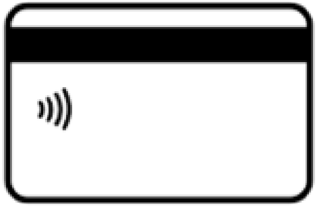
Credit Card Fraud Detection

Group 4: Jiayi Yang, Neal Xu, Suenkei Chan, Yao Zhang, Zhuofan Dong



Agenda

Business Problem

Definitions of Variables & Data Exploration

Methodology & Analytics Approach

Results & Recommendation

Business Problem

- **Background:** Fraudulent activity—**problem** for financial institutions that issue **credit cards.**
- **Problem: Insufficient detection** → **loss of customers' confidence** in the security mechanisms

  → **churn** of customers.

  Excessive alerts → **lack of confidence in** the detection algorithms as unreliable
- **Value of Solution: increase** customers' **stickiness**
- **Purpose and Objective:** develop **anomaly detection mechanisms**（(Extreme Imbalanced Dataset): make the right fraud detection and reduce false fraud alerts.

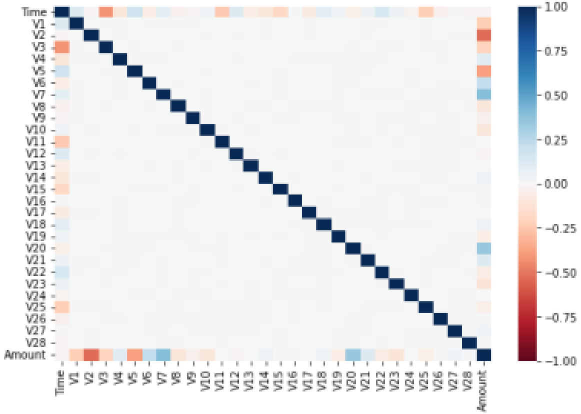
Exploratory Analysis

0: normal
1: fraud

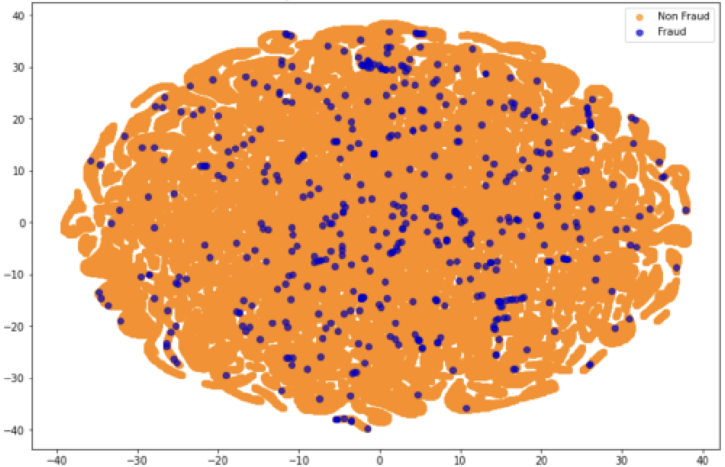
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1 [illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1								[illegible]	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								[illegible]	-0.206010	0.502292	0.219422	0.215153	69.99	0

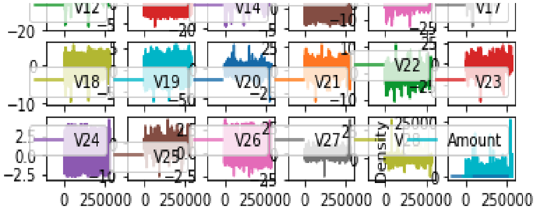
5 rows × 31 columns

	Time	V1	V2	V3		V21	V22	V23	V24	V25
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.	848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.918649e-15	5.682686e-16	-8.761736e-15	2	.473120e-16	8.042109e-16	5.282512e-16	4.456271e-15	1.426896e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.	.345240e-01	7.257016e-01	6.244603e-01	6.056471e-01	5.212781e-01
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.	483038e+01	-1.093314e+01	-4.480774e+01	-2.836627e+00	-1.029540e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8	.283949e-01	-5.423504e-01	-1.618463e-01	-3.545861e-01	-3.171451e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1	.945017e-02	6.781943e-03	-1.119293e-02	4.097606e-02	1.659350e-02
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7	.863772e-01	5.285536e-01	1.476421e-01	4.395266e-01	3.507156e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.	720284e+01	1.050309e+01	2.252841e+01	4.584549e+00	7.519589e+00



TSNE plot of Fraud and Non Fraud

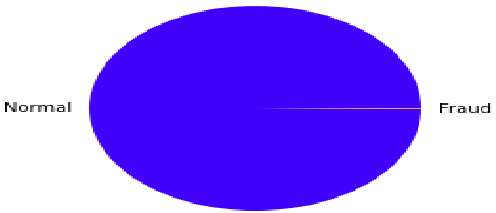




```
f['Class'].value_counts()
```

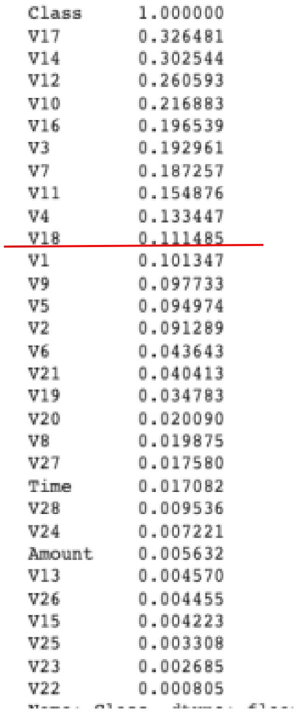
```
     284315
      492
ame: Class, dtype: int64
```

Normal vs fraudulent transactions



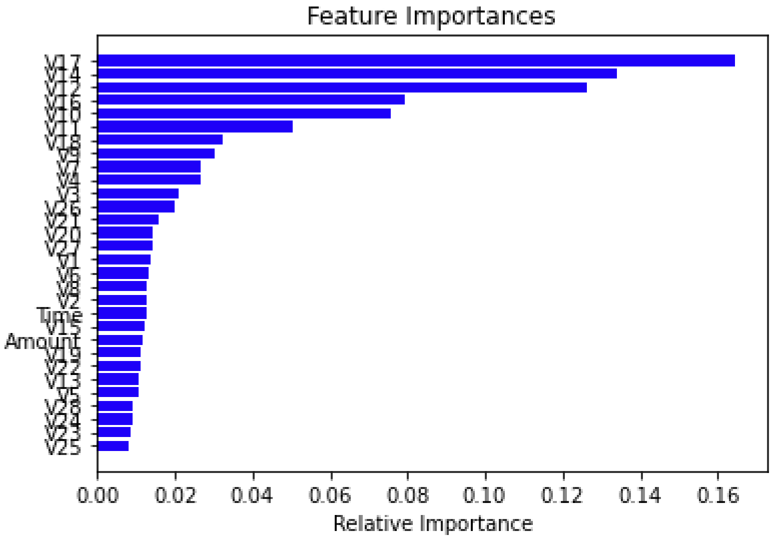
THE UNIVERSITY OF CHICAGO

Default Correlation with other Variables



```
Class     1.000000
V17       0.326481
V14       0.302544
V12       0.260593
V10       0.216883
V16       0.196539
V3        0.192961
V7        0.187257
V11       0.154876
V4        0.133447
V18       0.111485
V1        0.101347
V9        0.097733
V5        0.094974
V2        0.091289
V6        0.043643
V21       0.040413
V19       0.034783
V20       0.020090
V8        0.019875
V27       0.017580
Time      0.017082
V28       0.009536
V24       0.007221
Amount    0.005632
V13       0.004570
V26       0.004455
V15       0.004223
V25       0.003308
V23       0.002685
V22       0.000805
Name: Class, dtype: floa
```

Feature Selection



Generally, the features revealed from Random Forest are more convincing as it takes linear and nonlinear relationship into consideration.

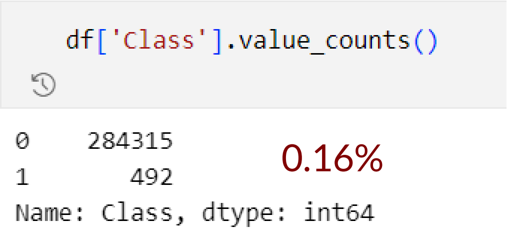
Models Performance

No Resampling

Over-Sampling (SMOTE)

Weighted Learning

Nonlinear:
XGboosting
Decision Tree
Random Forest
Neural Network
Linear:
Logistic Regression

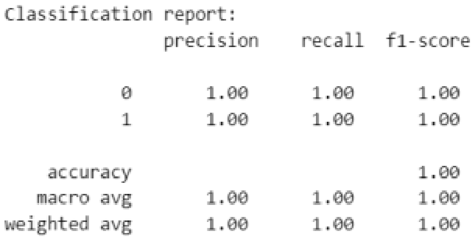


```
df['Class'].value_counts()
```

```
0    284315
1       492
Name: Class, dtype: int64
```

0.16%

All Leads to Overfitting:



```
Classification report:
              precision    recall  f1-score

           0       1.00      1.00      1.00
           1       1.00      1.00      1.00

    accuracy                           1.00
   macro avg       1.00      1.00      1.00
weighted avg       1.00      1.00      1.00
```

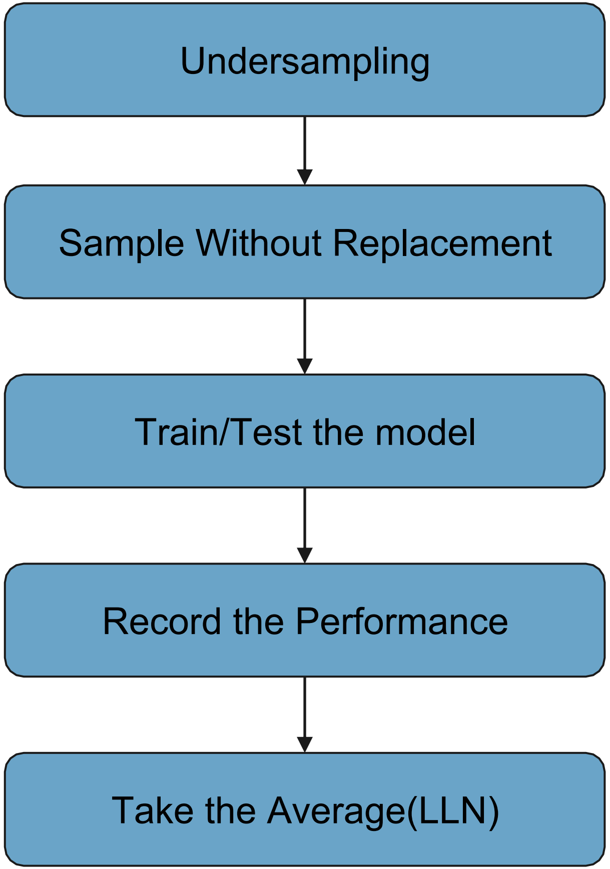
Model(for loop)

Advantages:

- Balanced data for both categories
- Train and test the models with different data in each loop without replacement
- Average performance score tells the truth of the model performance

Disadvantage:

- Size of training set may not be enough to fully train the models
- Hard to use the models with large computing resources such as deep learning



Undersampling

Sample Without Replacement

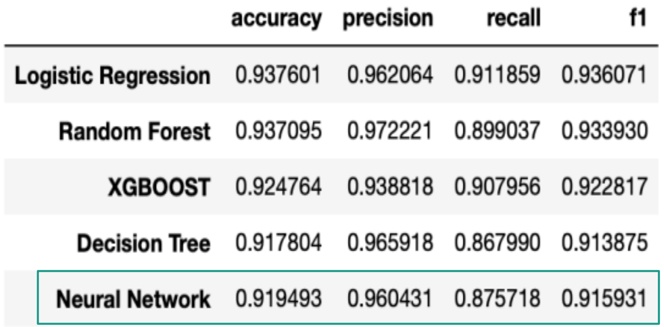
Train/Test the model

Record the Performance

Take the Average(LLN)

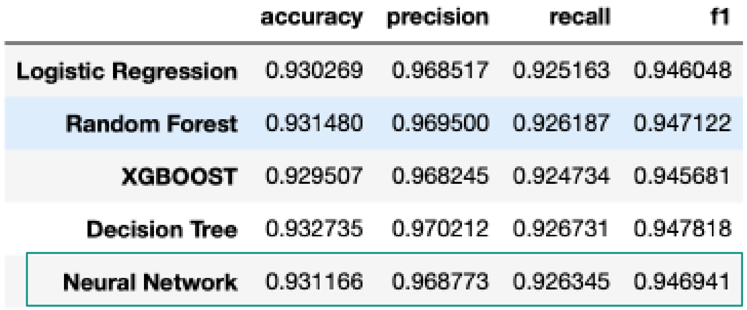
Model(for loop)

Label 0: 492 Label1: 492

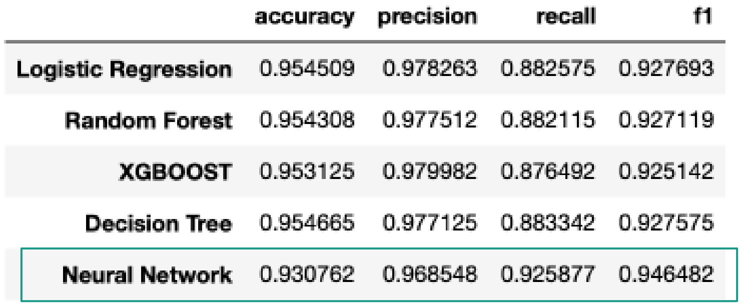


	accuracy	precision	recall	f1
Logistic Regression	0.937601	0.962064	0.911859	0.936071
Random Forest	0.937095	0.972221	0.899037	0.933930
XGBOOST	0.924764	0.938818	0.907956	0.922817
Decision Tree	0.917804	0.965918	0.867990	0.913875
Neural Network	0.919493	0.960431	0.875718	0.915931

Label 0: 250 Label1: 492



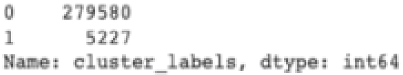
	accuracy	precision	recall	f1
Logistic Regression	0.930269	0.968517	0.925163	0.946048
Random Forest	0.931480	0.969500	0.926187	0.947122
XGBOOST	0.929507	0.968245	0.924734	0.945681
Decision Tree	0.932735	0.970212	0.926731	0.947818
Neural Network	0.931166	0.968773	0.926345	0.946941



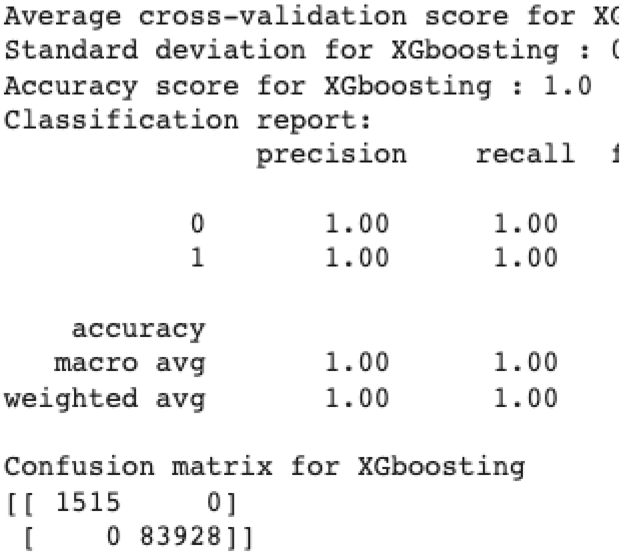
	accuracy	precision	recall	f1
Logistic Regression	0.954509	0.978263	0.882575	0.927693
Random Forest	0.954308	0.977512	0.882115	0.927119
XGBOOST	0.953125	0.979982	0.876492	0.925142
Decision Tree	0.954665	0.977125	0.883342	0.927575
Neural Network	0.930762	0.968548	0.925877	0.946482

Label 0: 1000 Label1: 492

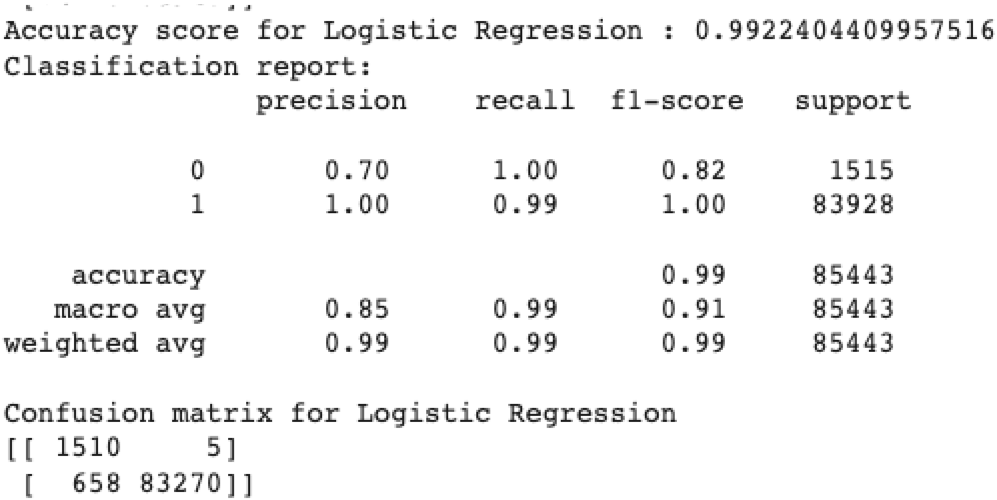
Model in Semi-Supervised Learning



```
0    279580
1      5227
Name: cluster_labels, dtype: int64
```



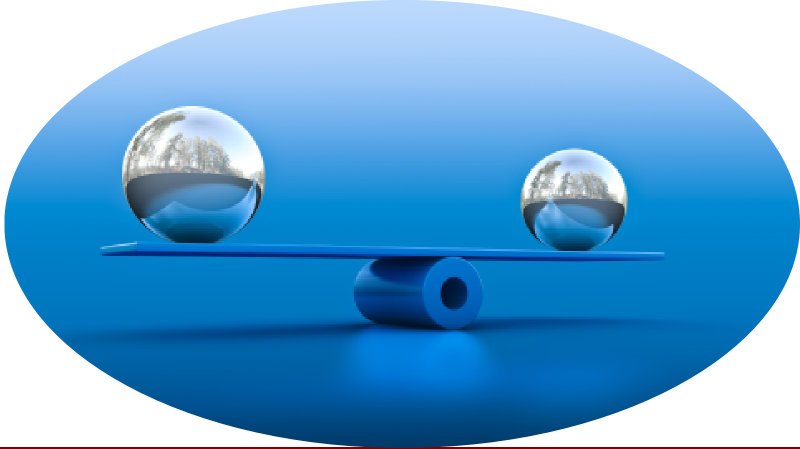
```
Average cross-validation score for XG
Standard deviation for XGboosting : 0
Accuracy score for XGboosting : 1.0
Classification report:
              precision    recall  f

           0       1.00      1.00
           1       1.00      1.00

    accuracy
   macro avg       1.00      1.00
weighted avg       1.00      1.00

Confusion matrix for XGboosting
[[ 1515     0]
 [    0 83928]]
```



```
Accuracy score for Logistic Regression : 0.9922404409957516
Classification report:
              precision    recall  f1-score   support

           0       0.70      1.00      0.82      1515
           1       1.00      0.99      1.00     83928

    accuracy                           0.99     85443
   macro avg       0.85      0.99      0.91     85443
weighted avg       0.99      0.99      0.99     85443

Confusion matrix for Logistic Regression
[[ 1510     5]
 [  658 83270]]
```

Result Summary

- Useful in situation where fraud has a high cost (We have a high recall around 0.98)
- Approaches and models in supervised learning
- Approaches in semi-supervised learning
- Problematic
- Limited size of positive case; deeply imbalance dataset
- The dataset is continent-based
- The label is inaccurate



Future Improvement

- Different Dataset
- Advanced Models to detect the hidden patterns (Deep Learning)
- Expected Loss
- Novelty Detection