Credit Card Fraud Detection

RASHMI S-21BDA02
APARNA K-21BDA24
TONY TOMY-21BDA44

- Trust your instincts with us

The Need

- In 2019, credit card fraud losses amounted to $28.65 billion worldwide

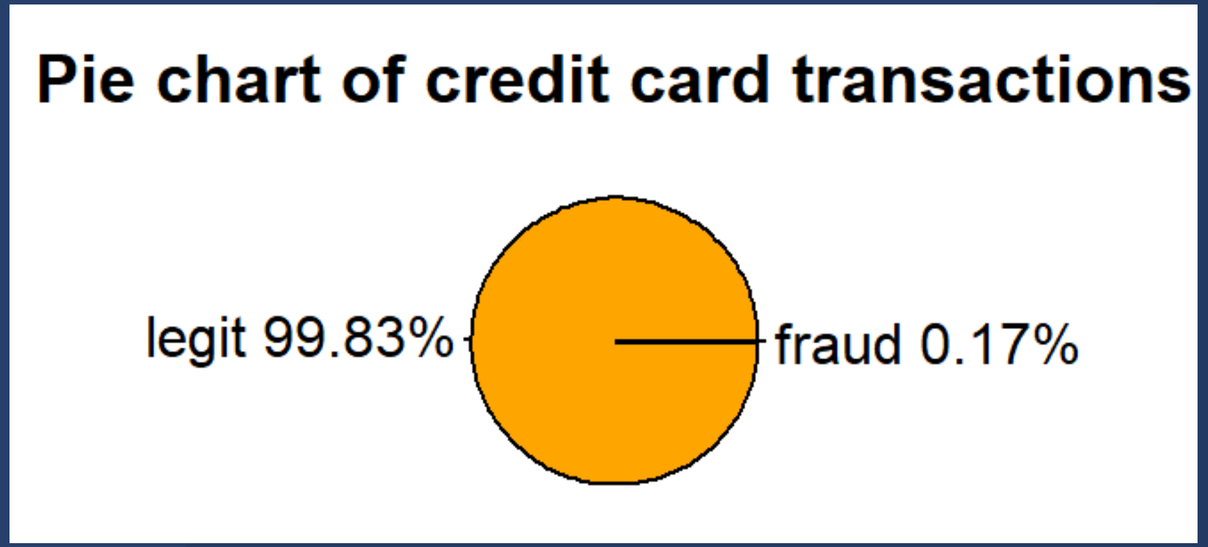
- Companies, such as VISA, are looking for AI/ML-based solutions to combat this issue.

- The goal of this project was to automate the fraud detection process in order to save time on verification methods. To find correlations in our data so as to make accurate predictions.

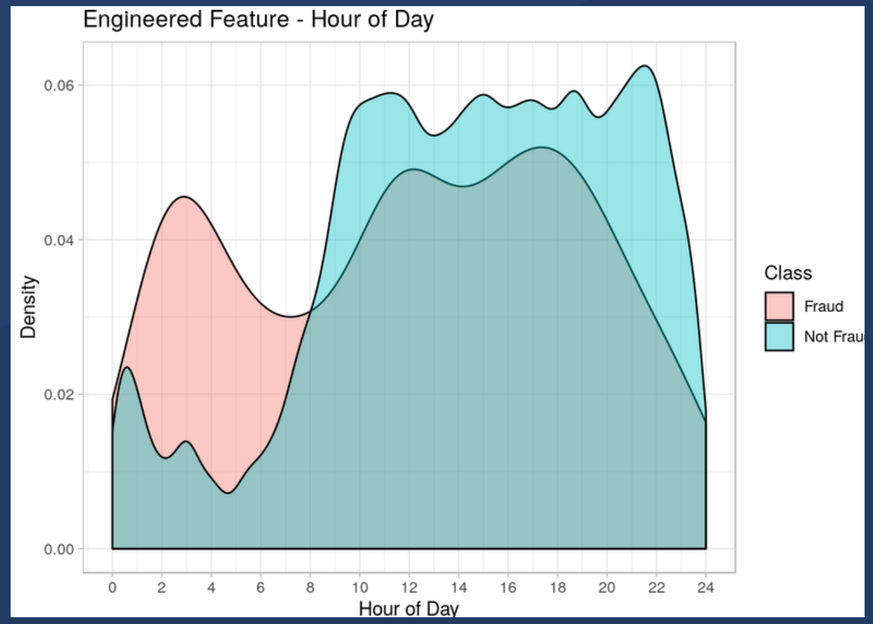
Let's Understand our data!

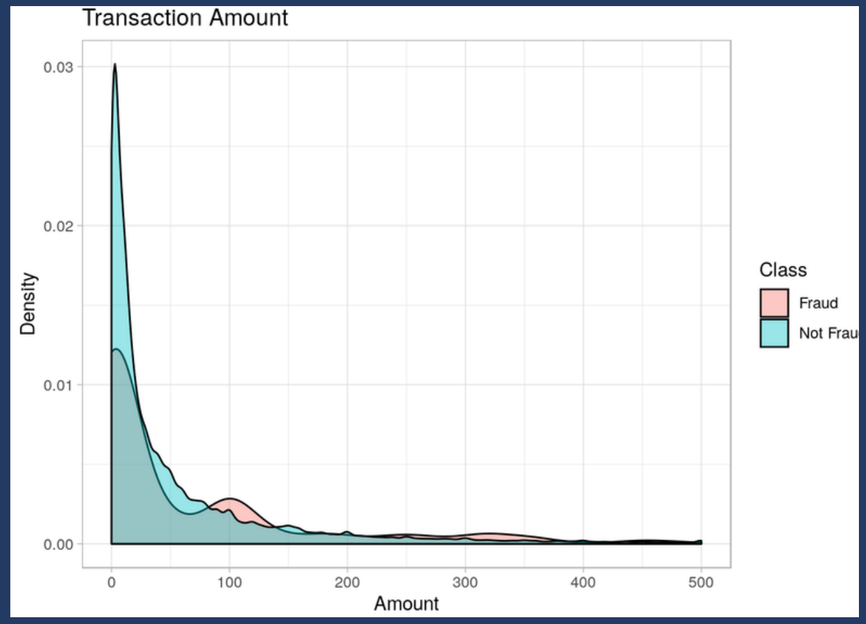
- **(V1 — V28) - masked using Principal Component Analysis (PCA) (confidential)**
- **Time - Represents the seconds elapsed between the current transaction and the first transaction in the dataset**
- **Amount - Represents the total transaction value (In thousands)**
- **Class - consists of two values:**
  1 for fraudulent cases **0 for non-fraudulent cases**

Oshh…
The data is imbalanced



Insights…



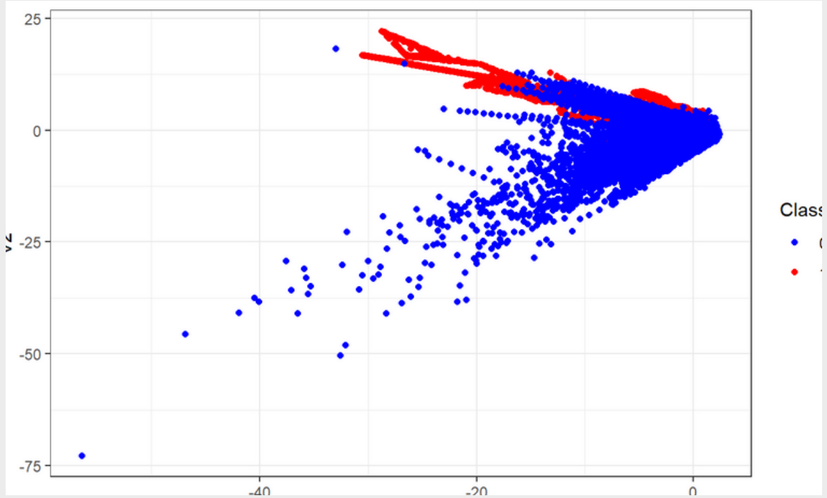


Smote The Rescuer

- **SMOTE (Synthetic Minority Oversampling Technique) is an oversampling approach to the minority class.**

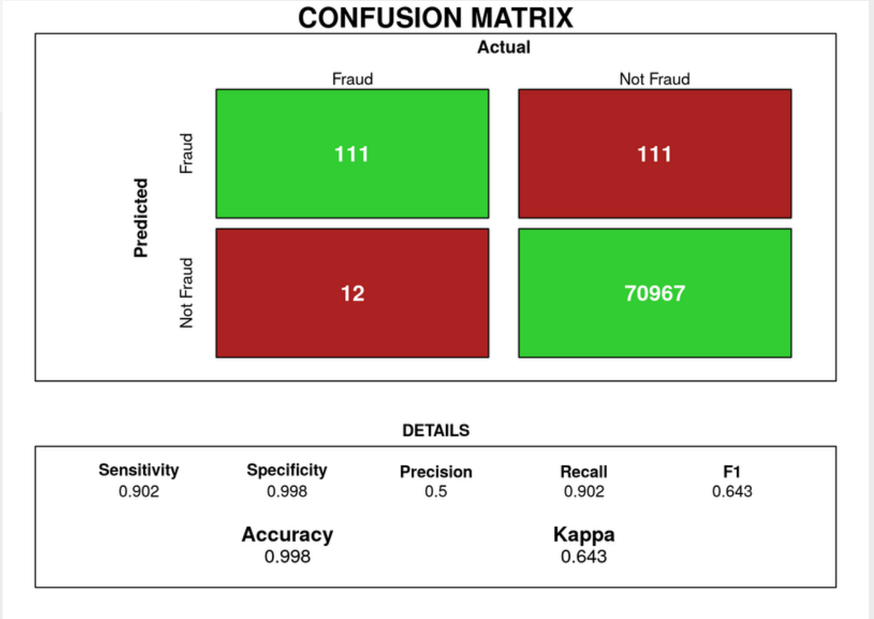
- **In context, it would mean randomly increasing fraud examples by "artificially" replicating to have a more balanced class distribution. It uses the KNN approach**

- **To balance the dataset, we tried Random oversampling, Random Under sampling and finally found SMOTE to be the most efficient technique to balance the dataset.**



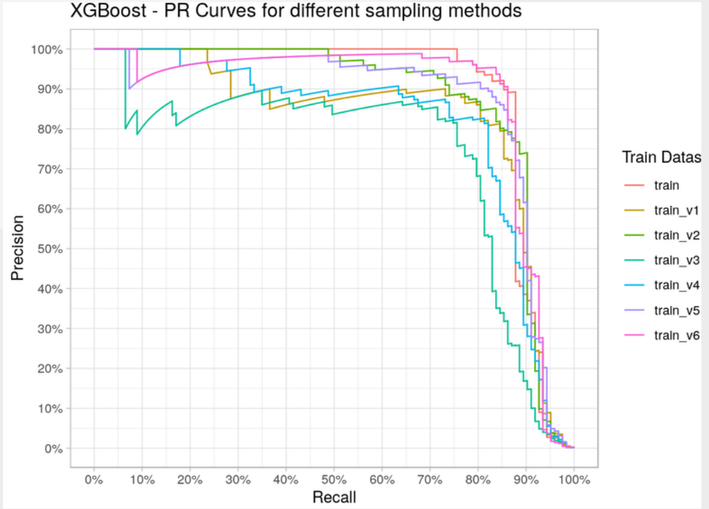
XGboost

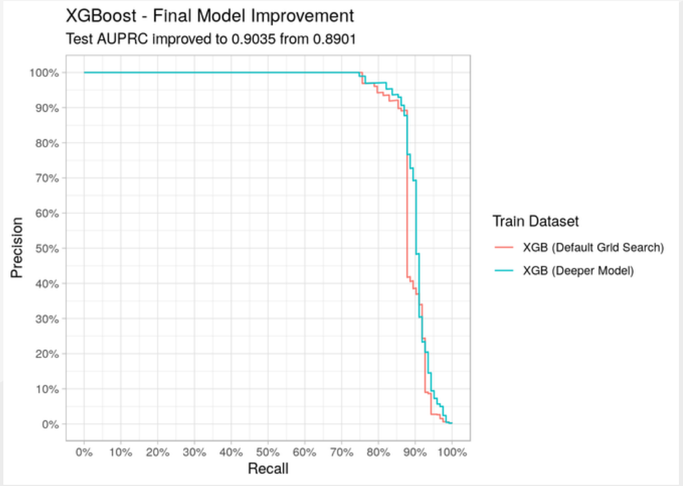
- A supervised learning algorithm based on the concept of trees
- The area under the PR curve was found to be 0.903
- A fraud-minority dataset with more up-sampling



Details of the methadology

name <chr>	num_obs <dbl>	frauds <dbl>	frauds_perc <dbl>	weighting <chr>
train	213606	369	0.00172748	original (very imbalar
train_v1	3690	1845	0.50000000	balanced
train_v2	22140	11070	0.50000000	balanced
train_v3	2460	1845	0.75000000	mostly fraud
train_v4	14760	11070	0.75000000	mostly fraud
train_v5	7380	1845	0.25000000	mostly non-fraud
train_v6	44280	11070	0.25000000	mostly non-fraud





- eta = 0.05
- nrounds = 1400
- max_depth = 5
- min_child_weight = 1
- gamma = 0
- colsample_bytree = 0.8
- subsample = 0.6

Let's be assured!

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Extreme Gradient Boosting	0.9995	0.9839	0.7982	0.9059	0.8437	0.8435
1	CatBoost Classifier	0.9995	0.9831	0.7773	0.914	0.831	0.8308
2	Random Forest Classifier	0.9994	0.9333	0.7418	0.9357	0.8172	0.8169
3	Extra Trees Classifier	0.9994	0.9504	0.7509	0.9285	0.8223	0.822
4	Decision Tree Classifier	0.9993	0.8899	0.78	0.8352	0.8015	0.8012
5	Ada Boost Classifier	0.9993	0.9612	0.6918	0.8713	0.7676	0.7672
6	Linear Discriminant Analysis	0.9993	0.9906	0.7618	0.8522	0.7997	0.7994
7	Gradient Boosting Classifier	0.999	0.7742	0.5973	0.7861	0.6594	0.6589
8	Logistic Regression	0.9988	0.919	0.5955	0.6875	0.6337	0.6332
9	Ridge Classifier	0.9988	0	0.4555	0.7949	0.5717	0.5712
10	K Neighbors Classifier	0.9982	0.5763	0	0	0	-0
11	SVM - Linear Kernel	0.9981	0	0	0	0	-0.0002
12	Naive Bayes	0.9937	0.9826	0.6745	0.1734	0.275	0.273
13	Light Gradient Boosting Machine	0.9933	0.5241	0.23	0.0878	0.1205	0.1183
14	Quadratic Discriminant Analysis	0.9835	0.9762	0.9145	0.0699	0.1635	0.1609

Conclusion

- 111 + 12 = 123 frauds, therefore ~ 246 frauds per day
- Before model cost = (£145.61)(246)(365) = £13,074,440.52
- With our final model, we achieved half-day errors of 12 FN’s and 49 FP’s, with a half-day business cost of £1874.74
- New annual business cost = (£1874.74)(2)(365) = £1,368,557.28

Therefore, an upper bound estimate for annual cost-savings is:
£13,074,440.52 - £1,368,557.28 =~ £11.7M
This is a reduction in fraud-handling costs of ~90%!

Thank you