Credit Card Fraud Detection

Noah Monastersky | Data Scientist

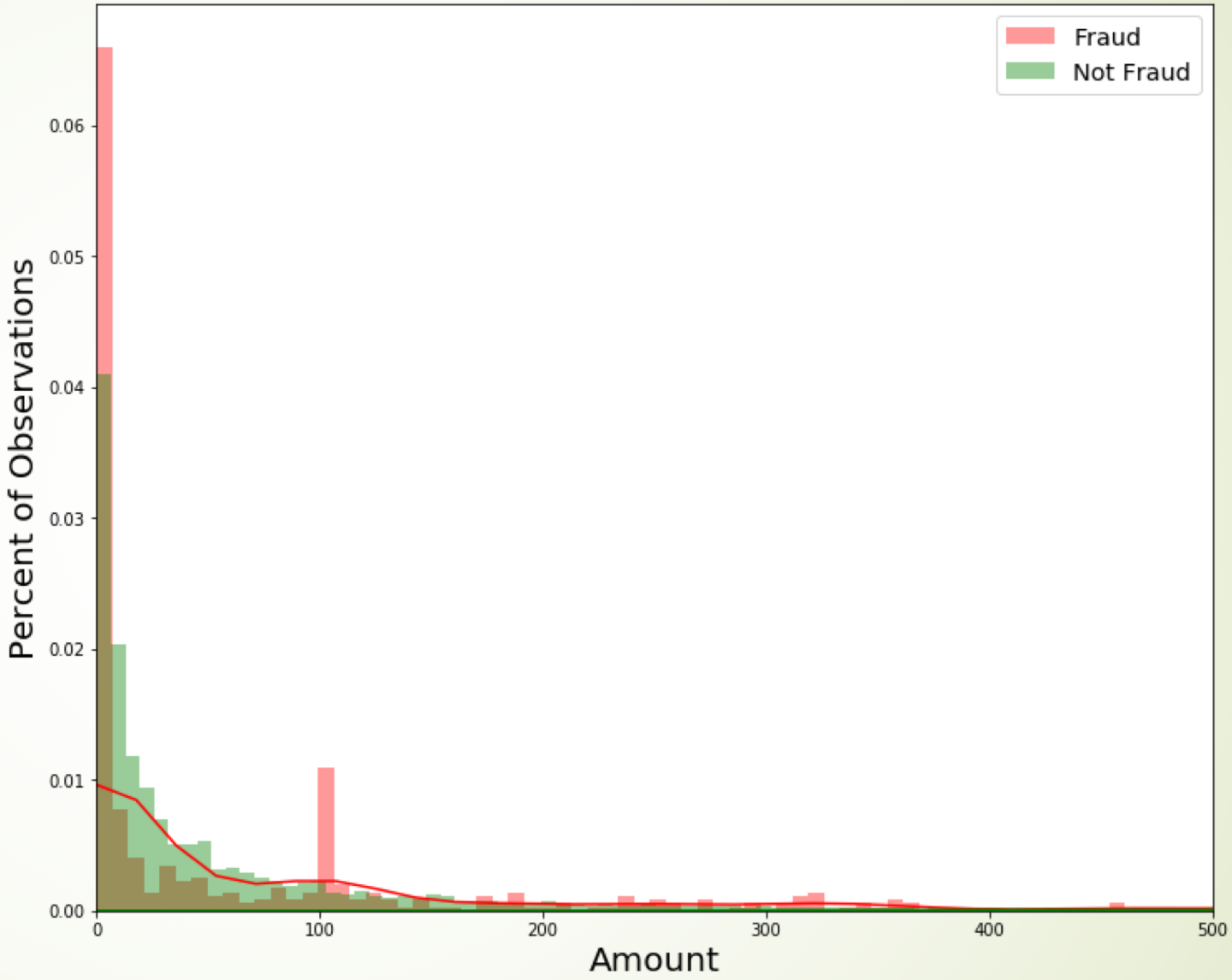
Problem Statement

- Can I build a model that predicts whether a credit card charge is fraudulent or not?
- This has a clear use cases because financial institutions are liable when fraudulent charges occur.

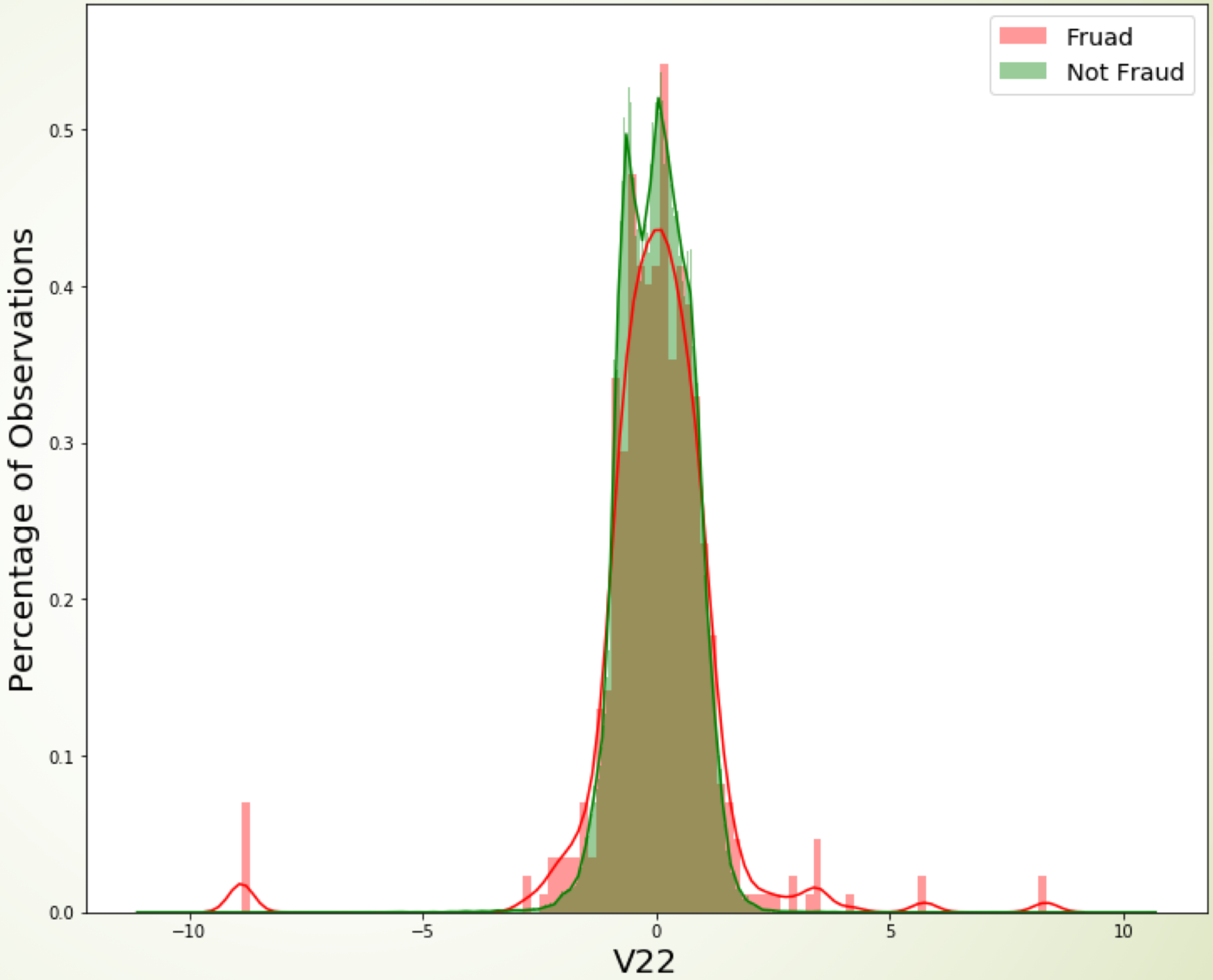
The Data

- 284,807 observations from European cardholders in 2013
- 492 Fraud | 284,315 non- fraud
- Amount and Time column
- V1 – V28 PCA'd columns.

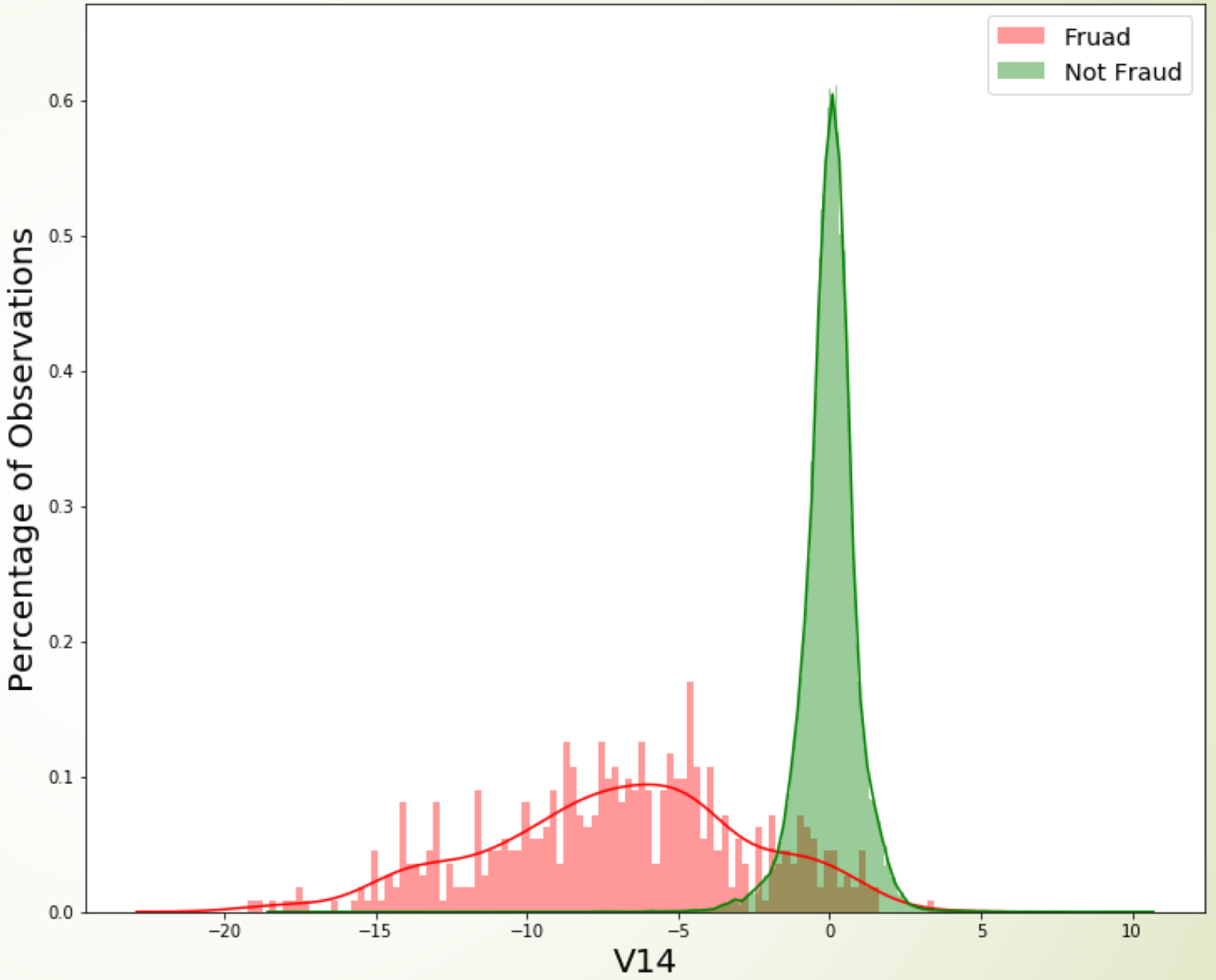
Most Charges are low dollar amount.

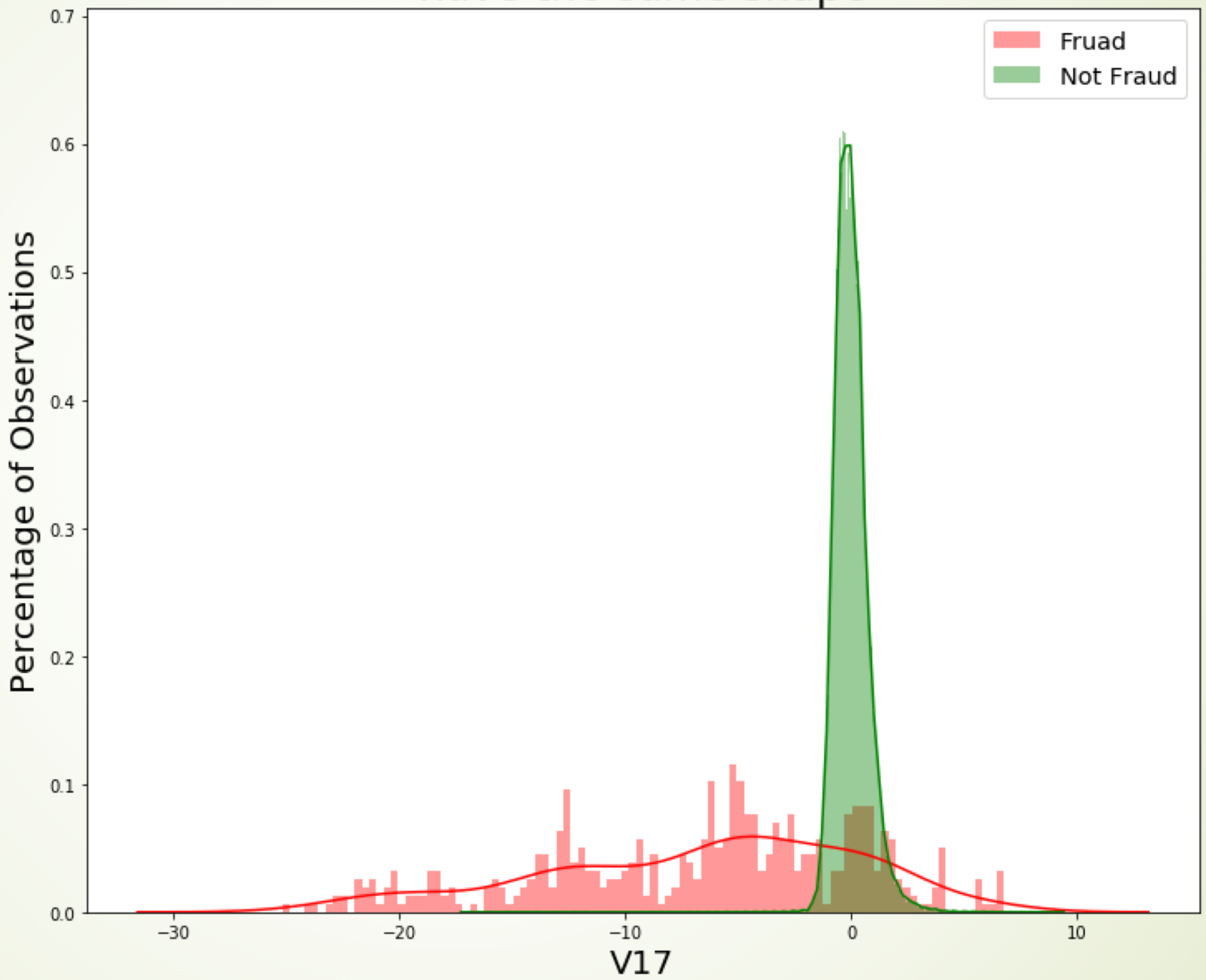


Variables with low correlation to class like V22 follow roughly the same shape.

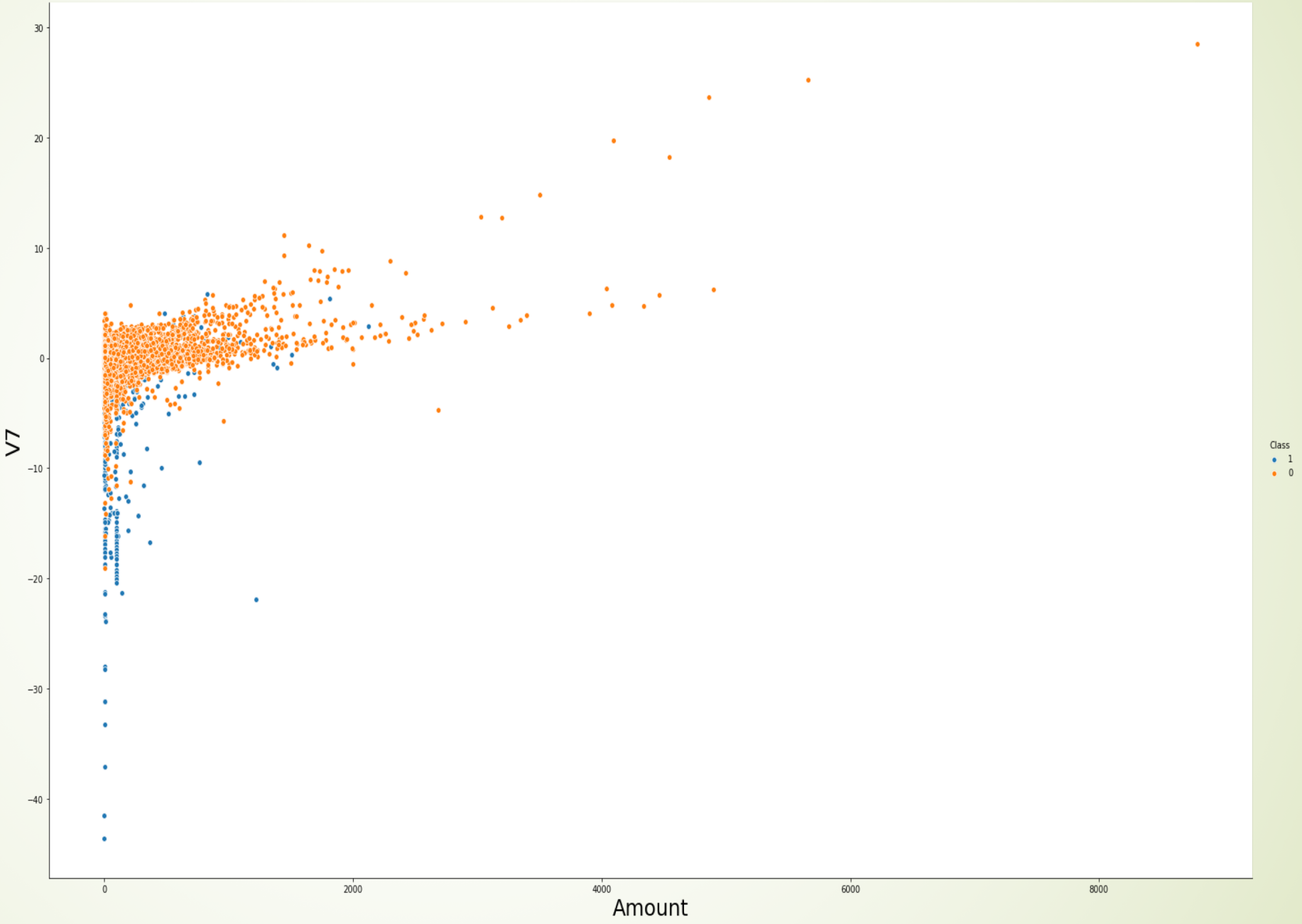


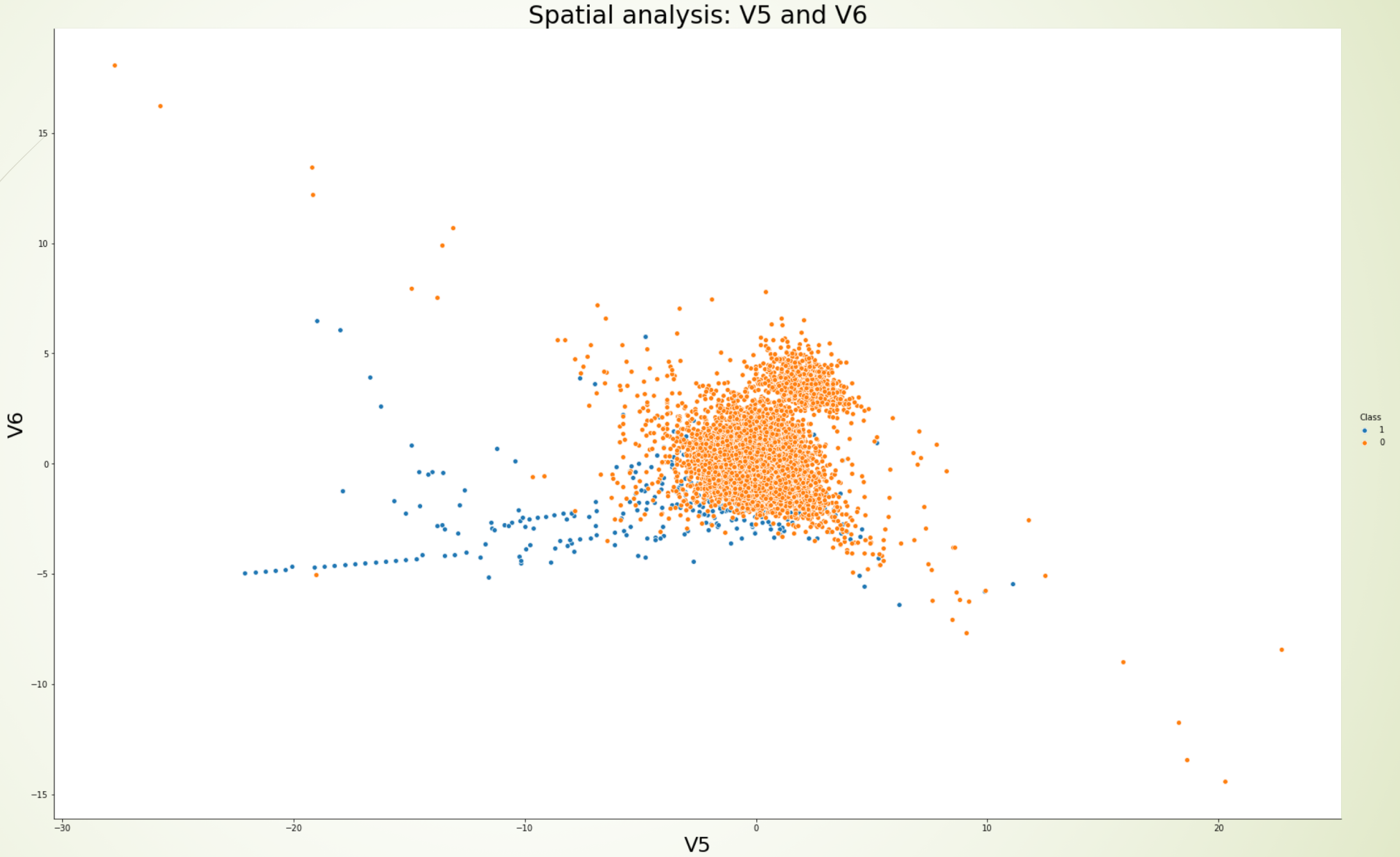
Variables with high correlation to class like V14 follow different shapes for fraud and non-fraud




V17's Fraud and Non-Fraud do not have the same shape
Fruad
Not Fraud
Percentage of Observations
V17
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
−30
−20
−10
0
10

Are there any noticeable clusters?




Spatial analysis: V5 and V6
V6
V5
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
Class
1
0

What does my model want to optimize

- False negative: Fraud occurred and was not caught
- False positive: Fraud did not occur but we acted as it did.

- False negatives are worst case scenario so we want a model that optimizes our recall score.

Training and Testing my Data

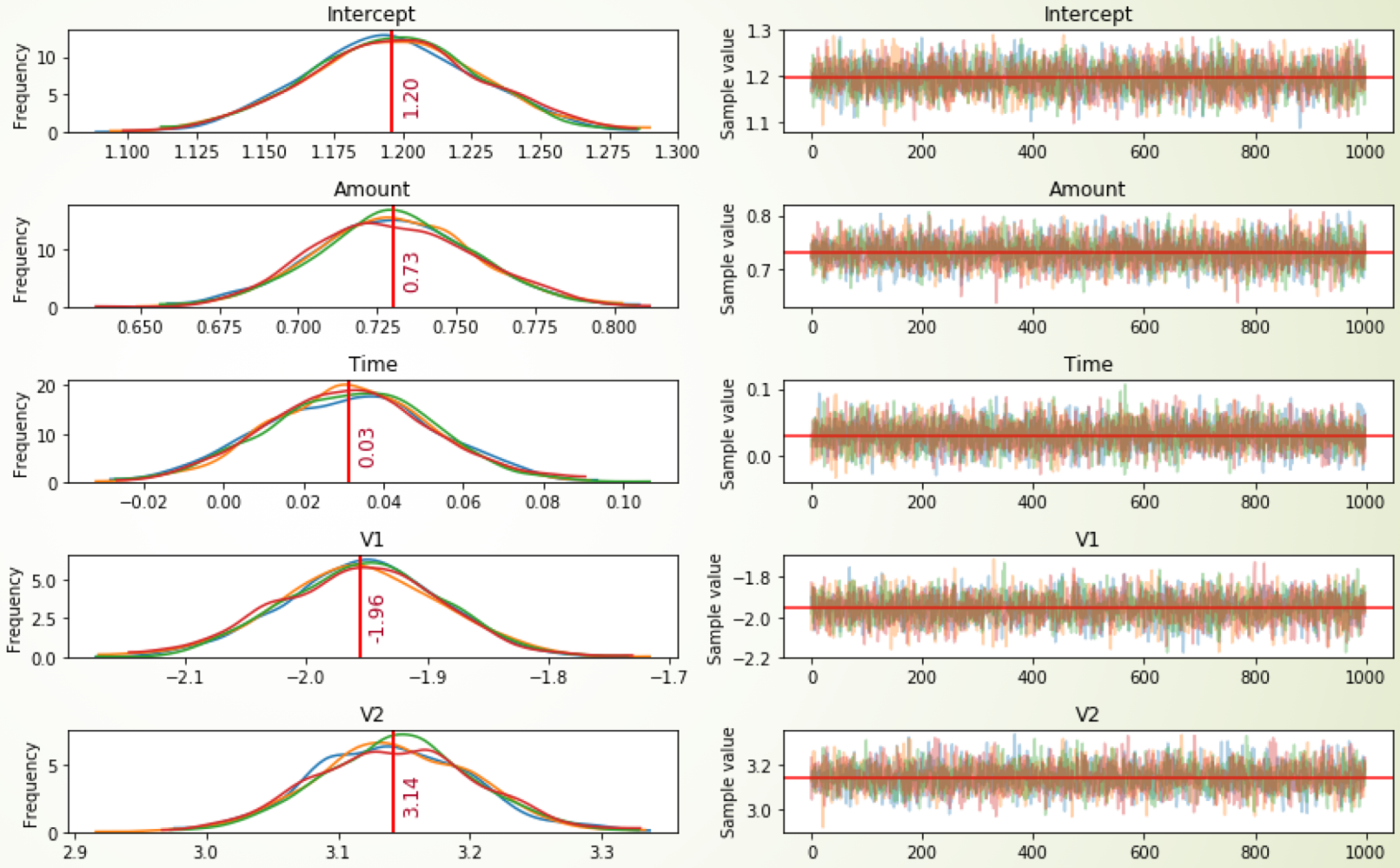
- I created a subset of my data by randomly selecting 50,000 rows and training my data and then testing my models on the entire dataset.
- I also used SMOTE to underrepresent the majority class and overrepresent the minority class.
- Allowed me to have 50 50 classes.

Frequentist Models Used

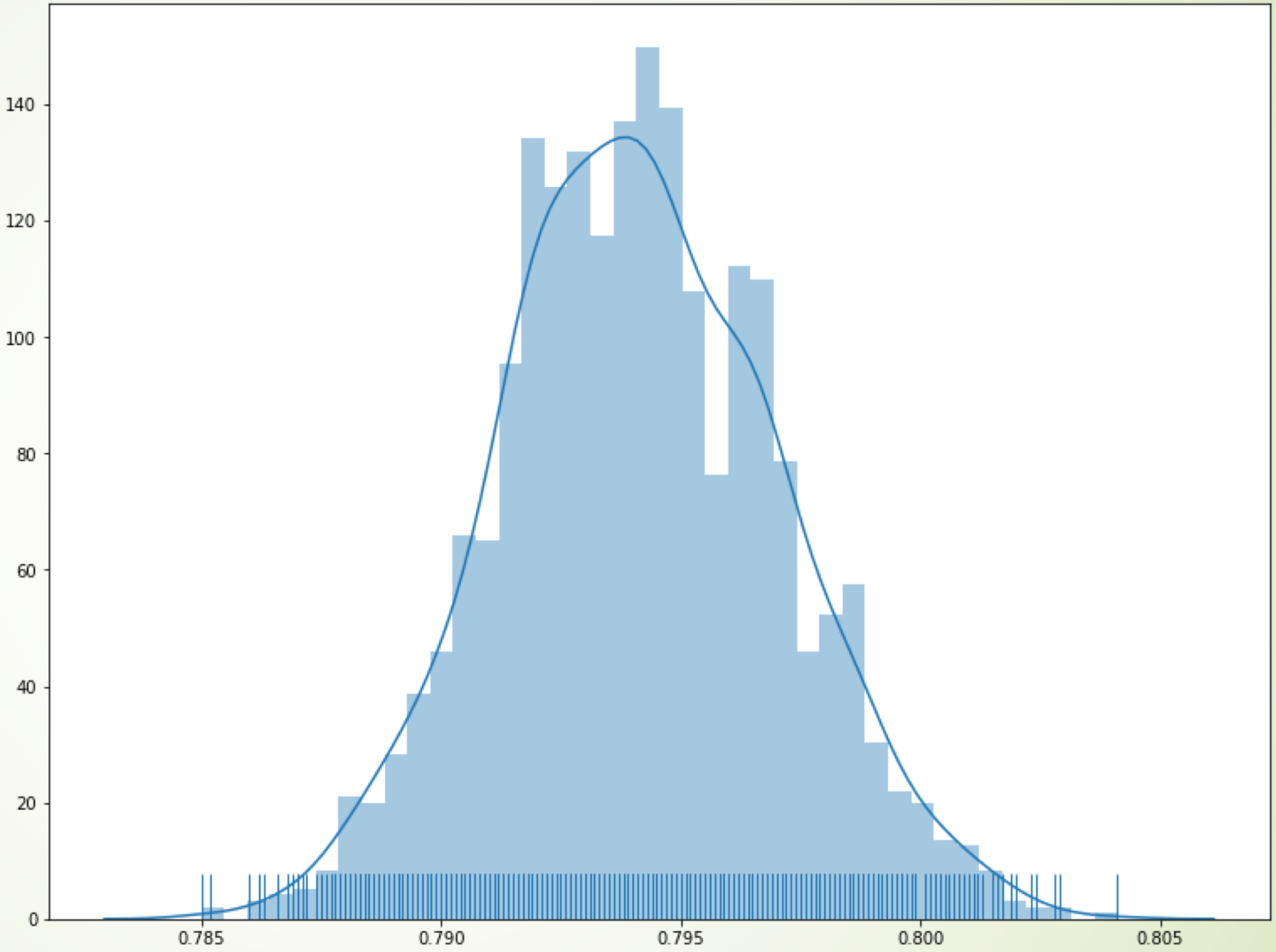
- Logistic Regression (50 false negatives, 2512 false positives)
- Random Forest (0 false negatives, 164 false positives)
- Gradient Boosting (0 false negatives, 273 false positives)
- XG Boost (0 false negatives, 182 false positives)
- K Nearest Neighbors (0 false negatives, 1312 false positives)
- Support Vector Machine (51 false negatives, 2029 false positives)
- K-Means (unsupervised, grouped them all together)

Bayesian Logistic Regression

- For each Beta there is a distribution for possible values it could observe.

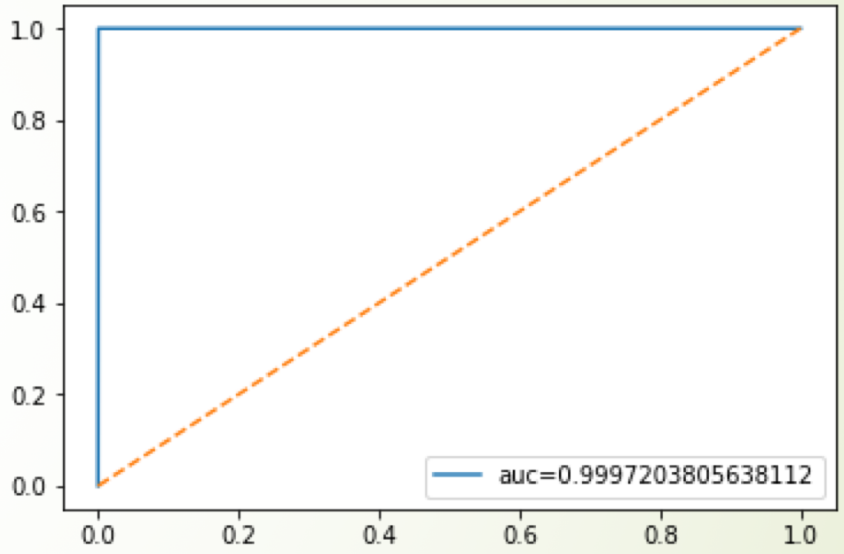


Results



Conclusions

- Use Random Forest to model as the accuracy score is .9994 and I am able to get 0 false negatives.
- Boosting methods work well but do not outperform random forest.
- Unsupervised models such as K-Means do not work well
- Bayesian models did not outperform their frequentist counterparts.



Going Forward

- See if my Bayesian can improve by changing my priors
- Use flask so this model could be implemented.
- Get information on what the PCA'd columns represent.