Credit Card Fraud Detection

Thang Truong

Introduction

114, 348

cases in the US

$30 billion dollars

global credit card loss

Leleko, S., & Holoborodko, Y. (2024, April 16). Credit Card Fraud Detection Using Machine Learning. Retrieved from https://spd.tech/machine-learning/credit-card-fraud-detection/

Data Compiling

- The dataset obtained from Kaggle contains transactions made by credit cards in September 2013 by European cardholders.

- **31 columns** and **284,807 rows.**

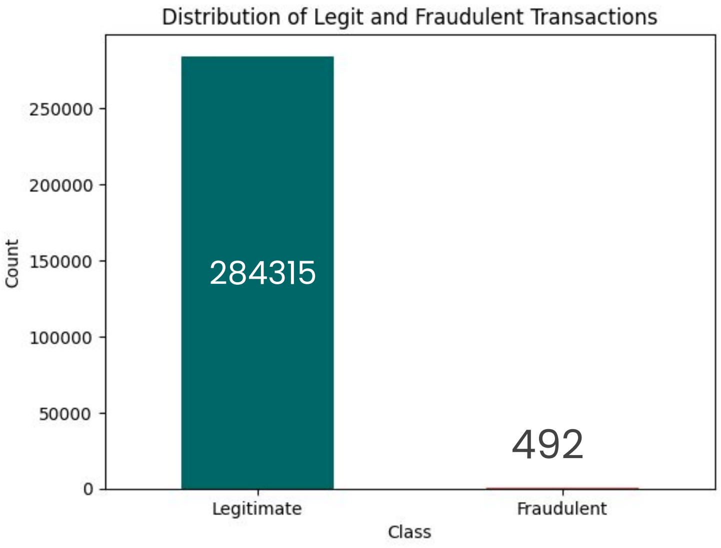
- Contains only numeric input variables which are the result of a PCA transformation (due to confidentiality)

	Time	V1	V2	V3	V4	V5	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0.0
1	0	1.191857	0.266151	0.166480	0.448154	0.060018	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0.0
2	1	-1.358354	-1.340163	1.773209	0.379780	-0.503198	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0.0
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0.0
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0.0

5 rows × 31 columns

Data Validation

- No missing values detected
- Extreme imbalance of class for the dataset




```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

MinMax Scale

Before

	Time	V1	V2	V3	V4	V5	V6	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.84807
mean	94813.859575	1.168375e-15	3.416908e-16	-1.379537e-15	2.074095e-15	9.604066e-16	1.487313e-15	-5.55646
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.54075
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.01030
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.70436
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895

After

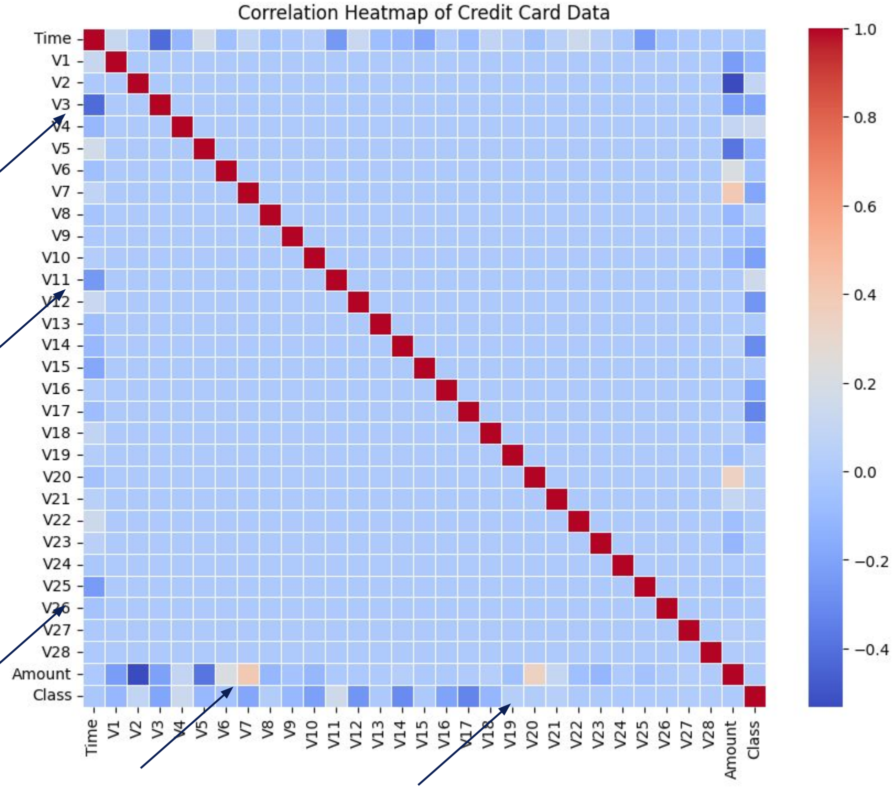
	Time	V1	V2	V3	V4	V5	V6	
count	284807.000000	284807.000000	284807.000000	284807.000000	284807.000000	284807.000000	284807.000000	284
mean	0.548717	0.958294	0.767258	0.837414	0.251930	0.765716	0.263020	
std	0.274828	0.033276	0.017424	0.026275	0.062764	0.009292	0.013395	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.313681	0.942658	0.760943	0.821985	0.214311	0.761060	0.255295	
50%	0.490138	0.958601	0.767949	0.840530	0.251050	0.765351	0.260263	
75%	0.806290	0.980645	0.775739	0.855213	0.284882	0.769836	0.267027	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

8 rows × 31 columns

Correlation Heatmap

- V7, V20 positively correlates with Amount column.

- V3, V11, V25 negatively correlates with Time column.



SQL Exploration with PySpark

```
[ ] # Q2: What is the average amount in the class 0 (non-fraudulent transactions)?
    # 88.29
    result = spark.sql("SELECT AVG(AMOUNT) FROM credit_card_fraud_table WHERE Class = 0")
    result.show()
```

```
+-----------------+
|      avg(AMOUNT)|
+-----------------+
|88.29102242233286|
+-----------------+
```

```
# Q3: What is the average amount in the class 1 (fraudulent transactions)?
# 122.2
result = spark.sql("SELECT AVG(AMOUNT) FROM credit_card_fraud_table WHERE Class = 1")
result.show()
```

```
+------------------+
|       avg(AMOUNT)|
+------------------+
|122.21132113821136|
+------------------+
```

Data balancing method

Random Undersampling

Randomly deletes examples from the majority class.
656 rows for training set. Class 0: 328 rows. Class 1: 328 rows

NearMiss Undersampling

Deletes examples from the majority class that are closest to the minority class in the feature space (smallest average distance)
656 rows for training set. Class 0: 328 rows. Class 1: 328 rows

SMOTE Oversampling

Synthetic Minority Over-sampling Technique generates synthetically the minority class by interpolating between existing examples.
379086 rows for training set. Class 0: 189542 rows. Class 1:

Model

Random Forest

- Constructing multiple decision trees and output a class that is the mode or mean of those trees

Logistic Regression

- Predicts the probability of a given input belongs to one of the two outcomes

Neural Network (Multilayer Perceptron)

- A class of Artificial Neural Network that composed of multiple layers of interconnected nodes (neurons)

Evaluation

Logistic Regression

3-fold cross validation

	Mean Accuracy	Mean Precision	Mean Recall	Mean F-1 Score	Mean AUC Score
SMOTE Oversampling	0.952	0.061	**0.914**	0.111	0.9750
NearMiss Undersampling	0.98	0.377	0.839	0.45	0.970
Random Undersampling	**0.993**	**0.426**	0.829	**0.498**	**0.9751**

Random Forest

	Accuracy	Precision	Recall	F-1 Score	AUC
SMOTE Oversampling	0.846	**0.632**	0.542	**0.281**	0.924
NearMiss Undersampling	0.41	0.002	**0.95**	0.006	0.945
Random Undersampling	**0.943**	0.054	0.91	0.114	**0.976**

Neural Network

	Accuracy	Precision	Recall	F-1 Score	AUC
SMOTE Oversampling	0.929	0.06	**0.917**	0.106	**0.978**
NearMiss Undersampling	0.829	0.0526	0.900	0.09	0.969
Random Undersampling	**0.953**	**0.0855**	0.898	**0.149**	0.973

Comparing between different ML models

	Mean Accuracy	Mean Precision	Mean Recall	Mean F1 score	AUC Score
Random Undersampling (Logistic Regression)	**0.993**	0.426	0.829	**0.498**	**0.9751**
SMOTE (Random Forest)	0.846	**0.632**	0.542	0.281	0.924
Random Undersampling (Neural Network)	0.953	0.0855	**0.898**	0.149	0.973

Limitation and Future Improvement

- Class imbalance in the test set

-> Finding data that overcome the imbalance in the class of test set

- Computationally expensive (because of k-fold cross validation with complicated ML)

-> Utilize better cloud computing, experiment with simpler model, reduce dimensionality

Conclusion

The recommended approach that balances between F-1 score, AUC score and other metrics is **random undersampling with logistic regression.**

Recommendation for credit card fraud detection includes

- **Real time monitoring of fraud detection** with real-time feature extraction and model inference
- **Continuous reevaluation of models** to adapt with evolving fraud patterns
- **Overcome imbalance of dataset**