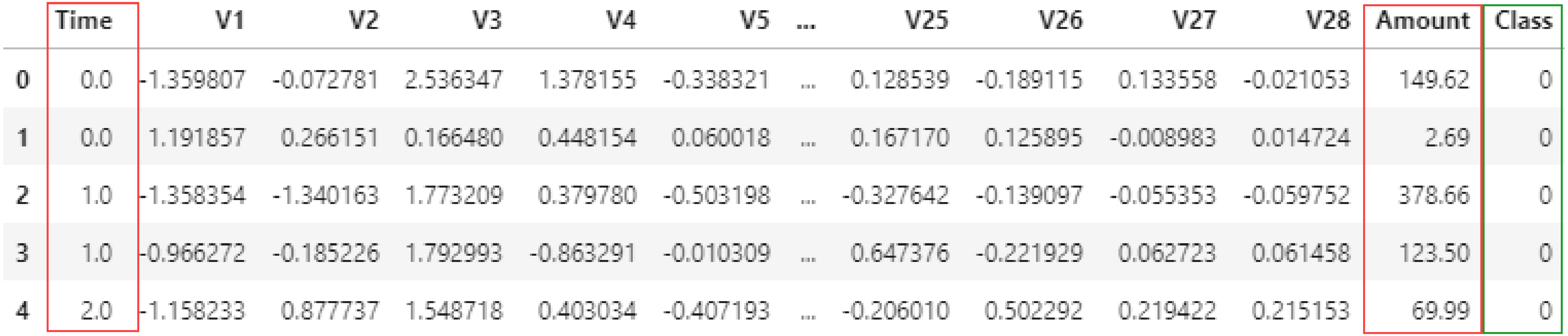
Exploring Class Imbalance with Fraud Detection

James Bush
Springboard
Milestone Report 2

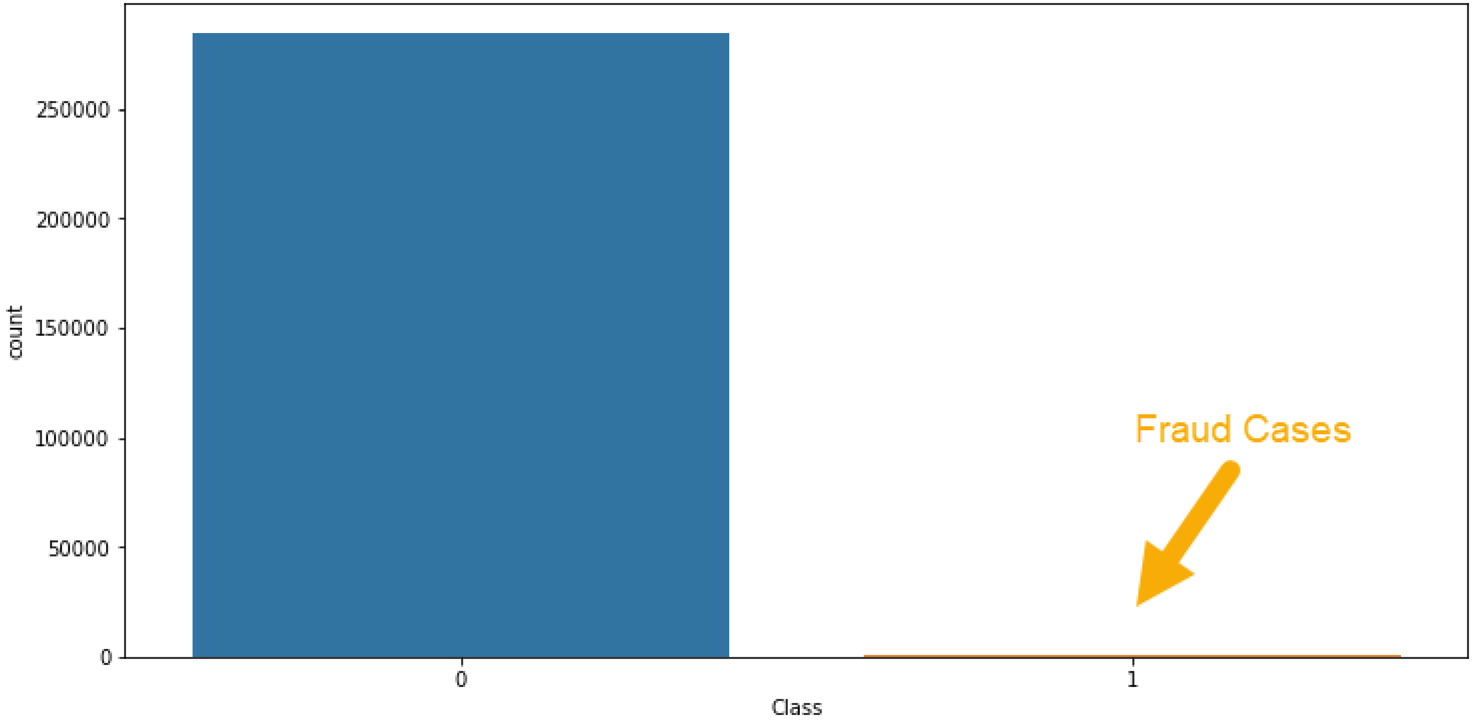
Features



	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

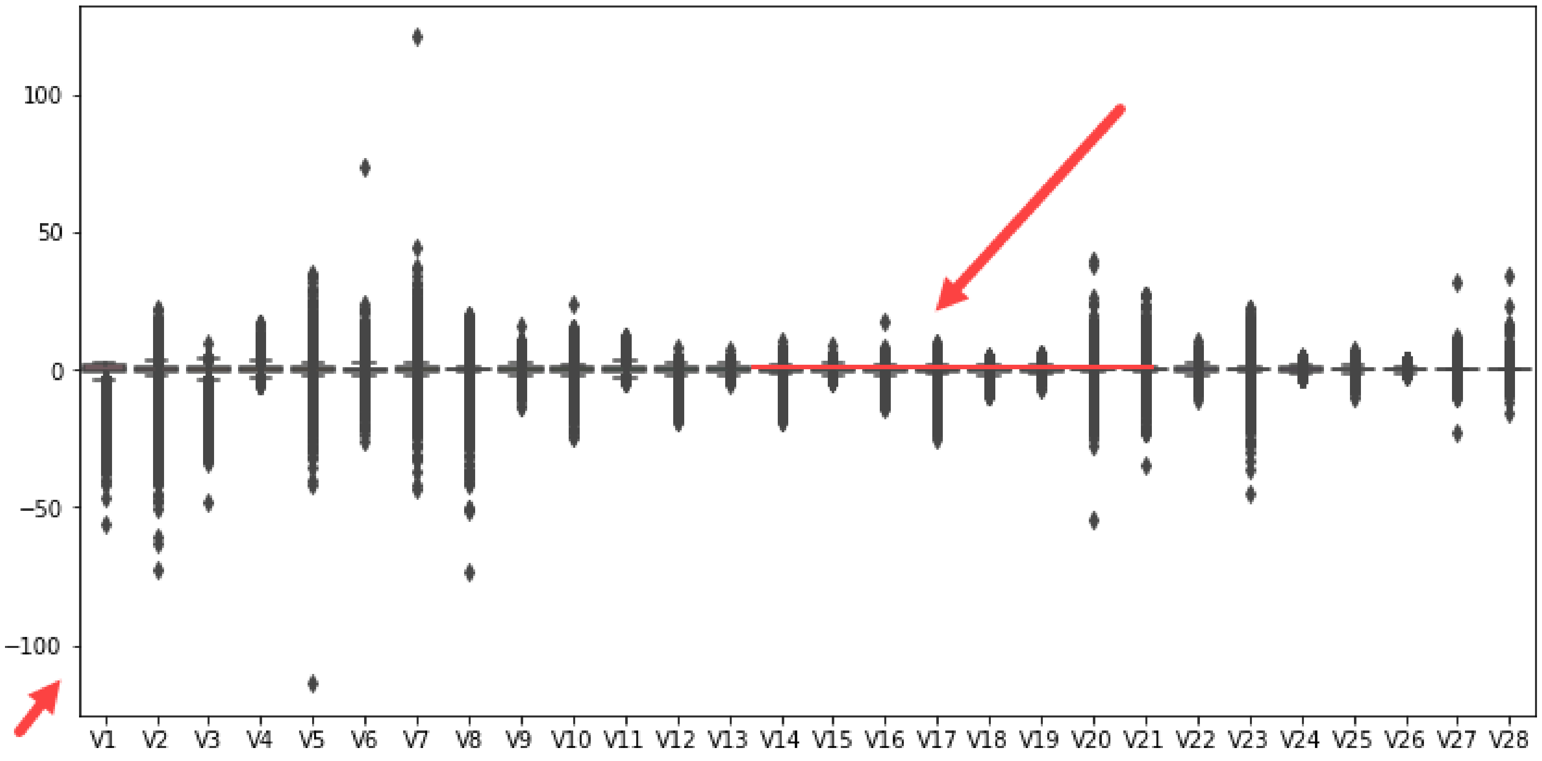
- 30 Independent Variables
- 28 Transformed with PCA (V1-V28)

Transaction Counts



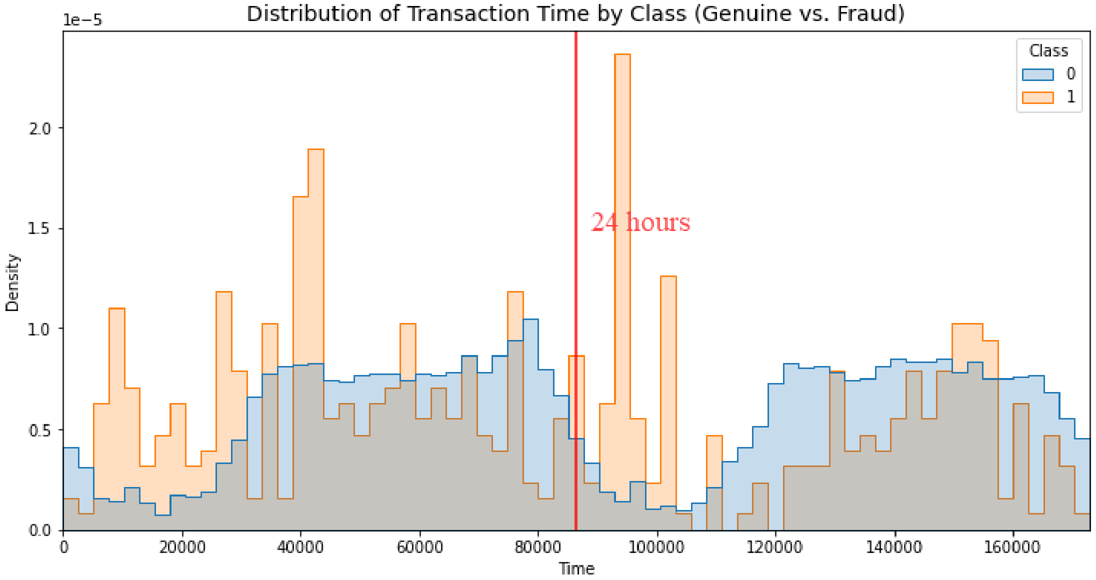
- 284,807 Total Transactions
- 284,315 Majority Class (0)
- 492 Minority Class (1)

Transformed Feature Boxplots



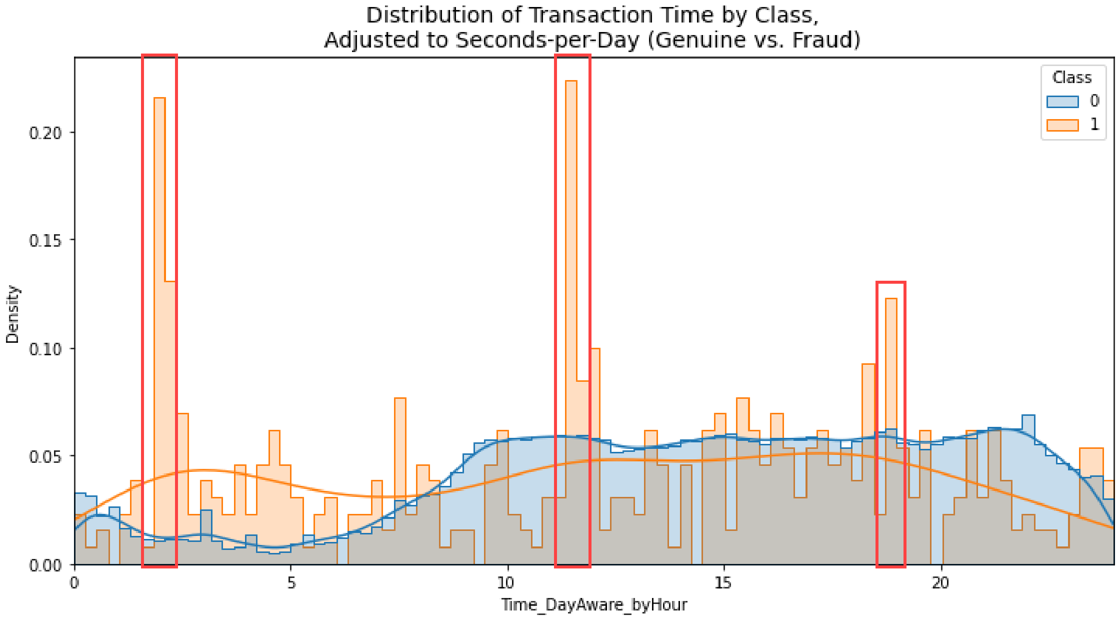
- PCA-transformed features centered on 0
- Values range 150 to -150
- Values have a higher density between 50 to -50

Transaction Time



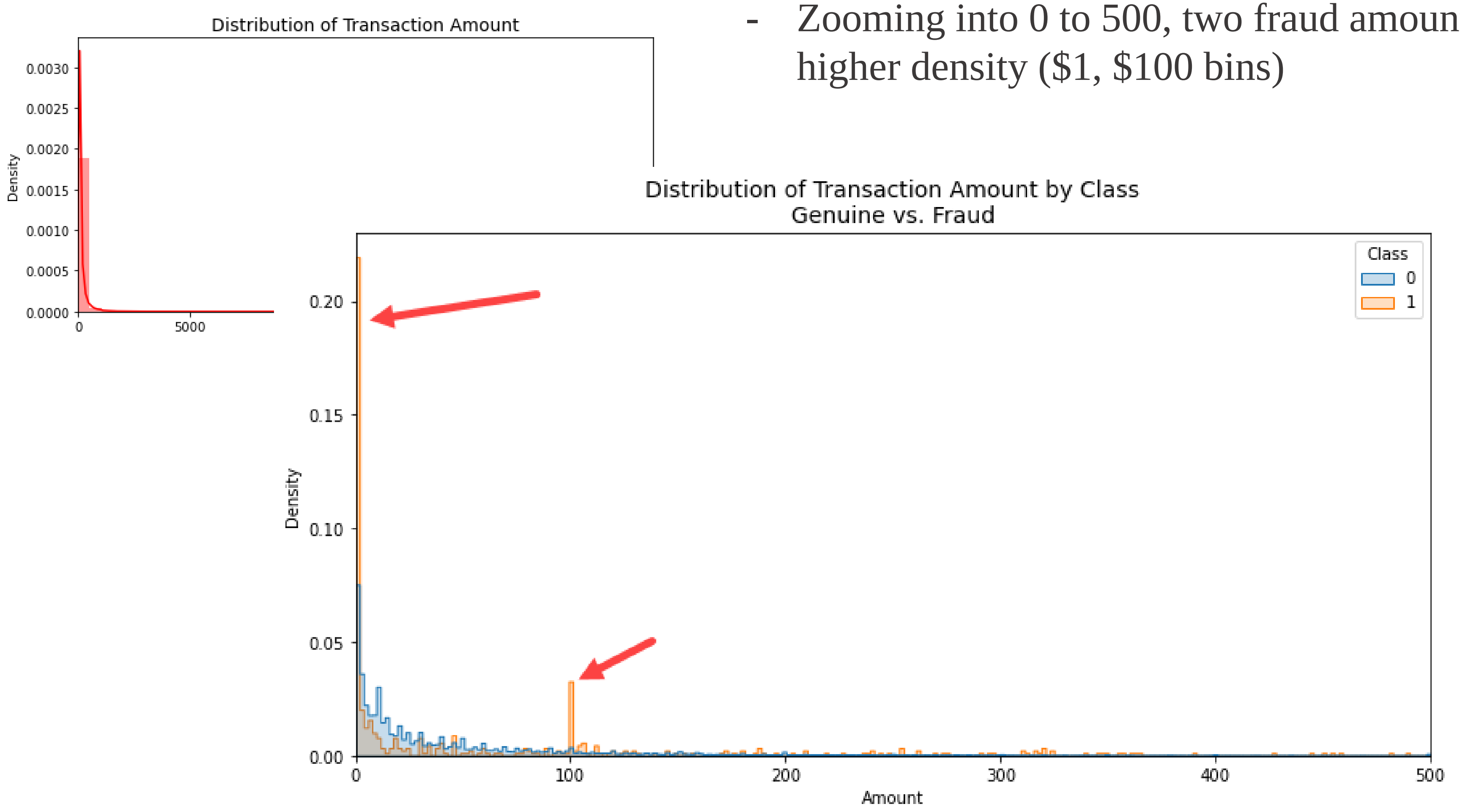
- Transaction Time is given as the number of seconds from transaction 0
- Total time adds up to 48 hours

- 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

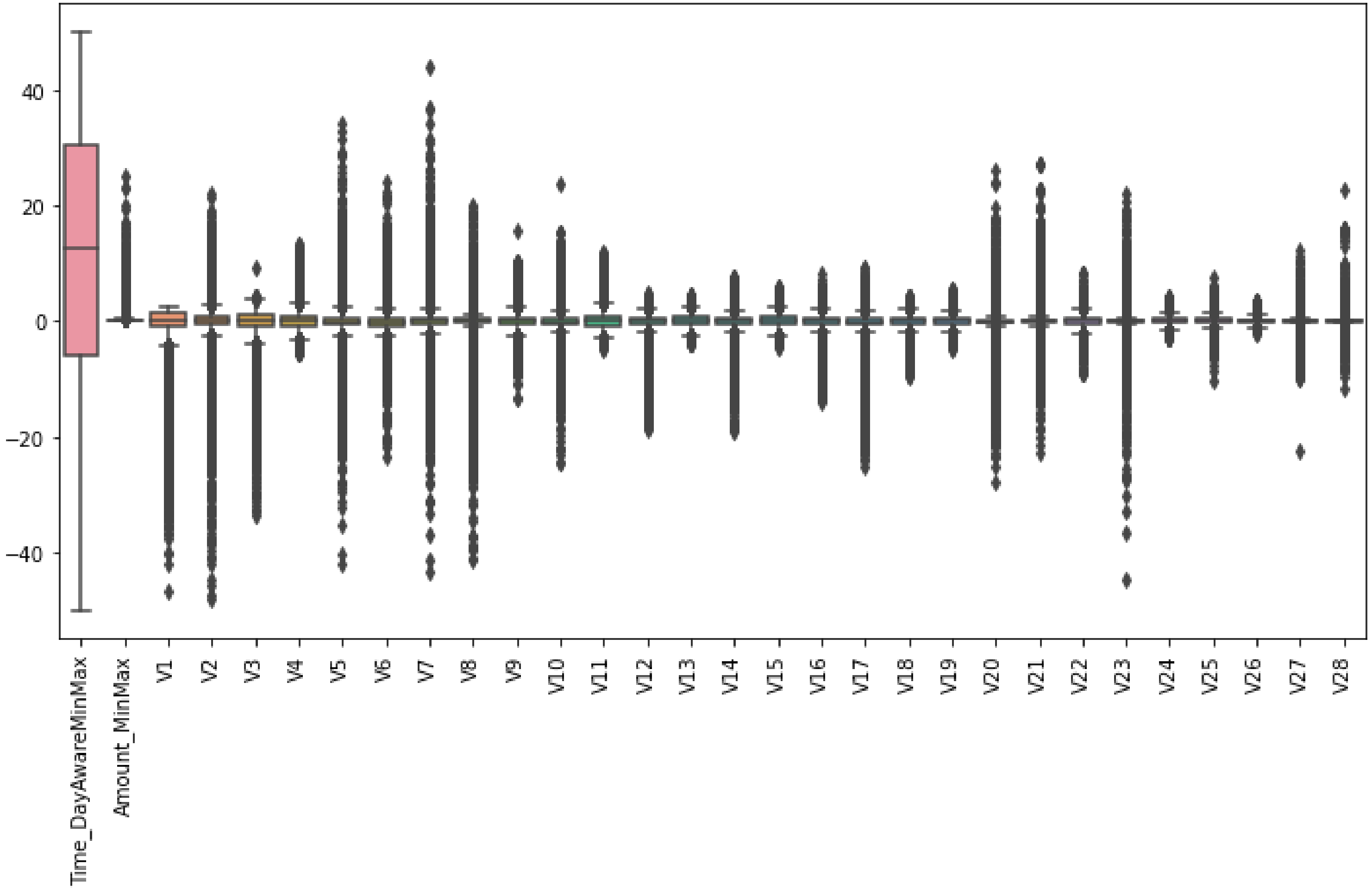


Transaction Amount

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

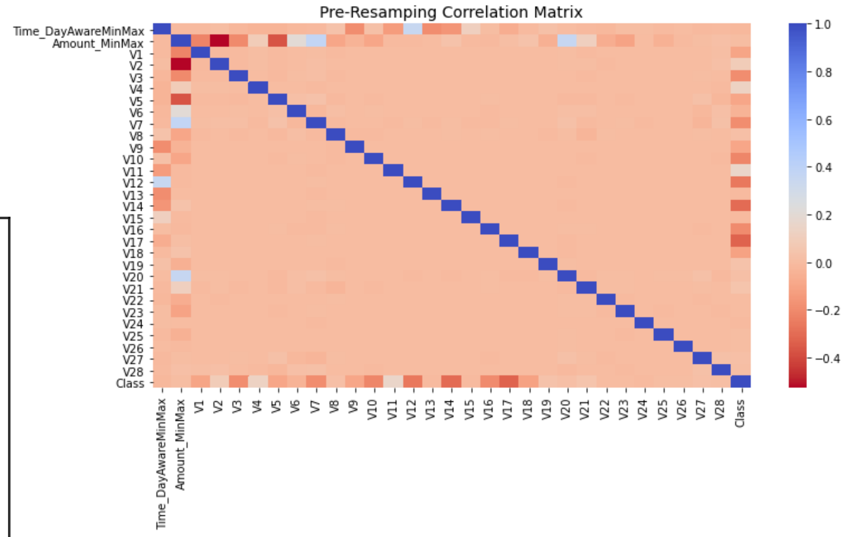


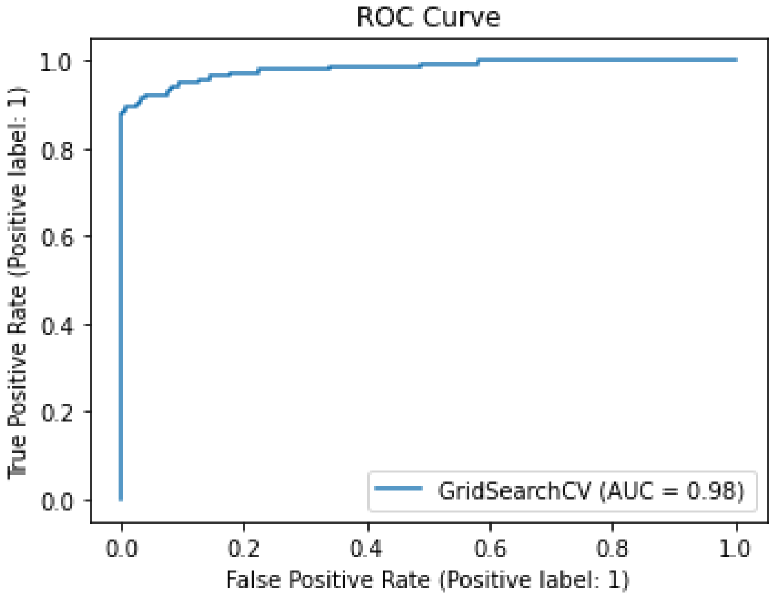
Scaling Time, Amount

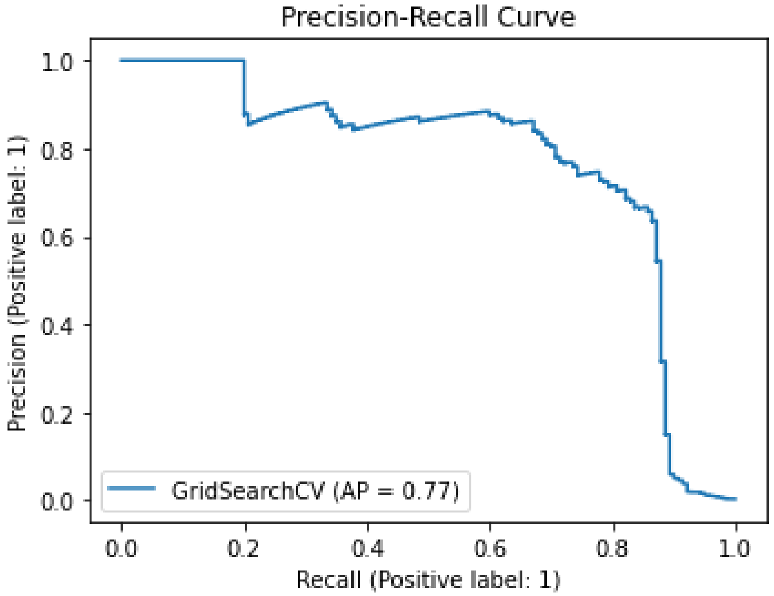


Boxplots show the result after Feature Scaling, Dropping Outliers.

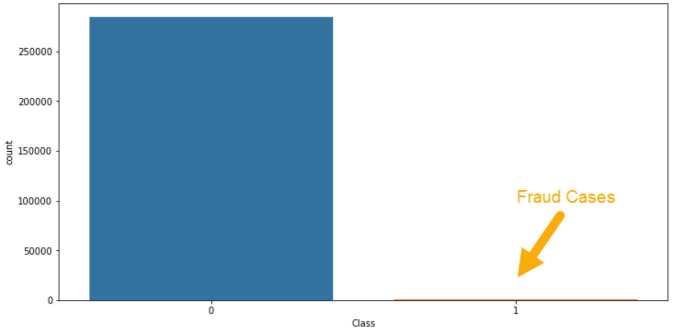
Pre-Sampling LogR





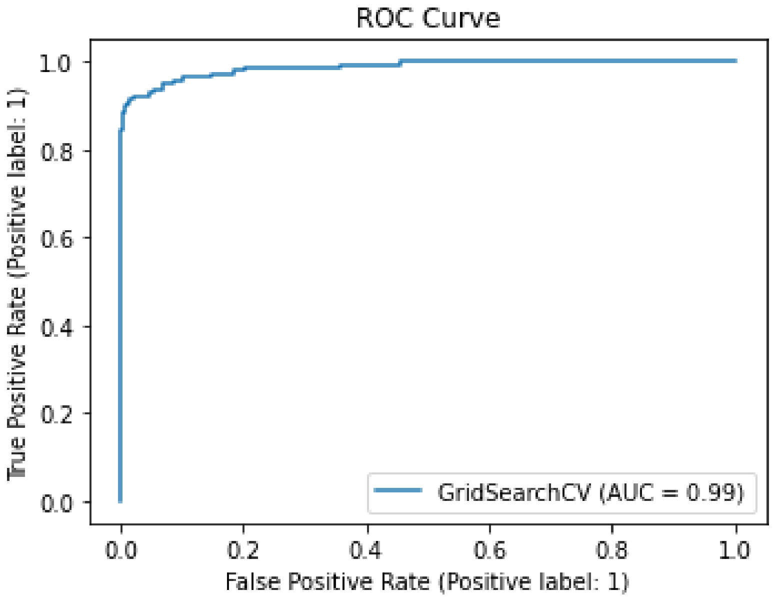


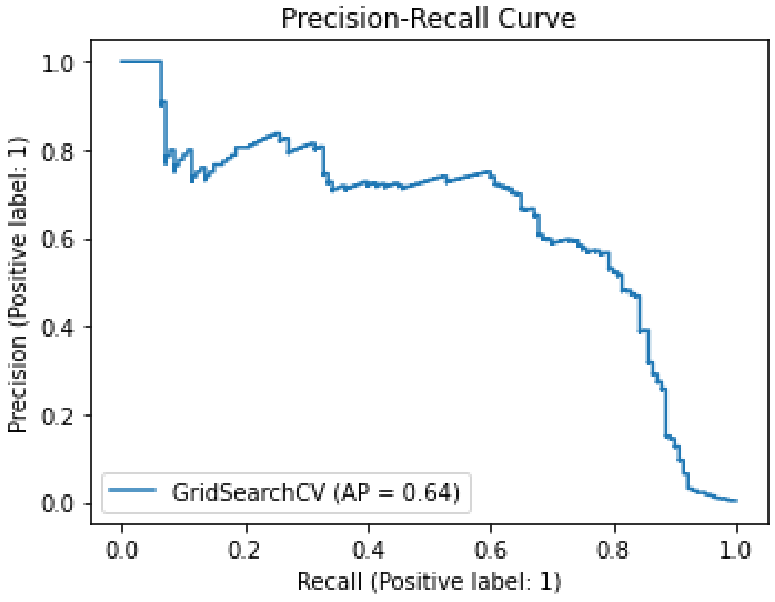
```
Accuracy         = 1.00
Precision        = 0.86
Recall           = 0.63
F1 Score         = 0.73
```

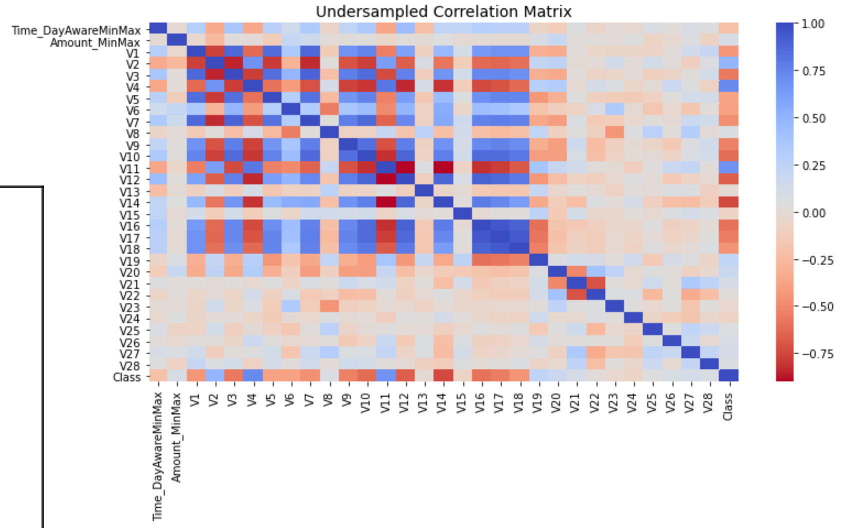


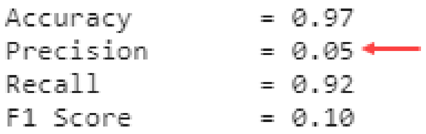
Fraud Cases

Undersampling

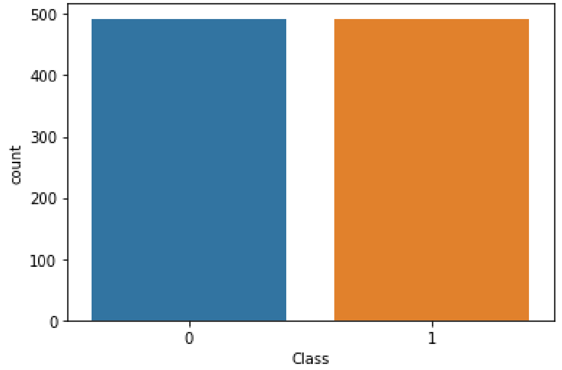




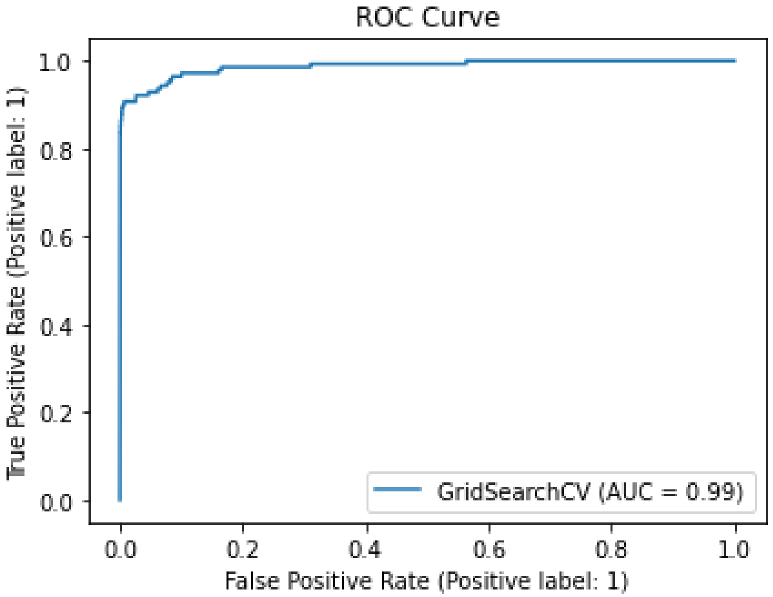


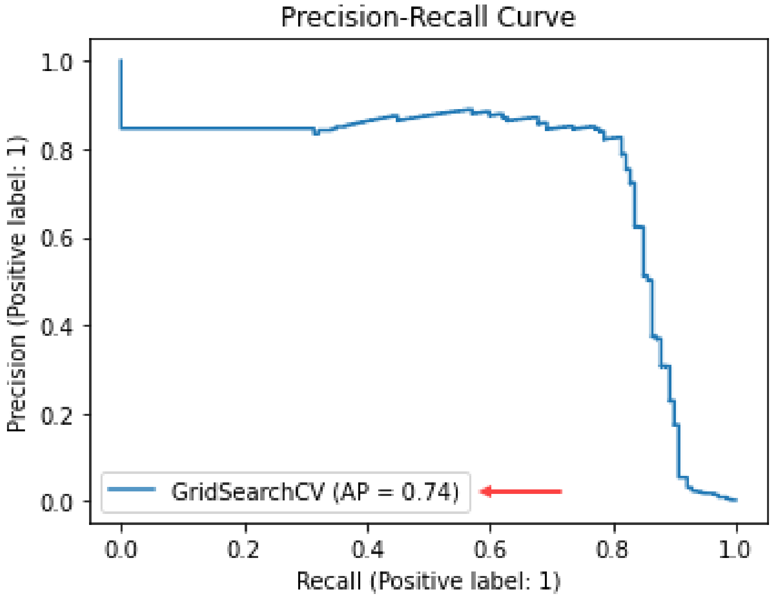


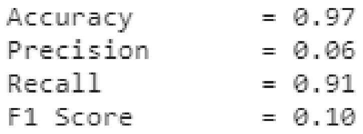
```
Accuracy        = 0.97
Precision       = 0.05
Recall          = 0.92
F1 Score        = 0.10
```



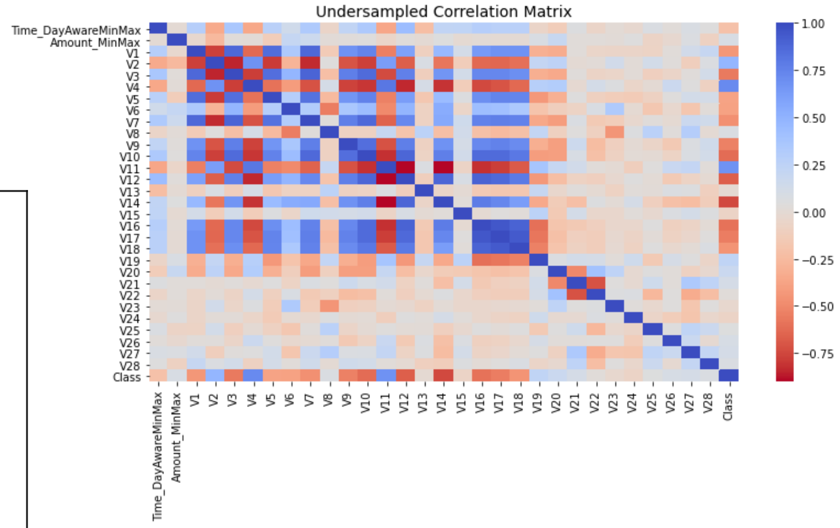
Oversampling

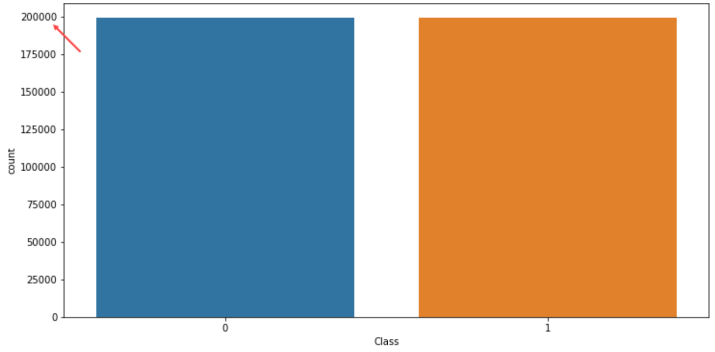




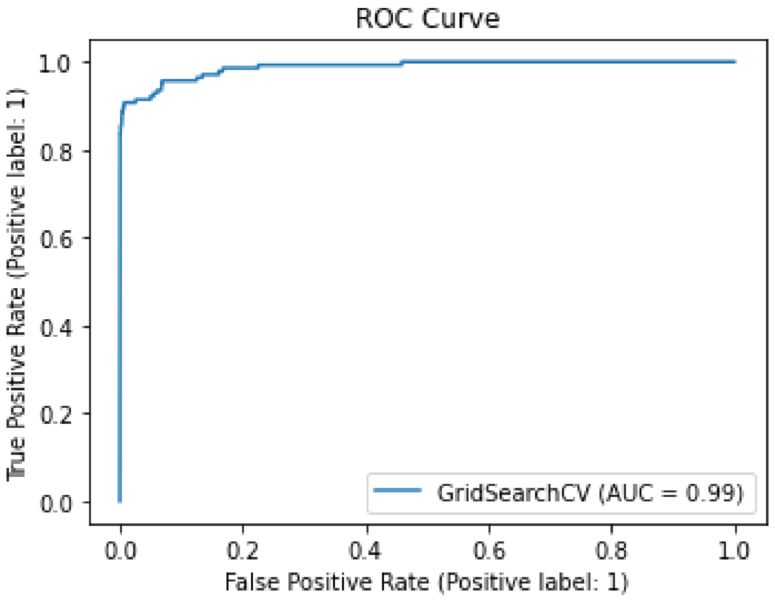


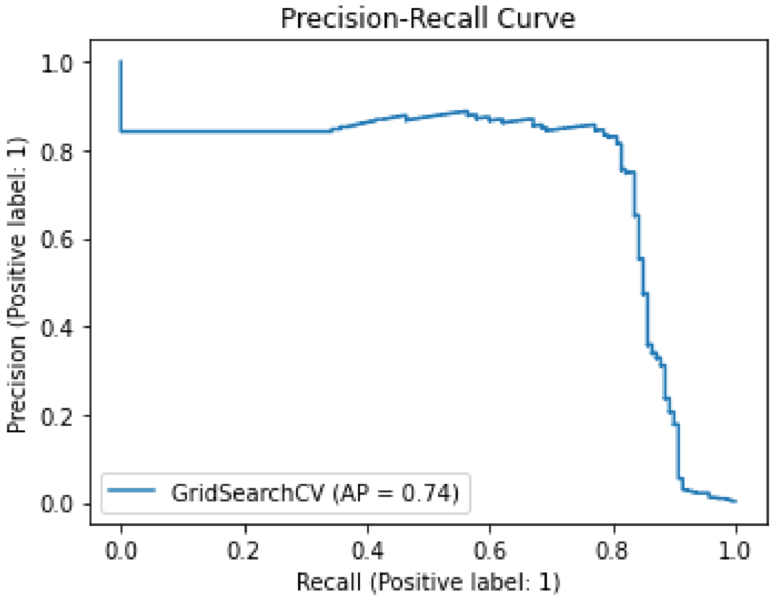
```
Accuracy         = 0.97
Precision        = 0.06
Recall           = 0.91
F1 Score         = 0.10
```

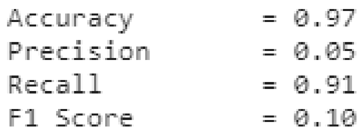




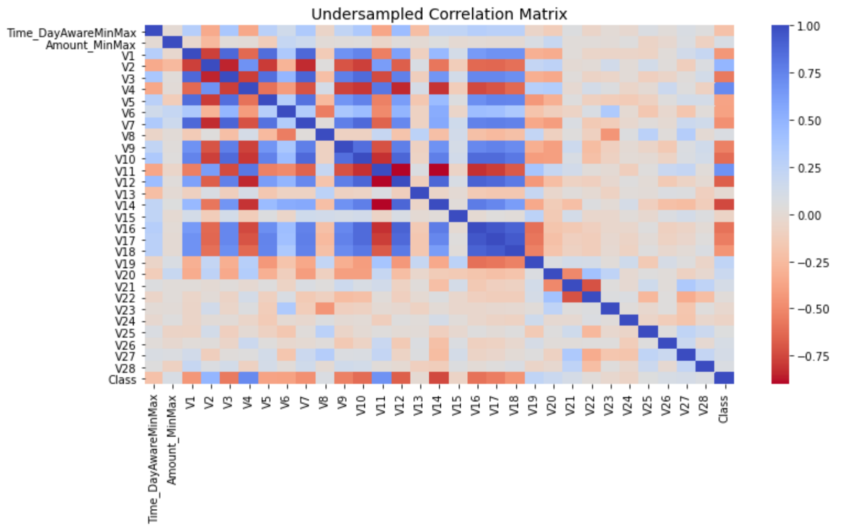
Synthetic Minority Oversamping Technique (SMOTE)

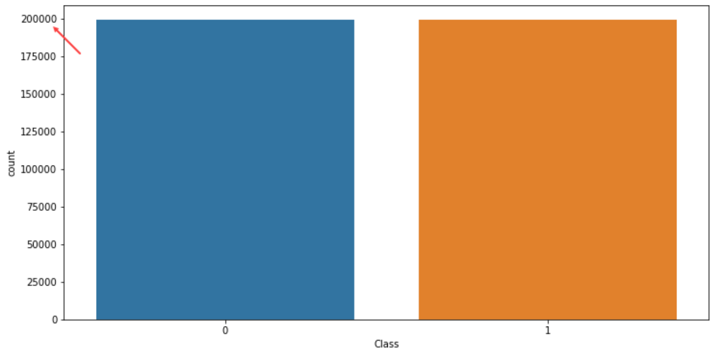






```
Accuracy        = 0.97
Precision       = 0.05
Recall          = 0.91
F1 Score        = 0.10
```

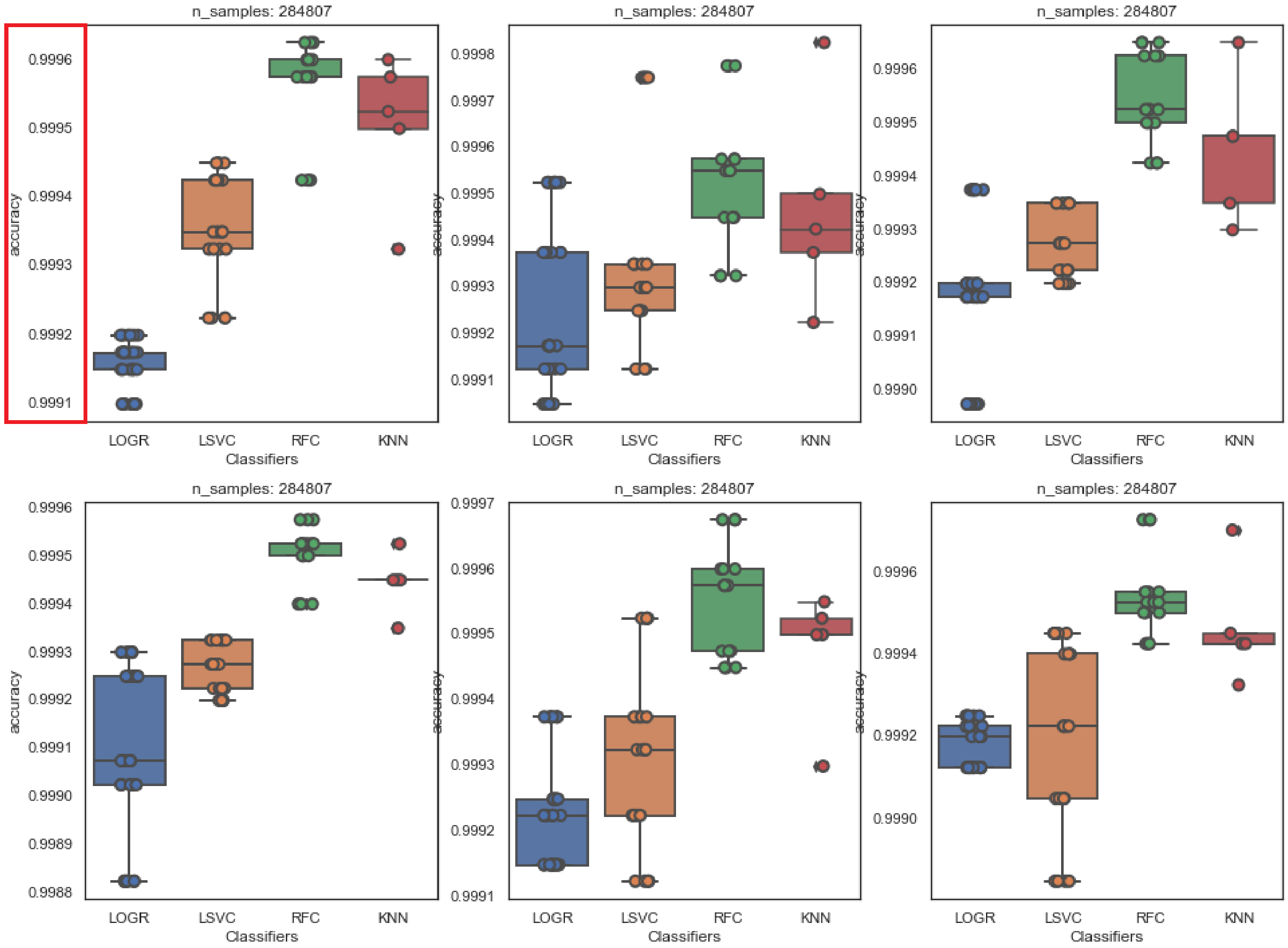




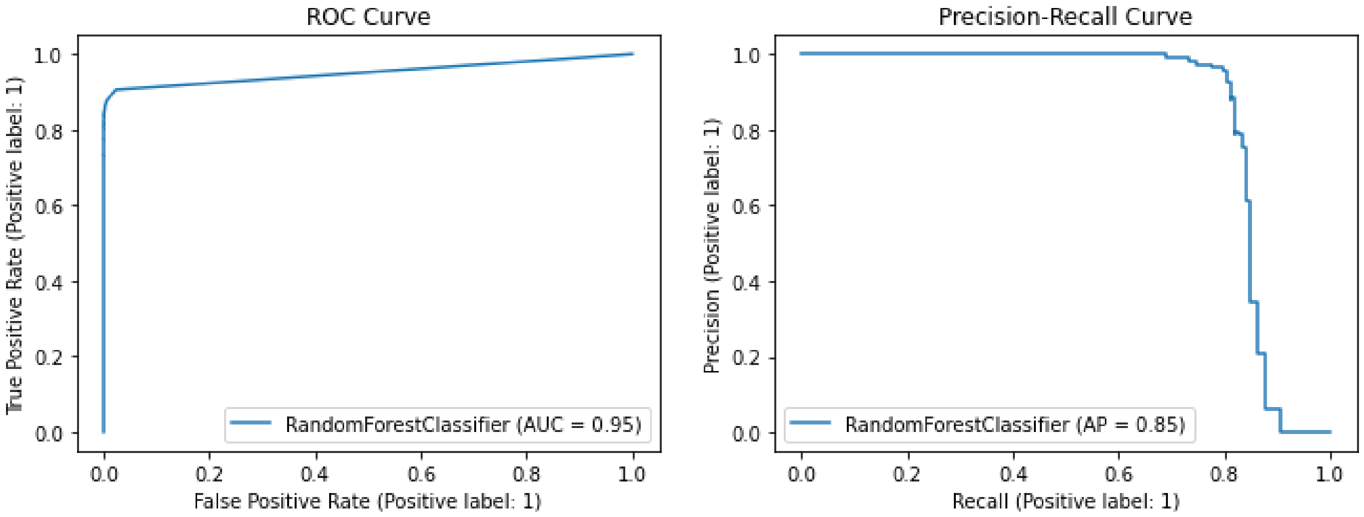
Model Evaluation

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

TN	FP
FN	TP

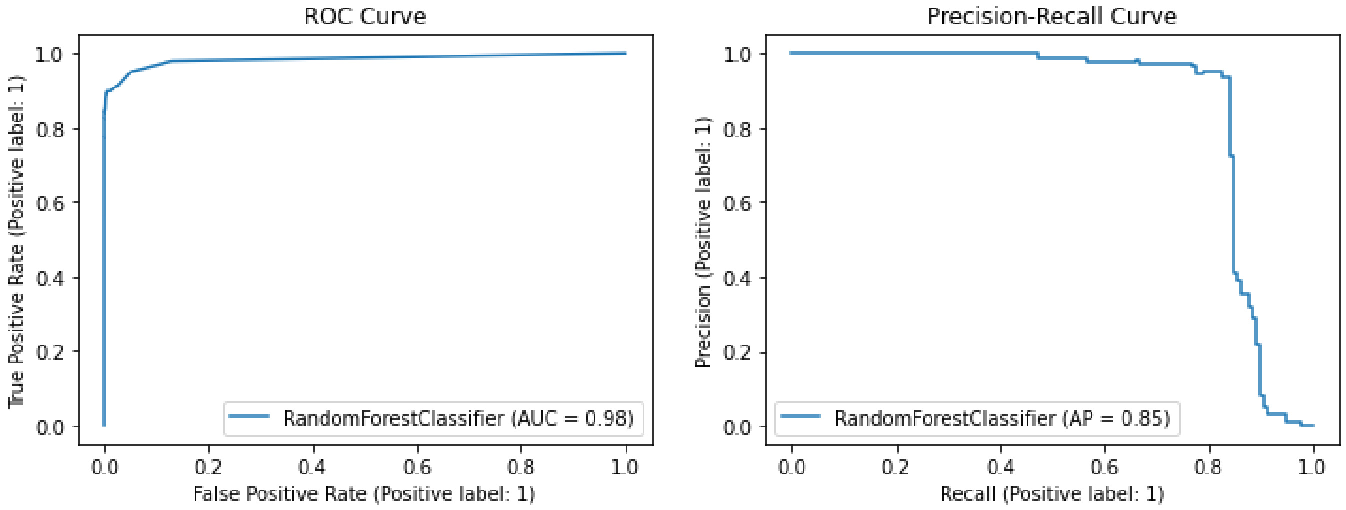


Implementing a Random Forest Classifier without scale or resampling



```
Accuracy         = 1.00
Precision        = 0.97
Recall           = 0.78
F1 Score         = 0.86
```

Implementing Random Forest Classifier with SMOTE



```
Accuracy         = 1.00
Precision        = 0.90
Recall           = 0.84
F1 Score         = 0.87
```