Exploring Class Imbalance with Fraud Detection

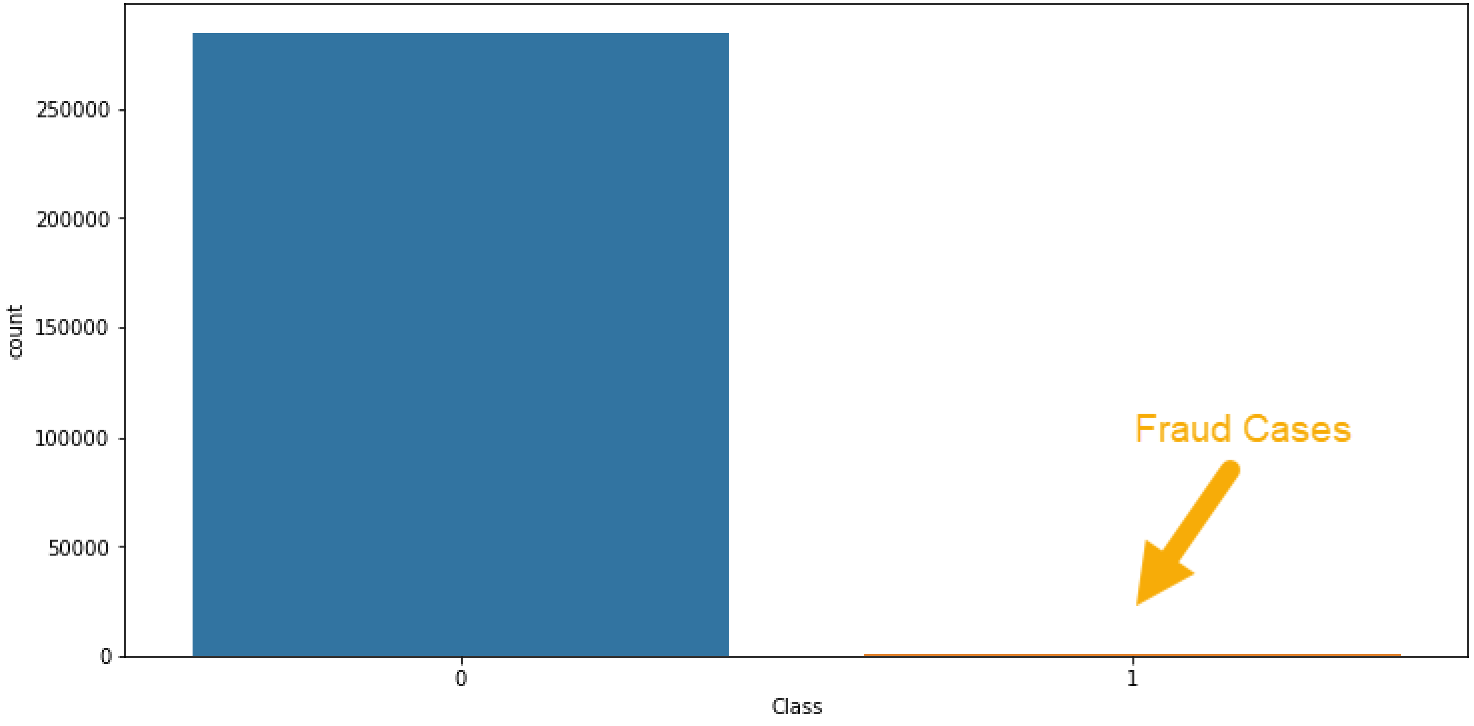
James Bush
Springboard
Capstone Presentation 2

Features

	Time	V1	V2	V3	V4	V5	...	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	...	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	...	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	...	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	...	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	...	-0.206010	0.502292	0.219422	0.215153	69.99	0

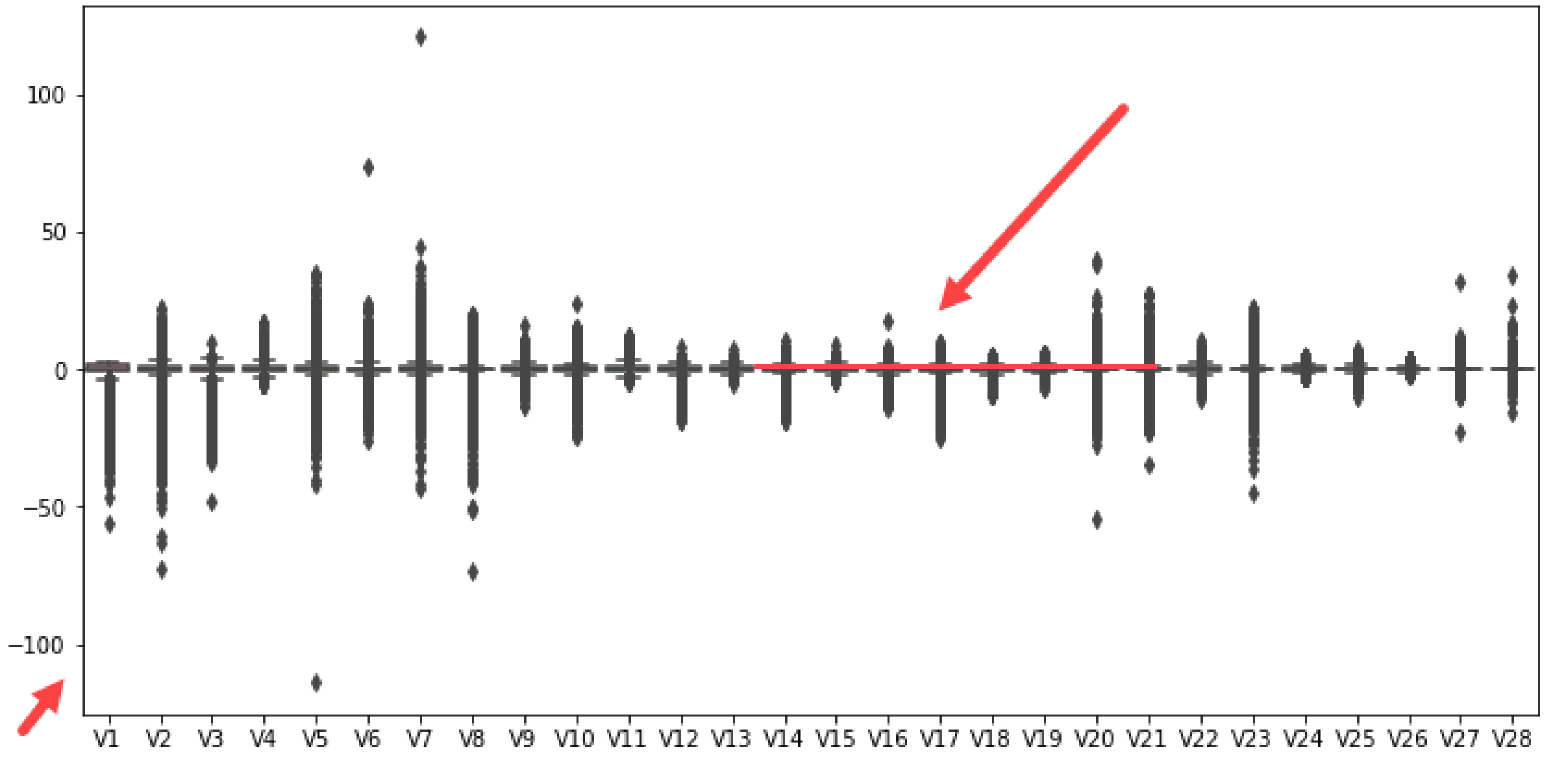
- 30 Independent Variables
- 28 Transformed with PCA (V1-V28)

Transaction Counts



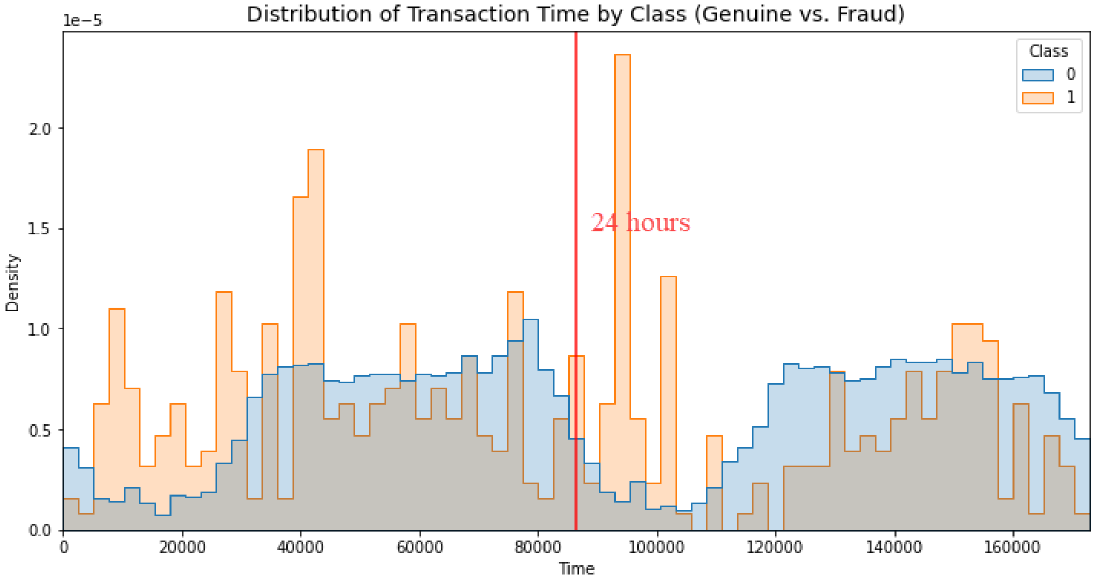
- 284,807 Total Transactions
- 284,315 Majority Class (0)
- 492 Minority Class (1)

Transformed Feature Boxplots

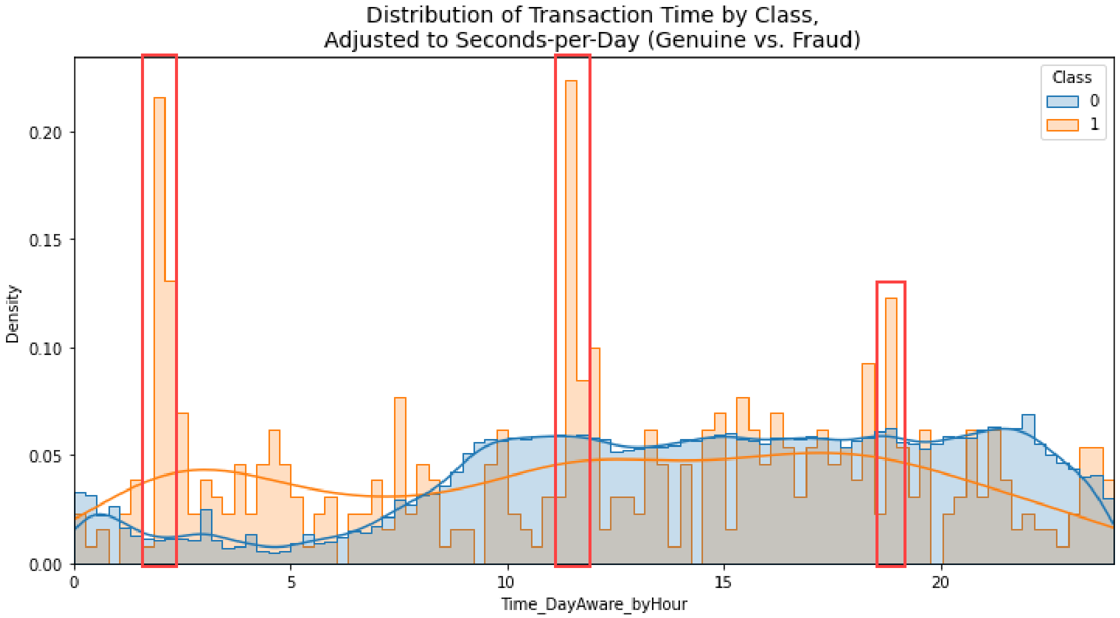


- PCA-transformed features centered on 0
- Values range 150 to -150
- Values have a higher density between 50 to -50

Transaction Time



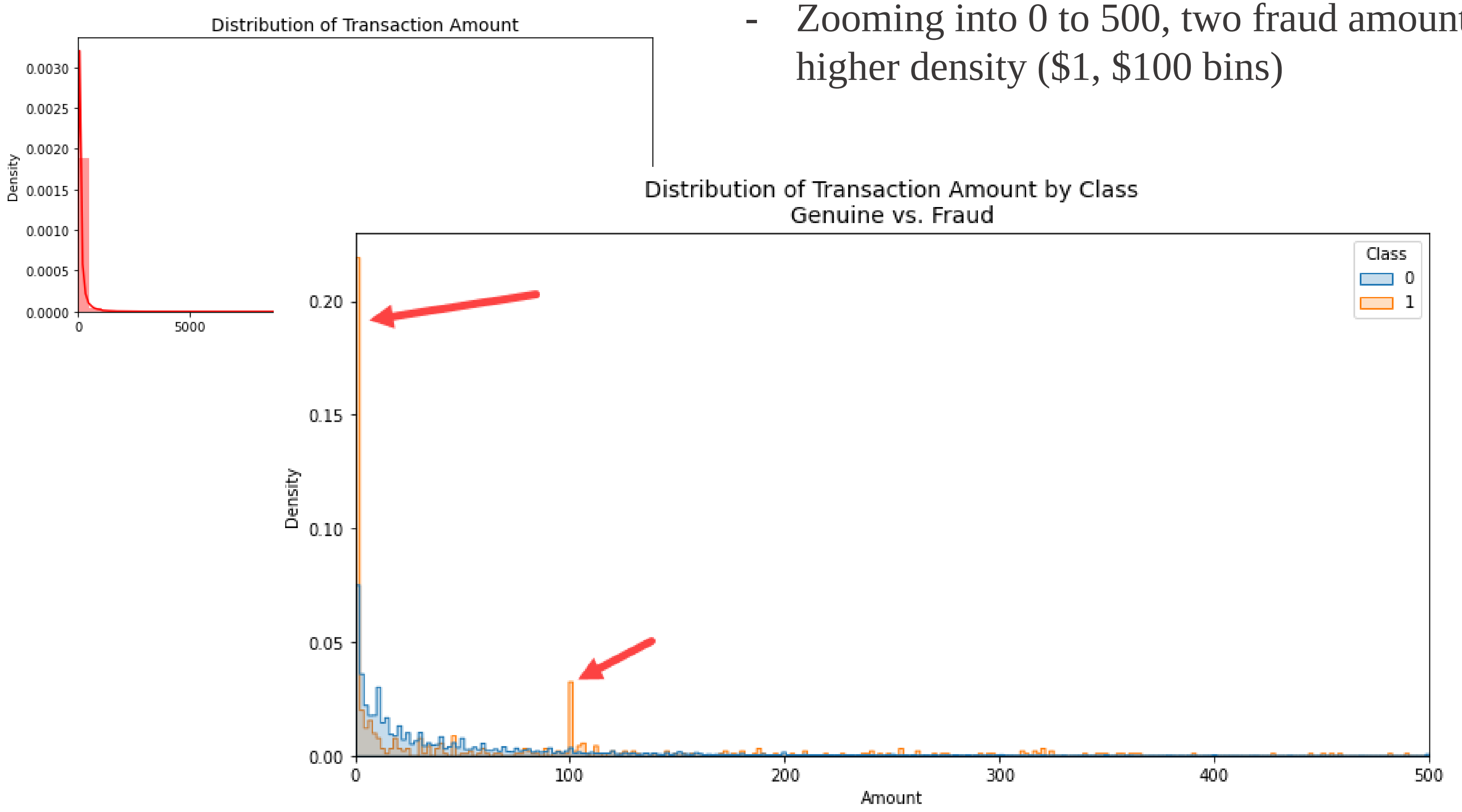
- Transaction Time is given as the number of seconds from transaction 0
- Total time adds up to 48 hours



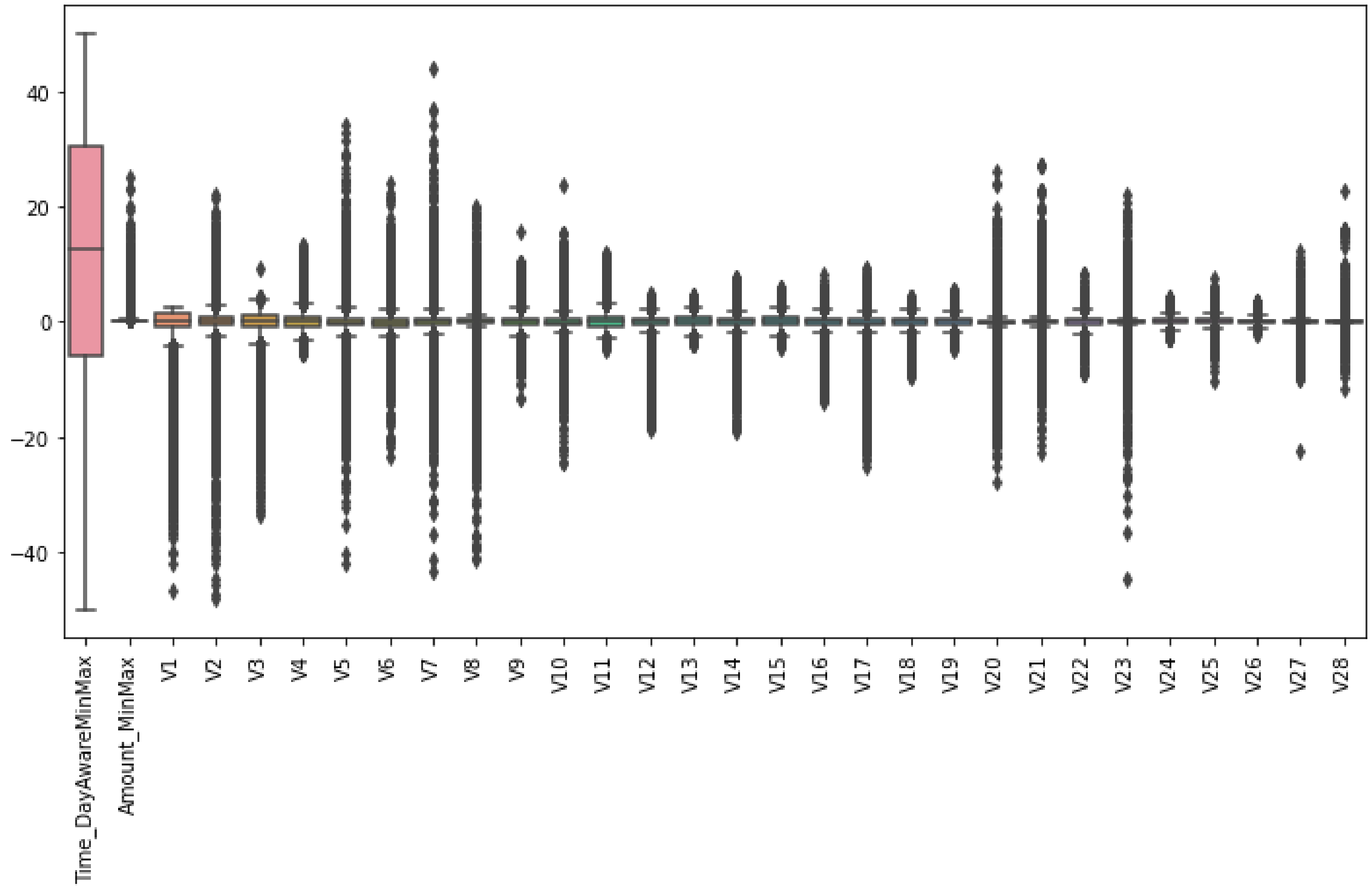
- Time recalculated into 24-hour periods show 3 instances of higher density for fraud class
- Time might not be an excellent feature with only 2 days worth of information

Transaction Amount

- Zooming into 0 to 500, two fraud amounts have a higher density ($1, $100 bins)

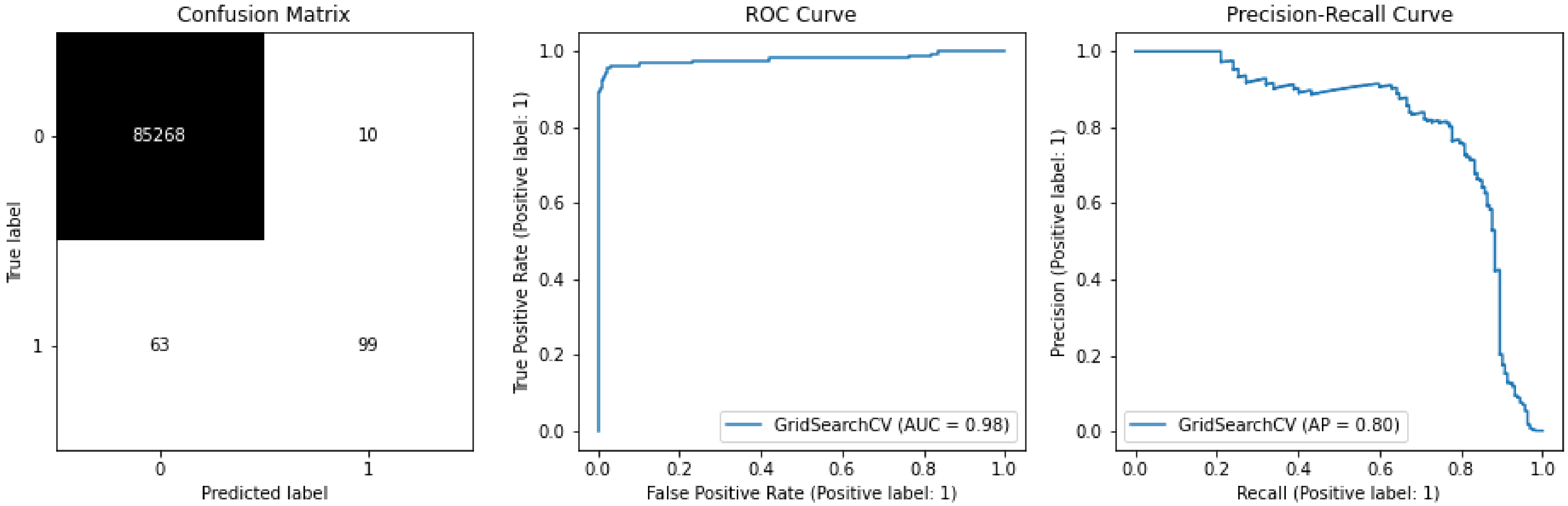


Scaling Time, Amount



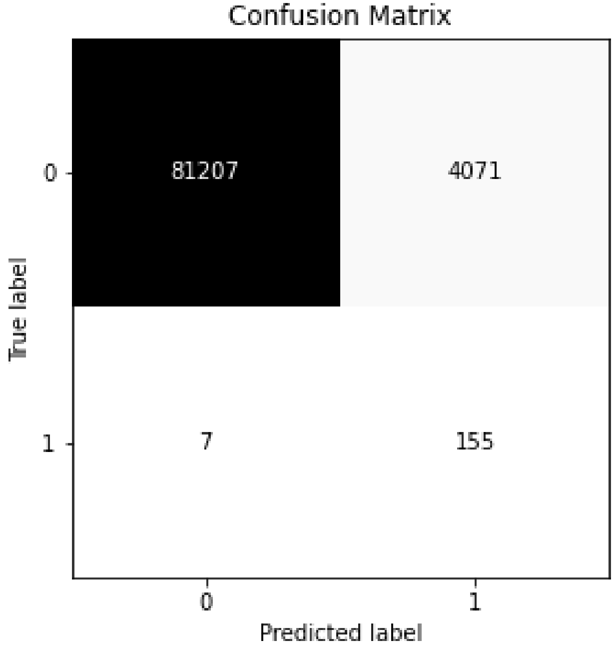
Boxplots show the result after Feature Scaling, Dropping Outliers.

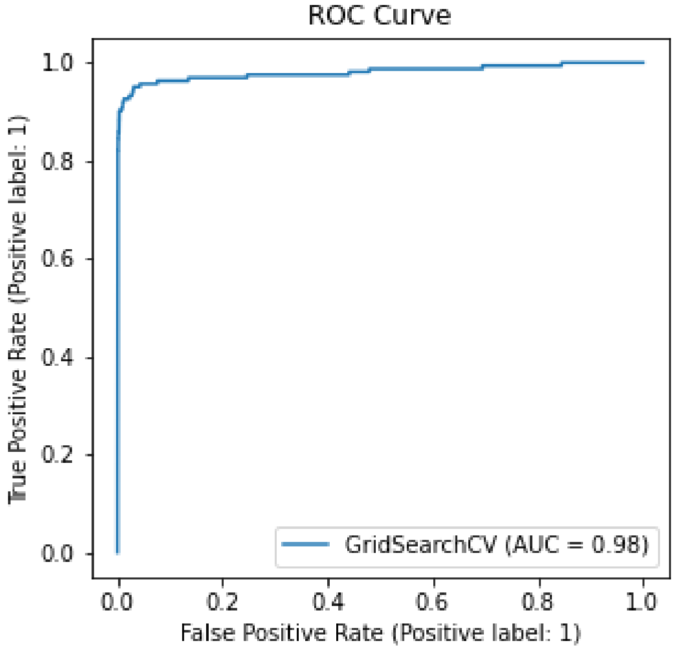
Pre-Sampling LogR

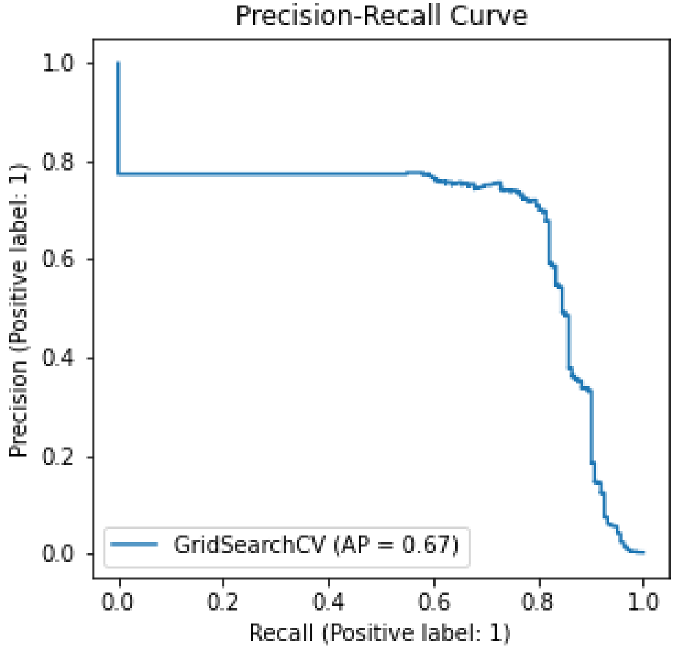



```
Accuracy        = 1.00
Precision       = 0.91
Recall          = 0.61
F1 Score        = 0.73
```

Undersampling

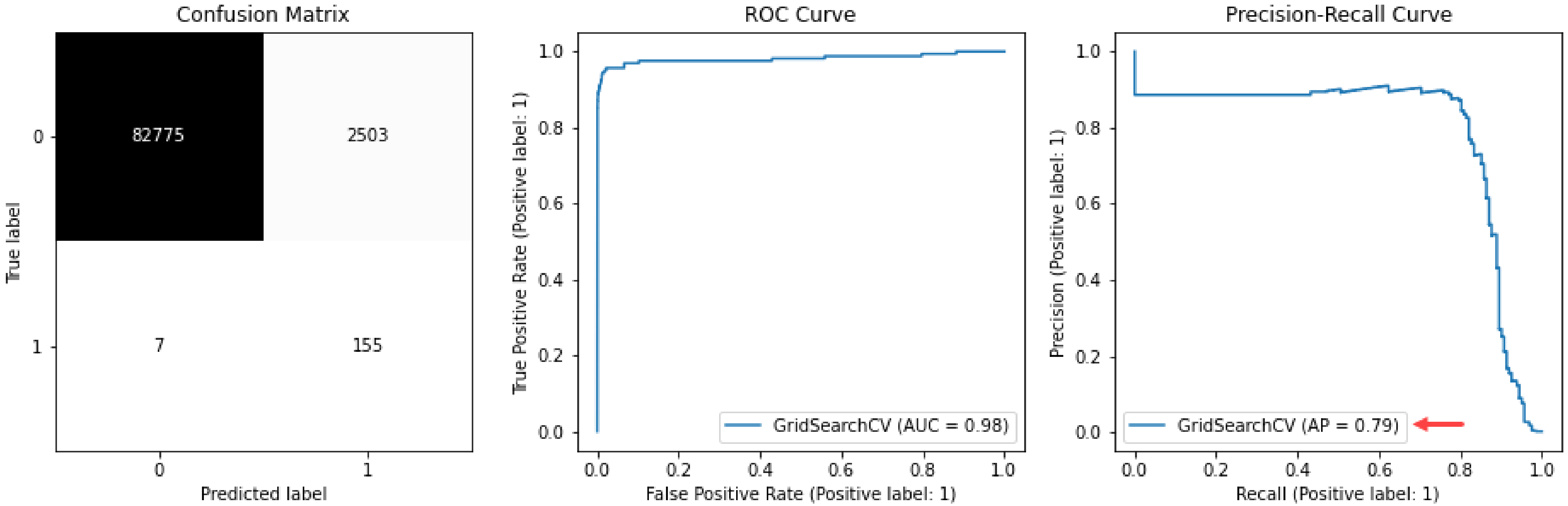






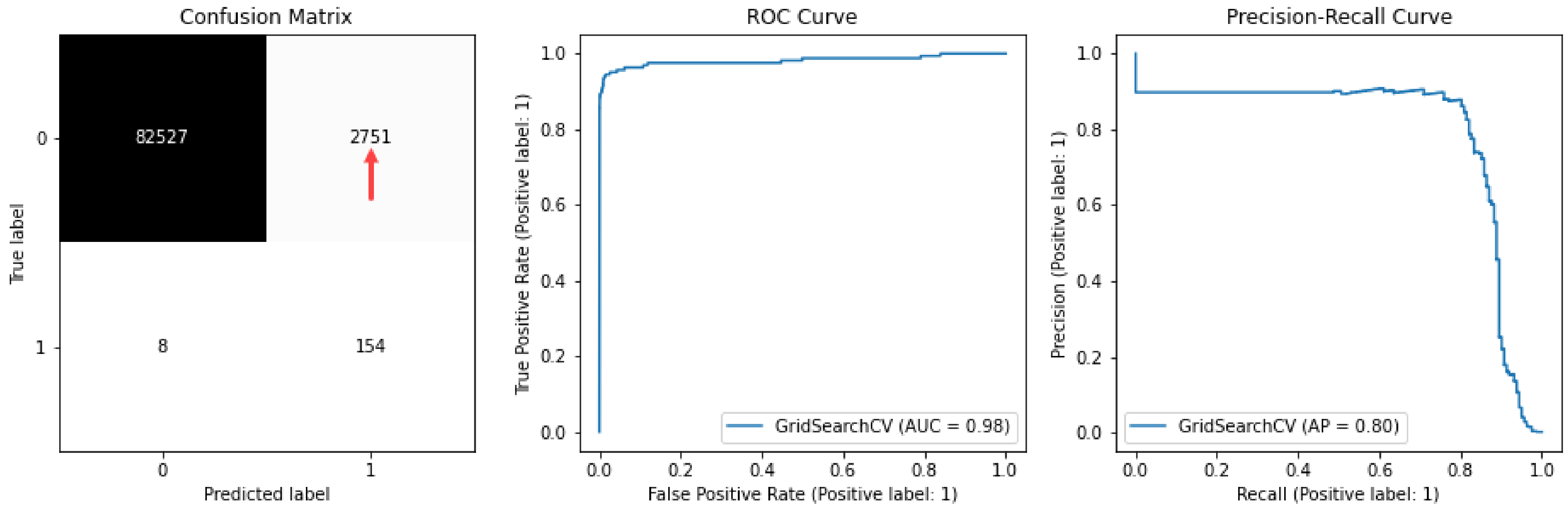

```
Accuracy        = 0.95
Precision       = 0.04
Recall          = 0.96
F1 Score        = 0.07
```

Oversampling




```
Accuracy        = 0.97
Precision       = 0.06
Recall          = 0.96
F1 Score        = 0.11
```

Synthetic Minority Oversamping Technique (SMOTE)

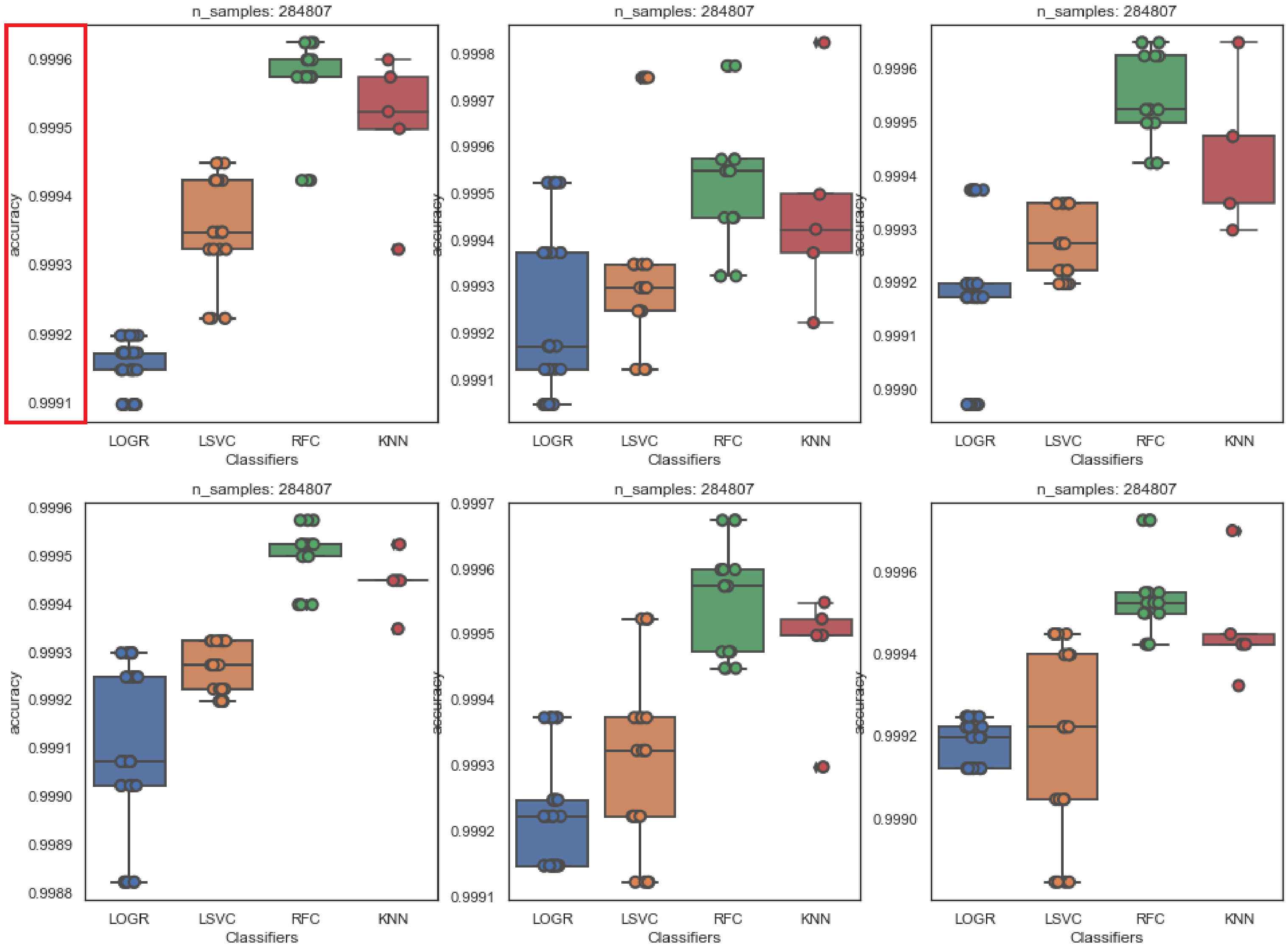



```
Accuracy        = 0.97
Precision       = 0.05
Recall          = 0.95
F1 Score        = 0.10
```

Model Evaluation

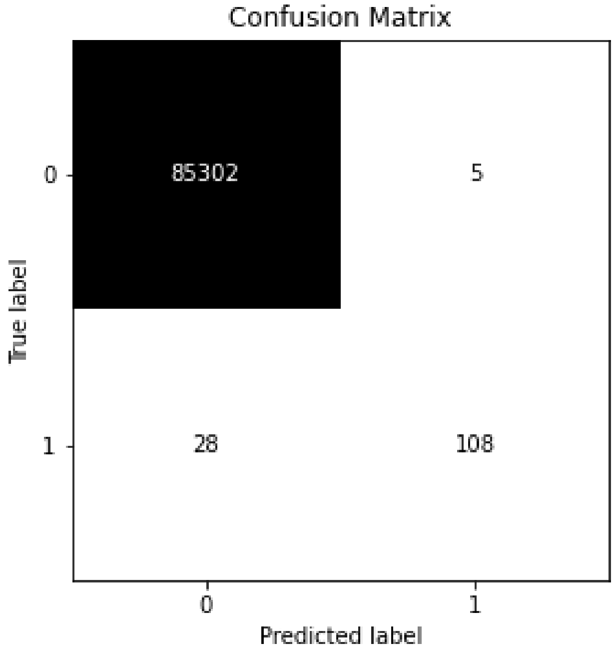
$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

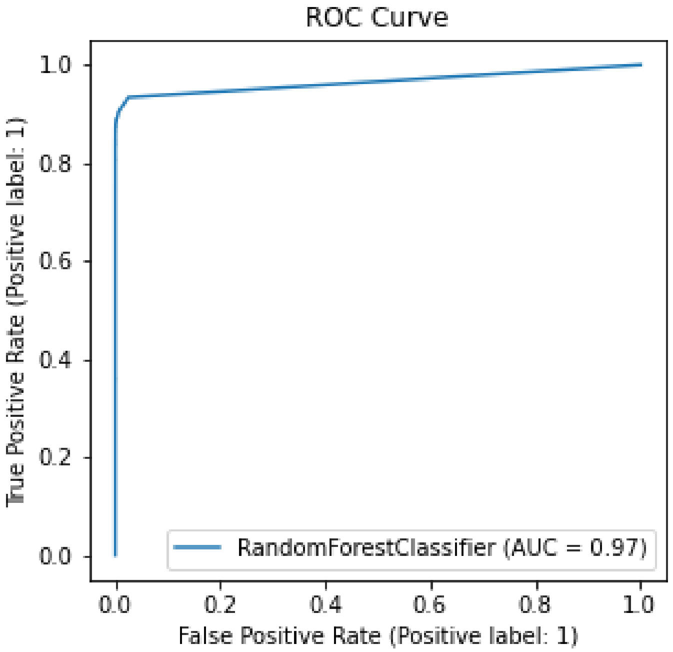
TN	FP
FN	TP

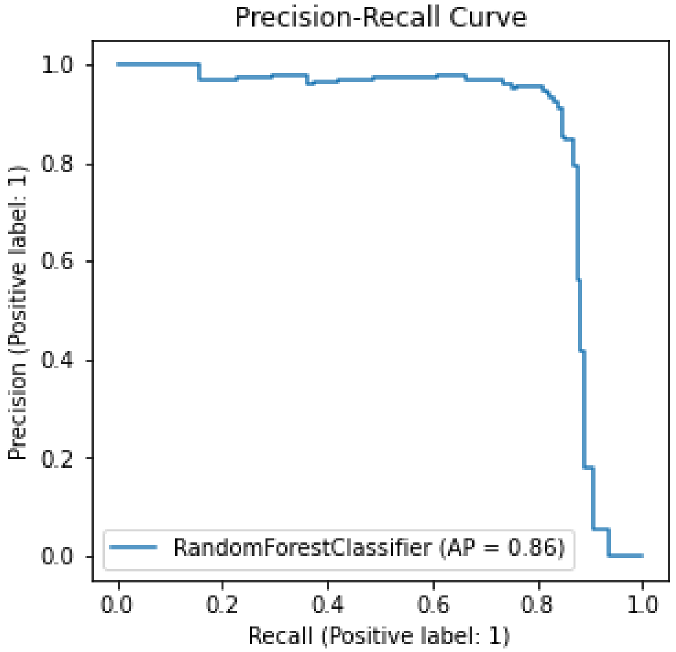


Random Forest Classifier

No scaling, no resampling



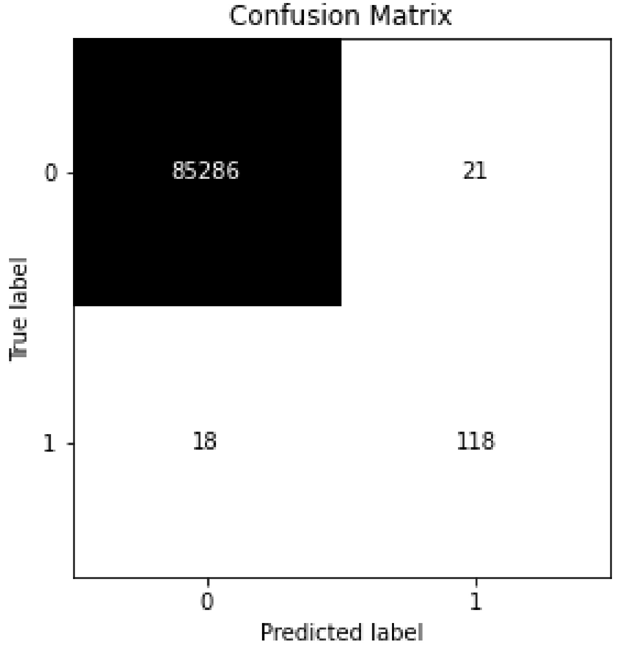


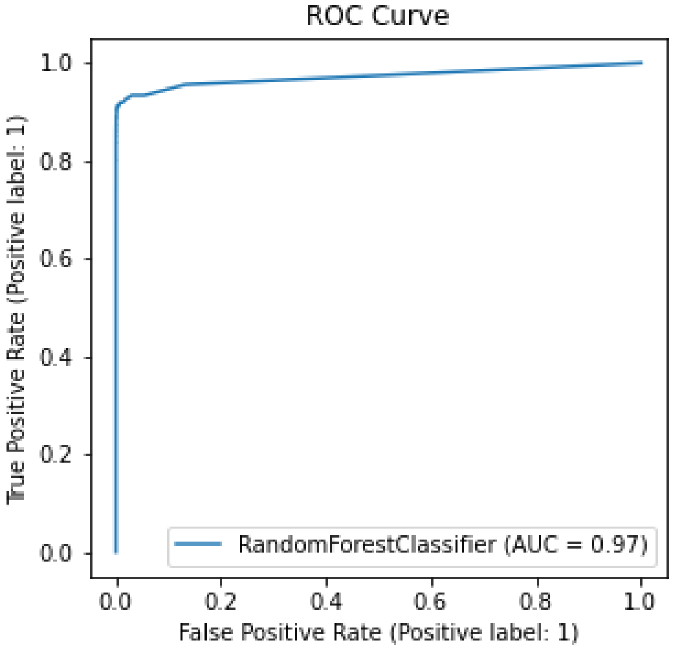


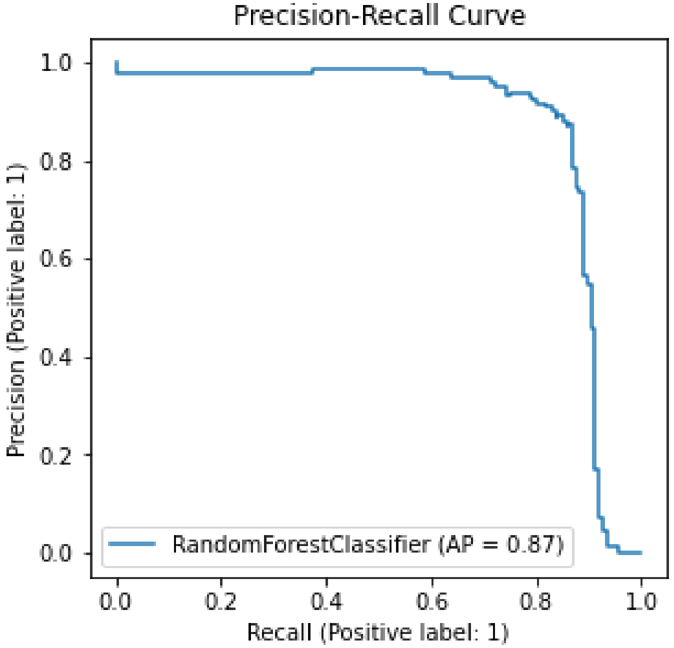

```
Accuracy       = 1.00
Precision      = 0.96
Recall         = 0.79
F1 Score       = 0.87
```

Random Forest Classifier

No scaling, SMOTE resampling





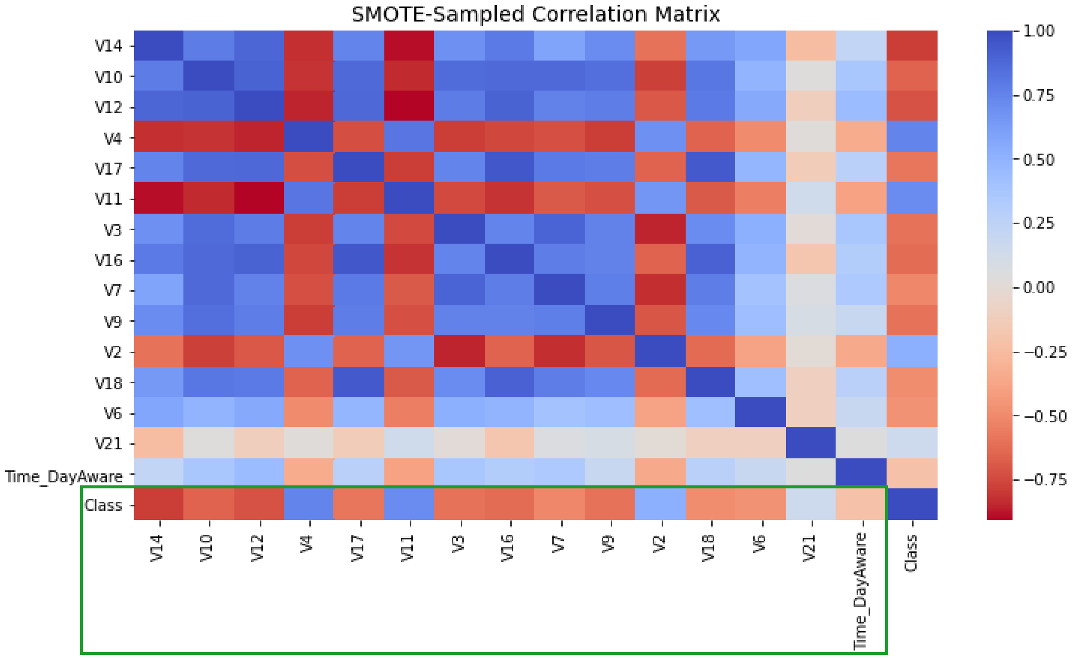


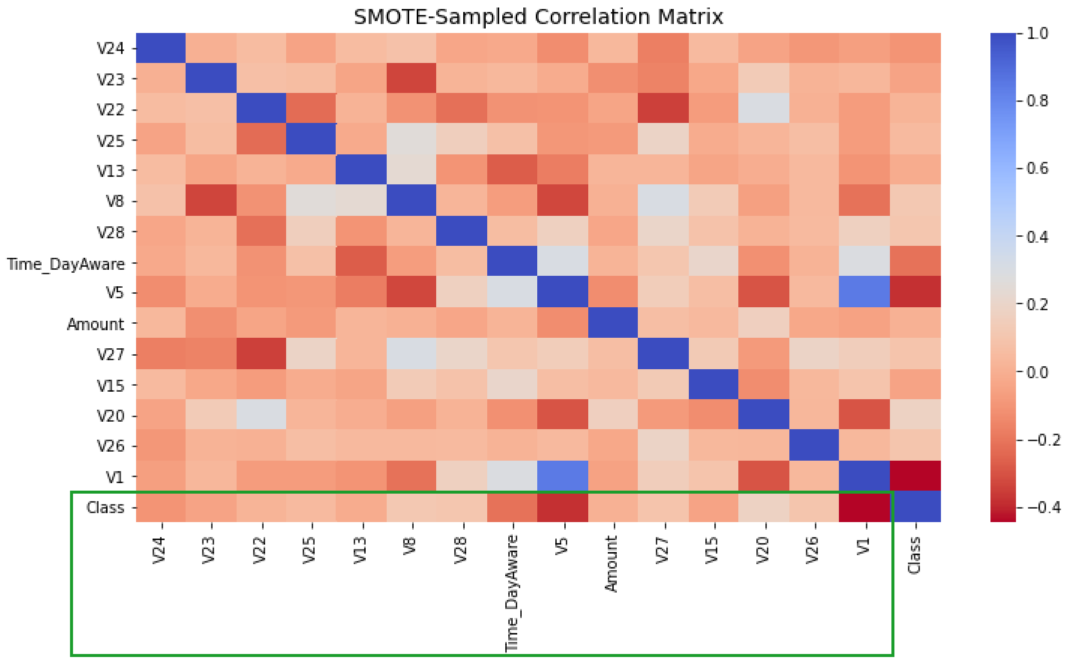

```
Accuracy        = 1.00
Precision       = 0.85
Recall          = 0.87
F1 Score        = 0.86
```

Feature Selection

Correlation Matrix

*Correlation Matrix for the **Top 15** Important Features from Mean Decrease in Impurity.*

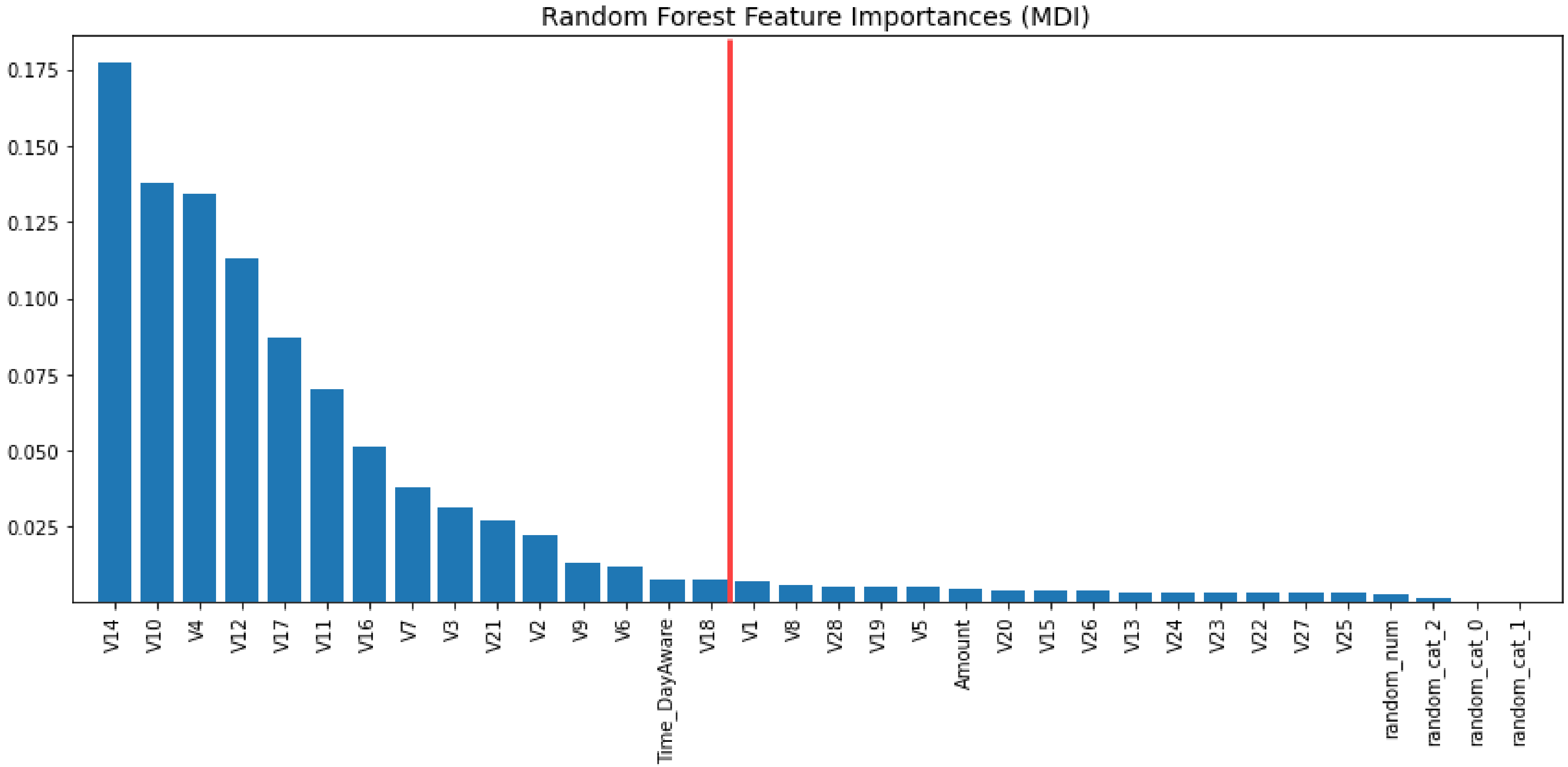




*Correlation Matrix for the **Bottom 15** Important Features from Mean Decrease in Impurity.*

Feature Selection

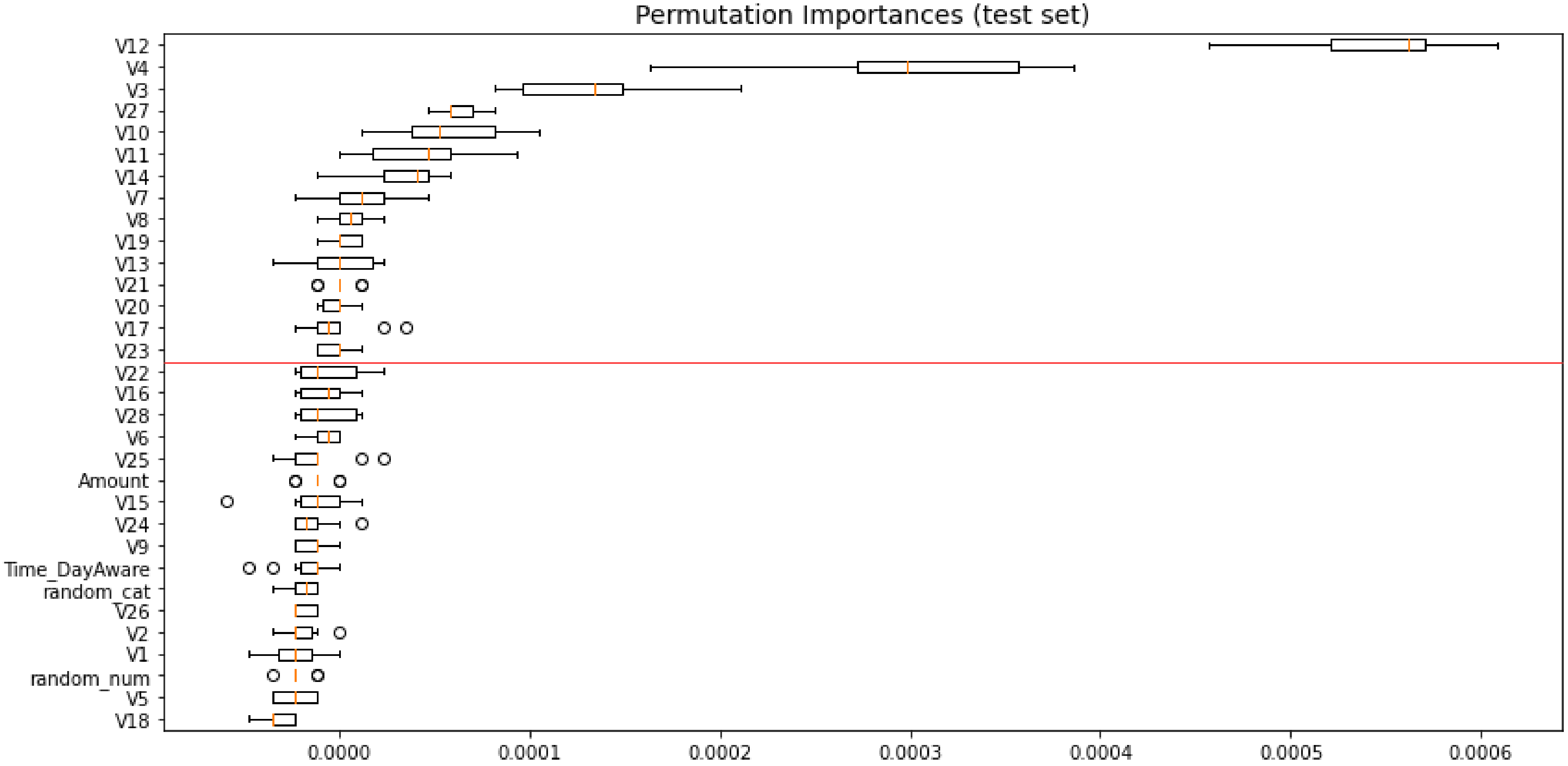
Mean Decrease in Impurity (MDI)



Gini Importance Returns Feature Importance from the SMOTE RFC Fit.

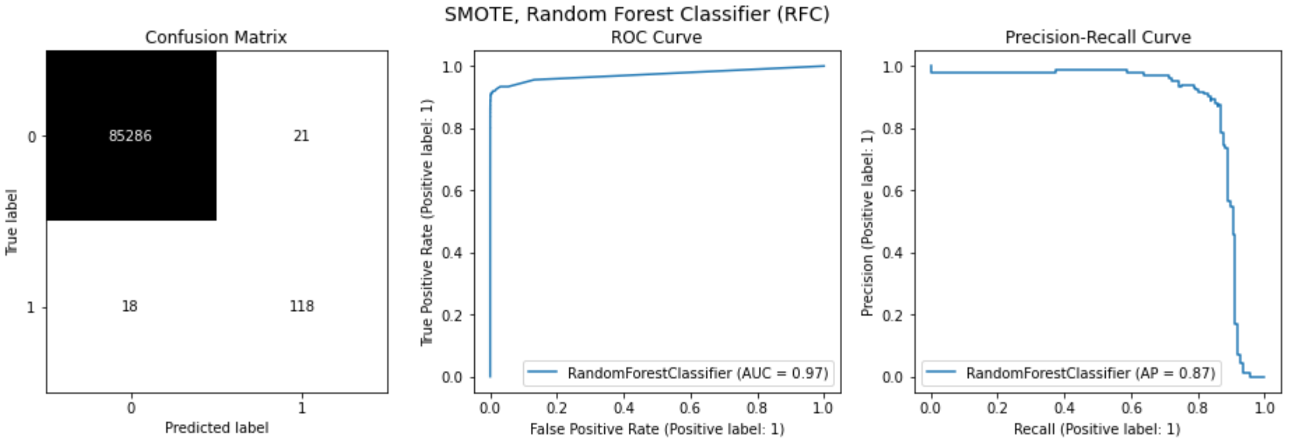
Feature Selection

Permutation Importance

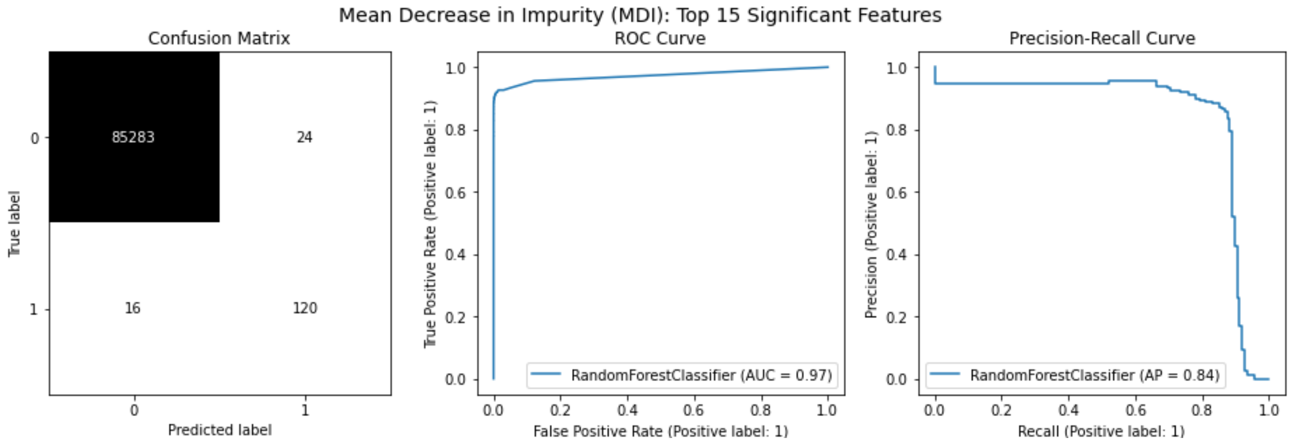


MDI can inflate the importance of numerical features, and the importance can be even higher for features that are not predictive of the target variable – making Permutation Importance a better alternative for feature selection (scikit-learn, 2020).

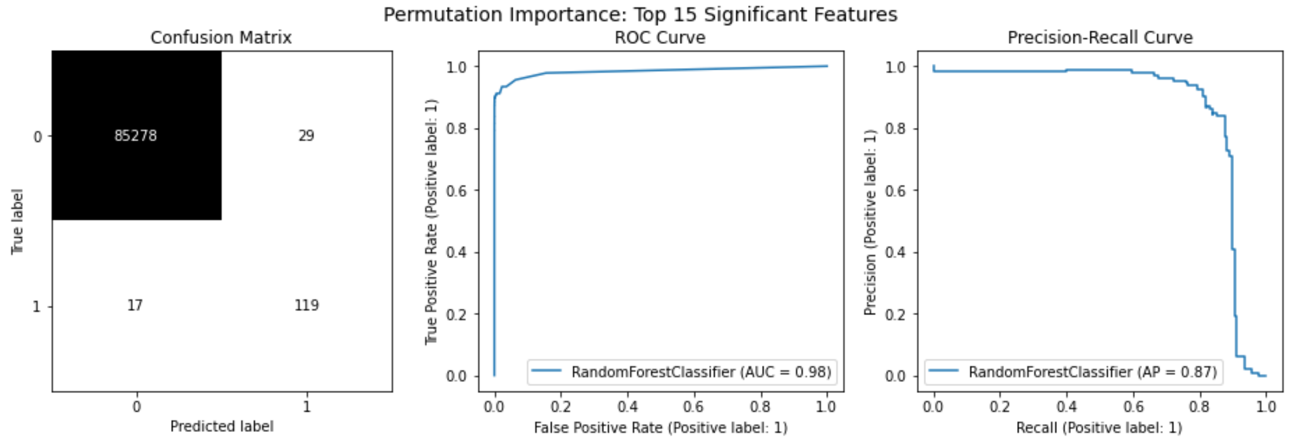
No Feature Selection



Mean Decrease in Impurity



Permutation Importance



Summary

Business Case

- *True Positive*: Correctly Identifying Fraud
- *False Negative*: Missed Identifying Fraud
- *False Positive*: Identify a Genuine Transaction as Fraud
- *True Negative*: Correctly Identify Genuine Transactions

A business runs on profit. Fraud creates significant cost. Fraud Detection Model needs to identify fraud (Maximize True Positives) and minimize missed fraud (Minimize False Negatives) to best control costs. False Alarms can be significant at higher quantities also.

Significant Observations

1.) Implementing the Random Forest Classifier improved model performance
2.) SMOTE resampling improved model performance

Machine Learning vs. Learning Fraudsters

- Fraudsters learn, models need to be equally adaptable