E-SHOP CLOTHING DATA ANALYSIS

FINAL TERM PROJECT

CS 5805: MACHINE LEARNING 1

PRADYUMNA
KOMBETHOTA RAMGOPAL

INTRODUCTION

- This project utilizes the E-Shop Clothing dataset, containing clickstream data from online stores catering to pregnant women.
- Our aim is to apply a range of machine-learning algorithms to this real-world dataset.
- Initially, feature engineering and exploratory data analysis techniques like PCA, SVD, VIF, Condition Number, and Random Forest Analysis were employed to reduce dimensions and address collinearity
- In the subsequent phase, stepwise regression was applied to eliminate irrelevant features based on a predefined threshold and make predictions on the dataset's continuous variable.
- Phase three involved utilizing various machine learning classifiers to forecast the dependent variable, ultimately recommending the best-performing classifier.
- Finally, clustering and association rule mining algorithms were employed to unveil valuable trends and patterns within the dataset.

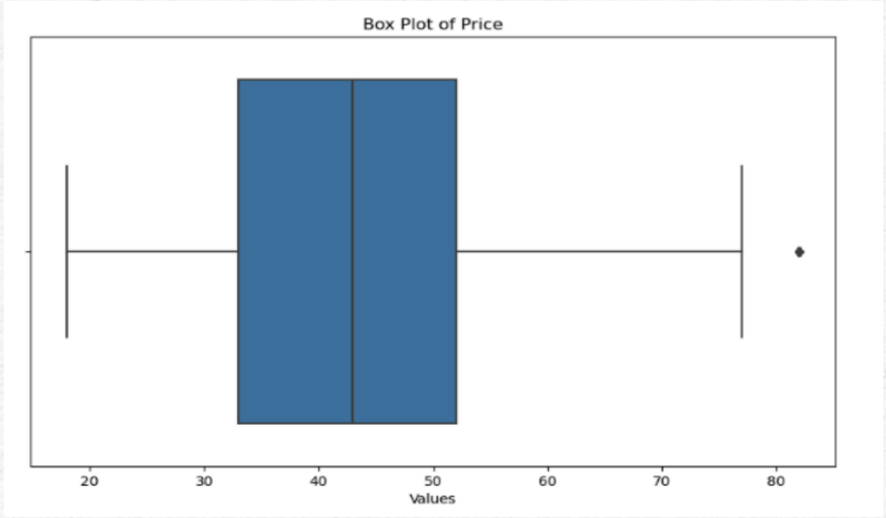
DATASET DESCRIPTION

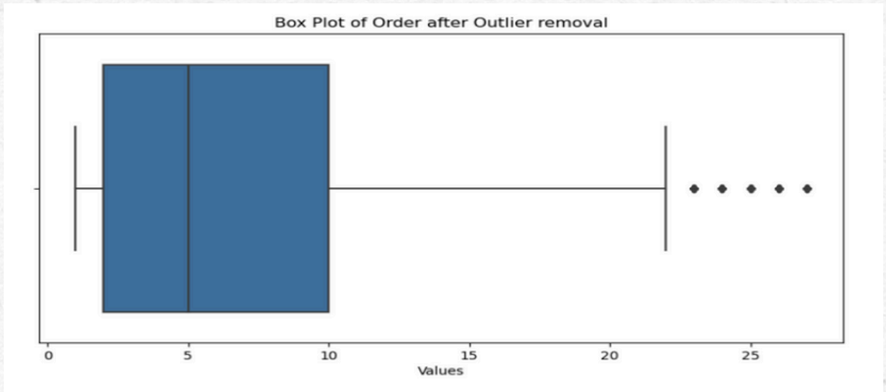
- The dataset is multivariate and sequential.
- It contains both numerical and categorical values with a total of 165,474 observations.
- It has fourteen features namely year, month, day, order, country, session id, page one, page two, color, location, model photography, price one, price two, and page.

FEATURE ENGINEERING

- The chosen dataset has no missing or nan values.
- No duplicated values in the dataset.
- Data aggregated for necessary operation.
- Label Encoded to convert categorical variables to numerical variables.
- Removed the outlier using IQR method.

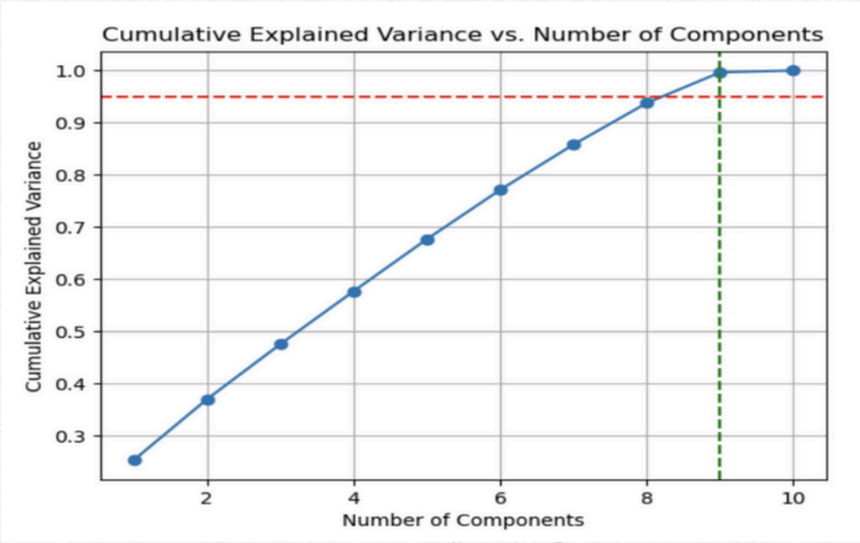
```
The missing values of the dataframe is:
year                          0
month                         0
day                           0
order                         0
country                       0
session ID                    0
page 1 (main category)        0
page 2 (clothing model)       0
colour                        0
location                      0
model photography             0
price                         0
price 2                       0
page                          0
dtype: int64
```





DIMENSIONALITY REDUCTION

- Performed PCA and condition number on the standardized dataset.
- The condition number drastically dropped to 8.4101 for numerical and 8.631 for categorical target.
- Singular value decomposition of the matrix was calculated.

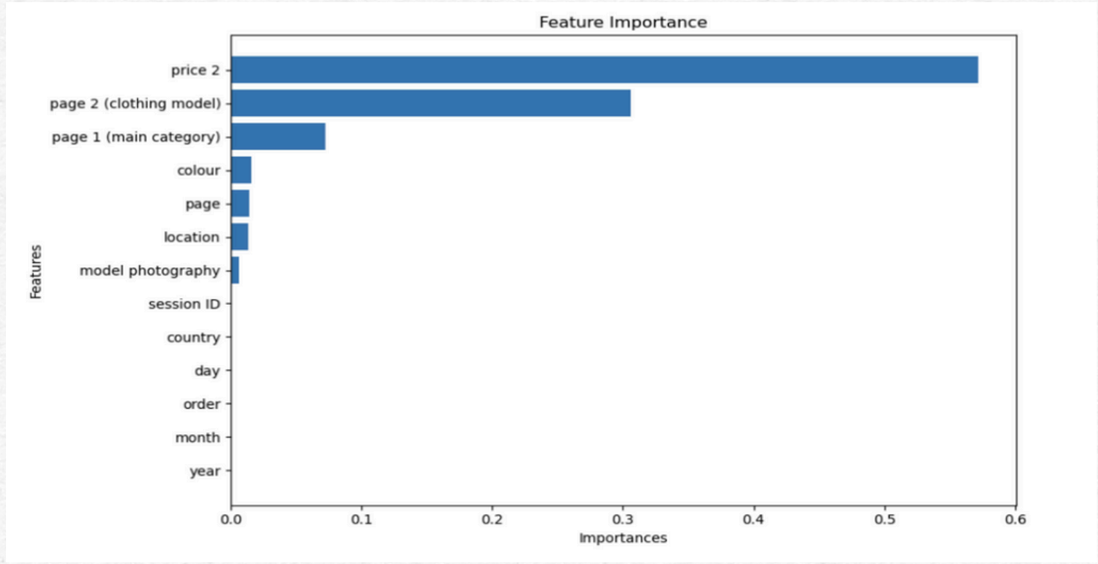



```
The Explained Variance Ratio is: [0.25309012 0.11679154 0.10552205 0.1015245  0.09977816 0.09458505
 0.08683162 0.0796398  0.05865891 0.00357825]
```

```
The SVD values of the features are: [637.71325566 410.31105655 404.46865482 390.41732552 383.54083877
 374.63021365 368.50208612 333.18073177 300.53871522  73.87814832]
```

RANDOM FOREST ANALYSIS

- We set the threshold of 0.001
- Seven features are important according to random forest analysis for numerical target.
- Four features are important for the categorical target.




```
The selected features are: ['price 2', 'page 2 (clothing model)', 'page 1 (main category)', 'colour', 'page', 'location', 'model photography']
```

```
The selected features are: ['price', 'page 1 (main category)', 'page 2 (clothing model)', 'colour']
```

VARIANCE INFLATION FACTOR

- Variance Inflation Factor, is a statistical measure used to assess multicollinearity in regression analysis.
- We found that year, session id and month have very high VIF.
- Threshold of five to drop the features which are contributing more to multicollinearity

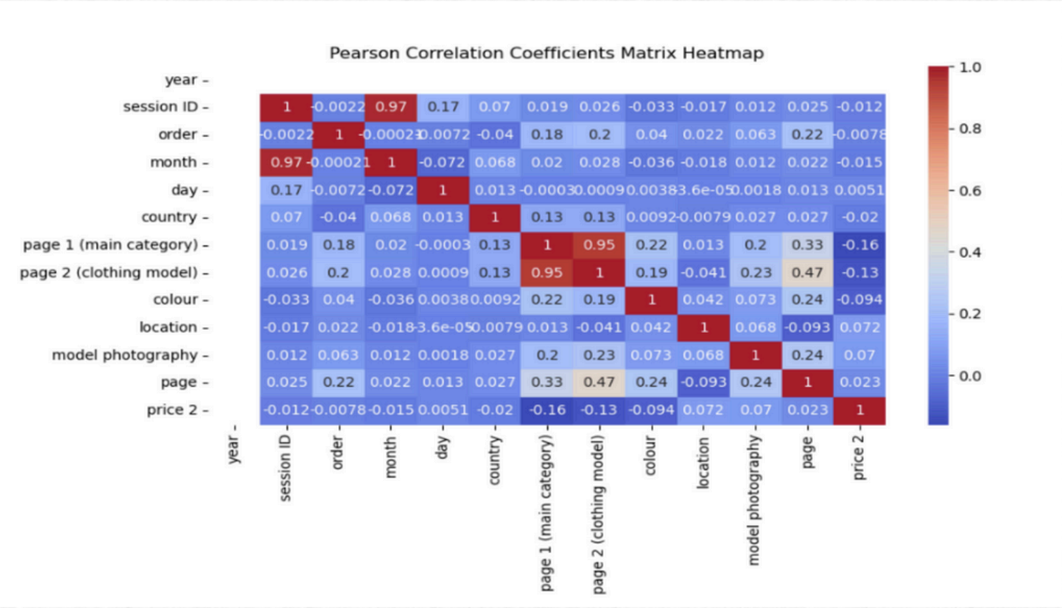
The VIFs are:

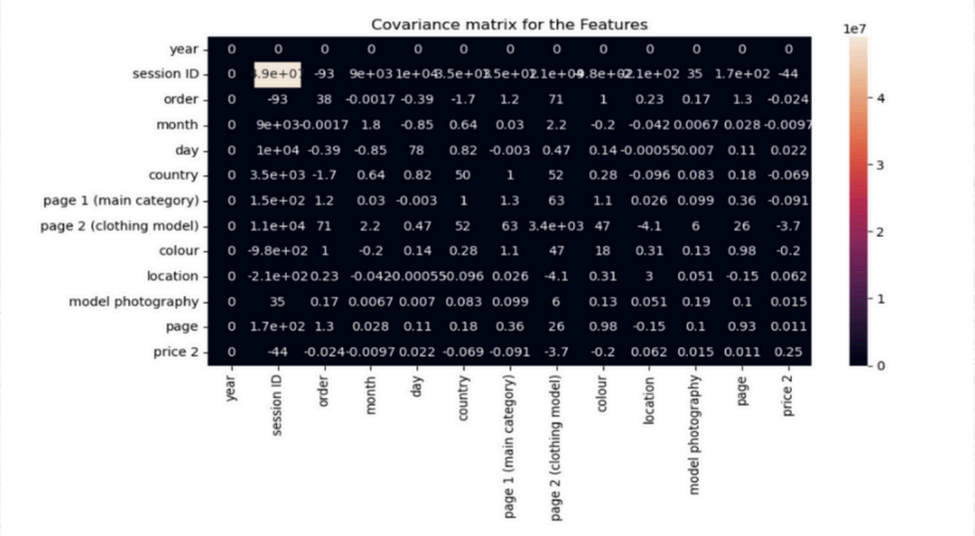
	Feature	VIF
0	year	3984.194388
1	session ID	489.122123
2	order	1.072189
3	month	478.467384
4	day	28.367786
5	country	1.026899
6	page 1 (main category)	13.301469
7	page 2 (clothing model)	14.832735
8	colour	1.144969
9	location	1.050649
10	model photography	1.102100
11	page	1.726990
12	price 2	1.057023

```
The features selected by VIF are:
['order', 'country', 'colour', 'location', 'model photography', 'page', 'price 2']
```

COVARIANCE MATRIX AND PEARSON CORRELATION COEFFICIENT

- Plotted the covariance matrix and pearson correlation coefficient for the dataset to understand the relation among features.





REGRESSION ANALYSIS

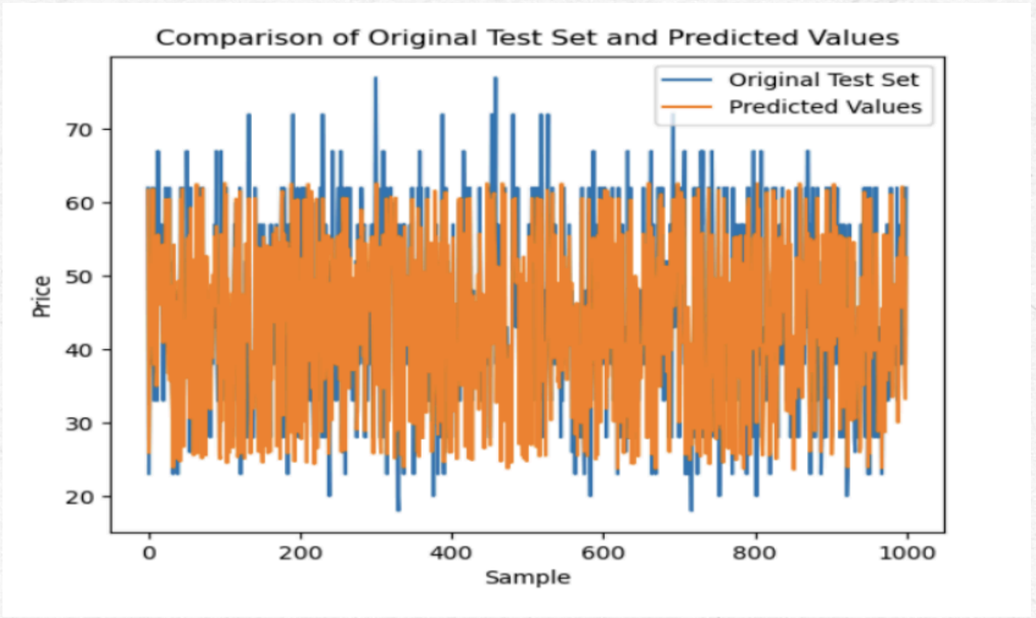
- We employed the backward stepwise regression model to predict a continuous numerical feature.
- We used a threshold of 0.01 and removed the features having p value greater than the threshold value

```
Backward Stepwise Regression Results:

  Eliminated Feature   P-value            AIC            BIC  \
0              order  0.833332  151589.932130  151696.733412
1           location  0.267314  151587.976415  151685.068489
2                day  0.244109  151587.207002  151674.589869

   Adjusted R-squared
0            0.796531
1            0.796532
2            0.796532
```

```
The final equation is:  (0.0053366507154653945 * month) + (0.004444628588451841 * country) + (-0.47042444828400876 * page 1 (main category)) + (-0.054593628007116216 *
 colour) + (-0.06527332401265237 * model photography) + (0.07627254278806014 * page) + (-0.8339717815755833 * price 2)
```

RESULTS

- Plot of the price target variable almost follows the same pattern which exists in the actual test set.
- Confidence interval for the coefficients were also calculated.



month	0.002793	0.007881
country	0.001883	0.007006
page 1 (main category)	-0.473228	-0.467621
colour	-0.057248	-0.051939
model photography	-0.067921	-0.062626
page	0.073482	0.079063
price 2	-0.836567	-0.831376

F-STATISTIC, T-STATISTIC, AND P VALUE

- A high f-statistic value confirms the usefulness and validity of the regression analysis.
- A high t-statistic value indicates that the coefficient of the corresponding predictor variable is statistically significant.
- P-value of zero, it typically indicates that the corresponding coefficient is highly statistically significant.

F-statistic:	6.805e+04

t	P>|t|
2.38e-13	1.000
4.112	0.000
3.401	0.001
-328.854	0.000
-40.306	0.000
-48.323	0.000
53.570	0.000
-629.770	0.000

CLASSIFICATION ANALYSIS

PRE-PRUNED DECISION TREE

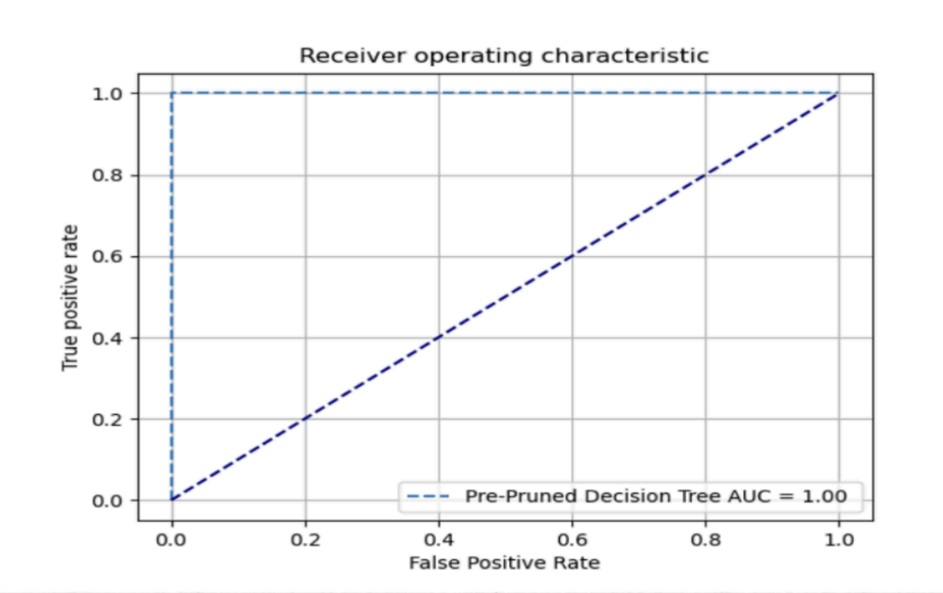
- Decision tree classifier on the data to predict the target with the hyper parameters max depth, min samples split, min sample leaf, max features, splitter, and criterion.
- GridSearCV which helps us to pick the best parameter for the model.

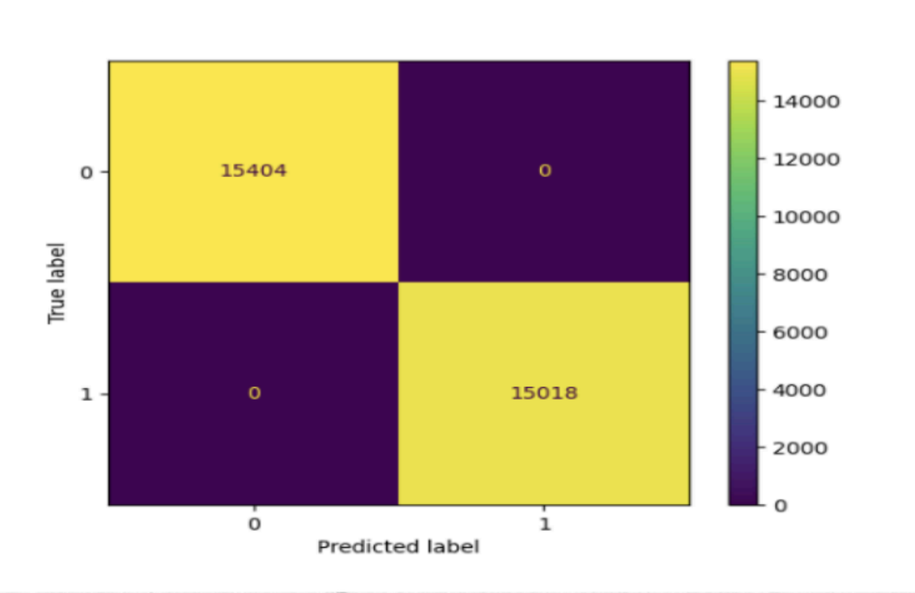
```
The accuracy of the pre-pruned model is:  1.0

The confusion matrix is:
 [[15404     0]
 [    0 15018]]

The precision of the pre-pruned model is:  1.0

The recall of the pre-pruned model 1.0

The specificity of the pre-pruned model is:  1.0

The f1 score of the pre-pruned model is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 0.9999671290513444
Accuracy: 0.9999342581026889
Accuracy: 0.998356452567221
Accuracy: 1.0
```

```
Best parameters set found on development {'criterion': 'gini', 'max_depth': 12, 'max_features': 3, 'min_samples_leaf': 1, 'min_samples_split': 4, 'splitter': 'best'}
```

ROC CURVE AND CONFUSION MATRIX

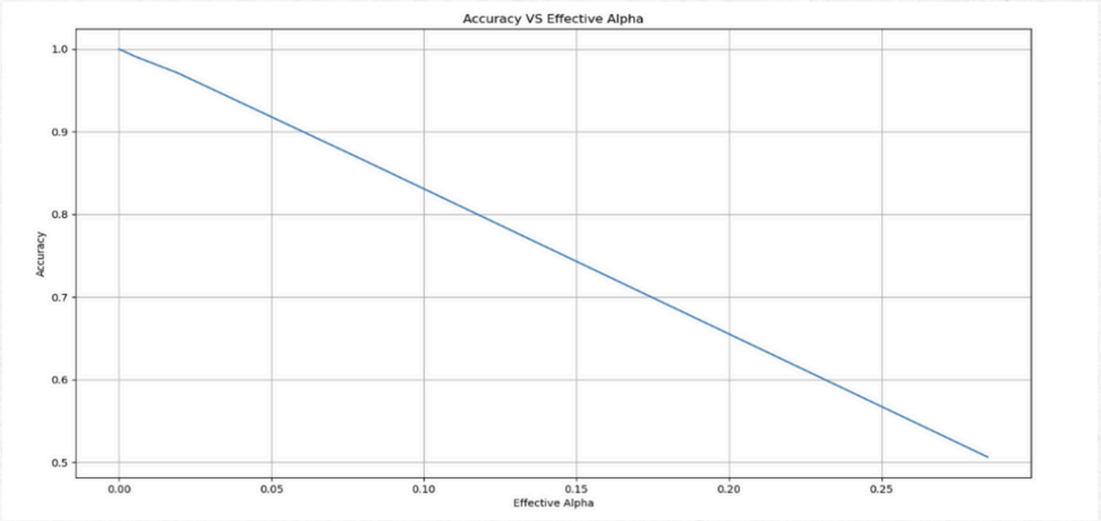
- The AUC of the pre pruned decision tree found to be 1.00.





POST-PRUNED DECISION TREE

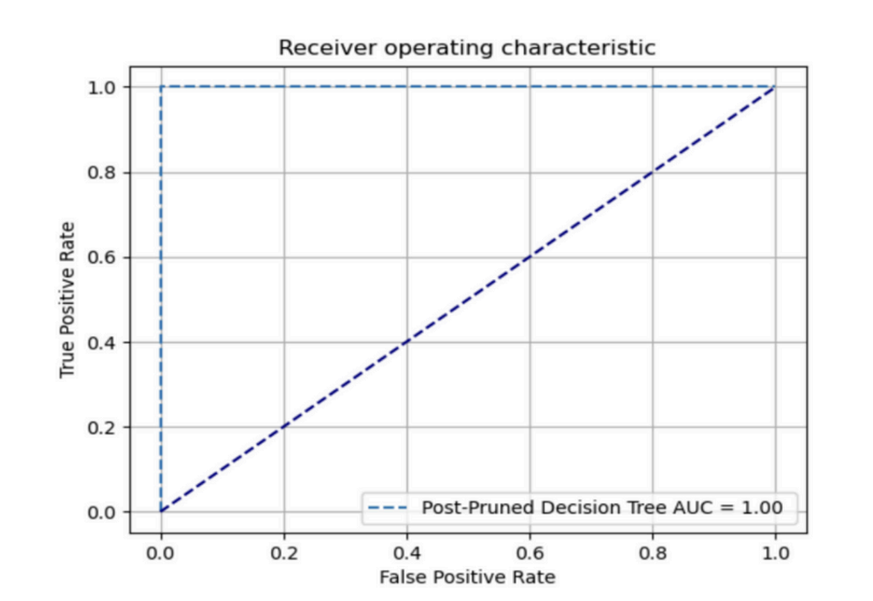
- In this approach we use cost complexity pruning path to determine the best value for alpha.

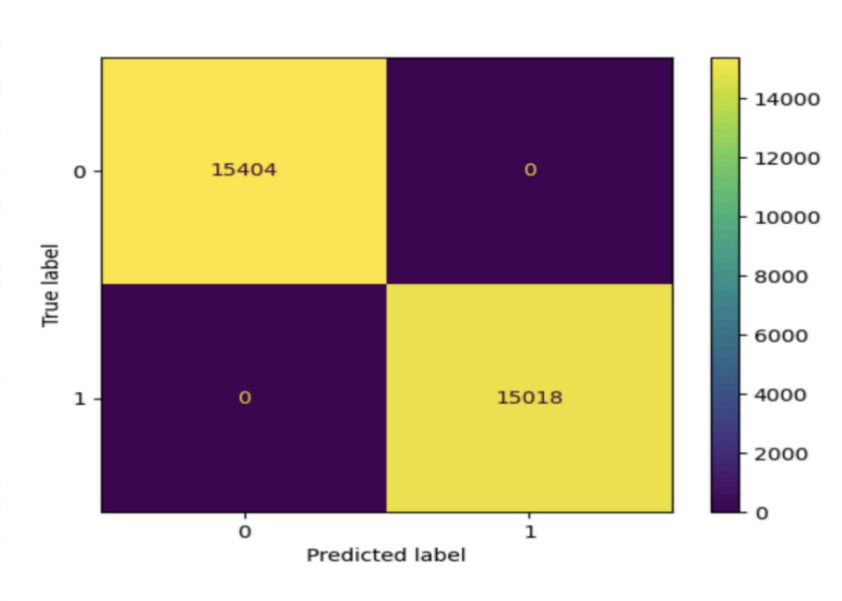



```
The accuracy of the post-pruned model is:  1.0

The confusion matrix is:
[[15404     0]
[    0 15018]]

The precision score of the post-pruned model is:  1.0

The recall score of the post-pruned model is:  1.0

The specificity of the post-pruned model is:  1.0

The f1 score of the post-pruned model is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- The AUC of the post pruned decision tree found to be 1.00.





LOGISTIC REGRESSION

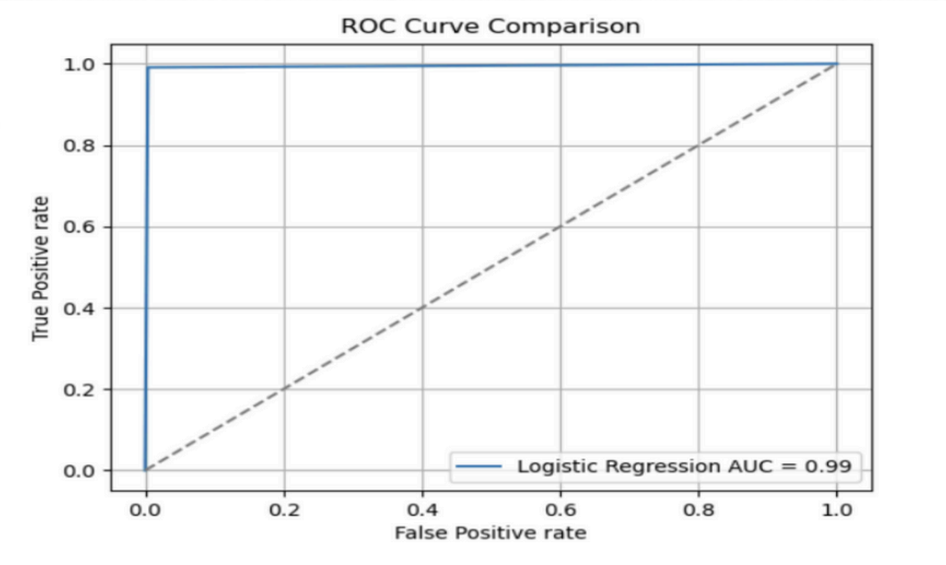
- We ran the logistic regression on the data to predict the target with the hyper parameters C and penalty

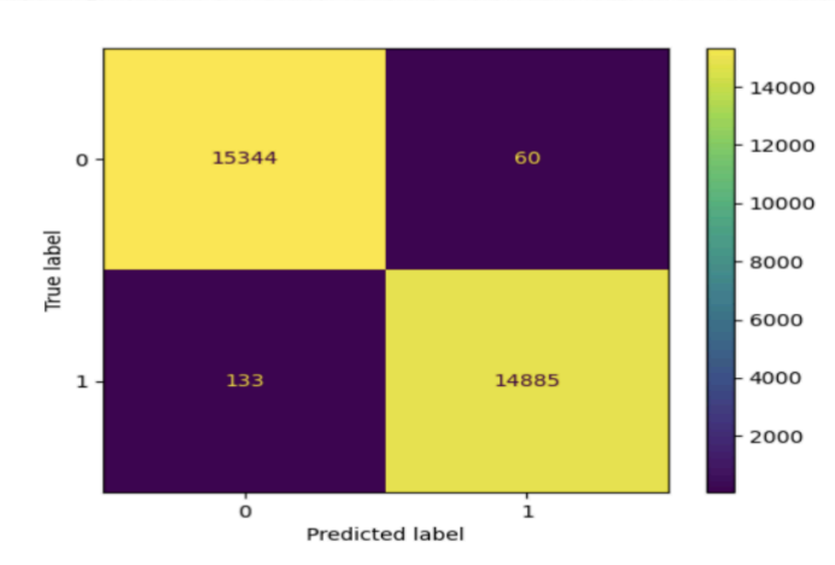
```
Best Parameters for the Logistic Regression: {'C': 100, 'penalty': 'l2'}
```

```
The test accuracy of the Logistic Regression model:  0.9936559069094734

The confusion matrix is:
 [[15344    60]
 [  133 14885]]

The precision score of the Logistic Regression model:  0.9959852793576447

 The recall score of the Logistic Regression model:  0.9911439605806366

The specificity of the Logistic Regression model:  0.9961049078161517

The f1 Score of the Logistic Regression model:  0.9935587224243234

The accuracy for each fold is:
Accuracy: 0.993064229833673
Accuracy: 0.9933600683715732
Accuracy: 0.992439681809217
Accuracy: 0.993688777858129
Accuracy: 0.9931954899575951
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the logistic regression model to be 0.9936
- AUC of the logistic regression is 0.99





K-NEAREST NEIGHBORS

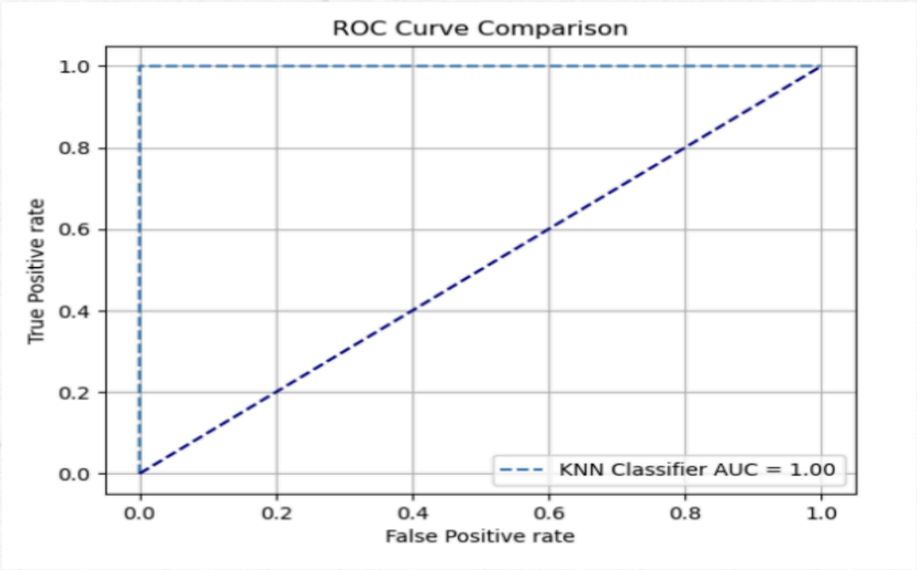
- We ran the K-Nearest Neighbors on the data to predict the target with the hyper parameters n_neighbors, p, weights, and algorithm.

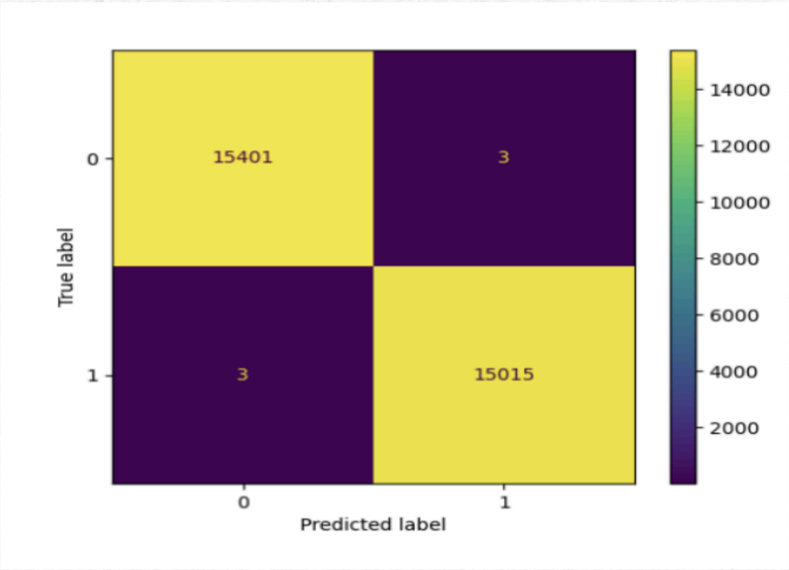
```
Best Parameters for the KNN classifier is:  {'algorithm': 'auto', 'n_neighbors': 9, 'p': 1, 'weights': 'distance'}
```

```
The accuracy of the KNN classifier model is:  0.9998027743080665

The confusion matrix of the KNN classifier is :
 [[15401     3]
 [    3 15015]]

The precision of the KNN classifier is :  0.9998002397123452

The recall of the KNN classifier is :  0.9998002397123452

The specificity of the KNN classifier is :  0.9998052453908076

The f1 score of the KNN classifier is :  0.9998002397123452

The accuracy for each fold is:
Accuracy: 0.999769903359411
Accuracy: 0.9998027743080665
Accuracy: 0.9999342581026889
Accuracy: 0.9997041614620998
Accuracy: 0.9997041517372868
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the KNN classifier to be 0.9980.
- We found AUC of the KNN classifier to be one.

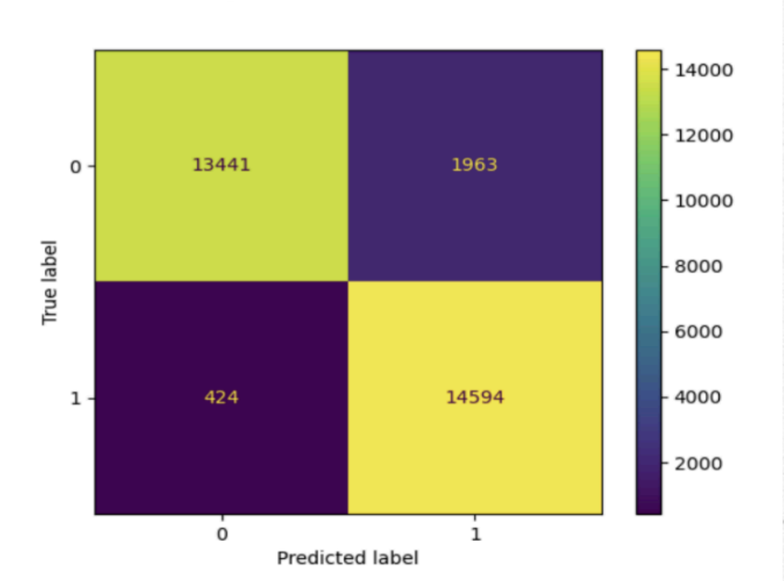


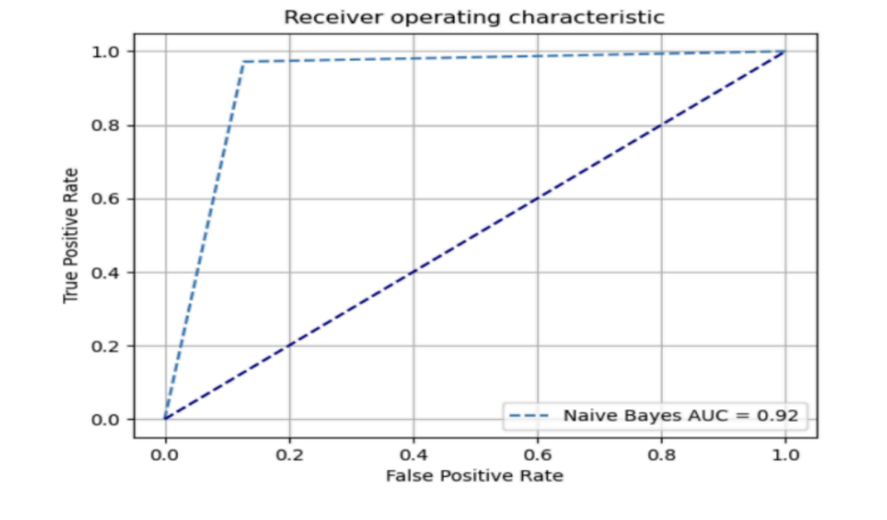


NAIVE BAYES CLASSIFIER

- We didn’t use GridSearchCV as the classifier doesn’t take any parameter values.
- The accuracy of the classifier is 0.925

```
The test accuracy of the Naive Bayes model:  0.9215370455591348

The confusion matrix of the Naive Bayes:
 [[13441  1963]
 [  424 14594]]

The precision score of the Naive Bayes: 0.8814398743733768

The recall score of the Naive Bayes: 0.9717672126781196

The specificity of the Naive Bayes:  0.8725655673850948

The f1 score of the Naive Bayes:  0.9244022169437845

The accuracy for each fold is:
Accuracy: 0.9212412070212347
Accuracy: 0.9230162382486359
Accuracy: 0.9196305305371113
Accuracy: 0.9249227532706594
Accuracy: 0.9216988264685579
```





SUPPORT VECTOR MACHINE (LINEAR)

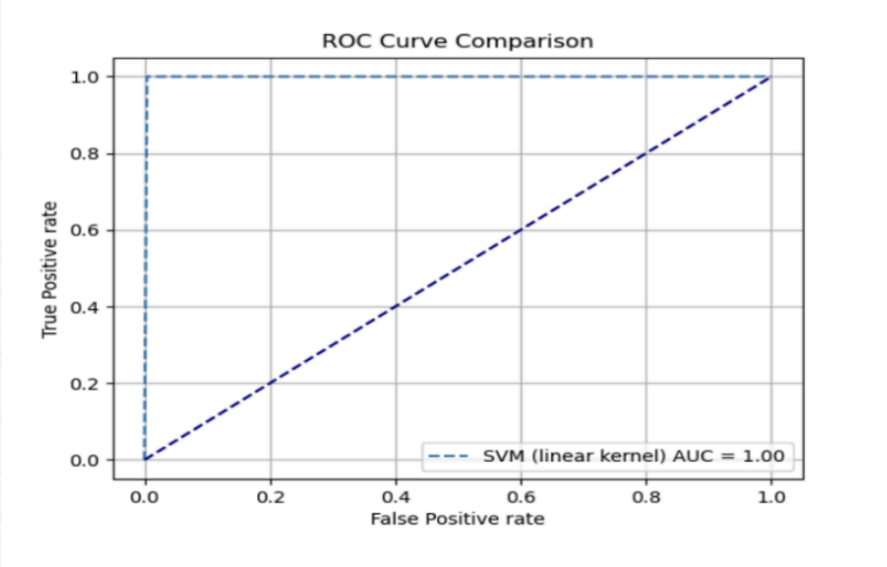
- We trained SVM with the linear kernel on the dataset with the parameters C and gamma.

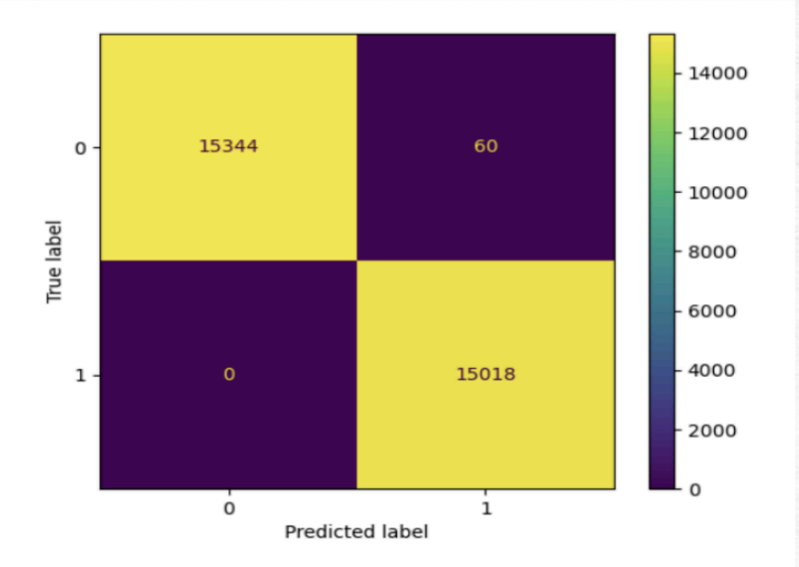
```
The best parameter for SVM (linear kernel) is:  {'C': 1, 'gamma': 'scale'}
```

```
The best parameter for SVM (linear kernel) is:  {'C': 1, 'gamma': 'scale'}

The test accuracy of SVM (linear kernel) is:  0.9980277430806653

The confusion matrix of SVM (linear kernel) is:
 [[15344    60]
 [    0 15018]]

The precision of SVM (linear kernel) is:  0.9960206923995225

The recall of SVM (linear kernel) is:  1.0

The specificity of SVM (linear kernel) is:  0.9961049078161517

The f1 score of SVM (linear kernel) is:  0.998006379585327

The accuracy for each fold is:
Accuracy: 0.9980934849779765
Accuracy: 0.9982907106699099
Accuracy: 0.9983893235158766
Accuracy: 0.9981920978239432
Accuracy: 0.9983235265112915
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the SVM (linear) to be 0.9980.





SUPPORT VECTOR MACHINE (POLYNOMIAL)

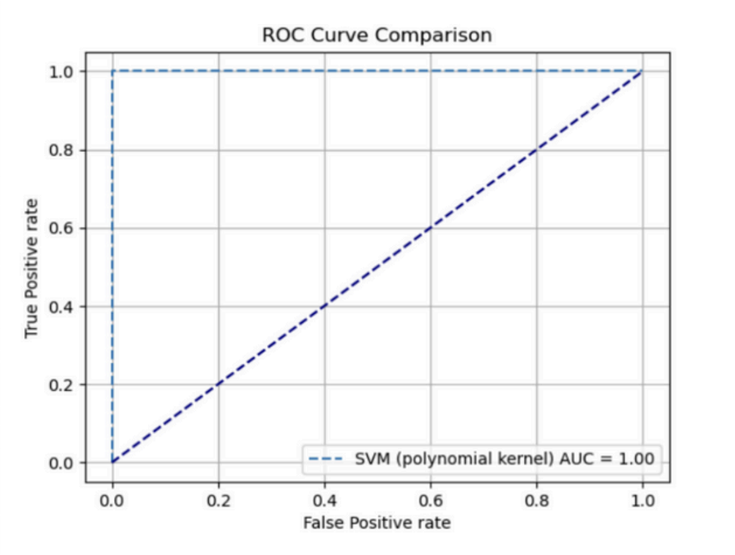
- We trained SVM with the polynomial kernel on the dataset with the parameters C, degree, and gamma.

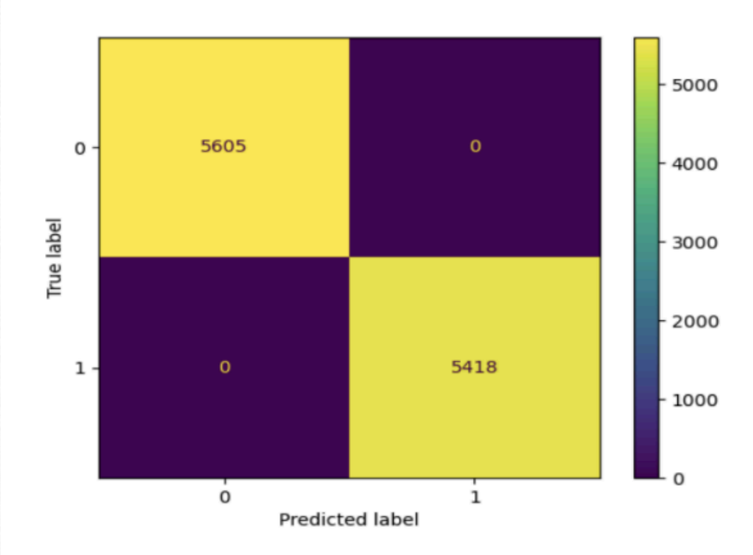
```
The best parameter for SVM (polynomial kernel) is:  {'C': 10, 'degree': 3, 'gamma': 'auto'}
```

```
The test accuracy of SVM (polynomial kernel) is:  1.0

The confusion matrix of SVM (polynomial kernel) is:
 [[5605    0]
 [   0 5418]]

The precision of SVM (polynomial kernel) is:  1.0

The recall of SVM (polynomial kernel) is:  1.0

The specificity of SVM (polynomial kernel) is:  1.0

The f1 score of SVM (polynomial kernel) is:  1.0

The accuracy for each fold is:
Accuracy: 0.9999092805951193
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the SVM (polynomial) to be 1.0.





SUPPORT VECTOR MACHINE (RBF)

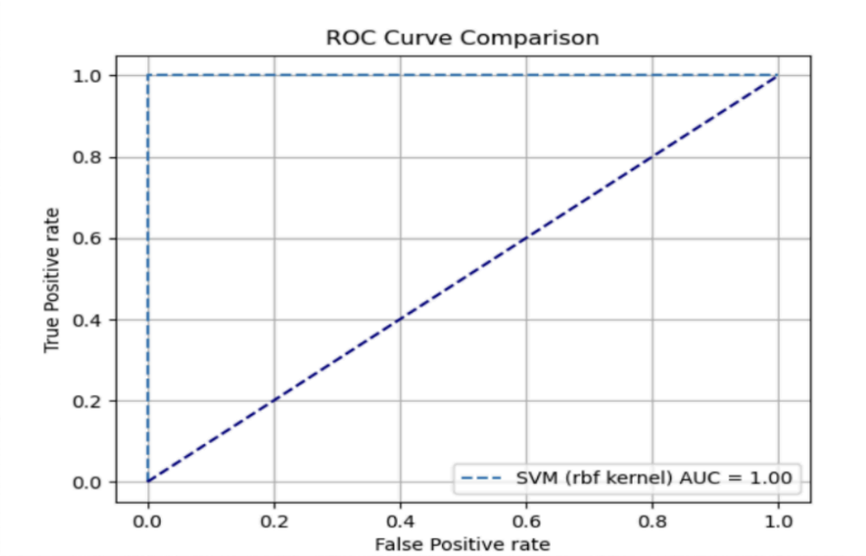
- We trained SVM with the RBF kernel on the dataset with the parameters C, and gamma.

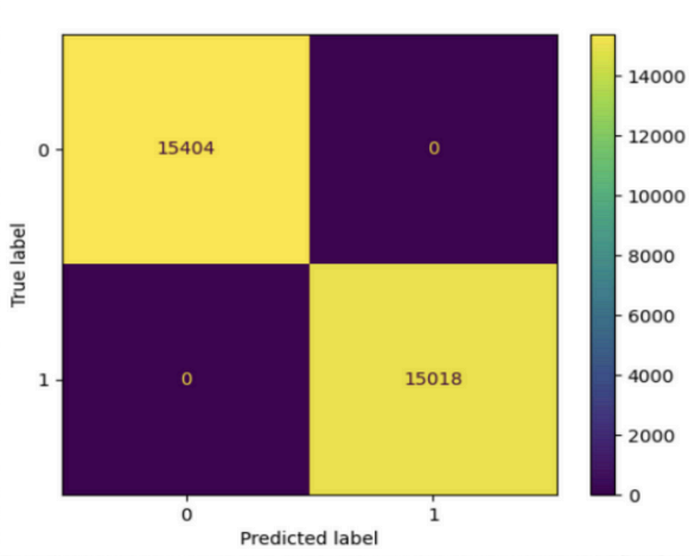
```
The best parameter for SVM (rbf kernel) is:  {'C': 10, 'gamma': 'scale'}
```

```
The test accuracy of SVM (rbf kernel) is:  1.0

The confusion matrix of SVM (rbf kernel) is:
 [[15404     0]
 [    0 15018]]

The precision of SVM (rbf kernel) is:  1.0

The recall of SVM (rbf kernel) is:  1.0

The specificity of SVM (rbf kernel) is:  1.0

The f1 score of SVM (rbf kernel) is:  1.0

The accuracy for each fold is:
Accuracy: 0.9999671290513444
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the SVM (RBF) to be 1.0.





RANDOM FOREST CLASSIFIER (BAGGING)

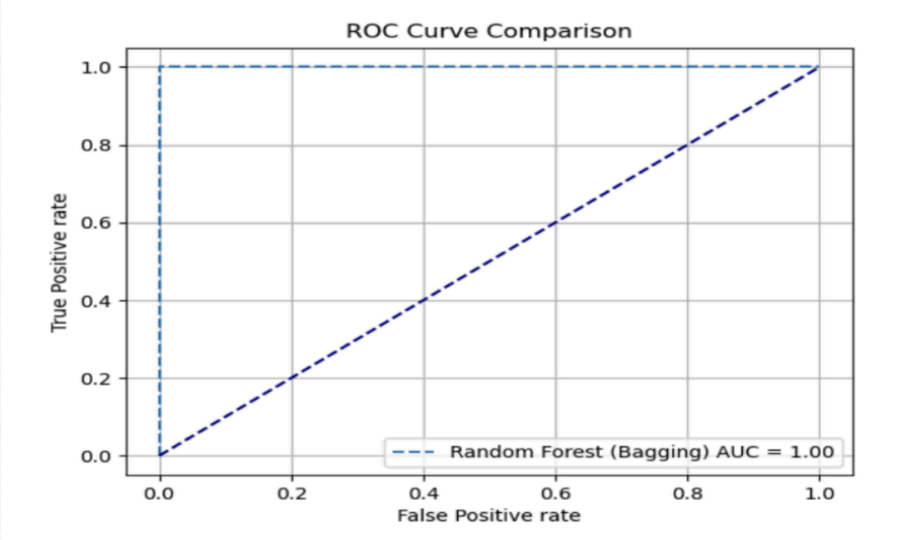
- We ran a combined Bagging classifier and Random Forest classifier to build our model with the parameters n_estimators, max samples, and max features.

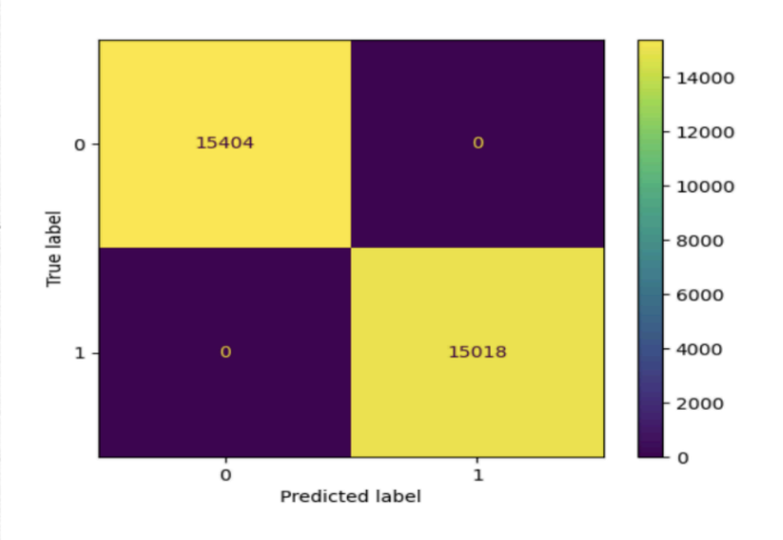
```
The best parameter for Random Forest (Bagging) is:  {'max_features': 8, 'max_samples': 0.5, 'n_estimators': 5}
```

```
The test accuracy of Random Forest (Bagging) is:  1.0

The confusion matrix of Random Forest (Bagging) is:
 [[15404     0]
 [    0 15018]]

The precision of Random Forest (Bagging) is:  1.0

The recall of Random Forest (Bagging) is:  1.0

The specificity of Random Forest (Bagging) is:  1.0

The f1 score of Random Forest (Bagging) is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the random forest classifier (Bagging) classifier to be 1.0





RANDOM FOREST CLASSIFIER (BOOSTING)

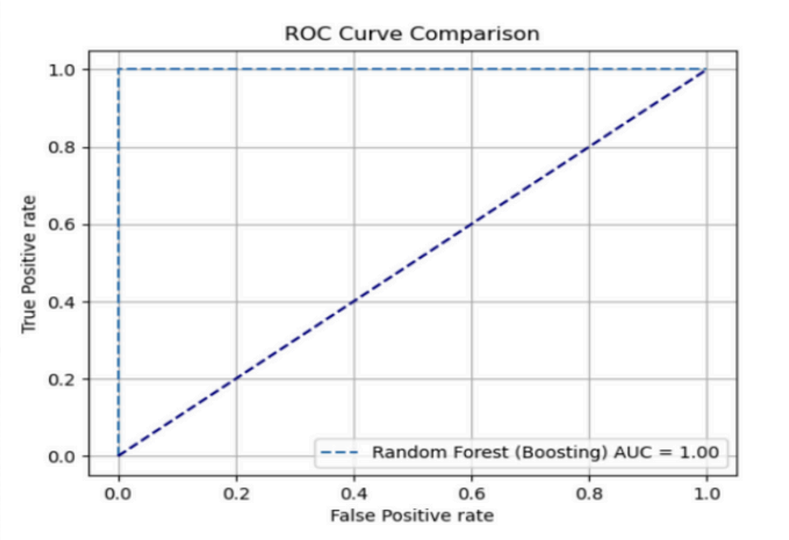
- We ran a random forest classifier (boosting) using AdaBoost classifier to build our model with the parameters n_estimators, and learning rate.

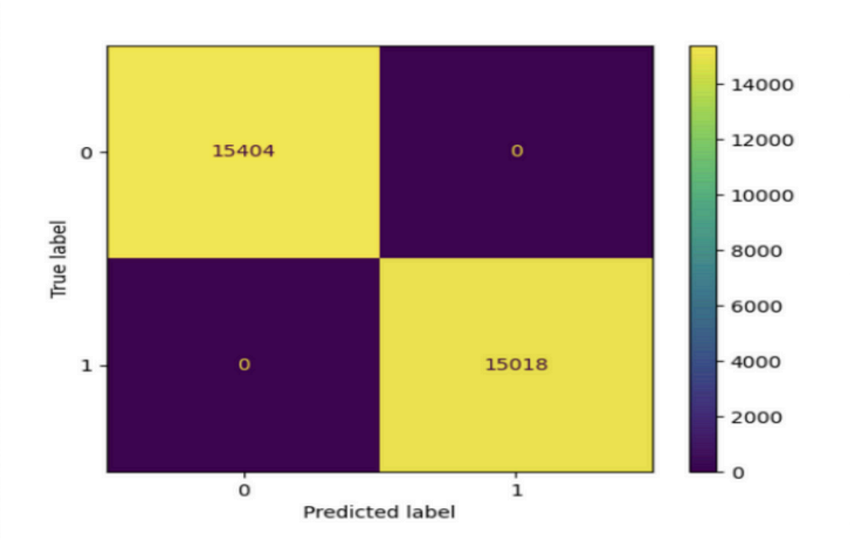
```
The best parameter for Random Forest (Boosting) is:  {'base_estimator__n_estimators': 5, 'learning_rate': 0.1, 'n_estimators': 5}
```

```
The test accuracy of Random Forest (Boosting) is:  1.0

The confusion matrix of Random Forest (Boosting) is:
 [[15404     0]
 [    0 15018]]

The precision of Random Forest (Boosting) is:  1.0

The recall of Random Forest (Boosting) is:  1.0

The specificity of Random Forest (Boosting) is:  1.0

The f1 score of Random Forest (Boosting) is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the random forest classifier (Boosting) classifier to be 1.0





RANDOM FOREST CLASSIFIER (STACKING)

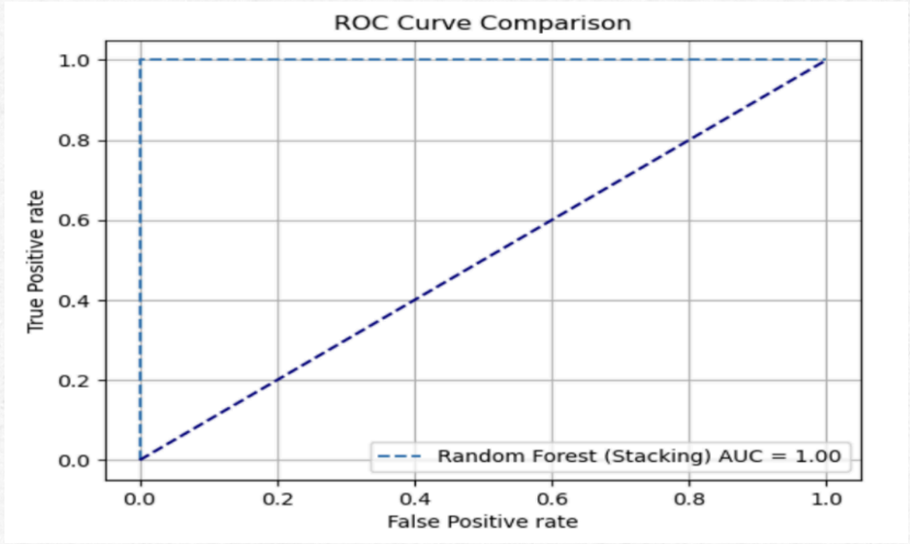
- We ran a random forest classifier (stacking) using GaussianNB and Logistic Regression to build our model with the parameters n_estimators, max features, and max samples.

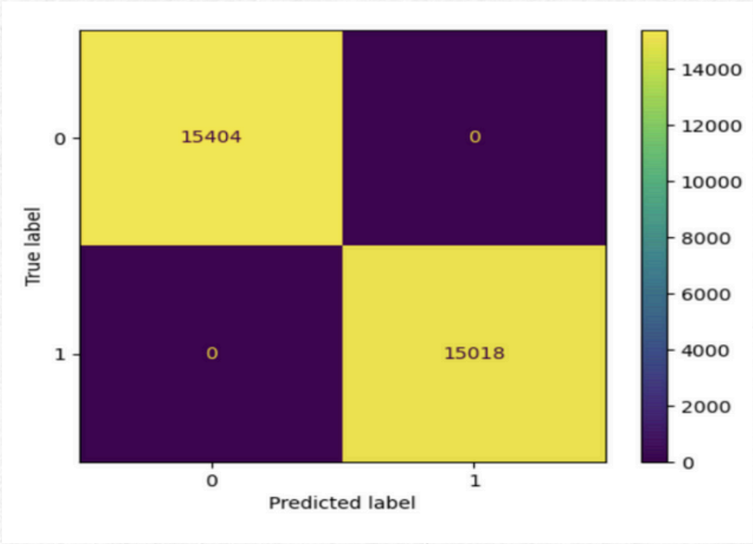
```
The best parameter for Random Forest (Stacking) is:  {'final_estimator__max_features': 2, 'final_estimator__max_samples': 0.7, 'final_estimator__n_estimators': 5}
```

```
The test accuracy of Random Forest (Stacking) is:  1.0

The confusion matrix of Random Forest (Stacking) is:
 [[15404     0]
 [    0 15018]]

The precision of Random Forest (Stacking) is:  1.0

The recall of Random Forest (Stacking) is:  1.0

The specificity of Random Forest (Stacking) is:  1.0

The f1 score of Random Forest (Stacking) is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 0.9999671290513444
Accuracy: 1.0
Accuracy: 0.9999671290513444
Accuracy: 0.9999671279708097
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the random forest classifier (Stacking) classifier to be 1.0





NEURAL NETWORKS - MULTI LAYER PERCEPTRON

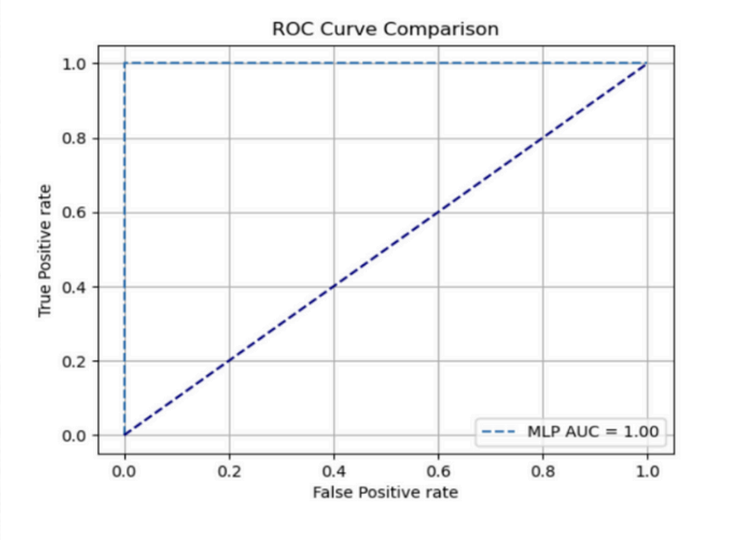
- We built a multi layer perceptron model with the parameters activation function and alpha.

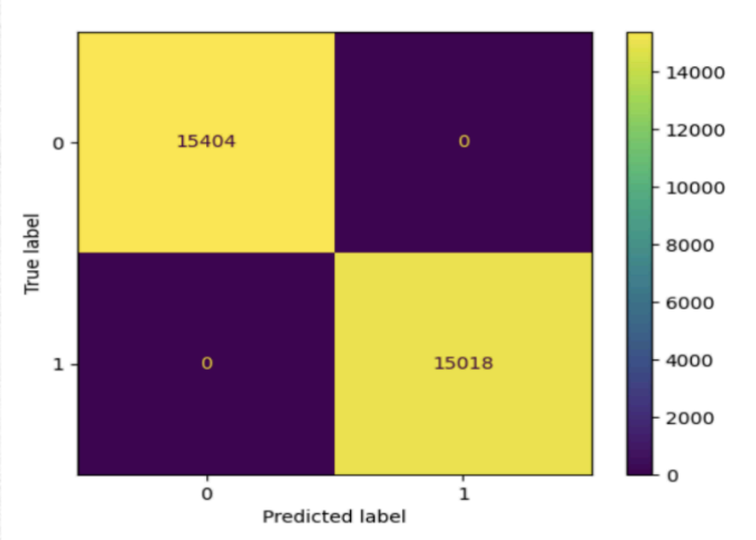
```
The best parameter for MLP is:  {'activation': 'logistic', 'alpha': 0.0001}
```

```
The test accuracy of MLP is:  1.0

The confusion matrix of MLP is:
 [[15404     0]
 [    0 15018]]

The precision of MLP is:  1.0

The recall of MLP is:  1.0

The specificity of MLP is:  1.0

The f1 score of MLP is:  1.0

The accuracy for each fold is:
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
Accuracy: 1.0
```

ROC CURVE AND CONFUSION MATRIX

- We found that accuracy of the multi layer perceptron to be 1.0





OVERVIEW OF CLASSIFIER PERFORMANCE

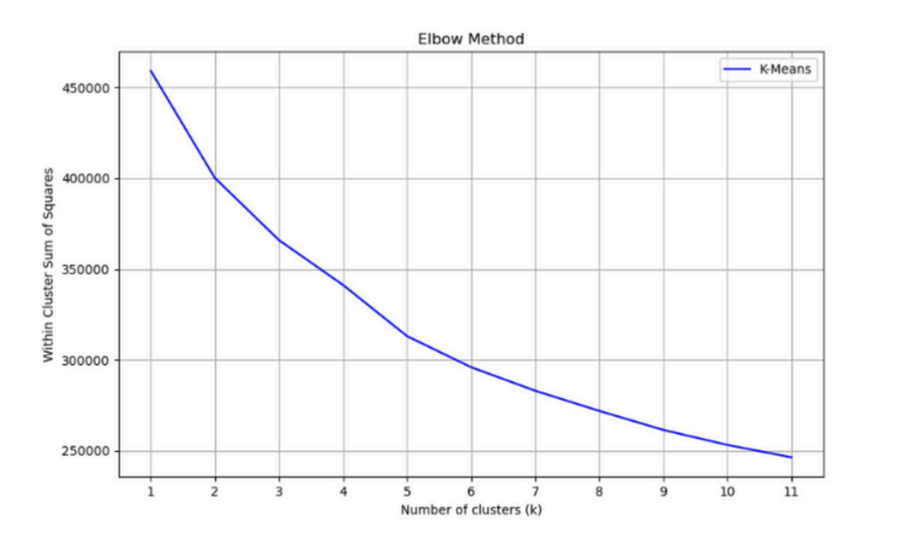
- We recommend Decision Tree (Post Pruned), Random Forest (Boosting), or MLP can be used for the given task.

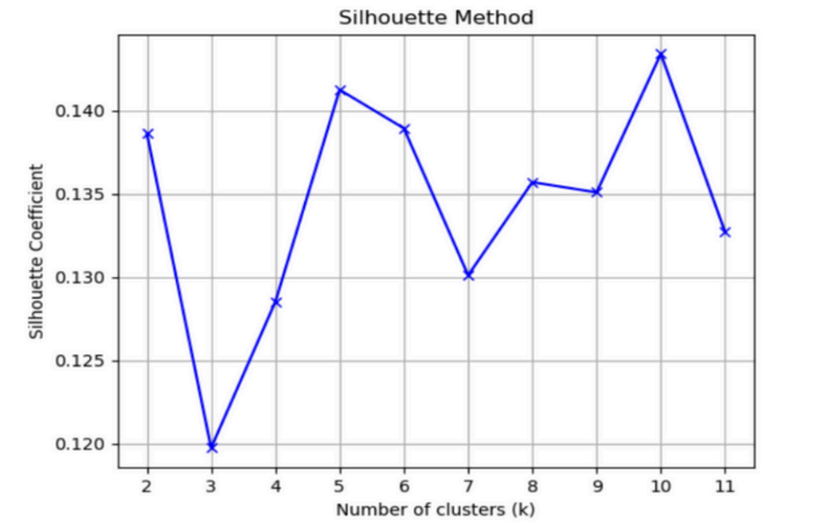
Classifier	Accuracy
Decision Tree (Pre Pruned)	1.0
Decision Tree (Post Pruned)	1.0
Logistic Regression	0.99
K-Nearest Neighbors	0.99
Naive Bayes Classifier	0.92
Support Vector Machine (Linear)	0.99
Support Vector Machine (Polynomial)	1.0
Support Vector Machine (RBF)	1.0
Random Forest Classifier (Bagging)	1.0
Random Forest Classifier (Boosting)	1.0
Random Forest Classifier (Stacking)	1.0
Multi Layer Perceptron	1.0

CLUSTERING AND ASSOCIATION RULE MINING

K-MEANS CLUSTERING

- K-means clustering can provide valuable insights into the structure of your data, uncover hidden patterns.





APRIORI ALGORITHM

- We mine association rules from a dataset using a specified confidence threshold (0.6) and the resulting association rules are then sorted based on confidence and lift.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
9	(sale)	(Poland)	0.234158	0.809571	0.218064	0.931272	1.150327	0.028497	2.770763
8	(profile)	(Poland)	0.260071	0.809571	0.215967	0.830417	1.025749	0.005421	1.122924
11	(trousers)	(en-face)	0.300603	0.739929	0.245670	0.817257	1.104507	0.023245	1.423152
2	(Page1)	(Poland)	0.564753	0.809571	0.453231	0.802530	0.991302	-0.003977	0.964341
7	(en-face)	(Poland)	0.739929	0.809571	0.593604	0.802244	0.990950	-0.005421	0.962950
0	(April)	(Poland)	0.291278	0.809571	0.233112	0.800307	0.988557	-0.002698	0.953608
13	(en-face, Page1)	(Poland)	0.447581	0.809571	0.356122	0.795660	0.982817	-0.006226	0.931923
4	(Page1)	(en-face)	0.564753	0.739929	0.447581	0.792525	1.071082	0.029704	1.253503
12	(Poland, Page1)	(en-face)	0.453231	0.739929	0.356122	0.785741	1.061914	0.020763	1.213816
10	(trousers)	(Poland)	0.300603	0.809571	0.224476	0.746753	0.922406	-0.018883	0.751949
1	(April)	(en-face)	0.291278	0.739929	0.216209	0.742277	1.003173	0.000684	1.009110
6	(Poland)	(en-face)	0.809571	0.739929	0.593604	0.733232	0.990950	-0.005421	0.974897
5	(trousers)	(Page1)	0.300603	0.564753	0.211181	0.702525	1.243950	0.041415	1.463137
14	(Page1)	(Poland, en-face)	0.564753	0.593604	0.356122	0.630580	1.062292	0.020883	1.100094
3	(en-face)	(Page1)	0.739929	0.564753	0.447581	0.604897	1.071082	0.029704	1.101603

RECOMMENDATIONS

- In the first phase we gained insights into our dataset's structure, relationships, and distributions and learned best practices for cleaning, transforming, and encoding data.
- In the second and third phase we understood the strengths and weaknesses of various regression and classification algorithms.
- In the last phase we gained experience in visualizing data and model results to communicate findings effectively.
- We can improve the performance of the classification by exploring more sophisticated feature engineering techniques to extract additional insights from the dataset.

THANK YOU VERY MUCH!