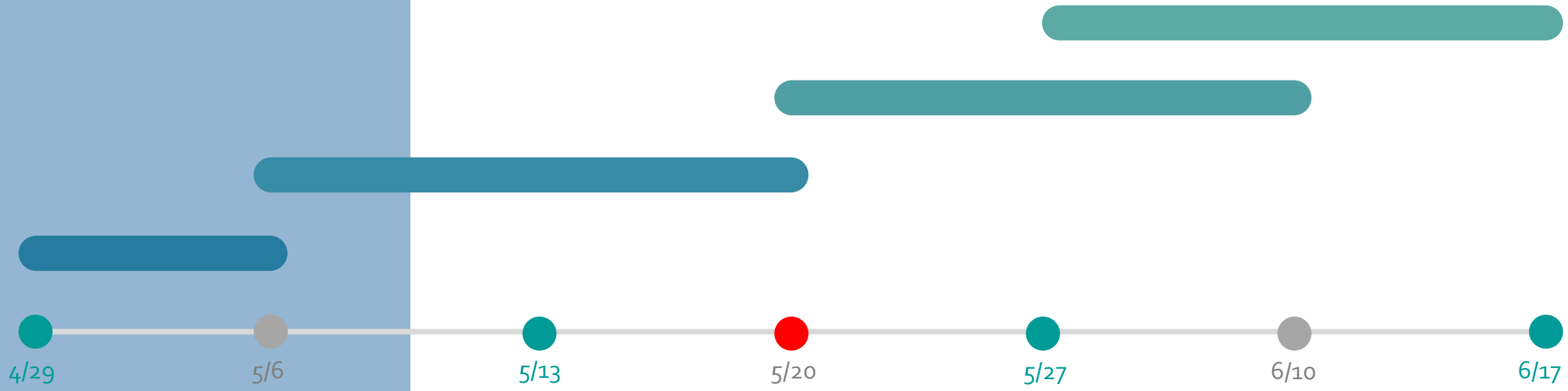
連假隊

7109018022 魏良育

7109018028 魏敏如

Gantt Chart



探索資料

資料視覺化
檢視資料遺失值
檢視資料不平衡狀況

資料前處理

標準化資料
Over-sampling
Under-sampling

模型建立

使用不同模型比較預測準確率

模型參數調整

此处添加详细文本描述，文字内容建议与标题相关尽量简洁生动……

Kaggle – **Credit Card Fraud Detection**

- 資料集來源 :
  https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- 簡介 :
  信用卡公司能夠識別欺詐性信用卡交易非常重要，這樣客戶就不會為他們沒有購買的商品付費。
- 信用卡欺詐，也作為信用欺詐，廣泛描述信貸或銀行信息的盜竊．盜賊使用訊息來製作欺詐性交易或獲得不當利益．

About dataset

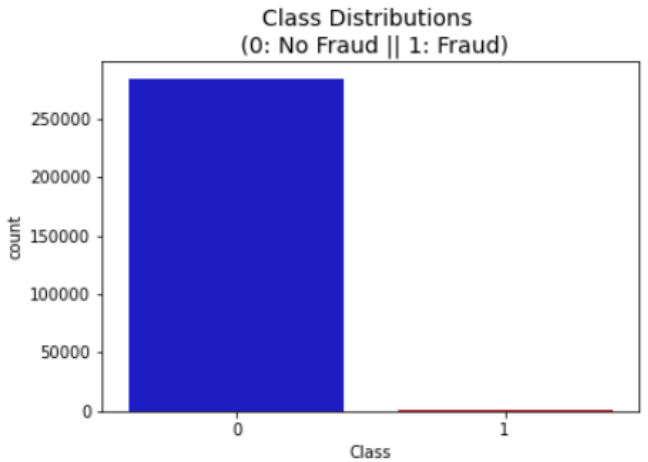
- 歐洲持卡人在 2013 年 9 月通過信用卡進行的交易。
- 284,807 筆交易中有 492 筆欺詐。
  數據集高度不平衡，正類（欺詐）佔所有交易的 0.172%。
- Feature :
  v1-v28 : 為使用PCA獲得的主成分
  Time : 每次交易與第一次交易之間經過的秒數
  Amount : 交易金額
- Label :
  Class : 1為Fraud、0:為Not Fraud

	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	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

5 rows × 31 columns

了解資料集

1. 交易金額相對較小。平均值約為88美元。
2. 沒有 “null”值。
3. 大多數交易是非欺詐（99.83%）的時間，
   而DataFrame中的時間發生欺詐事務（0.17%）。



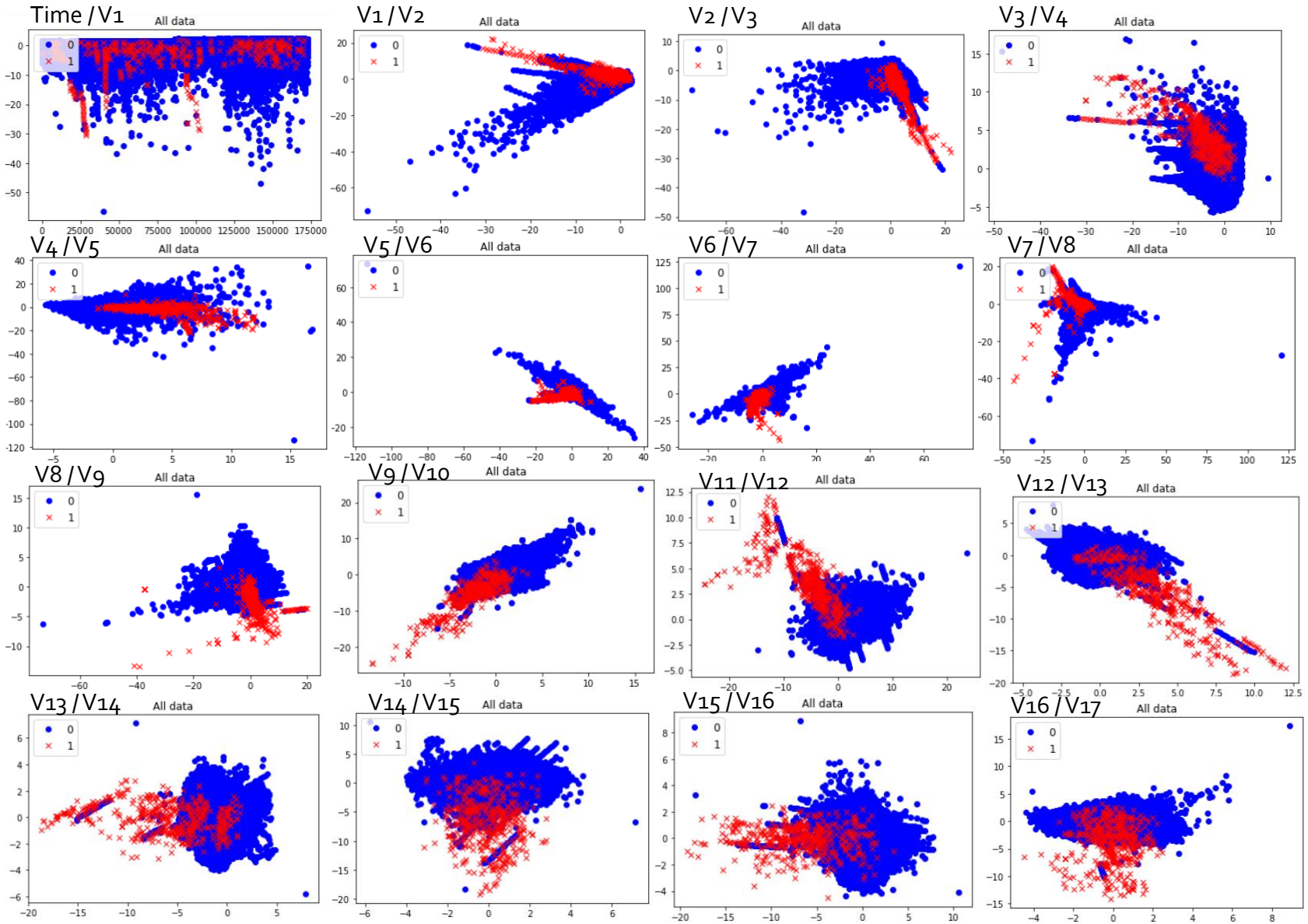
No Frauds 99.83 % of the dataset
Frauds 0.17 % of the dataset
Amount of Label 0: 284315
Amount of Label 1: 492

	Amount
count	284807.000000
mean	88.349619
std	250.120109
min	0.000000
25%	5.600000
50%	22.000000
75%	77.165000
max	25691.160000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 28480
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

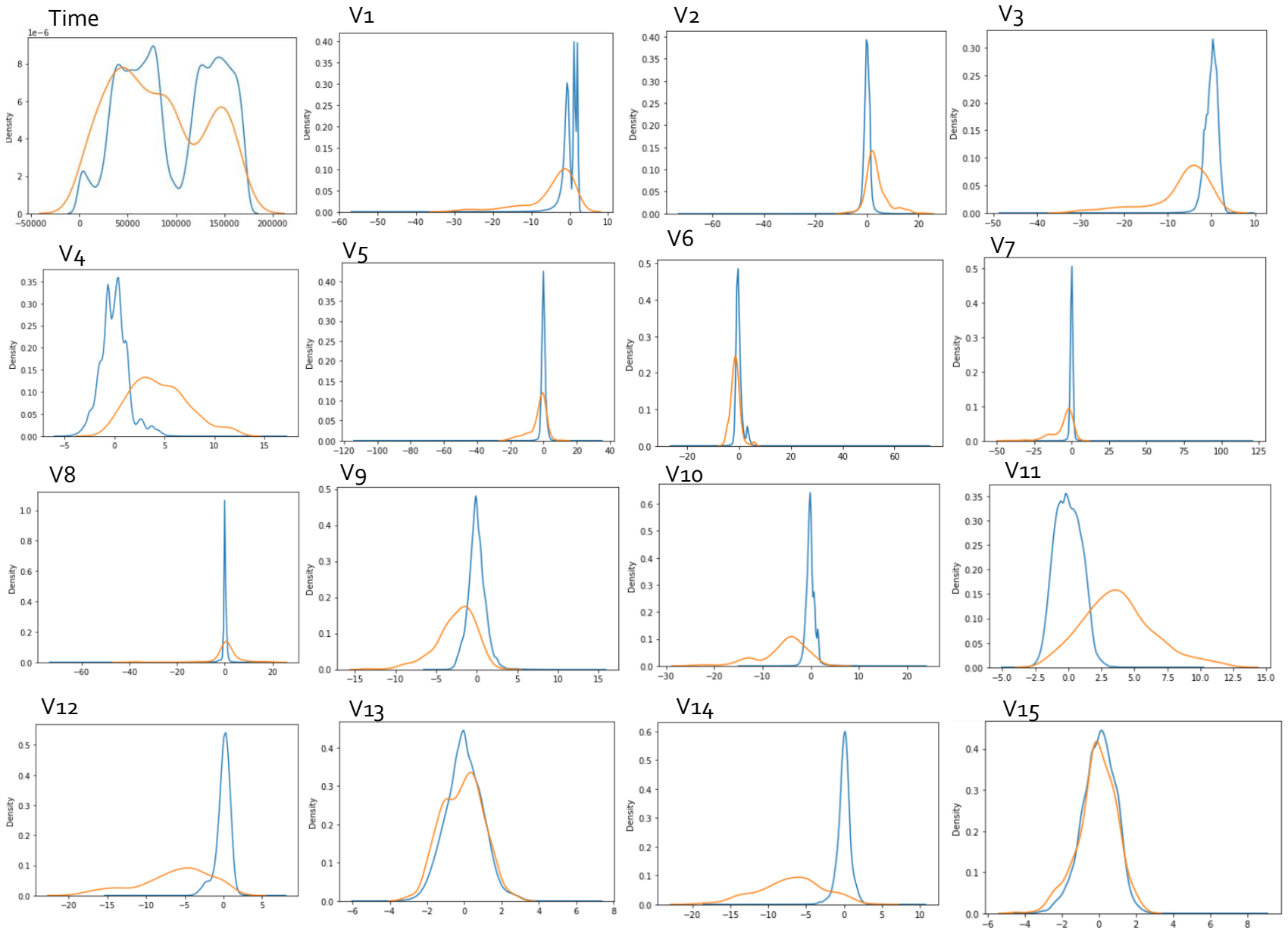
資料分布情況

-每兩feature的0與1分布



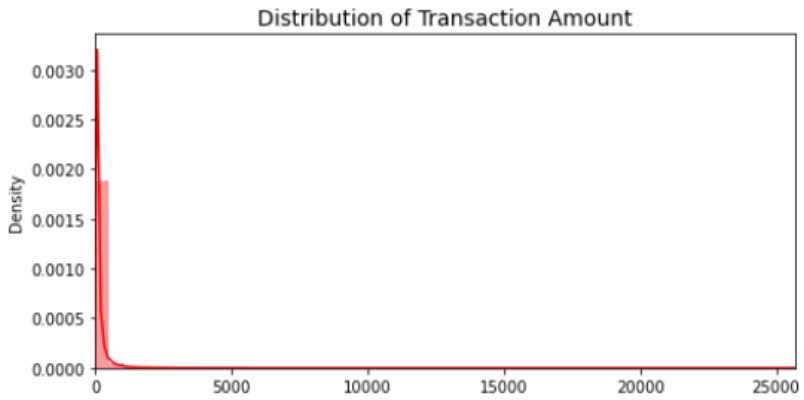
資料分布情況

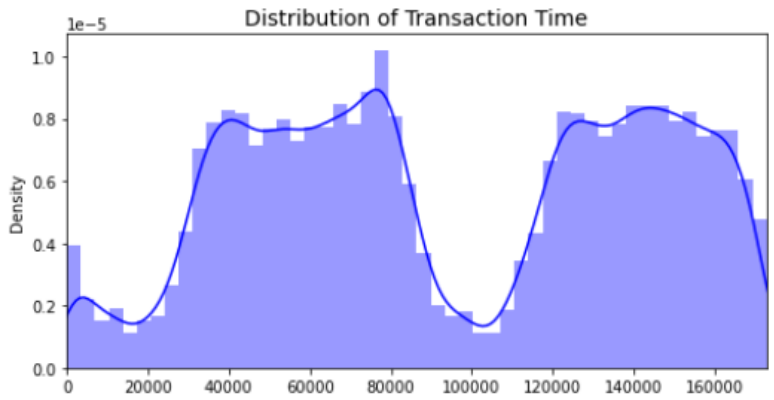
-每一feature的0與1分布



資料標準化

- 特徵v1~v28皆已做PCA轉換, 故已做Scaled
- 剩下Amount與Time尚未做Scaled





	Time	Amount
count	284807.000000	284807.000000
mean	94813.859575	88.349619
std	47488.145955	250.120109
min	0.000000	0.000000
25%	54201.500000	5.600000
50%	84692.000000	22.000000
75%	139320.500000	77.165000
max	172792.000000	25691.160000

資料標準化

- ## Method 1 : StandardScaler標準差標準化

標準化數據減去均值後, 除以標準差, 經過處理後數據符合標準常態分佈(mean = 0, std = 1)

$$x = \frac{x - mean}{std}$$

適用於 : 本身服從常態分佈的數據
Outlier : 基本可用於有Outlier的情況, 但在計算變異數和平均值時, Outlier仍會影響計算

- ## Method 2: MinMaxScaler極差標準化

將特徵縮放到給定的最小值和最大值之間, 也可以將每個特徵的最大絕對值轉換至單位大小, 對原始數據的線性轉換, 將數據歸一到[0,1]之間

$$x = \frac{x - min}{max - min}$$

適用於 : 分布範圍較穩定的數據, 當新數據的加入導致max/min變化, 則需重新定義
Outlier : 因為Outlier會影響最大值或最小值, 因此對outlier非常敏感

- ## Method 3 : RobustScaler 穩健標準化

刪除中位數, 並根據百分位數範圍(默認值為IQR:四分位間距)縮放數據

$$x = \frac{x - median}{75_{th}\ quntile - 25_{th}\ quantile}$$

適用於 :包含許多異常值的數據
Outlier : RobustScaler利用IQR進行縮放來弱化Outlier的影響

資料標準化

1. StandardScaler

	std_scaled_amount	std_scaled_time
0	0.244964	-1.996583
1	-0.342475	-1.996583
2	1.160686	-1.996562
3	0.140534	-1.996562
4	-0.073403	-1.996541

2. MinMaxScaler

	minmax_scaled_amount	minmax_scaled_time
0	0.005824	0.000000
1	0.000105	0.000000
2	0.014739	0.000006
3	0.004807	0.000006
4	0.002724	0.000012

3. RobustScaler

	rob_scaled_amount	robx_scaled_time
0	1.783274	-0.994983
1	-0.269825	-0.994983
2	4.983721	-0.994972
3	1.418291	-0.994972
4	0.670579	-0.994960

Sampling methods

Over Sampling

1. Random Oversampling(ROS)
2. SMOTE
3. Borderline SMOTE
4. SVM SMOTE
5. ADASYN

Under Sampling

1. Random Undersampling(RUS)
2. Ensemble Methods
3. NearMiss
4. Tomek Links
5. ENN

Sampling methods OverSampling

Random Oversampling

隨機地抽取少數類別的樣本，並將其複製後加入數據集當中

缺點 : 易造成過擬和

1

SMOTE

(*synthetic minority oversampling technique*)

對少數類樣本進行分析並根據少數類別的樣本來人工合成新樣本並添加到數據集中

缺點 :

1.選取的少數類樣本周圍也都是少數類樣本，則新合成的樣本不會提供太多有用信息

2.選取的少數類樣本周圍都是多數類樣本，這類的樣本可能是噪音

2

Borderline SMOTE 1

僅使用邊界上的少數樣本來合成新樣本, 從而改善樣本的類別分布

將少數樣本分為三類: Safe、Danger、Noise 僅對Danger樣本過採樣

1. 在K近鄰隨機選擇少數類樣本

2.在K近鄰隨機選擇任一樣本

3

Borderline SMOTE 2

在合成樣本時, 為使用k近鄰中的任一樣本

4

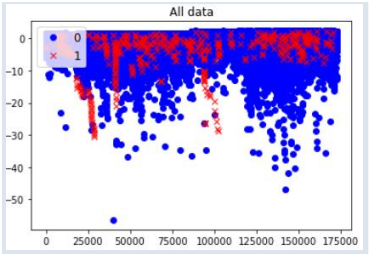
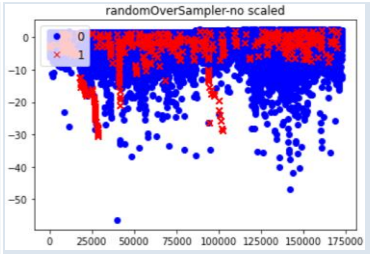
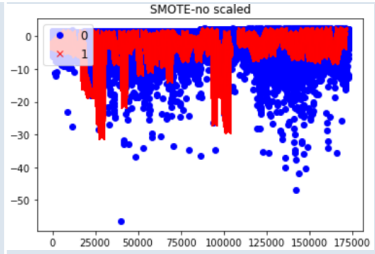
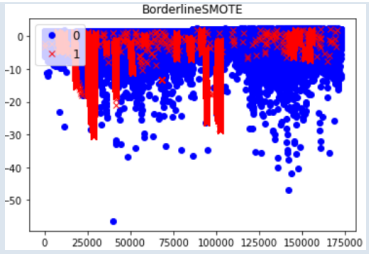
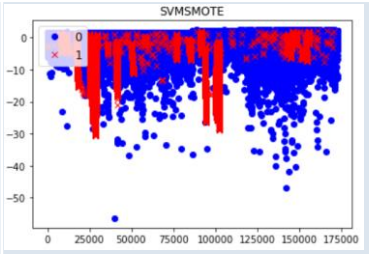
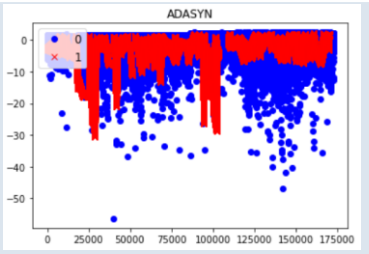
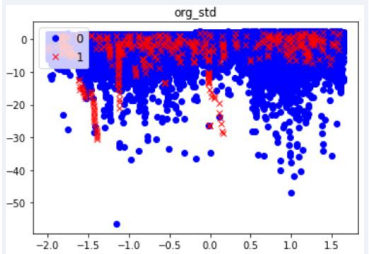
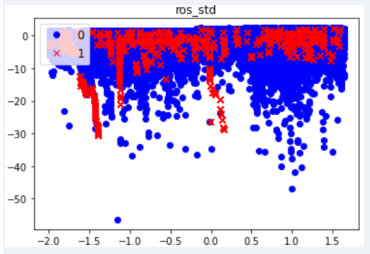
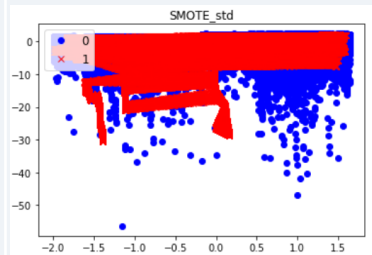
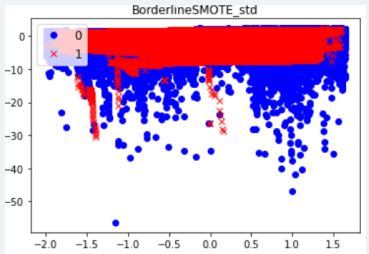
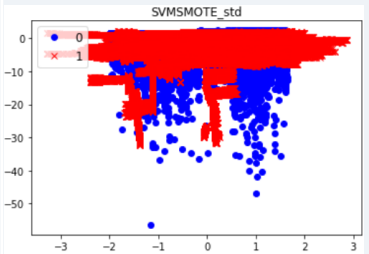
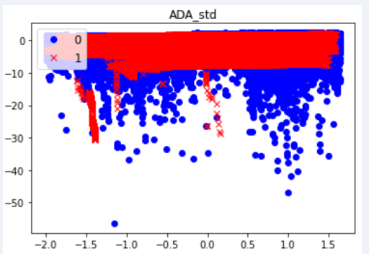
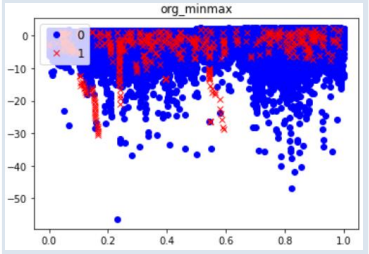
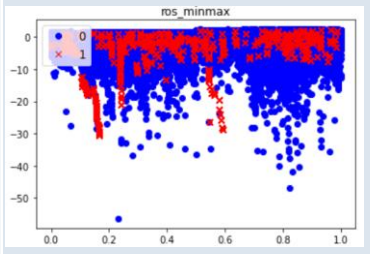
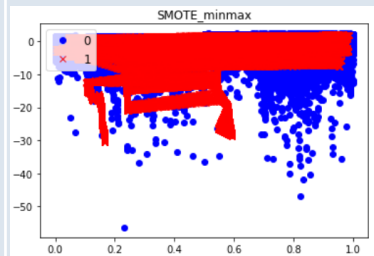
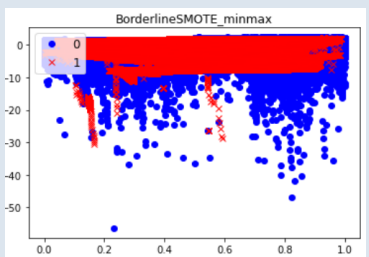
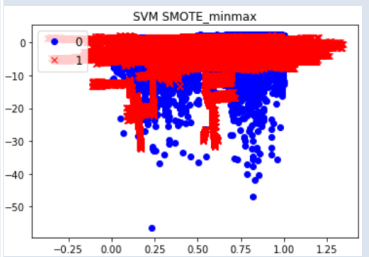
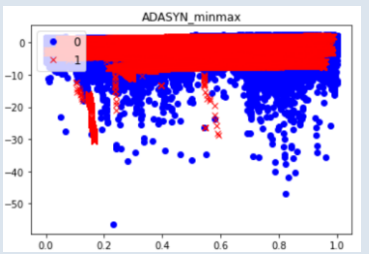
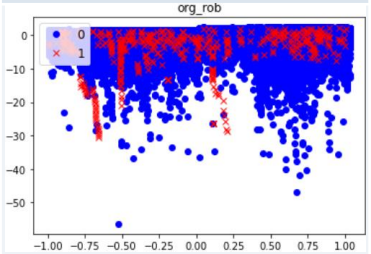
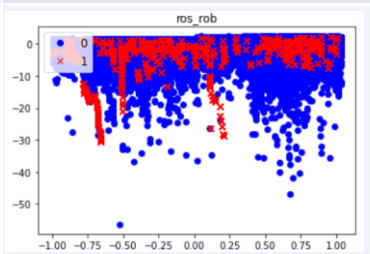
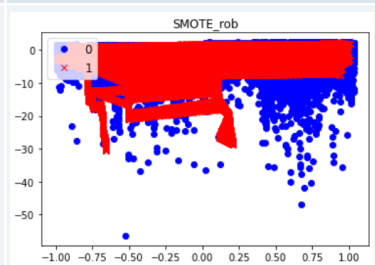
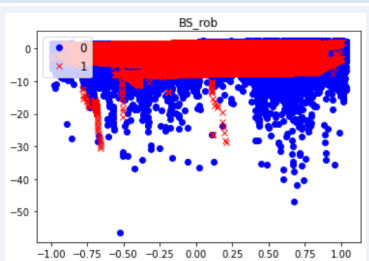
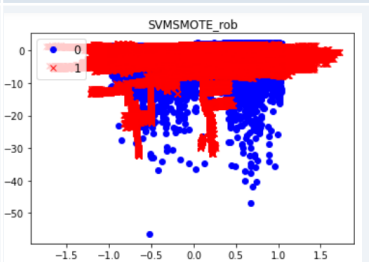
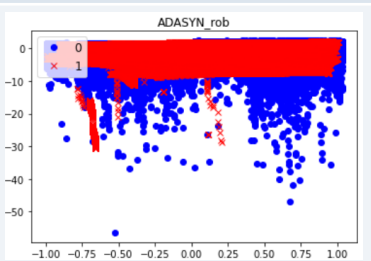
ADASYN

(*adaptive synthetic sampling*)

給每個少數類樣本施加一個權重, 周圍的多數類樣本越多則權重越高, 依此機制自動決定每個少數類樣本需要產生多少合成樣本

5

- Time/V1資料分布情況

	Orginal	ROS	SMOTE	Borderline-SMOTE 1	Borderline-SMOTE 2	ADASYN
未做 Scaled						
Standard scaler						
Minmax scaler						
Robuster Scaler						

Split train/test set

Train / test : 80/ 20

	未做Over sampling	Oversampling
Y_sampled	{0: 284315, 1: 492}	{0: 284315, 1: 284315}
X_train.shape	(227845, 30)	(454904, 30)
X_test.shape	(56962, 30)	(113726, 30)
y_train	{0: 227440, 1: 405}	{0: 227389, 1: 227515}
y_test	{0: 56875, 1: 87}	{0: 56926, 1: 56800}

Sampling methods UnderSampling

Random Under sampling

從多數類樣本中隨機選取一些剔除掉

缺點 : 剔除的樣本可能包含重要訊息, 致使學習出來的模型效果不好

1

ENN

(Edited Nearest Neighbours)

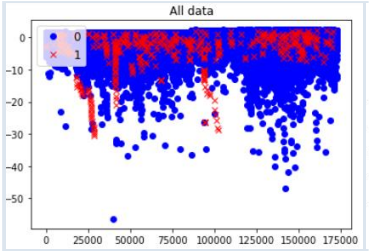
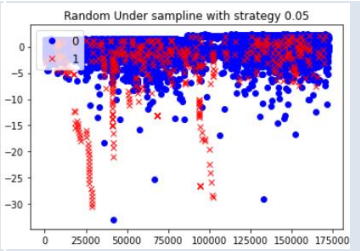
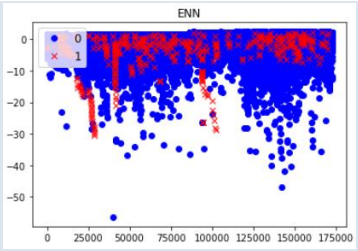
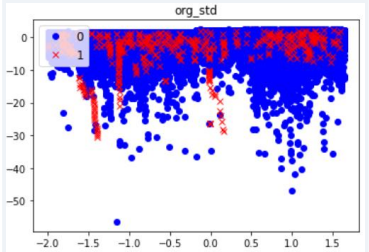
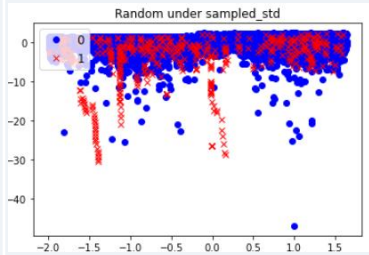
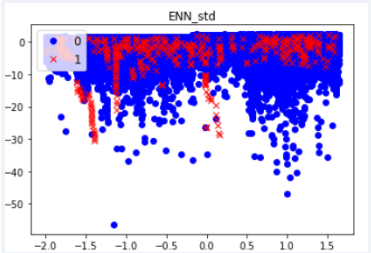
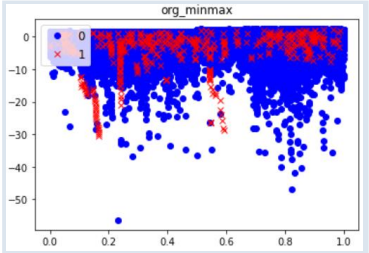
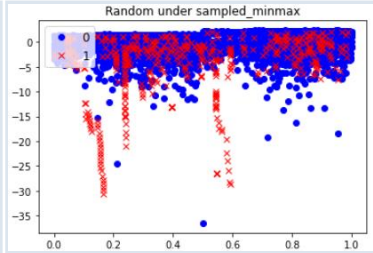
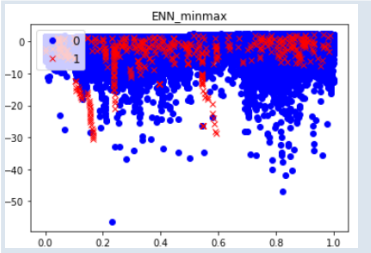
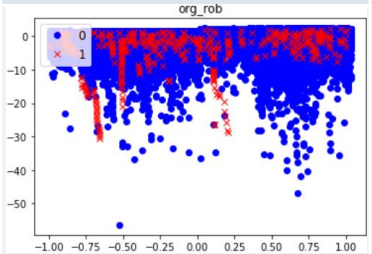
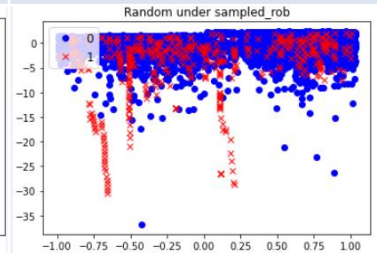
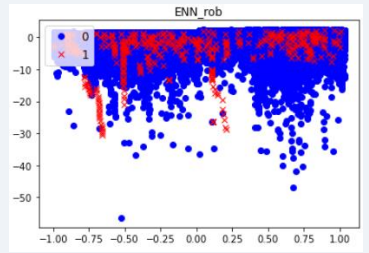
對於屬於多數類的一個樣本, 如果其K近鄰點有超過一半不屬於多數類, 則這個樣本會被剔除

缺點 : 無法控制欠採樣的數量

2

Sampling methods UnderSampling

- Time/V1資料分布情況

	Orginal	RUS	ENN
未做 Scaled			
Standard scaler			
Minmax scaler			
Robuster Scaler			

Model selection

Train / test : 80/ 20

	Original	RUS-0.05%	ENN
Y_sampled	{0: 284315, 1: 492}	{0: 9840, 1: 492}	{0: 283892, 1: 492}
X_train.shape	(227845, 30)	(8265, 30)	(227507, 30)
X_test.shape	(56962, 30)	(2067, 30)	(56877, 30)
y_train	{0: 227440, 1: 405}	{0: 7872, 1: 393}	{0: 227103, 1: 404}
y_test	{0: 56875, 1: 87}	{0: 1968, 1: 99}	{0: 56789, 1: 88}

XGBoost 預測

- XGBOOST使用預設參數，並且重複"抽樣->訓練"進行10次，再取平均計算各SCORE。
- SCORE包含以下:
  - Precision
  - Recall
  - F1-Score

Precision	Original		Over-sampling		SMOTE		Borderline-SMOTE		Borderline2-SMOTE	
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999661	0.960808	1	0.999865	0.999982	0.999754	0.999772	0.999912	0.999736	0.999807
StandardScaler 標準差標準化	0.999675	0.952663	1	0.999877	1	0.999490	0.999807	0.999859	0.999771	0.999684
MinMaxScaler 極差標準化	0.999645	0.947307	1	0.999877	0.999982	0.99931	0.999772	0.999771	0.999753	0.999702
RobustScaler 穩健標準化	0.999684	0.947500	1	0.999877	1	0.999542	0.999772	0.999806	0.999789	0.999614

Precision	Original		ADASYN							
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999661	0.960808	0.999982	0.999772						
StandardScaler 標準差標準化	0.999675	0.952663	1	0.999279						
MinMaxScaler 極差標準化	0.999645	0.947307	0.999982	0.999172						
RobustScaler 穩健標準化	0.999684	0.947500	0.999982	0.999065						

Recall	Original		Over-sampling		SMOTE		Borderline-SMOTE		Borderline2-SMOTE	
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999942	0.807385	0.999865	1	0.999754	0.999982	0.999912	0.999771	0.999806	0.999737
StandardScaler 標準差標準化	0.999930	0.813131	0.999877	1	0.999491	1	0.999859	0.999806	0.999683	0.999772
MinMaxScaler 極差標準化	0.999921	0.800198	0.999877	1	0.999332	0.999982	0.999772	0.999771	0.999701	0.999754
RobustScaler 穩健標準化	0.999926	0.808102	0.999877	1	0.999543	1	0.999807	0.999771	0.999612	0.999789

Recall	Original		ADASYN							
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999942	0.807385	0.999771	0.999982						
StandardScaler 標準差標準化	0.999930	0.813131	0.999279	1						
MinMaxScaler 極差標準化	0.999921	0.800198	0.999175	0.999982						
RobustScaler 穩健標準化	0.999926	0.808102	0.999071	0.999982						

F1-Score	Original		Over-sampling		SMOTE		Borderline-SMOTE		Borderline2-SMOTE	
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999801	0.877440	0.999932	0.999932	0.999868	0.999868	0.999842	0.999842	0.99971	0.99972
StandardScaler 標準差標準化	0.999802	0.877384	0.999939	0.999938	0.999745	0.999745	0.999833	0.999833	0.999727	0.999728
MinMaxScaler 極差標準化	0.999783	0.867560	0.999939	0.999938	0.999657	0.999657	0.999772	0.999771	0.999727	0.999728
RobustScaler 穩健標準化	0.999805	0.872267	0.999939	0.999938	0.999772	0.999771	0.999789	0.999789	0.999701	0.999702

F1-Score	Original		ADASYN							
	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1	Y=0	Y=1
資料無Scaled	0.999801	0.877440	0.999877	0.999877						
StandardScaler 標準差標準化	0.999802	0.877384	0.999639	0.999640						
MinMaxScaler 極差標準化	0.999783	0.867560	0.999579	0.999577						
RobustScaler 穩健標準化	0.999805	0.872267	0.999527	0.999524						

結論

- 經由不平衡資料處理後，資料的對於"1"的各種指標分數都明顯提升。
- 其中ADASYN的分數為最高，並且資料無經過標準化處理的分數為最高。