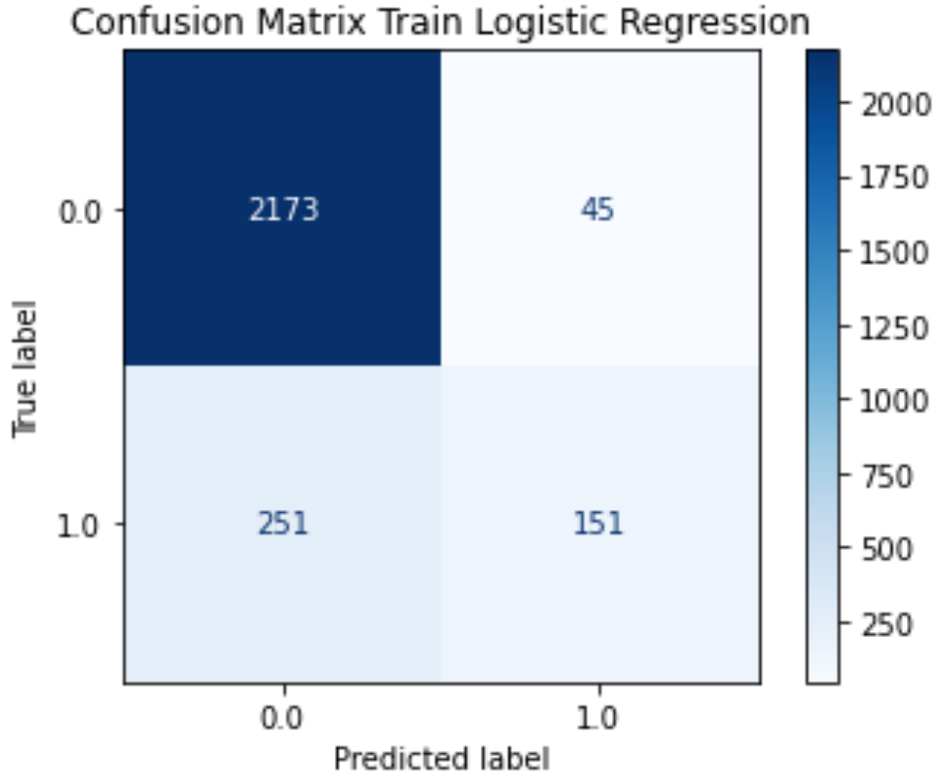
A5 Density plot



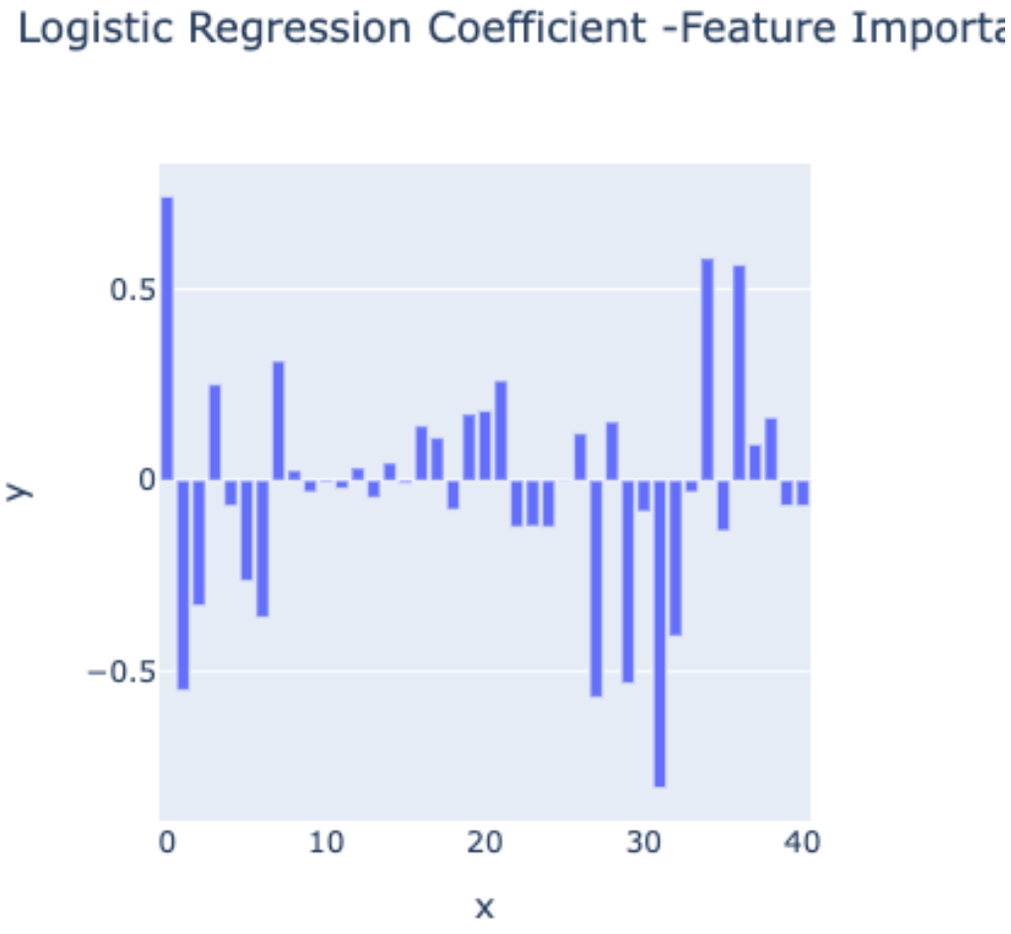
B1 PCA number OF Features by Explained Variance Threshold



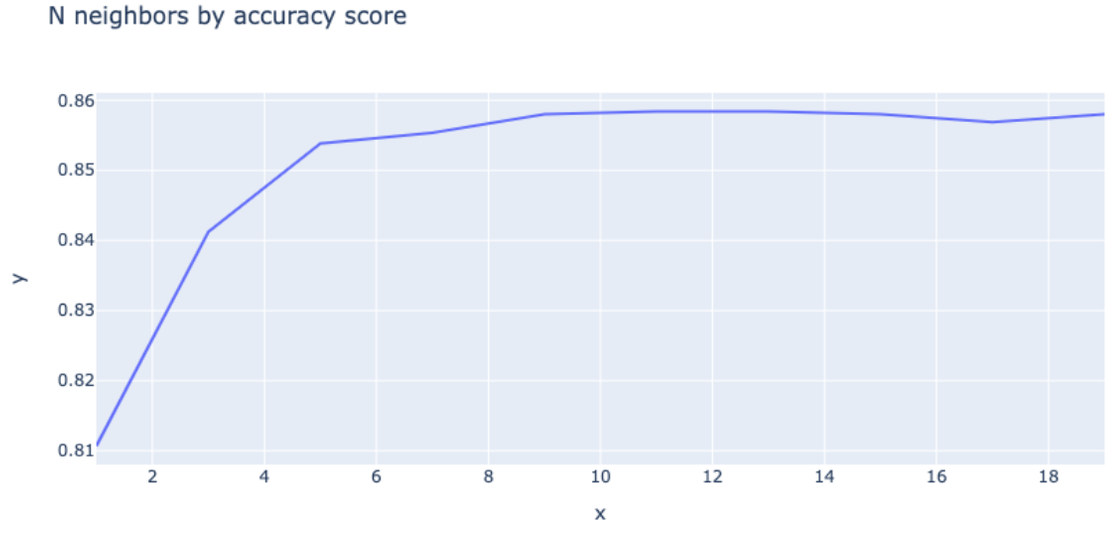
B2 Confusion Matrix Logistic Regression



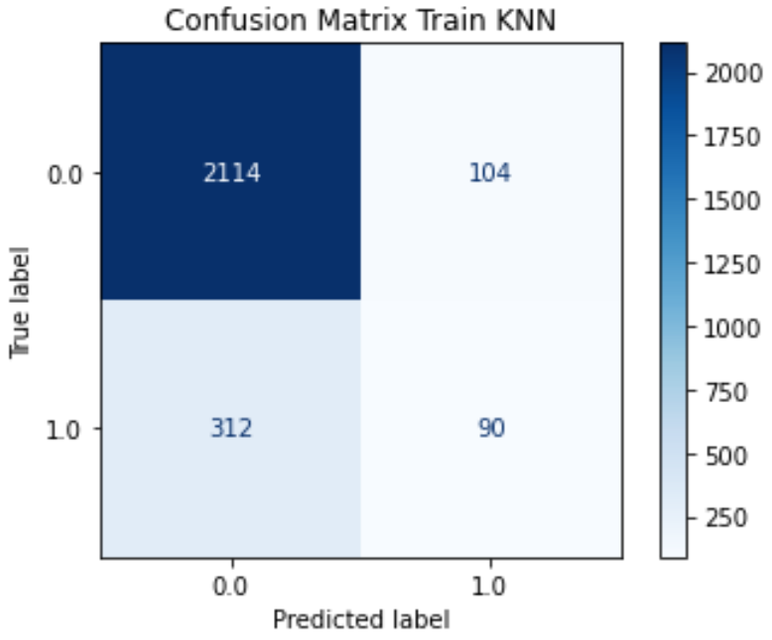
B3 Logistic Regression Coefficient- Feature Importance



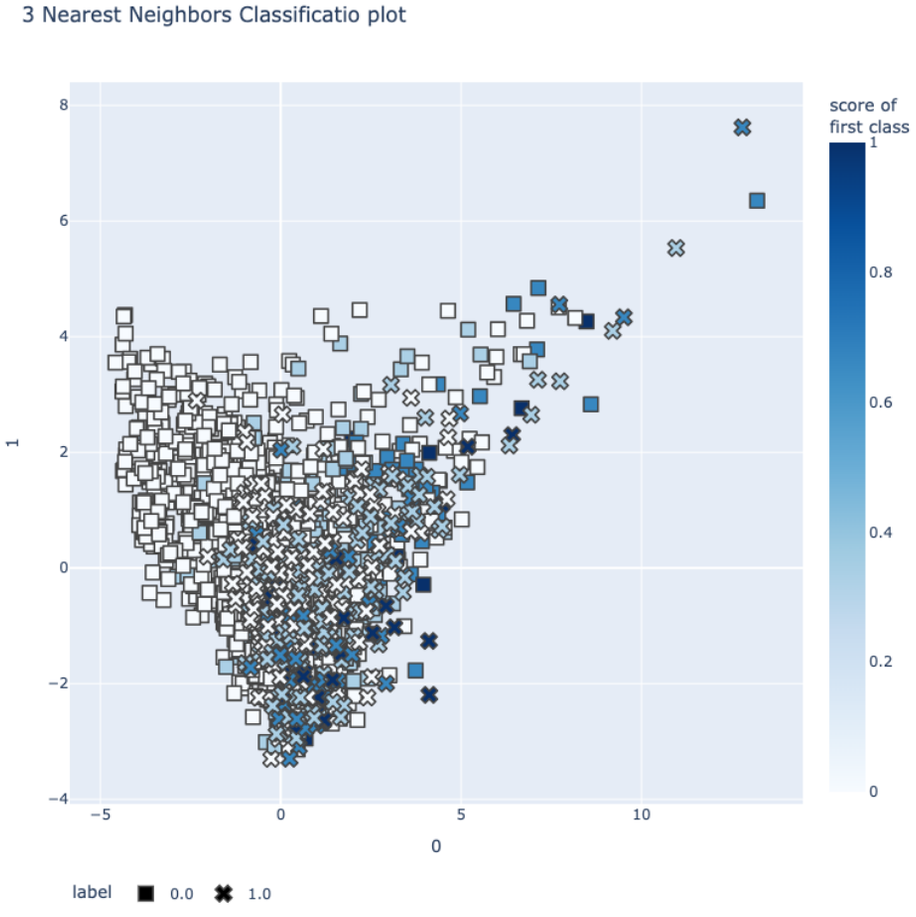
B4 N neighbors by accuracy score.



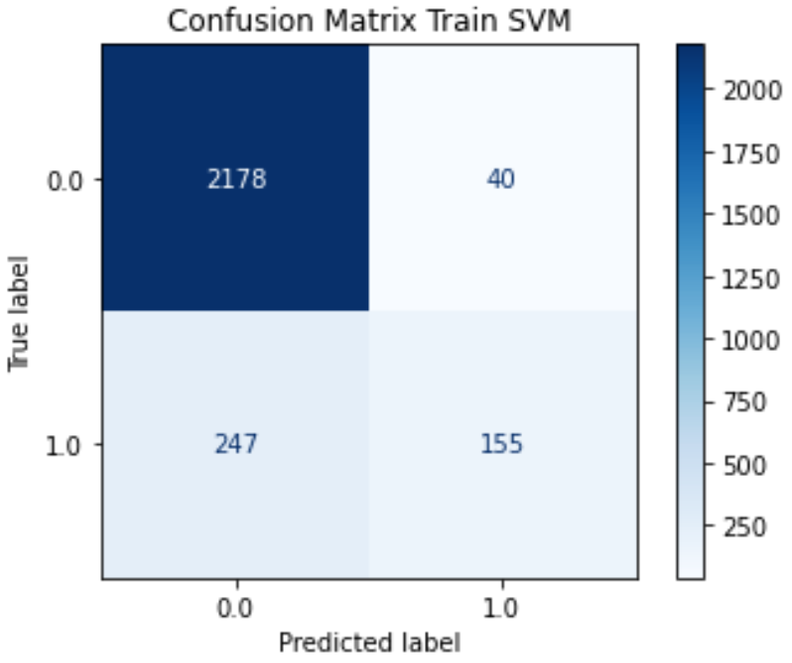
B5 Confusion Matrix KNN



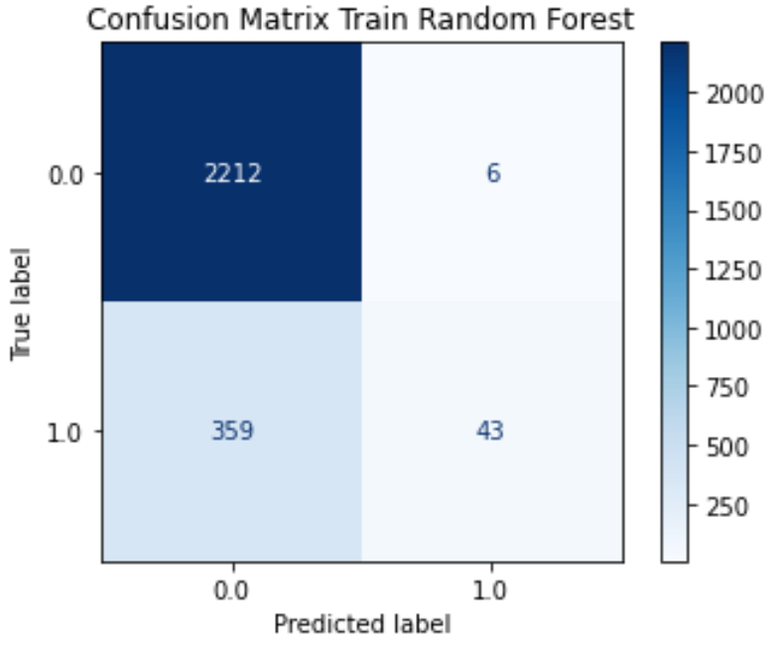
B6 Classification by 3 Nearest Neighbors



B7 svm confusion matrix

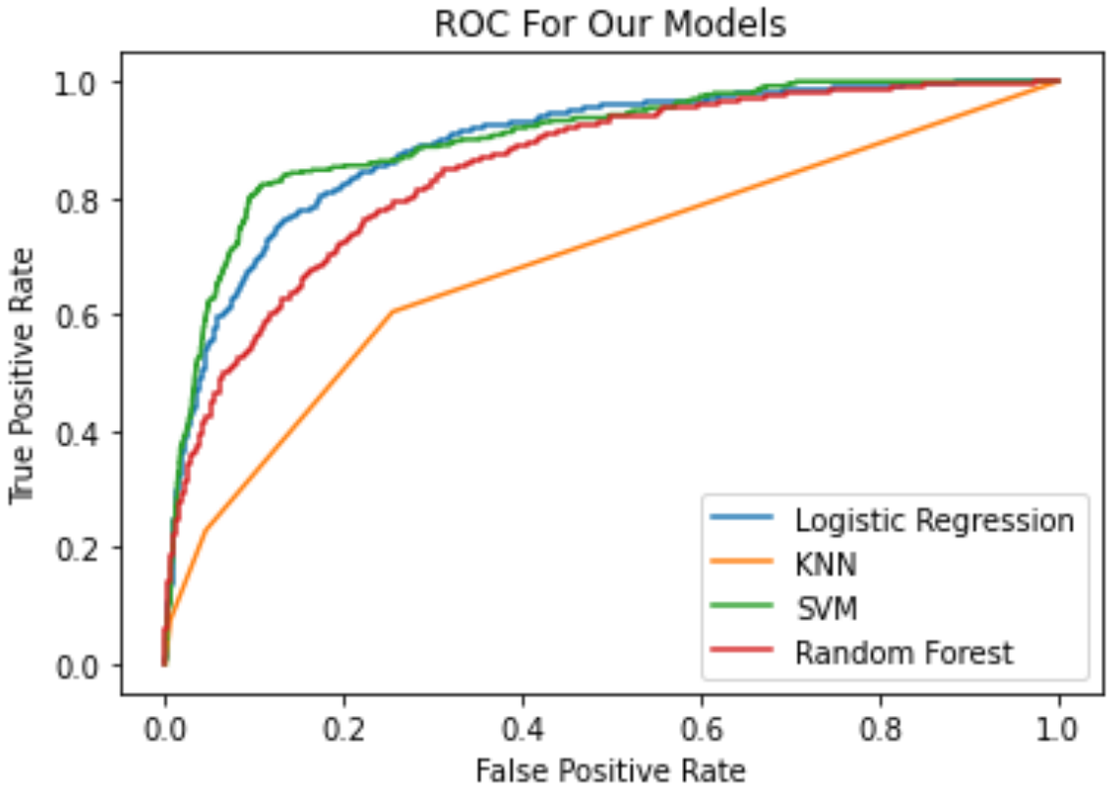


B8 RF confusion matrix

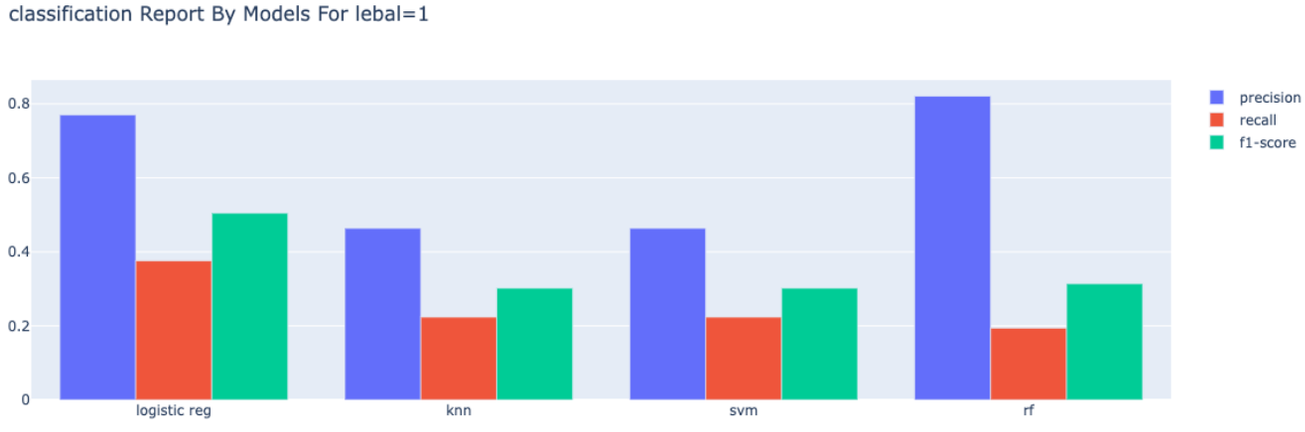


B9

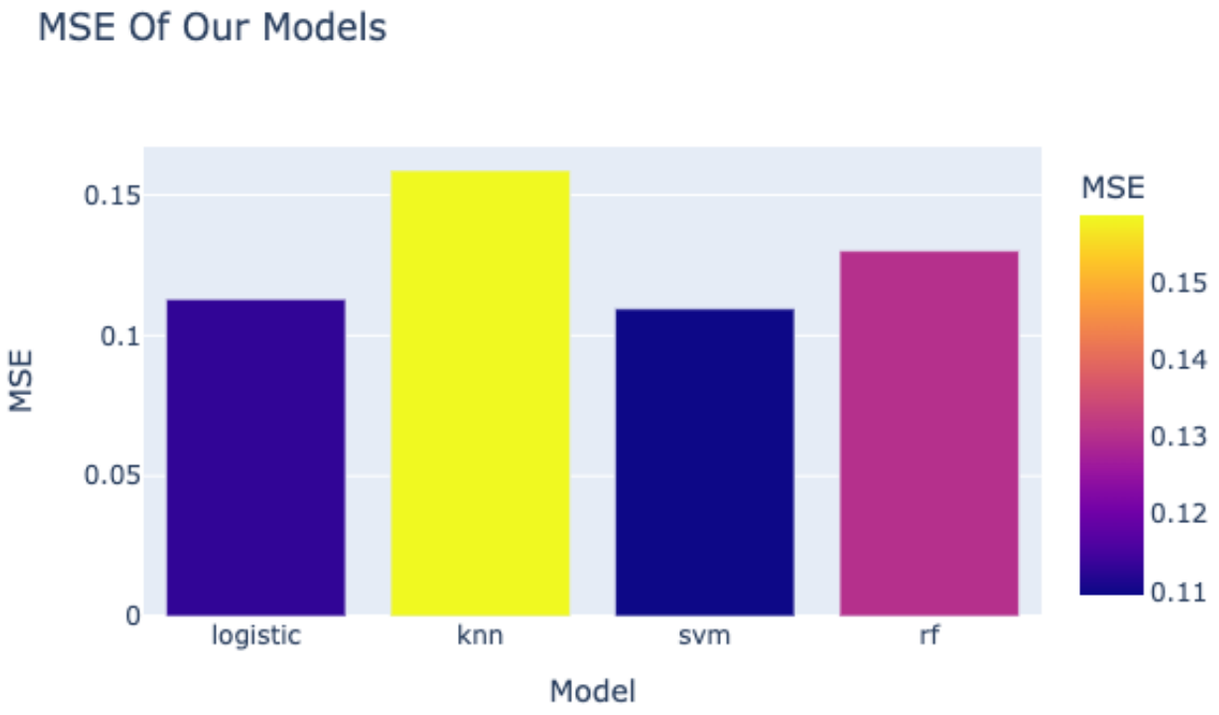
C1 Auc Roc



C2 Classification report by models for label=1



C3 Mse of our models


MSE Of Our Models
0.15
0.1
0.05
0
MSE
logistic
knn
svm
rf
Model
MSE
0.15
0.14
0.13
0.12
0.11