```python
# Project by Akhil Kumar
# PROJECT - CREDIT CARD FRAUD PREDICTION USING Random Forest
# https://www.linkedin.com/in/akhil-kumar-494951218/
# https://github.com/Akhil4005

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import gridspec

data=pd.read_csv("creditcard.csv")

data.head()
```

{"type":"dataframe","variable_name":"data"}

```python
data.columns
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9',
'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19',
'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28',
'Amount',
       'Class'],
      dtype='object')
```

```python
print(data.shape)
print(data.describe())
```

```
(49610, 31)
               Time            V1            V2            V3
V4  \
count  49610.000000  49610.000000  49610.000000  49610.000000
49609.000000
mean   28803.556239     -0.242569      0.012235      0.693009
0.185186
std    13097.468525      1.885867      1.630704      1.510559
1.400175
min        0.000000    -56.407510    -72.715728    -32.965346     -
5.172595
25%    21734.250000     -0.992845     -0.562967      0.217605     -
0.720957
50%    33390.000000     -0.247223      0.079282      0.797007
0.190288
75%    38852.750000      1.155638      0.732318      1.431013
1.067346
max    44135.000000      1.960497     18.183626      4.101716
16.491217
```

```
                  V5            V6            V7            V8
V9  \
count  49609.000000  49609.000000  49609.000000  49609.000000
49609.000000
mean      -0.257016      0.104114     -0.120255      0.053442
0.123490
std        1.413057      1.310705      1.283507      1.224245
1.213441
min      -42.147898    -26.160506    -26.548144    -41.484823     -
9.283925
25%       -0.866471     -0.635669     -0.605928     -0.146749     -
0.611499
50%       -0.287810     -0.150940     -0.076595      0.058406
0.012150
75%        0.283513      0.493918      0.424969      0.331555
0.819242
max       34.801666     22.529298     36.677268     20.007208
10.392889

       ...           V21           V22           V23           V24  \
count  ...  49609.000000  49609.000000  49609.000000  49609.000000
mean   ...     -0.028396     -0.107154     -0.040123      0.007997
std    ...      0.736050      0.637733      0.590810      0.594121
min    ...    -20.262054     -8.593642    -26.751119     -2.836627
25%    ...     -0.231664     -0.529531     -0.179110     -0.322243
50%    ...     -0.068396     -0.082137     -0.051560      0.061999
75%    ...      0.108082      0.307262      0.078474      0.401392
max    ...     22.614889      5.805795     17.297845      4.014444

                 V25           V26           V27           V28
Amount  \
count  49609.000000  49609.000000  49609.000000  49609.000000
49609.000000
mean       0.135954      0.020813      0.004792      0.004533
93.120688
std        0.439067      0.501438      0.388364      0.333225
253.265971
min       -7.495741     -1.577118     -8.567638     -9.617915
0.000000
25%       -0.127983     -0.330532     -0.063339     -0.006675
7.610000
50%        0.175766     -0.071826      0.008986      0.022155
25.000000
75%        0.421960      0.300180      0.083910      0.076342
85.000000
max        5.525093      3.517346     11.135740     33.847808
12910.930000

              Class
count  49609.000000
```

```
mean        0.002983
std         0.054539
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000

[8 rows x 31 columns]
```

```
fraud=data[data['Class']==1]
valid=data[data['Class']==0]
print(fraud.shape)
print(valid.shape)
```

```
(148, 31)
(49461, 31)
```

```
outlierfrac=len(fraud)/len(valid)
print(outlierfrac)
print("Fraud Cases: {}".format(len(data[data['Class']==1])))
print("Valid Cases: {}".format(len(data[data['Class']==0])))
```

```
0.0029922565253431995
Fraud Cases: 148
Valid Cases: 49461
```

```
print("Amount details of the fraudulent transaction")
print(fraud.Amount.describe())
```

```
Amount details of the fraudulent transaction
count     148.000000
mean      100.170676
std       233.347471
min         0.000000
25%         1.000000
50%         9.560000
75%        99.990000
max      1809.680000
Name: Amount, dtype: float64
```

```
print("Amount details of the valid transaction")
print(valid.Amount.describe())
```

```
Amount details of the valid transaction
count    49461.000000
mean        93.099593
std        253.325102
min          0.000000
25%          7.680000
50%         25.000000
```

```
75%          85.000000
max       12910.930000
Name: Amount, dtype: float64
```

```
data = data.dropna(axis=0)
```

```
data.columns
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9',
'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19',
'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28',
'Amount',
       'Class'],
      dtype='object')
```

```
# correlation heatmap
corrmat = data.corr()
print("Correlation Matrix:")
print(corrmat)
fig = plt.figure(figsize = (30, 20))
sns.heatmap(corrmat, vmax = .8, square =
True,annot=True,fmt='.2f',cmap='coolwarm')
plt.show()
```

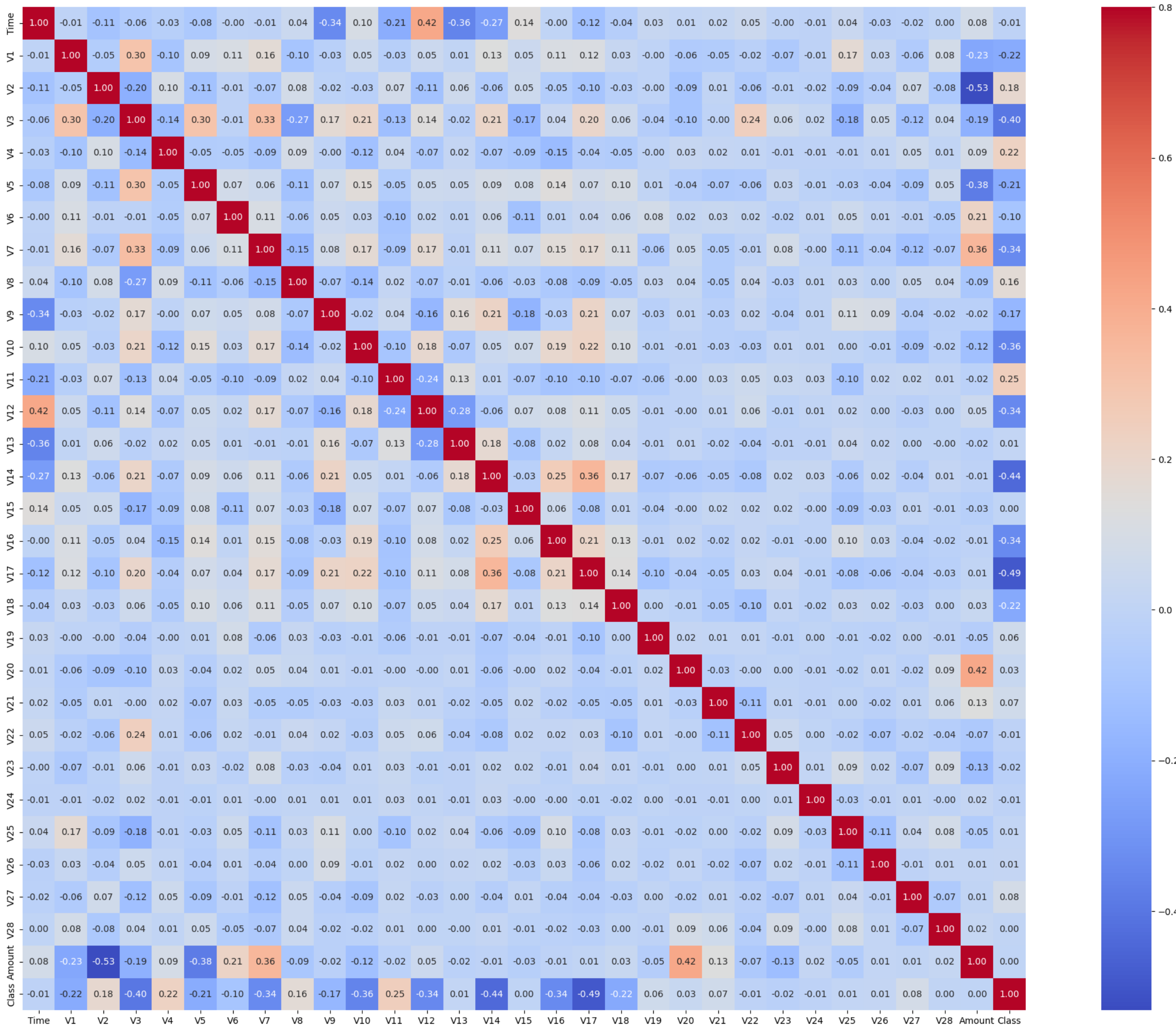
```
Correlation Matrix:
            Time        V1        V2        V3        V4        V5
V6  \
Time    1.000000 -0.005175 -0.108226 -0.059836 -0.034093 -0.075760 -
0.004925
V1     -0.005175  1.000000 -0.049820  0.300701 -0.098096  0.090339
0.105287
V2     -0.108226 -0.049820  1.000000 -0.196895  0.097895 -0.111988 -
0.008590
V3     -0.059836  0.300701 -0.196895  1.000000 -0.142634  0.301734 -
0.008739
V4     -0.034093 -0.098096  0.097895 -0.142634  1.000000 -0.054294 -
0.053092
V5     -0.075760  0.090339 -0.111988  0.301734 -0.054294  1.000000
0.074200
V6     -0.004925  0.105287 -0.008590 -0.008739 -0.053092  0.074200
1.000000
V7     -0.009393  0.161688 -0.068698  0.331018 -0.093911  0.055122
0.107260
V8      0.041860 -0.098520  0.080488 -0.274999  0.090697 -0.113355 -
0.057708
V9     -0.338263 -0.031668 -0.016021  0.165447 -0.004522  0.066371
0.046479
V10     0.099452  0.053959 -0.029492  0.205716 -0.118667  0.154500
```

```
0.029207
V11    -0.207438 -0.032444  0.072121 -0.127144  0.042723 -0.050450 -
0.102553
V12     0.418547  0.049418 -0.114509  0.135027 -0.068964  0.053142
0.023249
V13    -0.357951  0.006966  0.056528 -0.022734  0.016483  0.048429
0.006559
V14    -0.274600  0.133917 -0.058967  0.205284 -0.068143  0.091397
0.057459
V15     0.143178  0.046241  0.051660 -0.170393 -0.087960  0.081652 -
0.113192
V16    -0.000080  0.108372 -0.048489  0.037232 -0.146260  0.143726
0.008841
V17    -0.118151  0.117190 -0.096705  0.201500 -0.043984  0.073159
0.035395
V18    -0.039746  0.028160 -0.030369  0.058809 -0.053431  0.098910
0.055007
V19     0.027877 -0.003892 -0.002112 -0.044978 -0.004861  0.005583
0.081333
V20     0.011022 -0.059297 -0.094929 -0.097018  0.029911 -0.042529
0.022876
V21     0.019920 -0.053706  0.005714 -0.004501  0.016099 -0.067656
0.033861
V22     0.046430 -0.024111 -0.063569  0.238296  0.009444 -0.063467
0.022461
V23    -0.003473 -0.072403 -0.012717  0.057011 -0.008755  0.029673 -
0.019349
V24    -0.009329 -0.010447 -0.020204  0.021617 -0.008273 -0.014630
0.011061
V25     0.037219  0.169276 -0.090963 -0.183251 -0.008550 -0.032281
0.052209
V26    -0.025842  0.026631 -0.038082  0.051466  0.012627 -0.040848
0.012034
V27    -0.019690 -0.059994  0.067834 -0.122109  0.052222 -0.087884 -
0.009431
V28     0.001609  0.076966 -0.075651  0.035135  0.005389  0.052375 -
0.049048
Amount  0.077079 -0.234080 -0.531718 -0.190591  0.092385 -0.384757
0.214355
Class  -0.008044 -0.215601  0.182382 -0.401646  0.224816 -0.209979 -
0.099531

              V7        V8        V9  ...       V21       V22
V23  \
Time   -0.009393  0.041860 -0.338263  ...  0.019920  0.046430 -
0.003473
V1      0.161688 -0.098520 -0.031668  ... -0.053706 -0.024111 -
0.072403
V2     -0.068698  0.080488 -0.016021  ...  0.005714 -0.063569 -
```

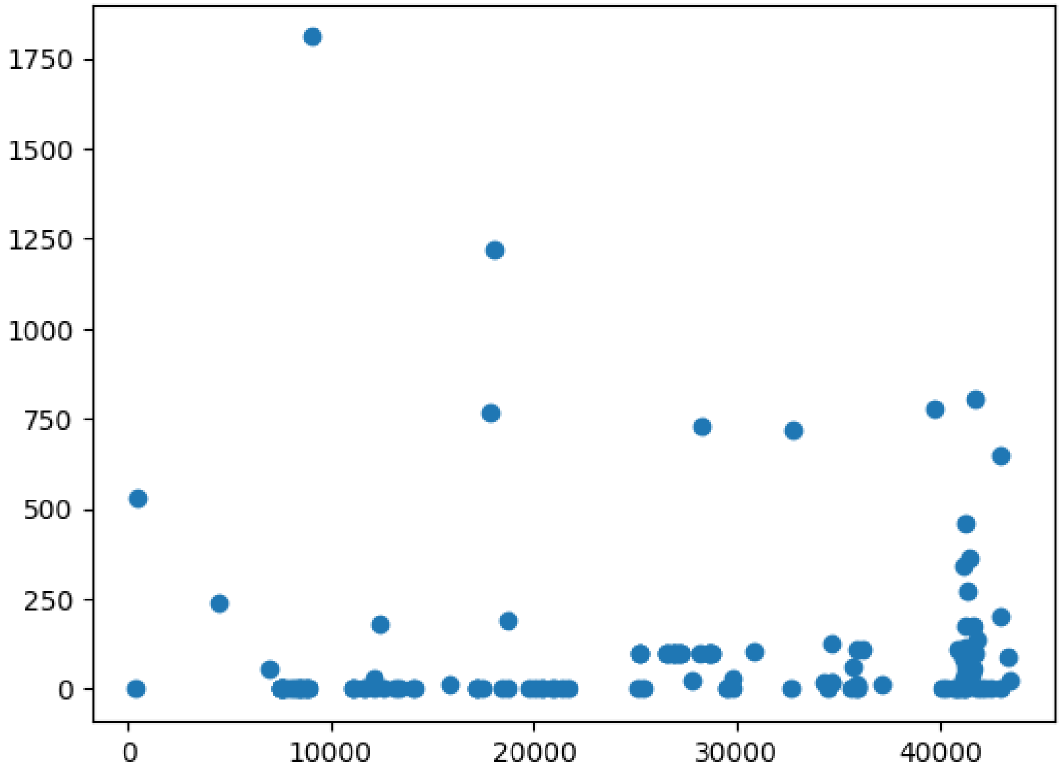
```
0.012717
V3      0.331018 -0.274999  0.165447  ... -0.004501  0.238296
0.057011
V4     -0.093911  0.090697 -0.004522  ...  0.016099  0.009444 -
0.008755
V5      0.055122 -0.113355  0.066371  ... -0.067656 -0.063467
0.029673
V6      0.107260 -0.057708  0.046479  ...  0.033861  0.022461 -
0.019349
V7      1.000000 -0.153186  0.076851  ... -0.051378 -0.010394
0.080515
V8     -0.153186  1.000000 -0.074469  ... -0.048375  0.036639 -
0.030369
V9      0.076851 -0.074469  1.000000  ... -0.027508  0.018340 -
0.036443
V10     0.166994 -0.140781 -0.017280  ... -0.031438 -0.033062
0.009294
V11    -0.094391  0.015349  0.043375  ...  0.031778  0.054577
0.032248
V12     0.167316 -0.069963 -0.163692  ...  0.013154  0.064202 -
0.014392
V13    -0.011330 -0.014827  0.156345  ... -0.018636 -0.041454 -
0.007433
V14     0.110164 -0.064441  0.214282  ... -0.046226 -0.078700
0.024063
V15     0.067465 -0.033211 -0.180986  ...  0.022224  0.019209
0.021520
V16     0.152385 -0.076662 -0.029312  ... -0.024429  0.018288 -
0.012865
V17     0.171099 -0.092459  0.210081  ... -0.053070  0.026943
0.038715
V18     0.113745 -0.045210  0.068701  ... -0.053633 -0.097208
0.009693
V19    -0.060346  0.026258 -0.028196  ...  0.014340  0.012482 -
0.013516
V20     0.053552  0.035217  0.009372  ... -0.027355 -0.004127
0.002557
V21    -0.051378 -0.048375 -0.027508  ...  1.000000 -0.113600
0.005512
V22    -0.010394  0.036639  0.018340  ... -0.113600  1.000000
0.045582
V23     0.080515 -0.030369 -0.036443  ...  0.005512  0.045582
1.000000
V24    -0.003736  0.005417  0.006944  ... -0.008572  0.001130
0.011456
V25    -0.110170  0.028412  0.105346  ...  0.000431 -0.022613
0.086504
V26    -0.038661  0.002126  0.093468  ... -0.023412 -0.073109
0.024786
```

```
V27    -0.118415  0.045239 -0.039278  ...  0.006007 -0.024493 -
0.071891
V28    -0.068748  0.041345 -0.024366  ...  0.063194 -0.037819
0.092185
Amount  0.361257 -0.089195 -0.022124  ...  0.131751 -0.072419 -
0.132672
Class  -0.338768  0.164694 -0.167273  ...  0.068019 -0.010136 -
0.022648

             V24       V25       V26       V27       V28    Amount
Class
Time   -0.009329  0.037219 -0.025842 -0.019690  0.001609  0.077079 -
0.008044
V1     -0.010447  0.169276  0.026631 -0.059994  0.076966 -0.234080 -
0.215601
V2     -0.020204 -0.090963 -0.038082  0.067834 -0.075651 -0.531718
0.182382
V3      0.021617 -0.183251  0.051466 -0.122109  0.035135 -0.190591 -
0.401646
V4     -0.008273 -0.008550  0.012627  0.052222  0.005389  0.092385
0.224816
V5     -0.014630 -0.032281 -0.040848 -0.087884  0.052375 -0.384757 -
0.209979
V6      0.011061  0.052209  0.012034 -0.009431 -0.049048  0.214355 -
0.099531
V7     -0.003736 -0.110170 -0.038661 -0.118415 -0.068748  0.361257 -
0.338768
V8      0.005417  0.028412  0.002126  0.045239  0.041345 -0.089195
0.164694
V9      0.006944  0.105346  0.093468 -0.039278 -0.024366 -0.022124 -
0.167273
V10     0.008769  0.003599 -0.011643 -0.094279 -0.019091 -0.123104 -
0.357204
V11     0.027978 -0.104126  0.016581  0.024410  0.010771 -0.020903
0.247225
V12     0.007852  0.018533  0.004278 -0.026409  0.001850  0.046627 -
0.340996
V13    -0.009883  0.038071  0.016368  0.002530 -0.000974 -0.020630
0.009606
V14     0.026719 -0.058908  0.024543 -0.042658  0.014604 -0.009175 -
0.440760
V15    -0.003571 -0.091924 -0.034257  0.009712 -0.014194 -0.032592
0.000805
V16    -0.001287  0.095913  0.030126 -0.035227 -0.018286 -0.011183 -
0.337406
V17    -0.013429 -0.076360 -0.063247 -0.044337 -0.031894  0.008506 -
0.493419
V18    -0.015571  0.026337  0.022005 -0.032318  0.003080  0.034751 -
0.224112
```

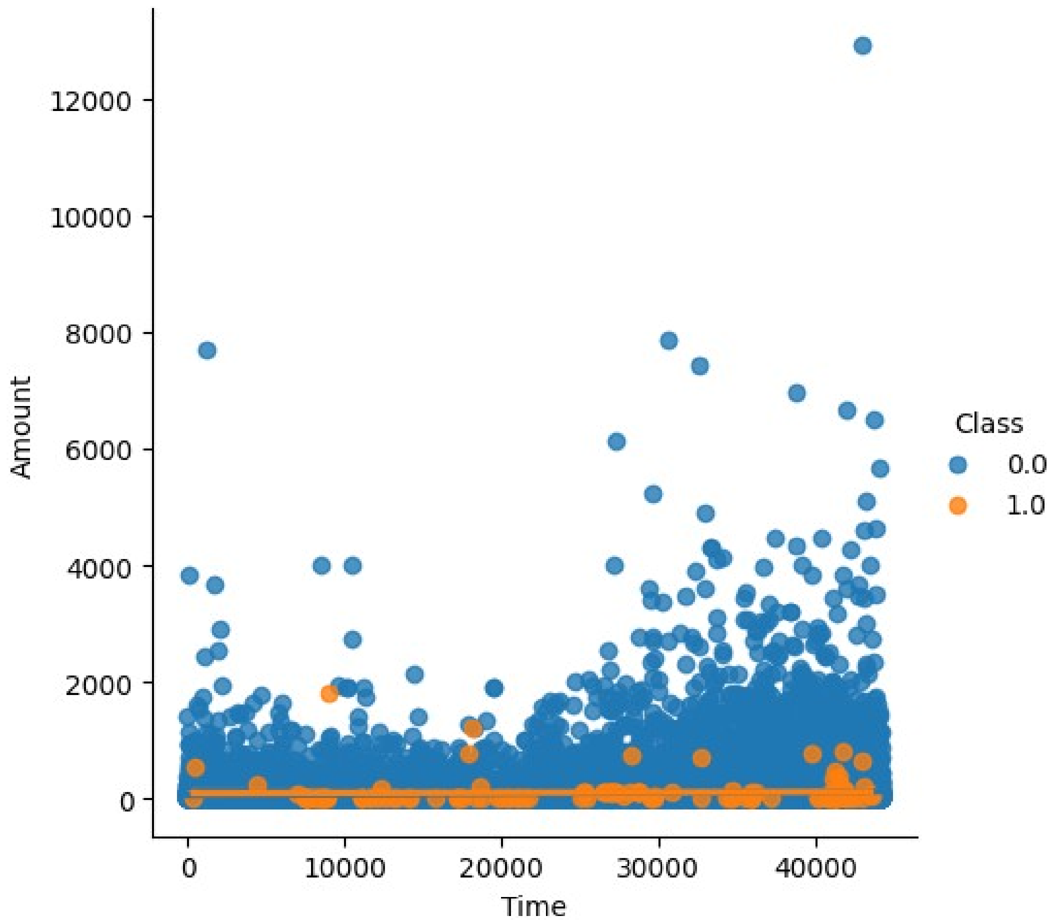
```
V19     0.000370 -0.012884 -0.020420  0.004168 -0.007324 -0.047529
0.063243
V20    -0.010626 -0.016502  0.011572 -0.020562  0.086370  0.415956
0.031868
V21    -0.008572  0.000431 -0.023412  0.006007  0.063194  0.131751
0.068019
V22     0.001130 -0.022613 -0.073109 -0.024493 -0.037819 -0.072419 -
0.010136
V23     0.011456  0.086504  0.024786 -0.071891  0.092185 -0.132672 -
0.022648
V24     1.000000 -0.034584 -0.009239  0.005001 -0.003981  0.015043 -
0.008538
V25    -0.034584  1.000000 -0.113226  0.036152  0.078435 -0.053647
0.013803
V26    -0.009239 -0.113226  1.000000 -0.008334  0.011121  0.011514
0.014691
V27     0.005001  0.036152 -0.008334  1.000000 -0.066067  0.008866
0.084283
V28    -0.003981  0.078435  0.011121 -0.066067  1.000000  0.022296
0.004289
Amount  0.015043 -0.053647  0.011514  0.008866  0.022296  1.000000
0.001523
Class  -0.008538  0.013803  0.014691  0.084283  0.004289  0.001523
1.000000

[31 rows x 31 columns]
```



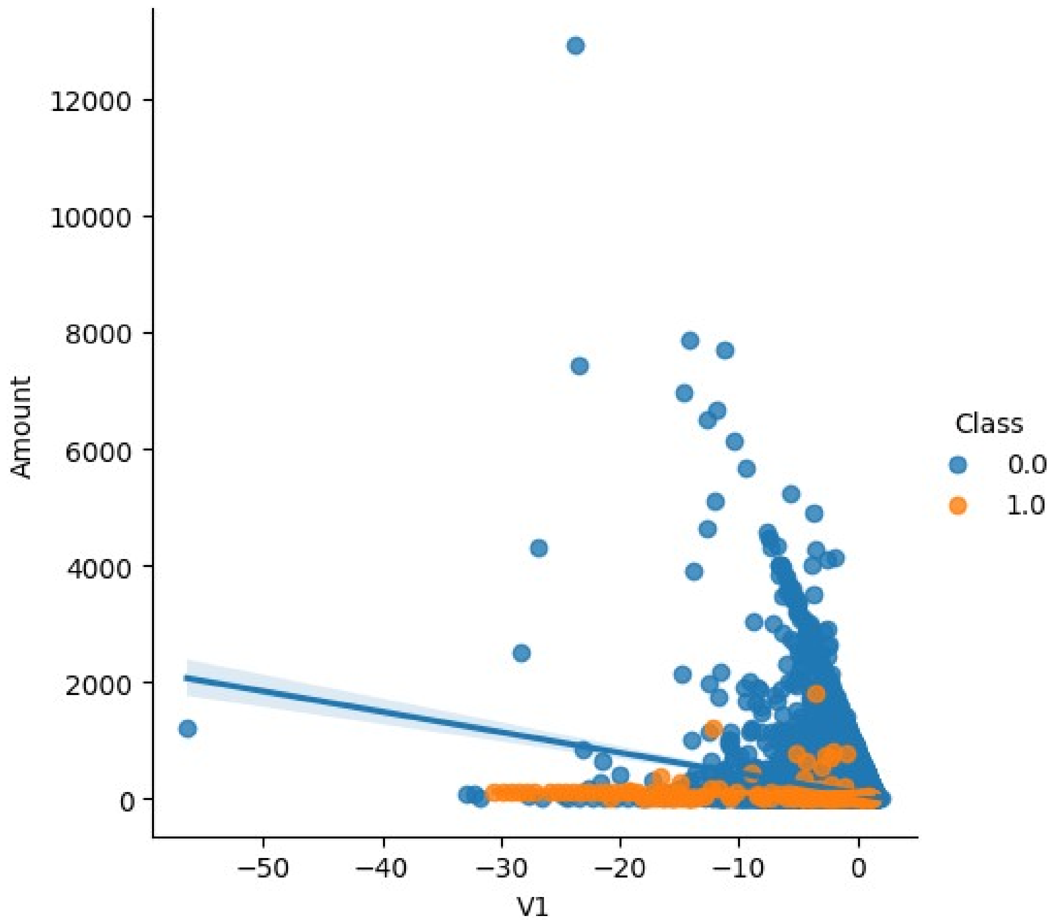

```
data.columns

Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9',
'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19',
'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28',
'Amount',
       'Class'],
      dtype='object')

#frauds with time
plt.scatter(x=data.loc[data['Class']==1]['Time'],
y=data.loc[data['Class']==1]['Amount'])
plt.show()
```



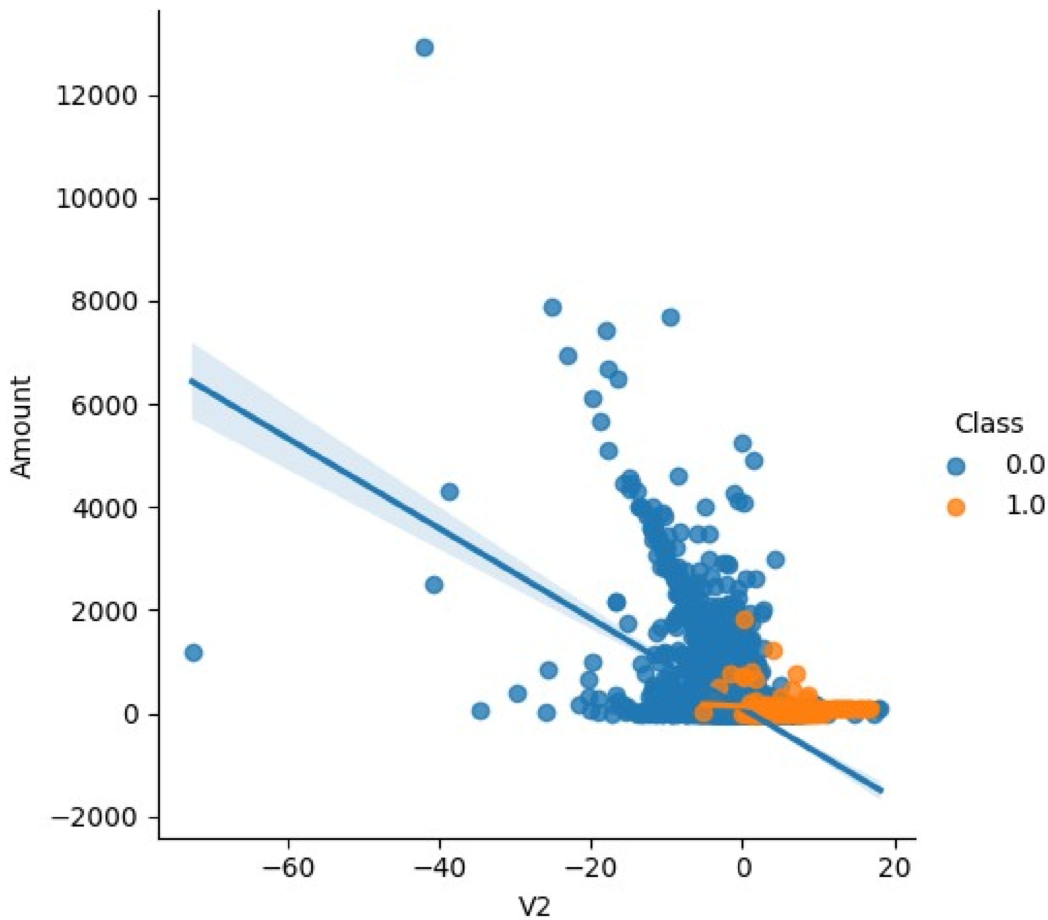

```
plt.gcf().set_size_inches(0.75, 0.5)
for col in data.columns:
  sns.lmplot(x=col, y='Amount', hue='Class', data=data)
  plt.show()

<Figure size 75x50 with 0 Axes>
```



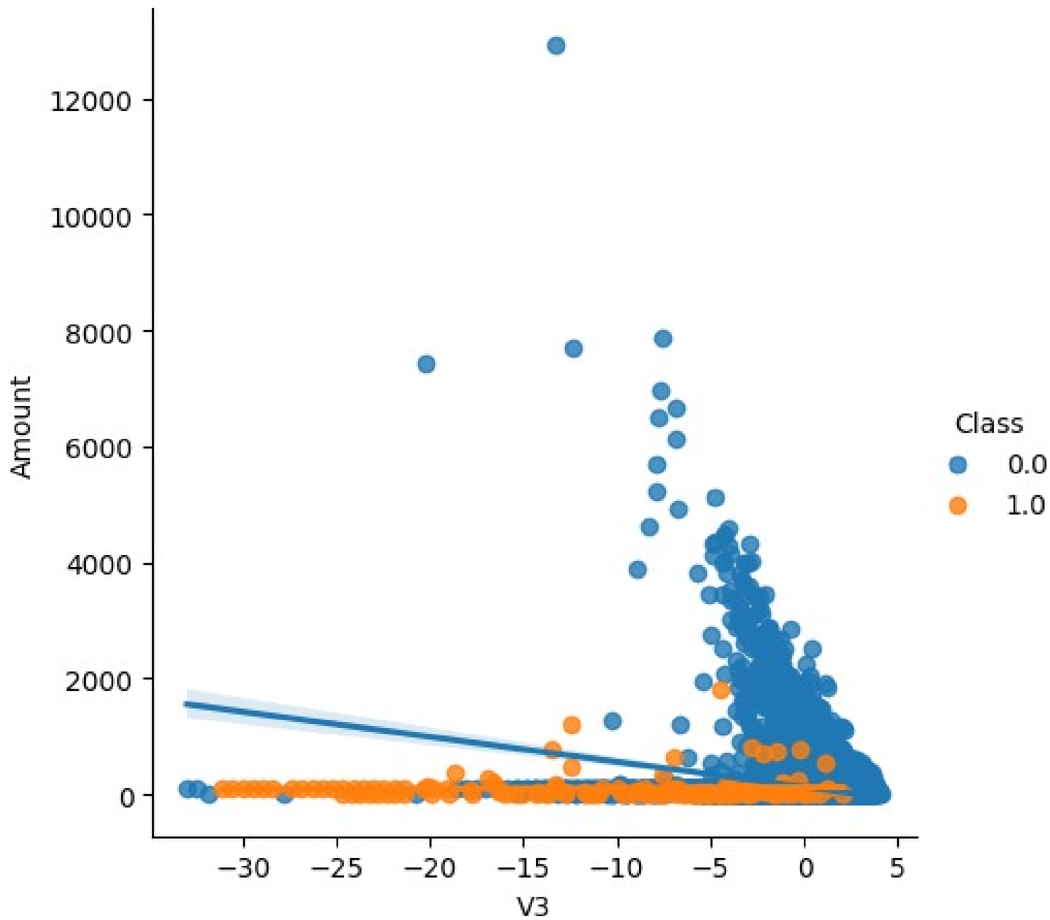
Amount
12000
10000
8000
6000
4000
2000
0
0
10000
20000
30000
40000
Time
Class
0.0
1.0



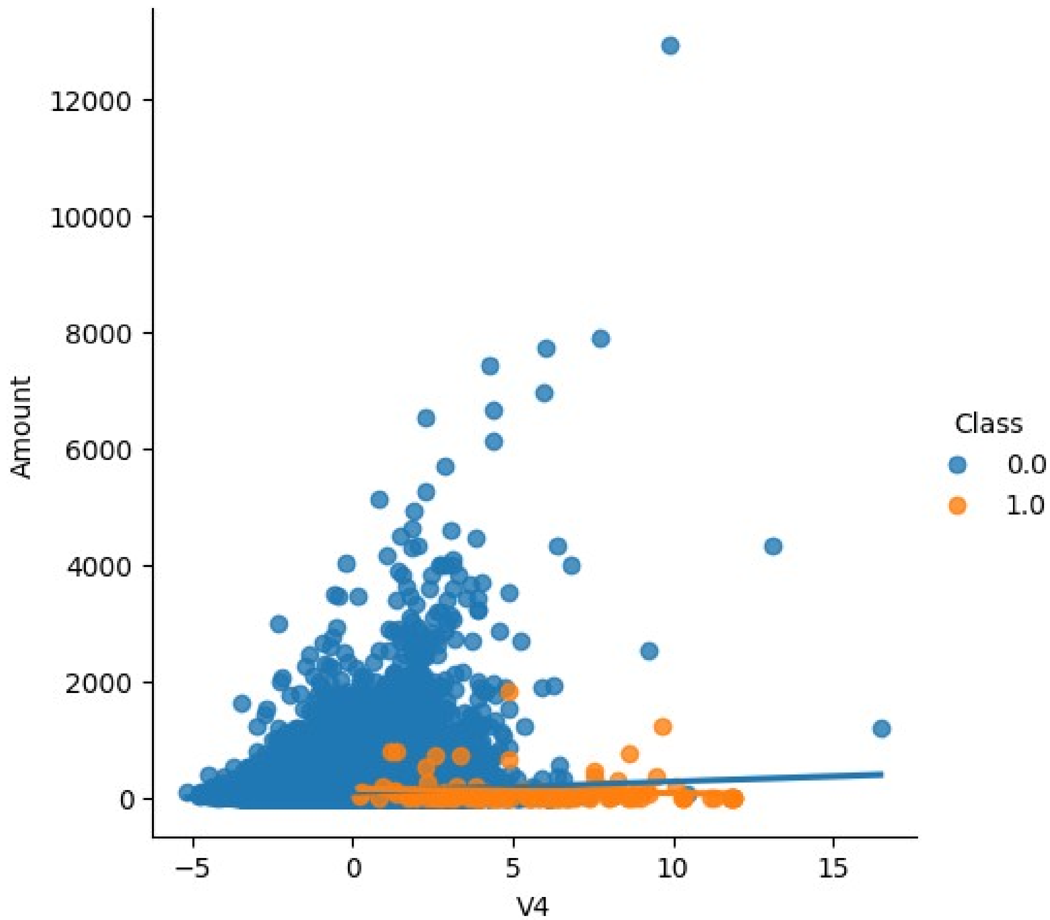
12000
10000
8000
6000
4000
2000
0
Amount
−50
−40
−30
−20
−10
0
V1
Class
0.0
1.0



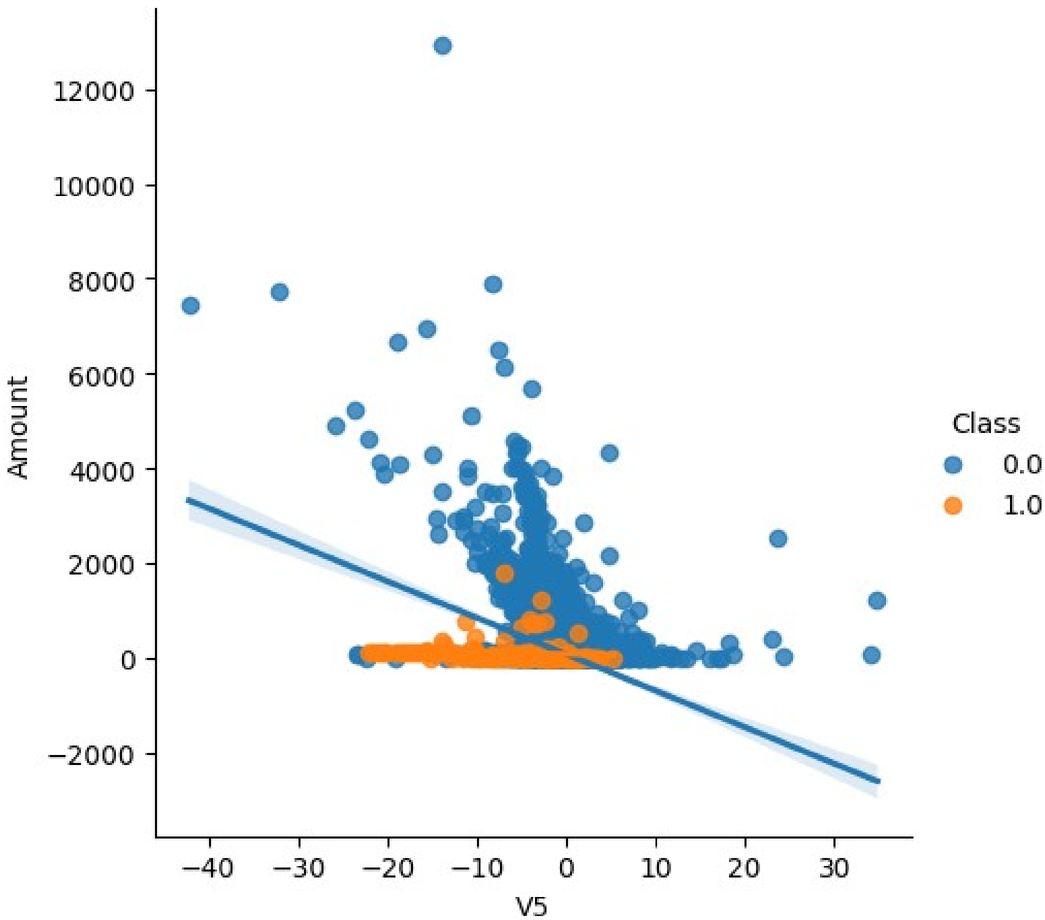
12000
10000
8000
6000
4000
2000
0
−2000
Amount
−60
−40
−20
0
20
V2
Class
0.0
1.0



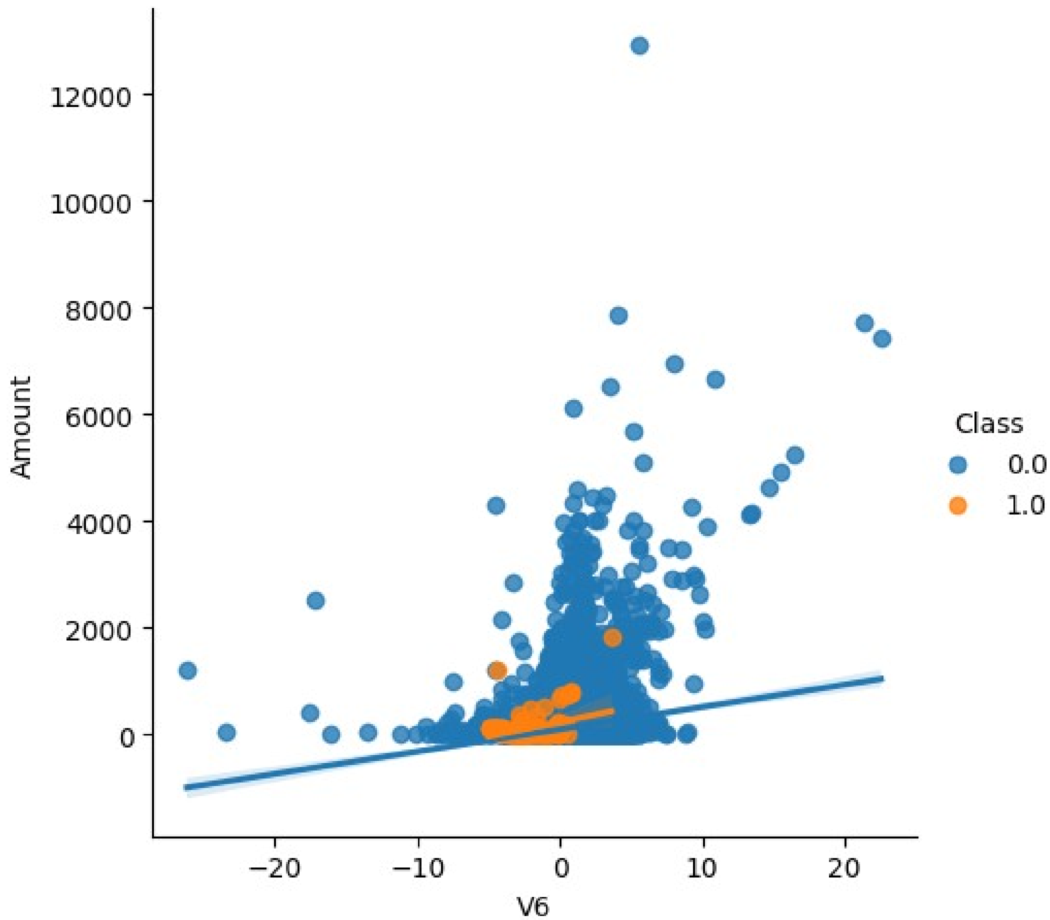
12000
10000
8000
6000
4000
2000
0
Amount
−30
−25
−20
−15
−10
−5
0
5
V3
Class
0.0
1.0



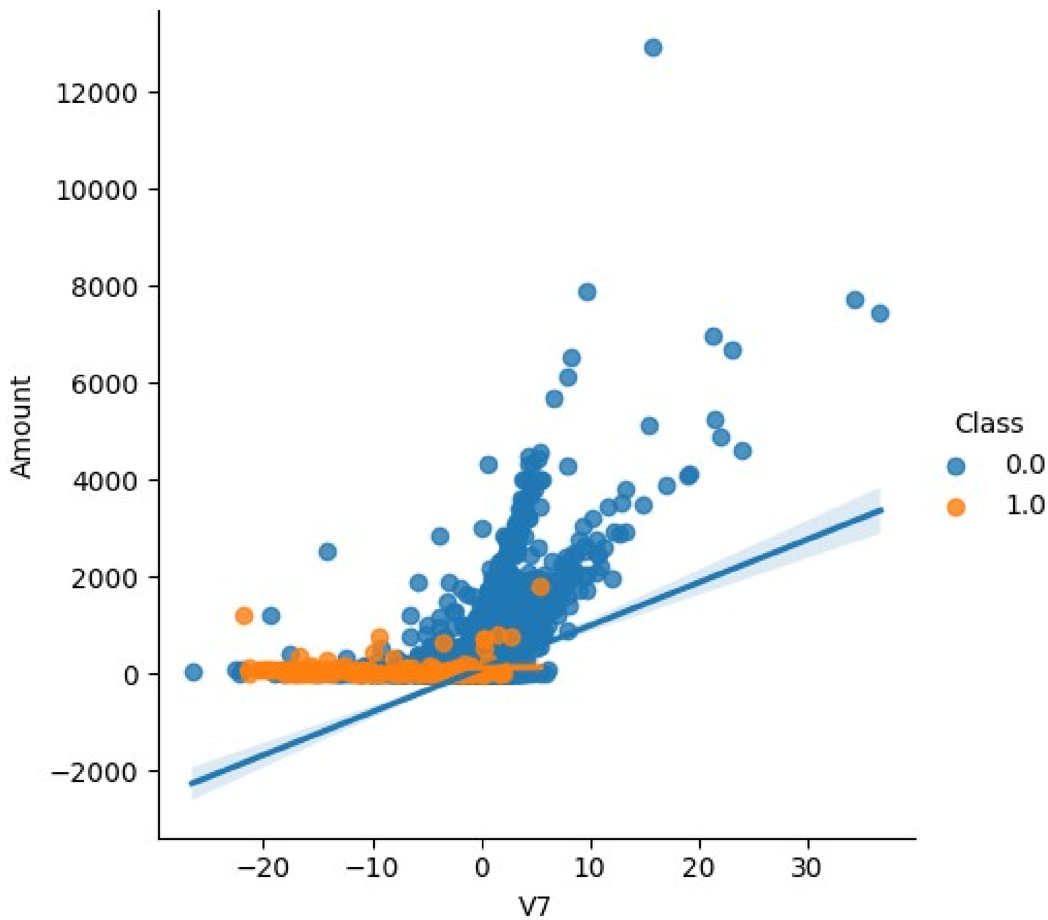
Amount
V4
Class
0.0
1.0
12000
10000
8000
6000
4000
2000
0
−5
0
5
10
15

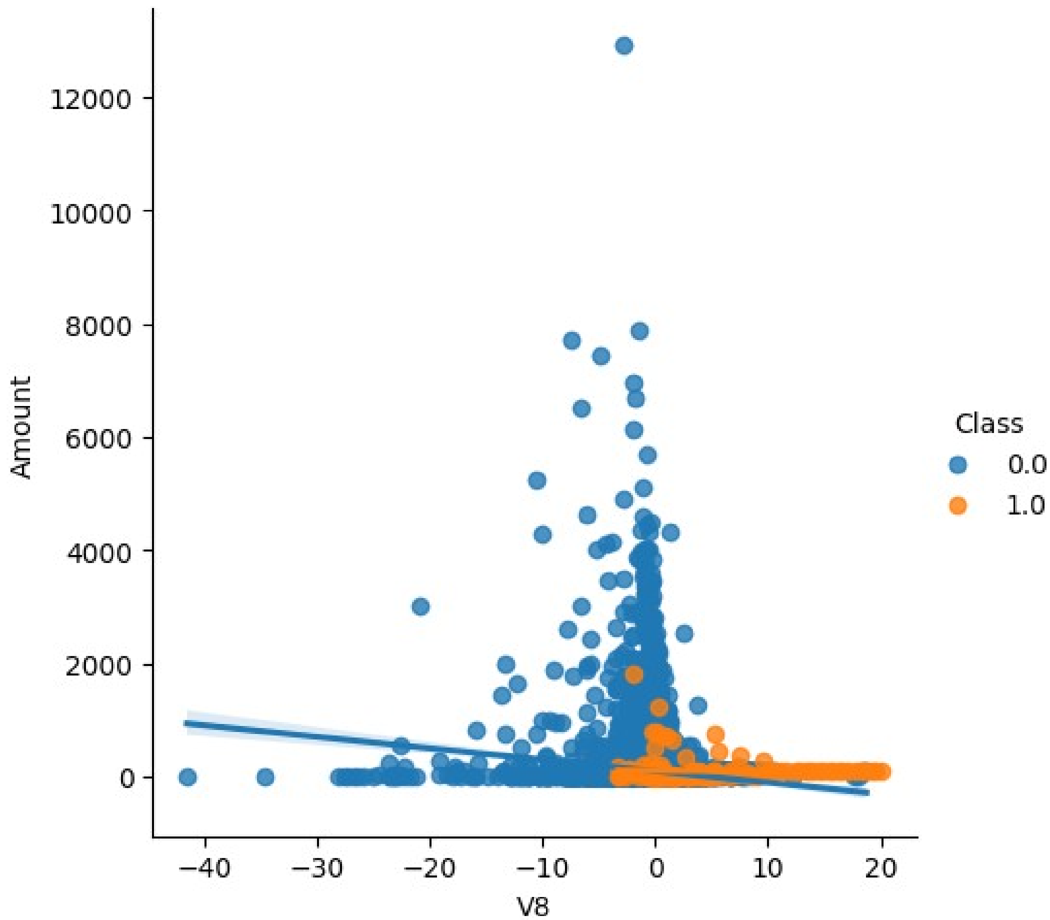


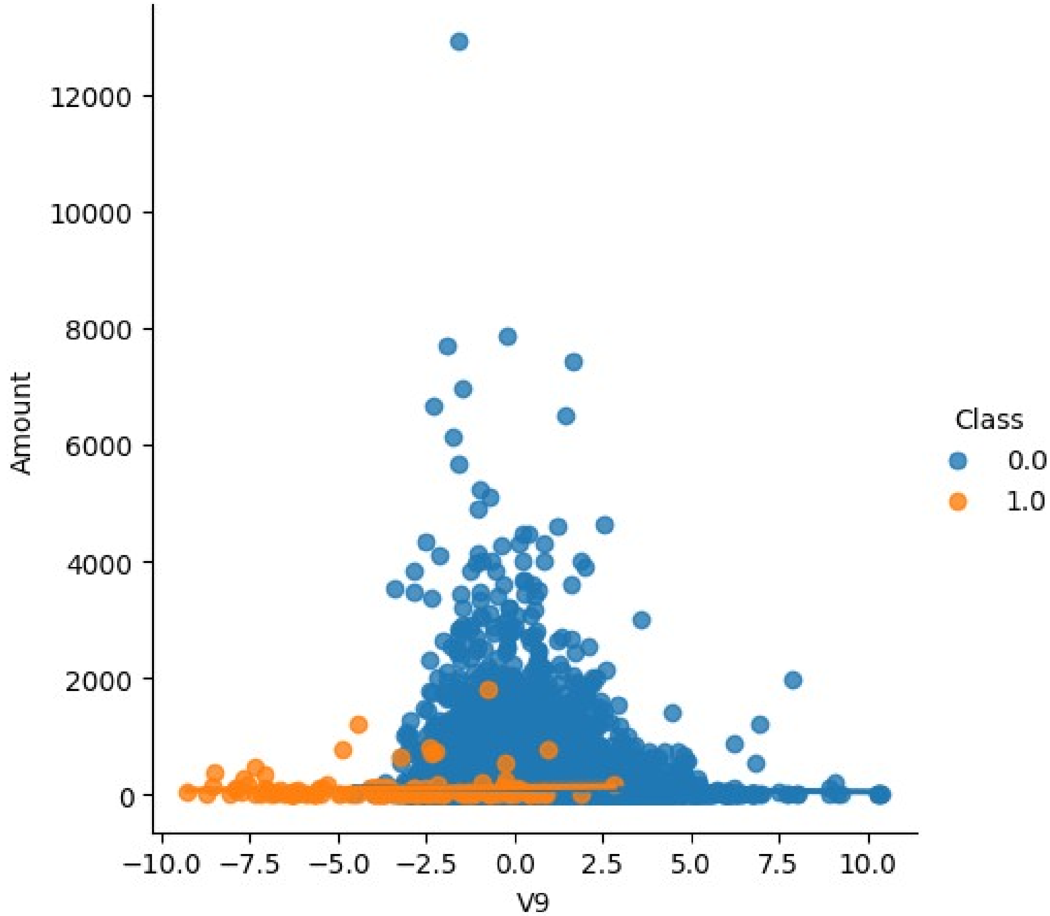
12000
10000
8000
6000
4000
2000
0
−2000
Amount
−40
−30
−20
−10
0
10
20
30
V5
Class
0.0
1.0



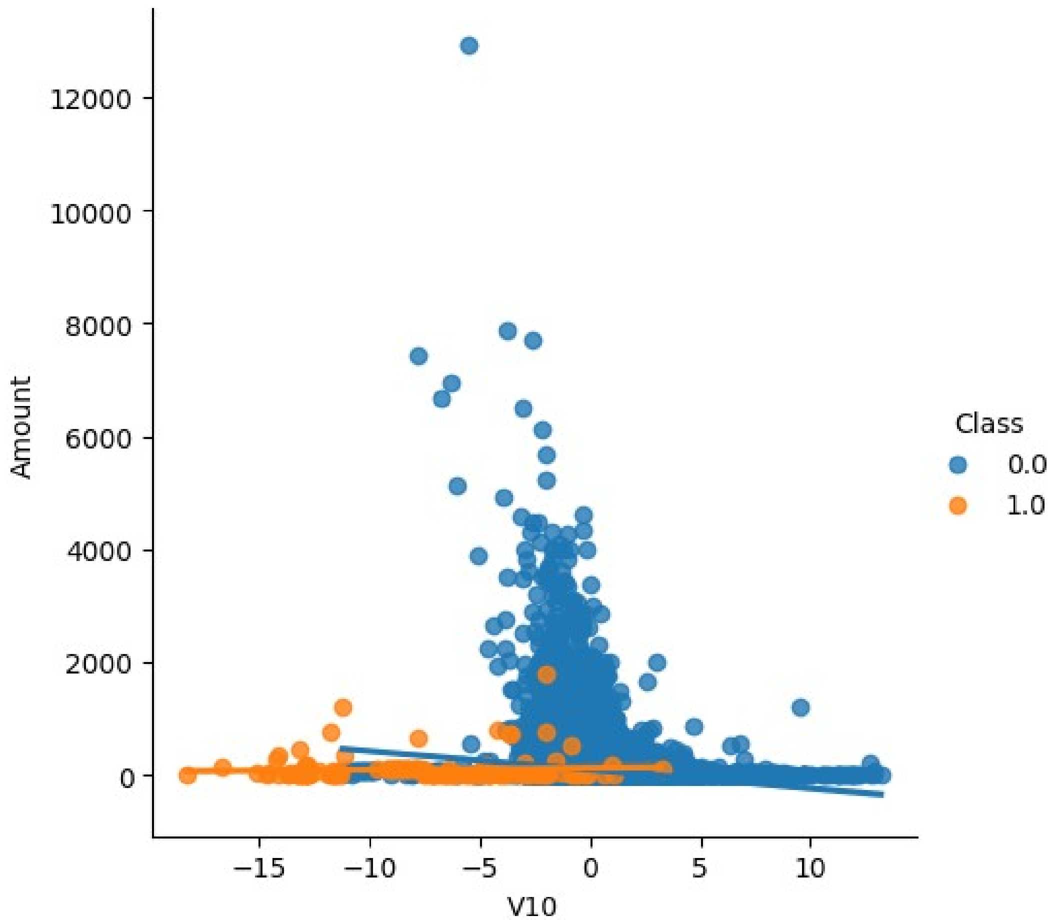
Amount
V6
Class
0.0
1.0
12000
10000
8000
6000
4000
2000
0
−20
−10
0
10
20


12000
10000
8000
6000
4000
2000
0
−2000
Amount
−20
−10
0
10
20
30
V7
Class
0.0
1.0

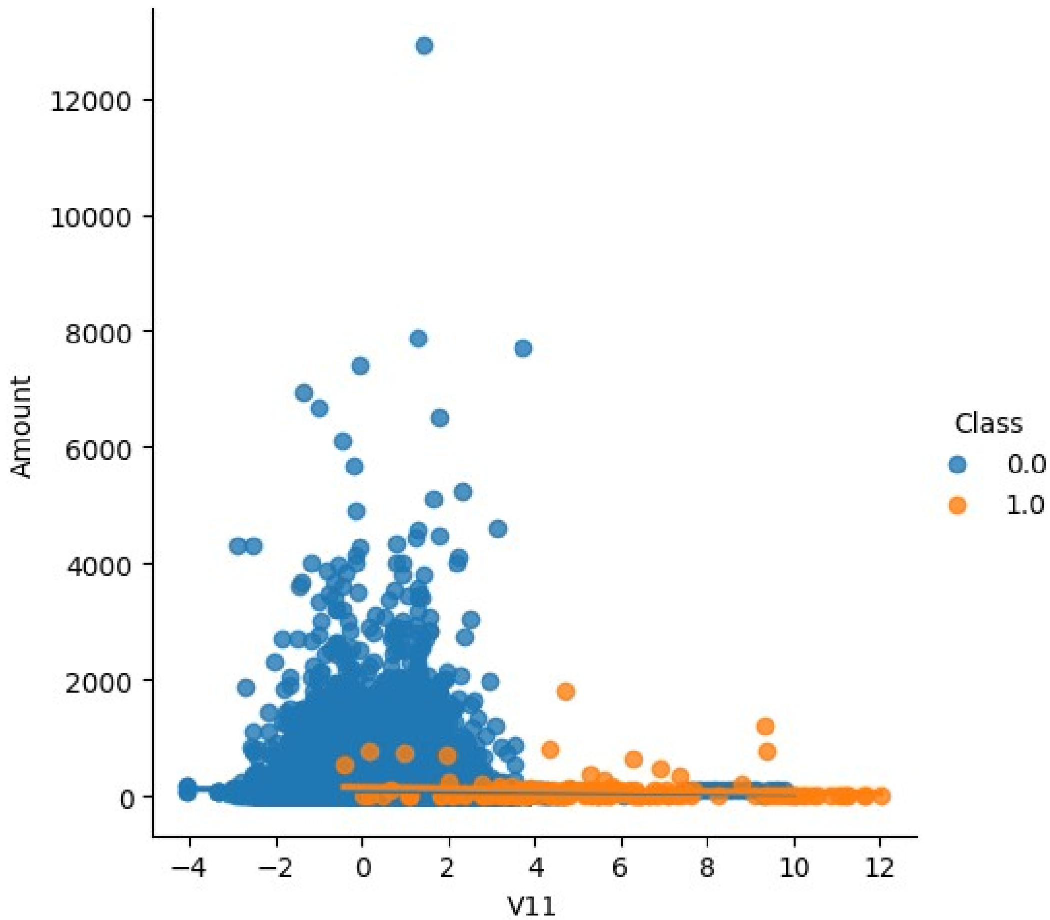

12000
10000
8000
6000
4000
2000
0
Amount
−40
−30
−20
−10
0
10
20
V8
Class
0.0
1.0



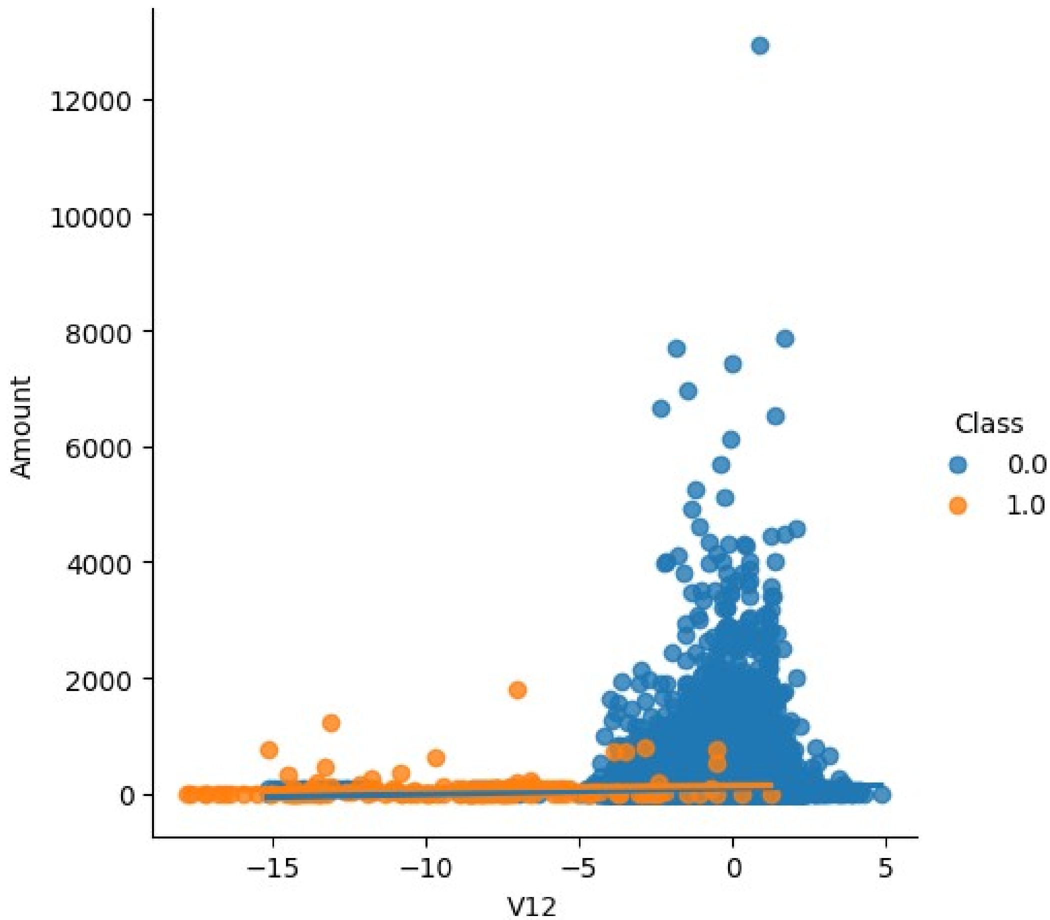
12000
10000
8000
6000
4000
2000
0
Amount
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
V9
Class
0.0
1.0

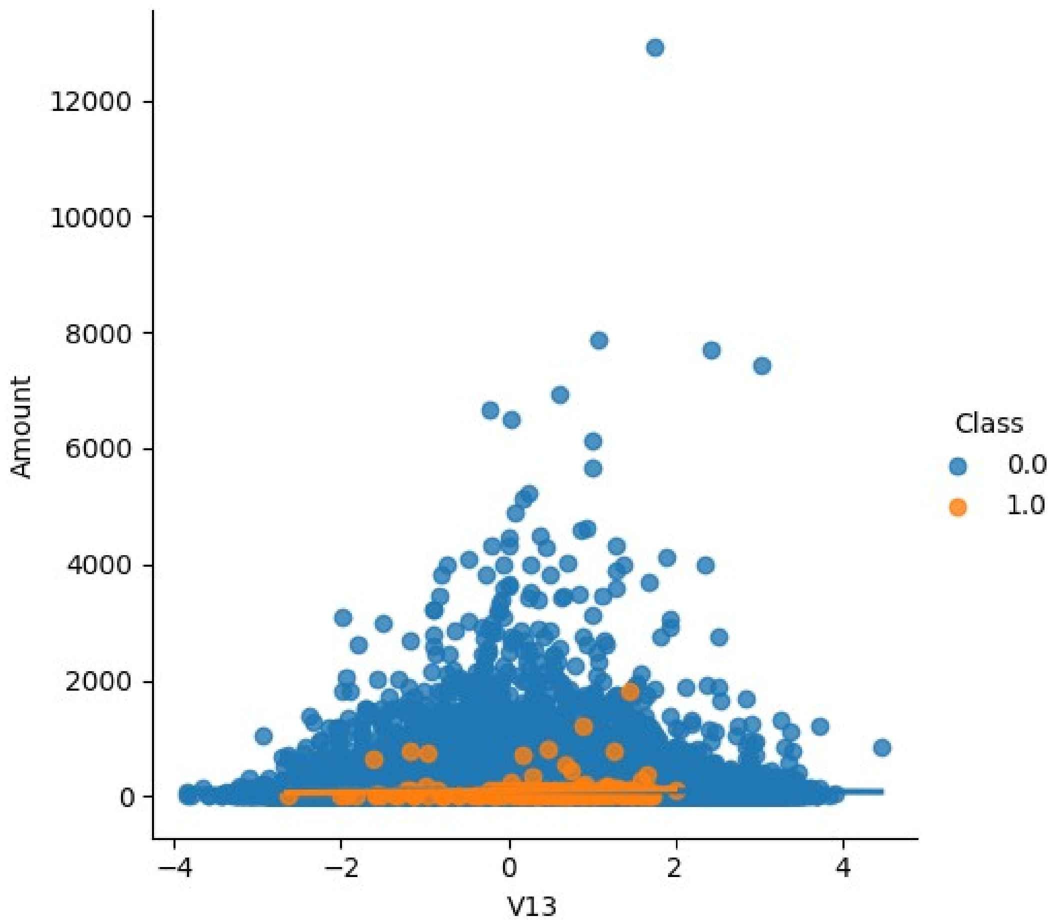


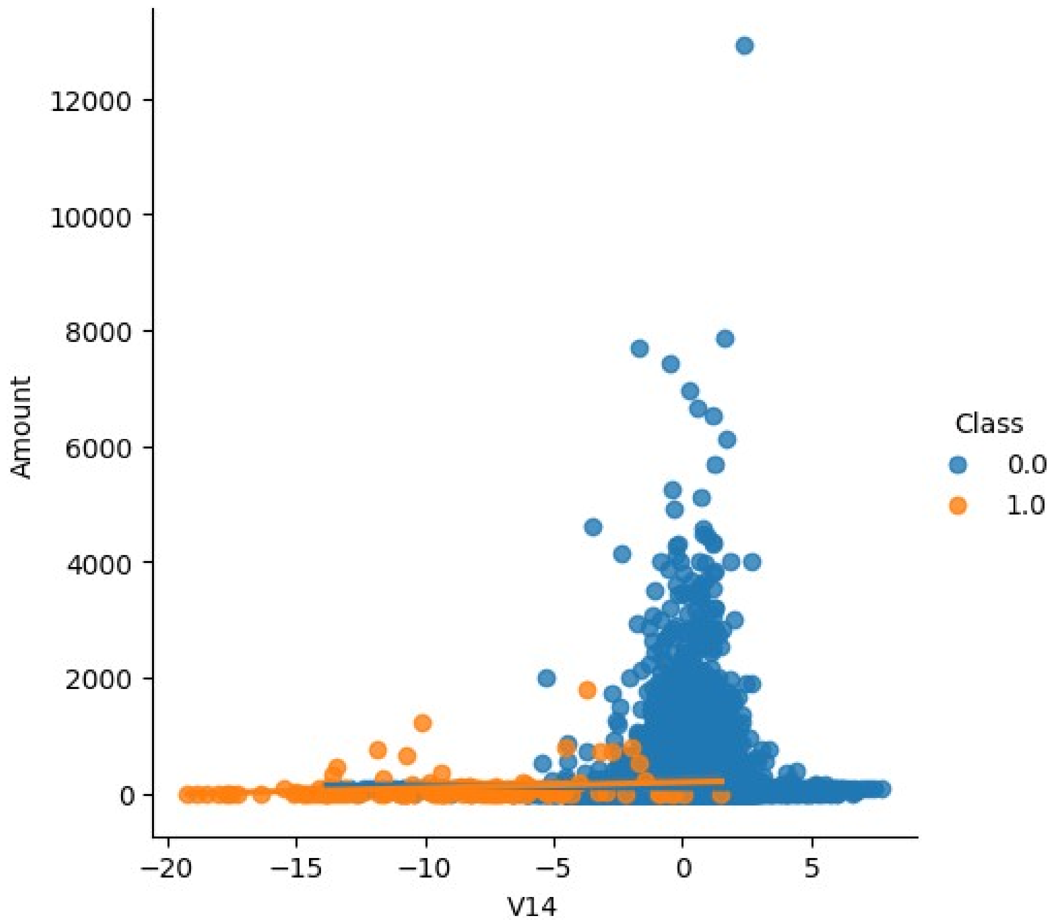
12000
10000
8000
6000
4000
2000
0
Amount
−15
−10
−5
0
5
10
V10
Class
0.0
1.0



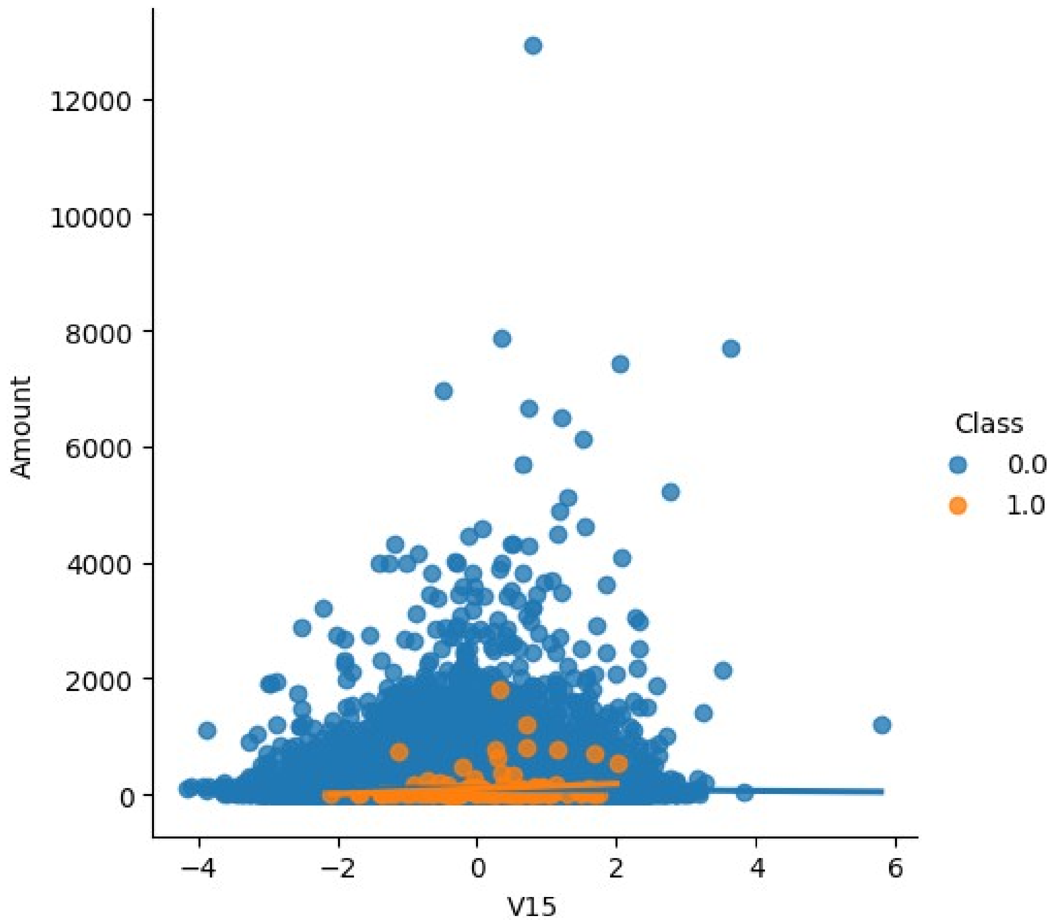
12000
10000
8000
6000
4000
2000
0
Amount
−4
−2
0
2
4
6
8
10
12
V11
Class
0.0
1.0


12000
10000
8000
6000
4000
2000
0
Amount
−15
−10
−5
0
5
V12
Class
0.0
1.0

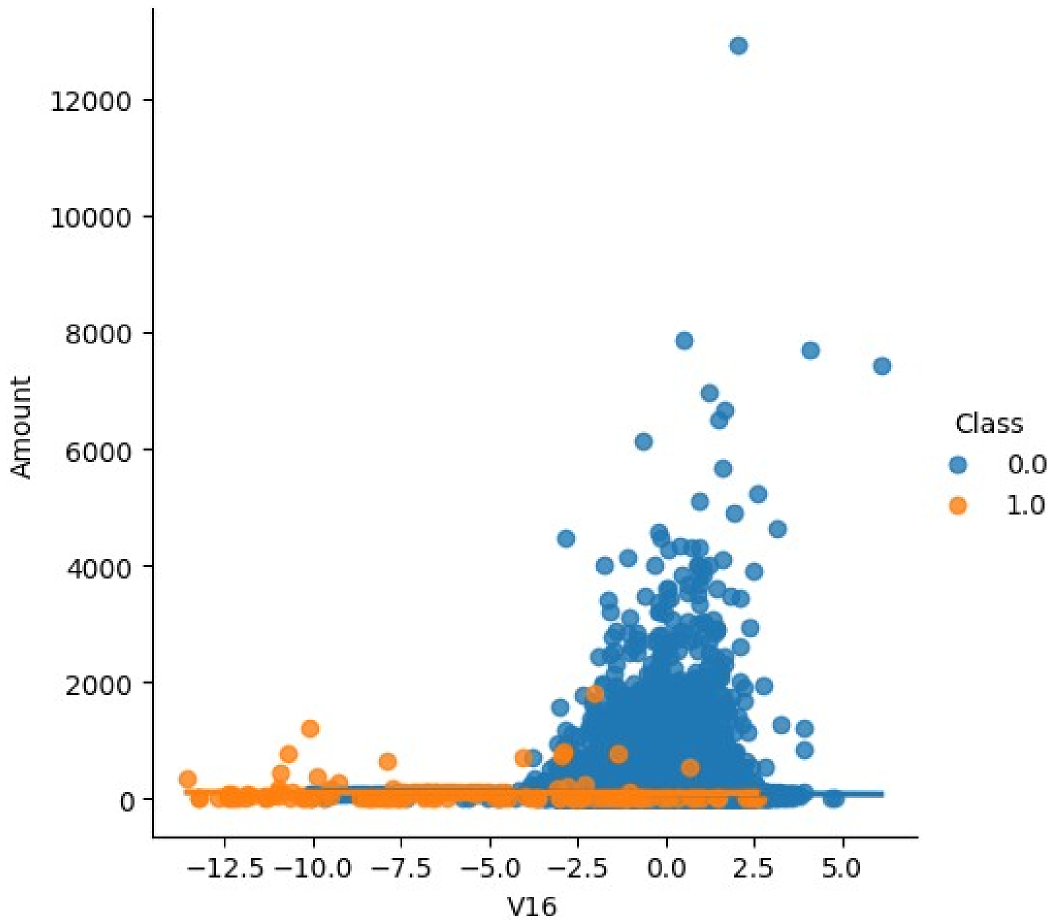

12000
10000
8000
6000
4000
2000
0
Amount
−4
−2
0
2
4
V13
Class
0.0
1.0

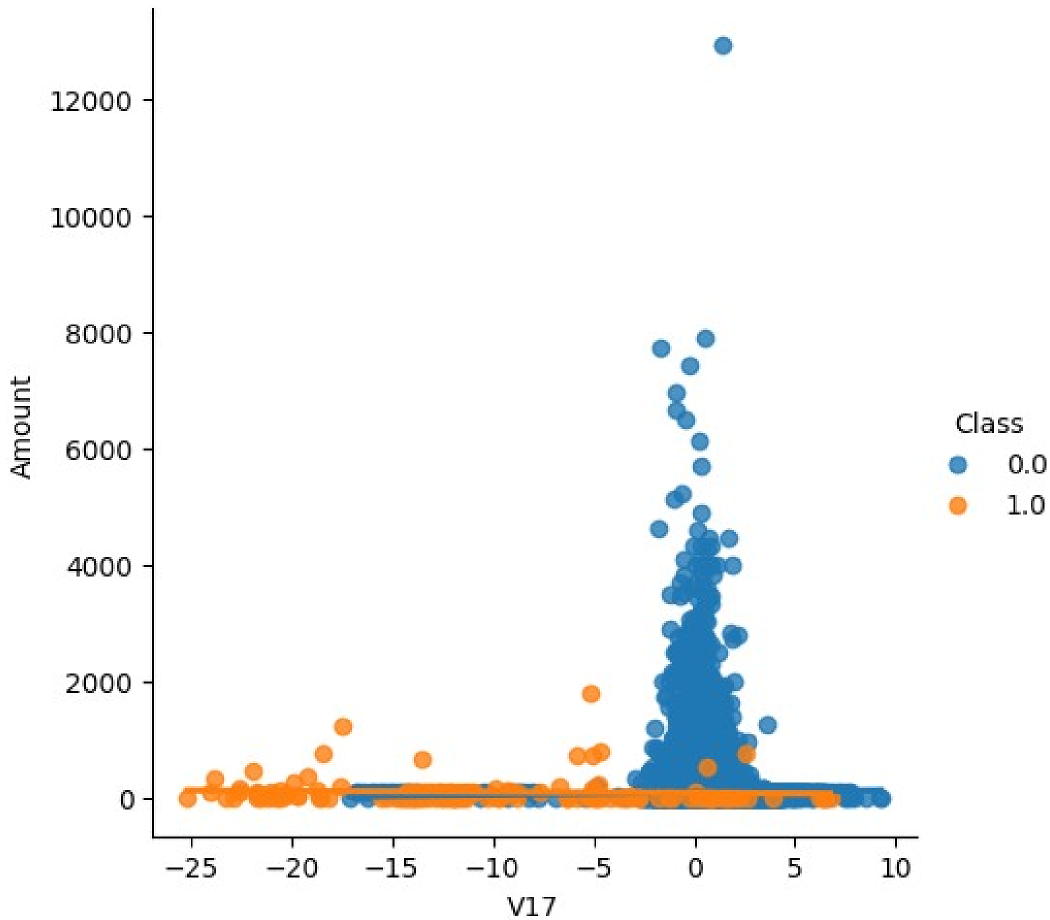


12000
10000
8000
6000
4000
2000
0
Amount
−20
−15
−10
−5
0
5
V14
Class
0.0
1.0

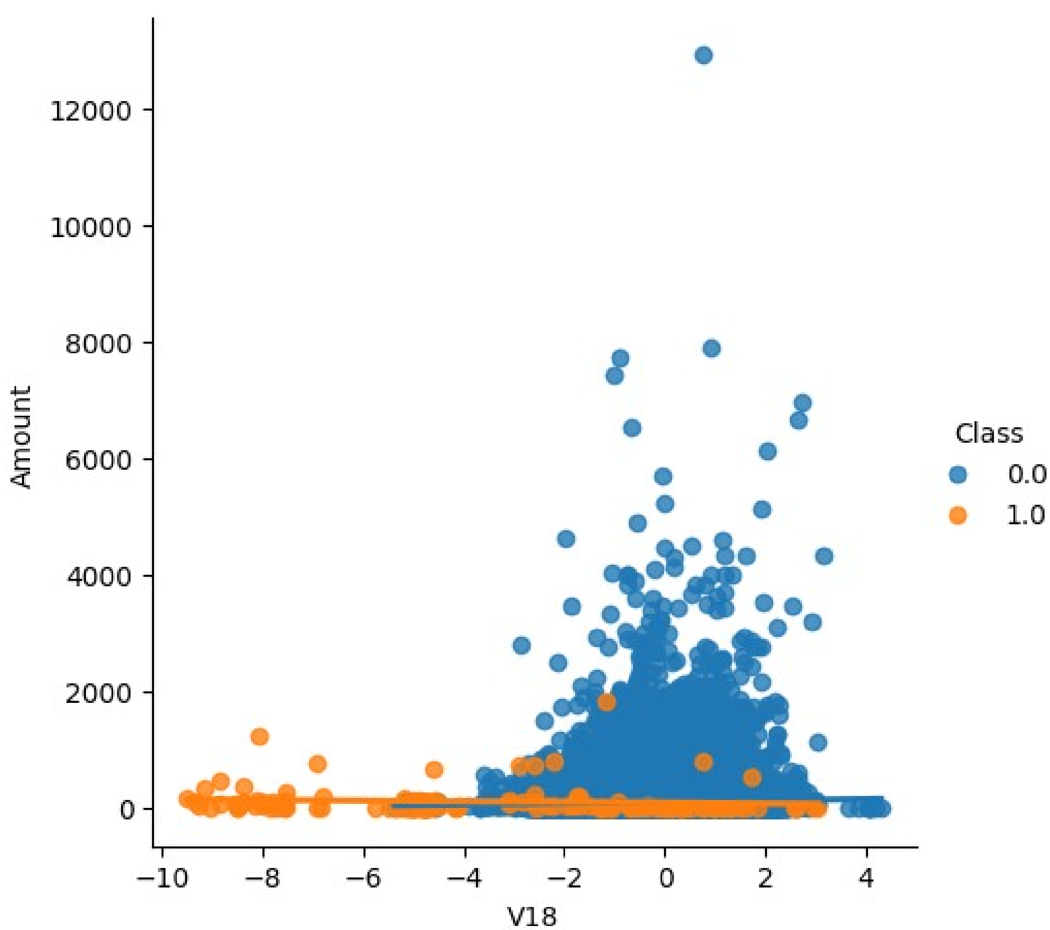


12000
10000
8000
6000
4000
2000
0
Amount
−4
−2
0
2
4
6
V15
Class
0.0
1.0

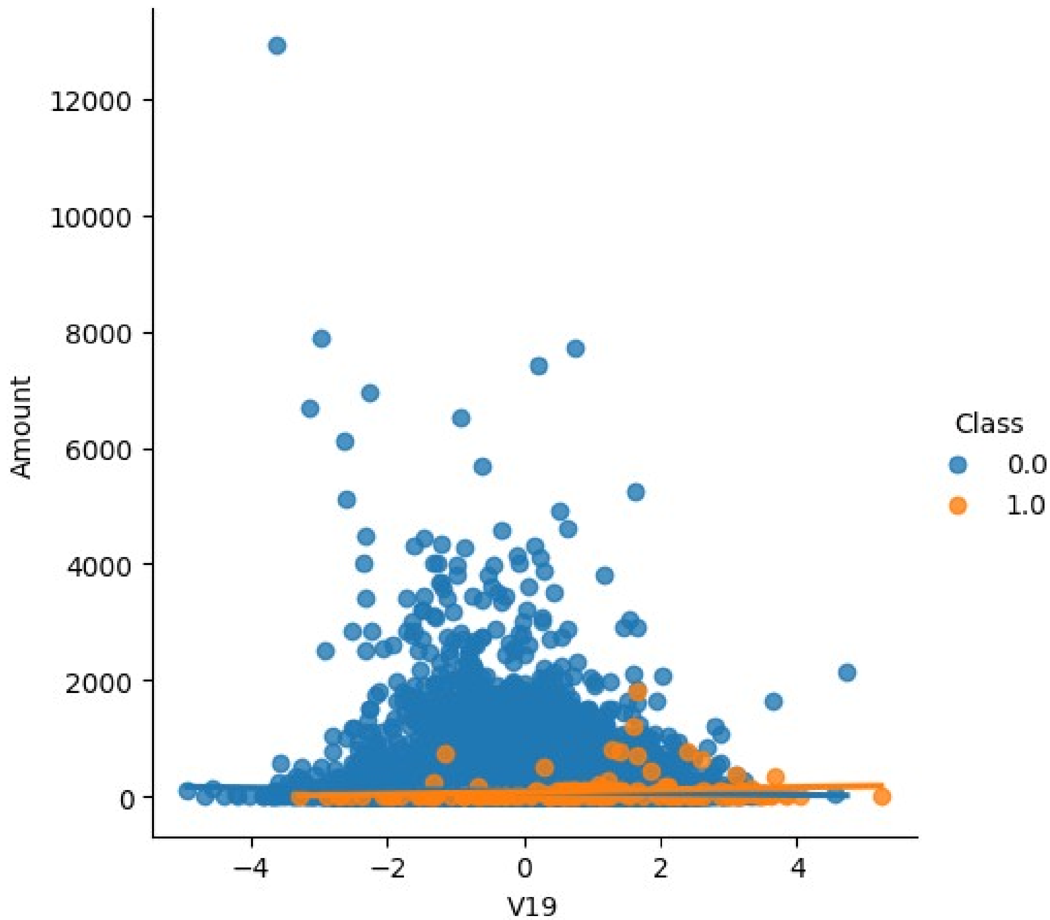

12000
10000
8000
6000
4000
2000
0
Amount
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
V16
Class
0.0
1.0



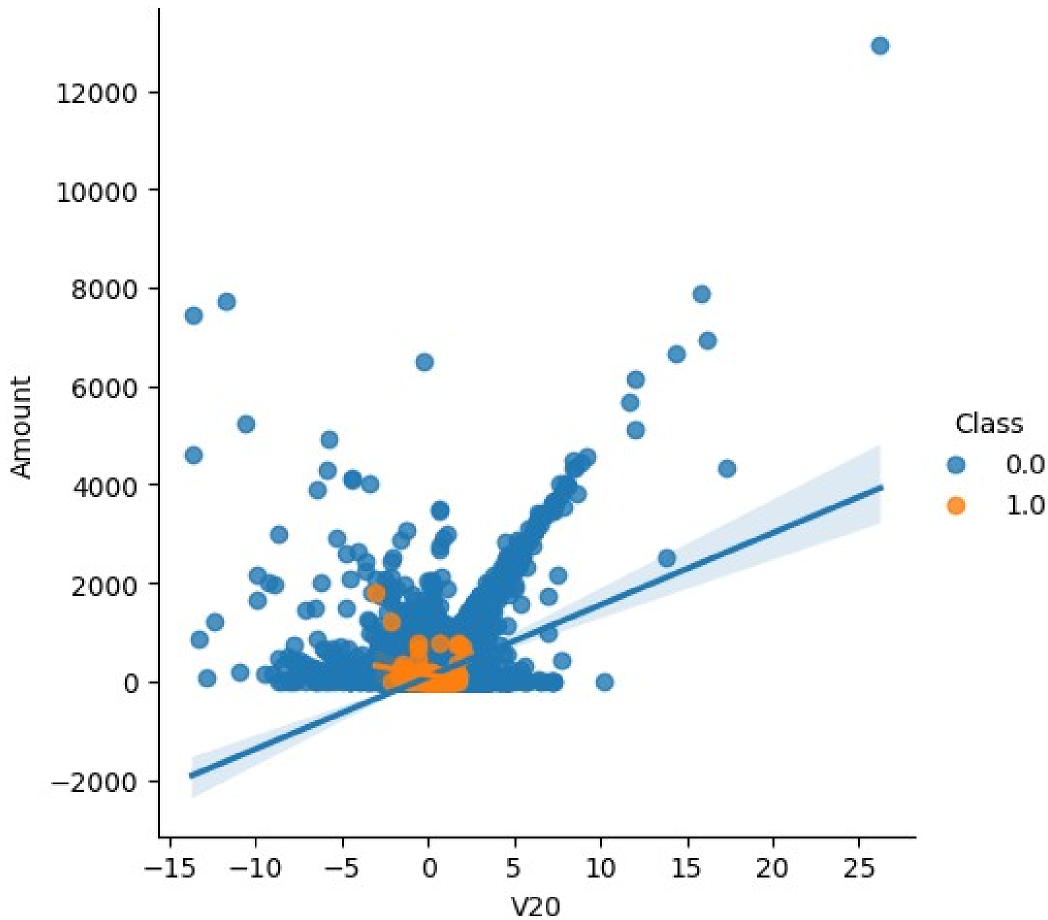
12000
10000
8000
6000
4000
2000
0
Amount
−25
−20
−15
−10
−5
0
5
10
V17
Class
0.0
1.0



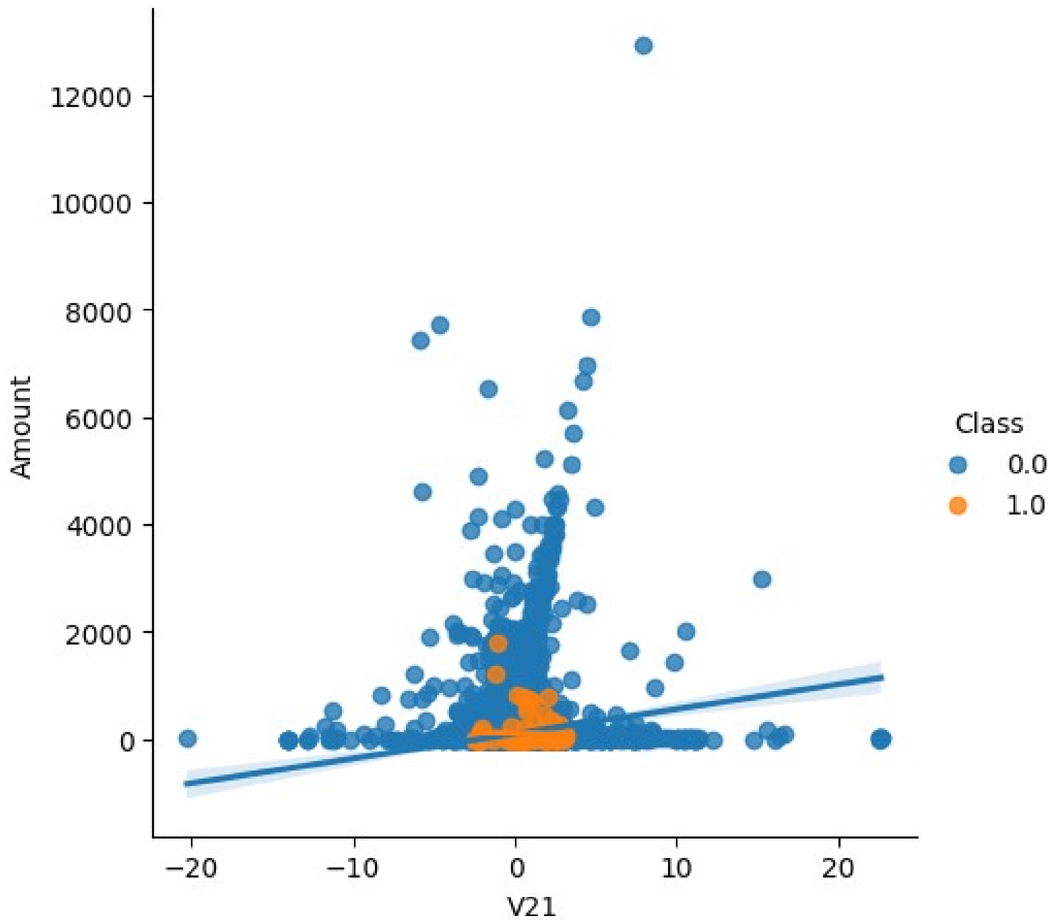
12000
10000
8000
6000
4000
2000
0
Amount
−10
−8
−6
−4
−2
0
2
4
V18
Class
0.0
1.0



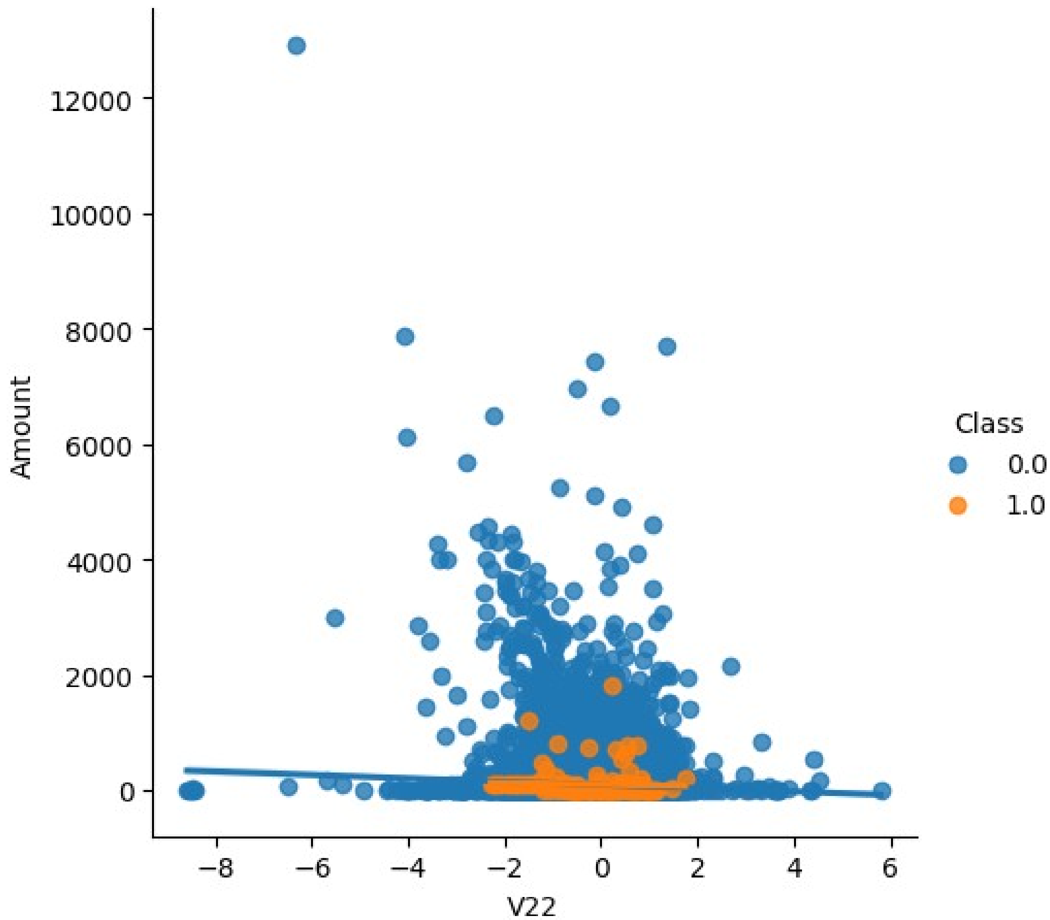
12000
10000
8000
6000
4000
2000
0
Amount
−4
−2
0
2
4
V19
Class
0.0
1.0

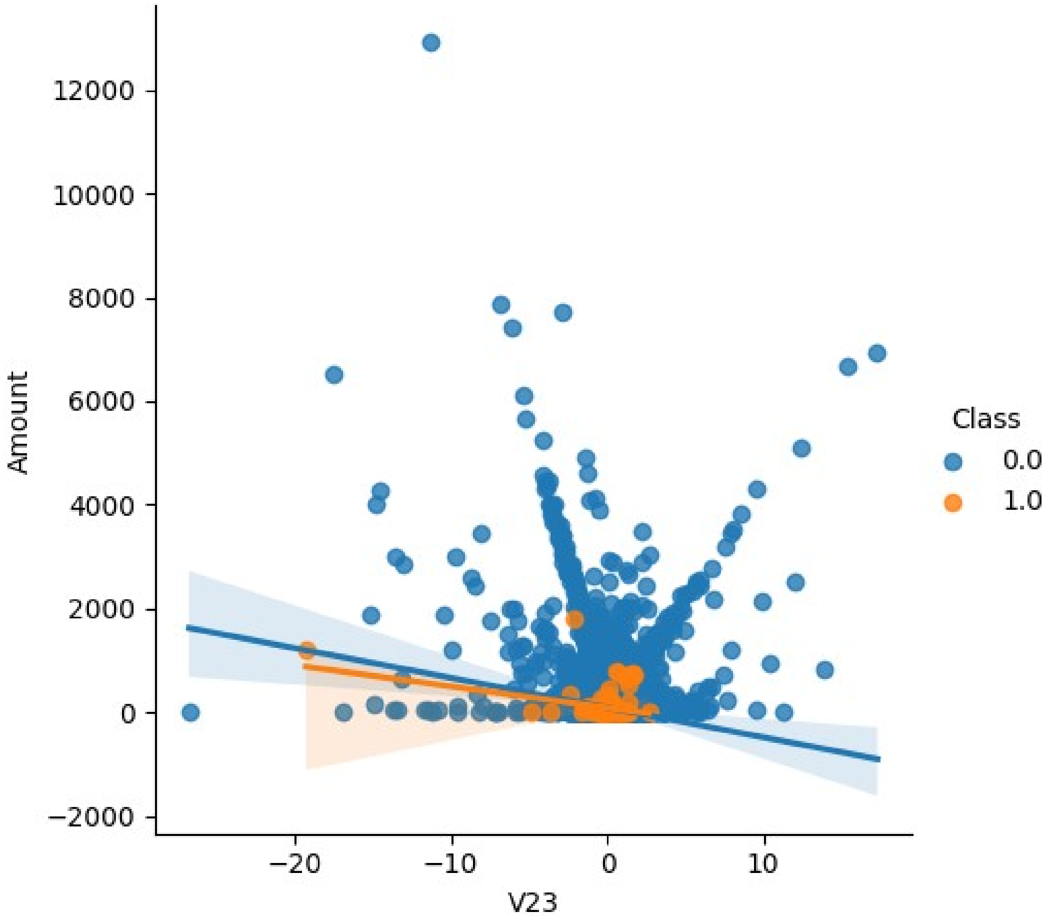


12000
10000
8000
6000
4000
2000
0
−2000
Amount
−15
−10
−5
0
5
10
15
20
25
V20
Class
0.0
1.0

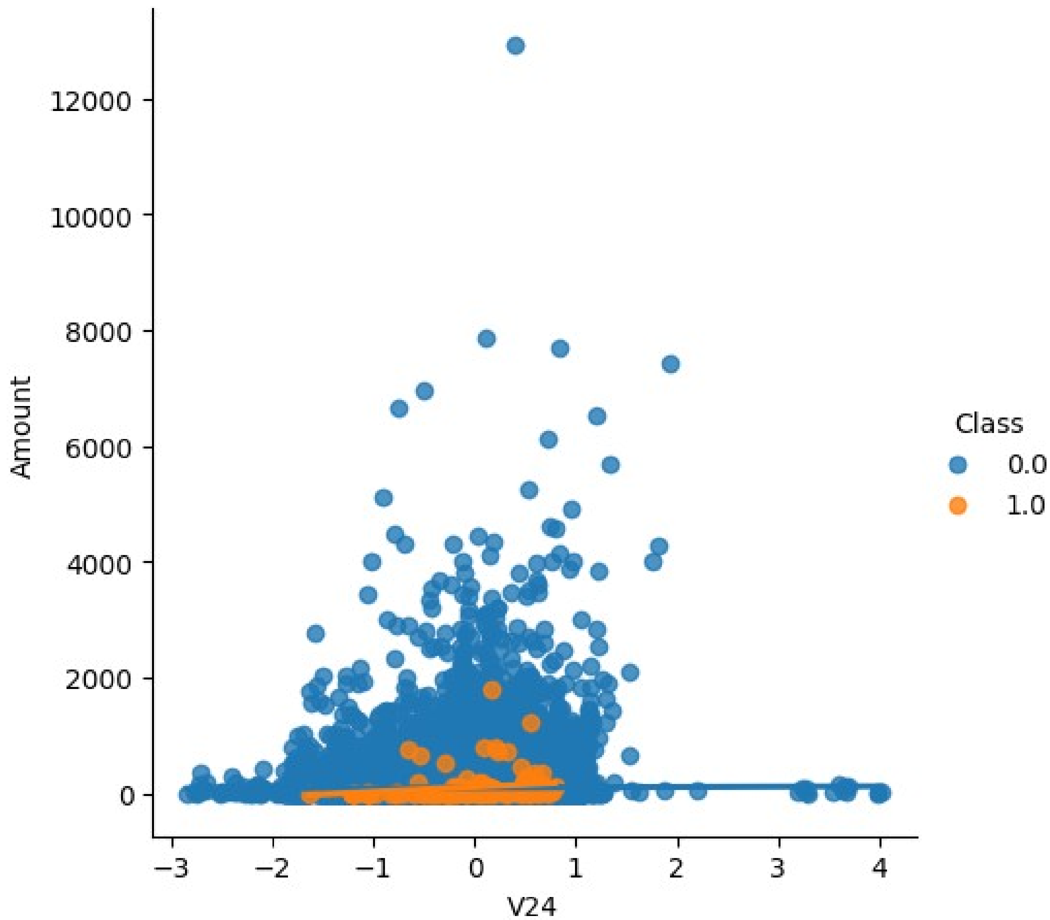


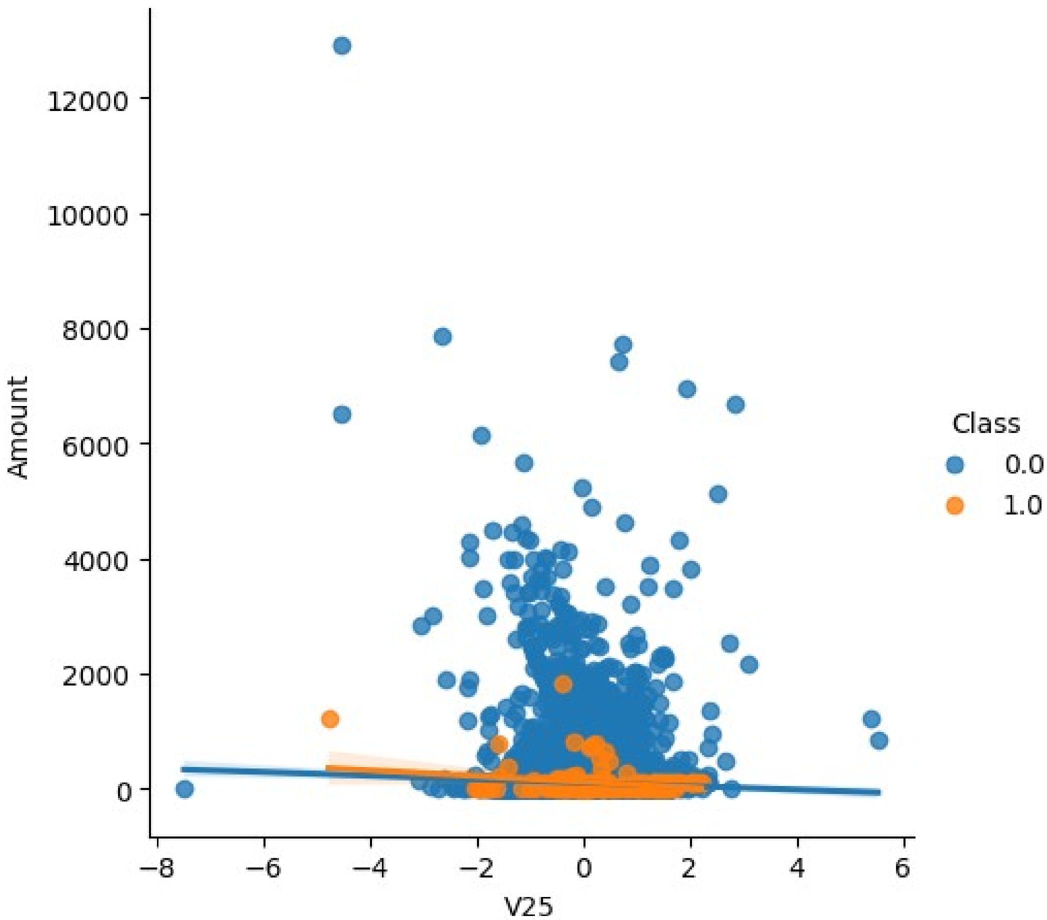
Amount
V21
Class
0.0
1.0


12000
10000
8000
6000
4000
2000
0
Amount
−8
−6
−4
−2
0
2
4
6
V22
Class
0.0
1.0

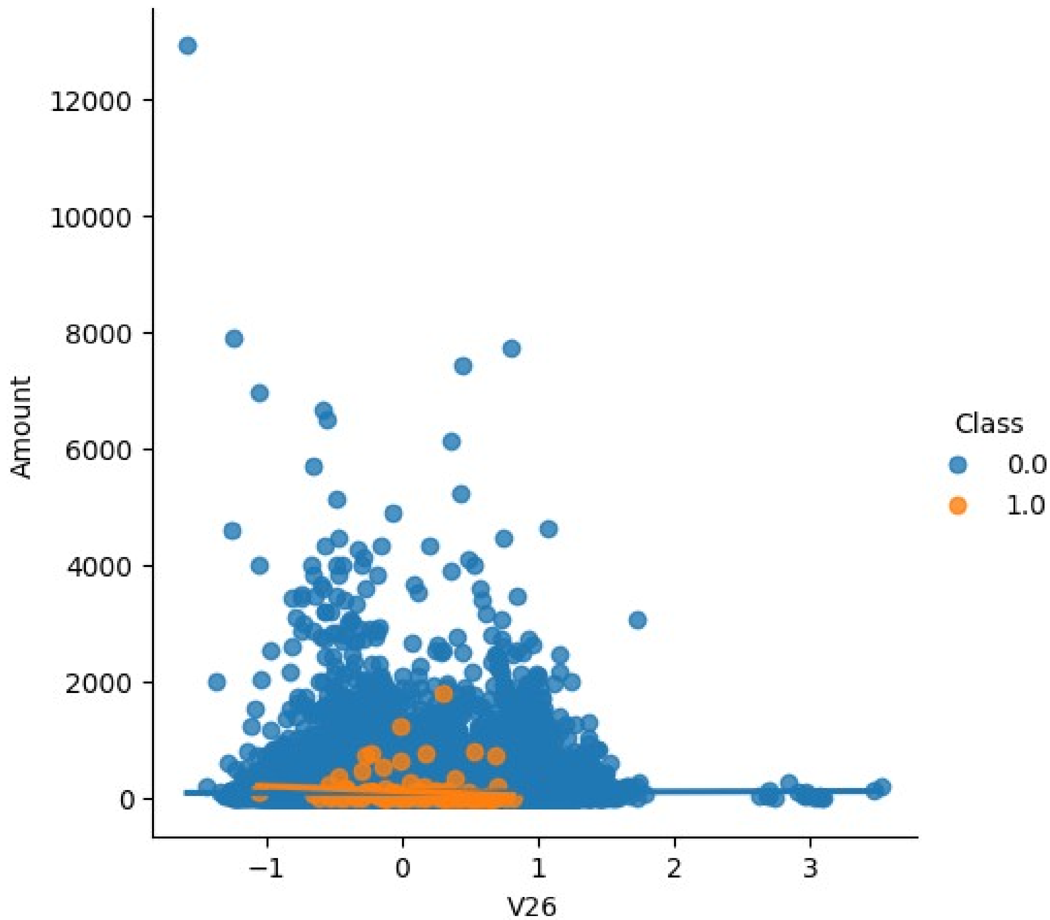


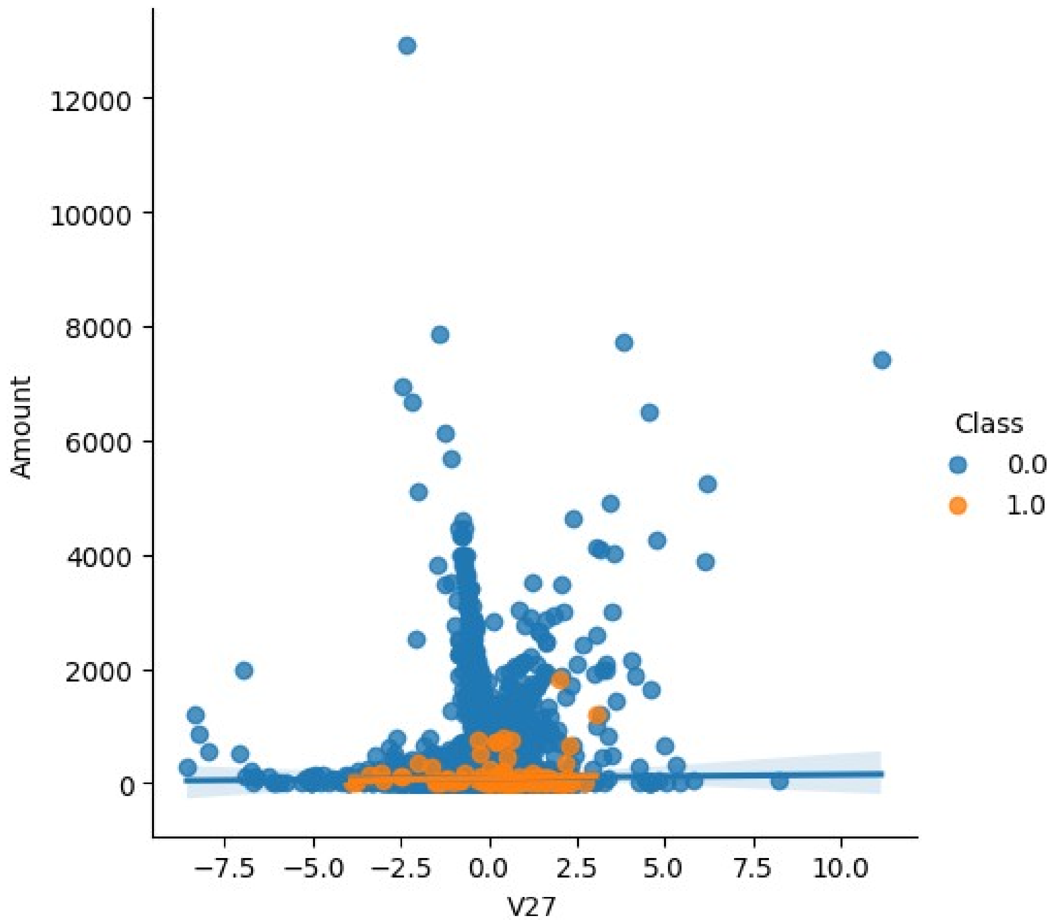
12000
10000
8000
6000
4000
2000
0
−2000
Amount
−20
−10
0
10
V23
Class
0.0
1.0


12000
10000
8000
6000
4000
2000
0
Amount
−3
−2
−1
0
1
2
3
4
V24
Class
0.0
1.0

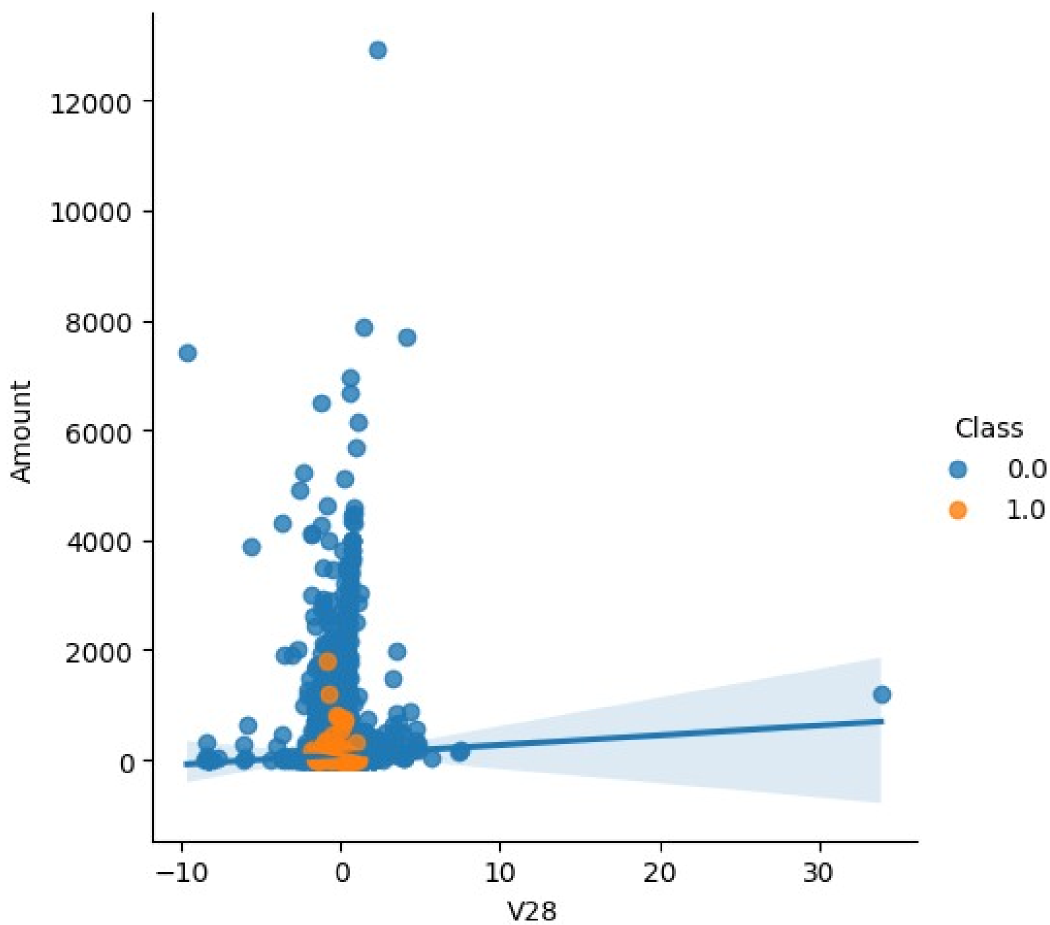


12000
10000
8000
6000
4000
2000
0
Amount
−8
−6
−4
−2
0
2
4
6
V25
Class
0.0
1.0

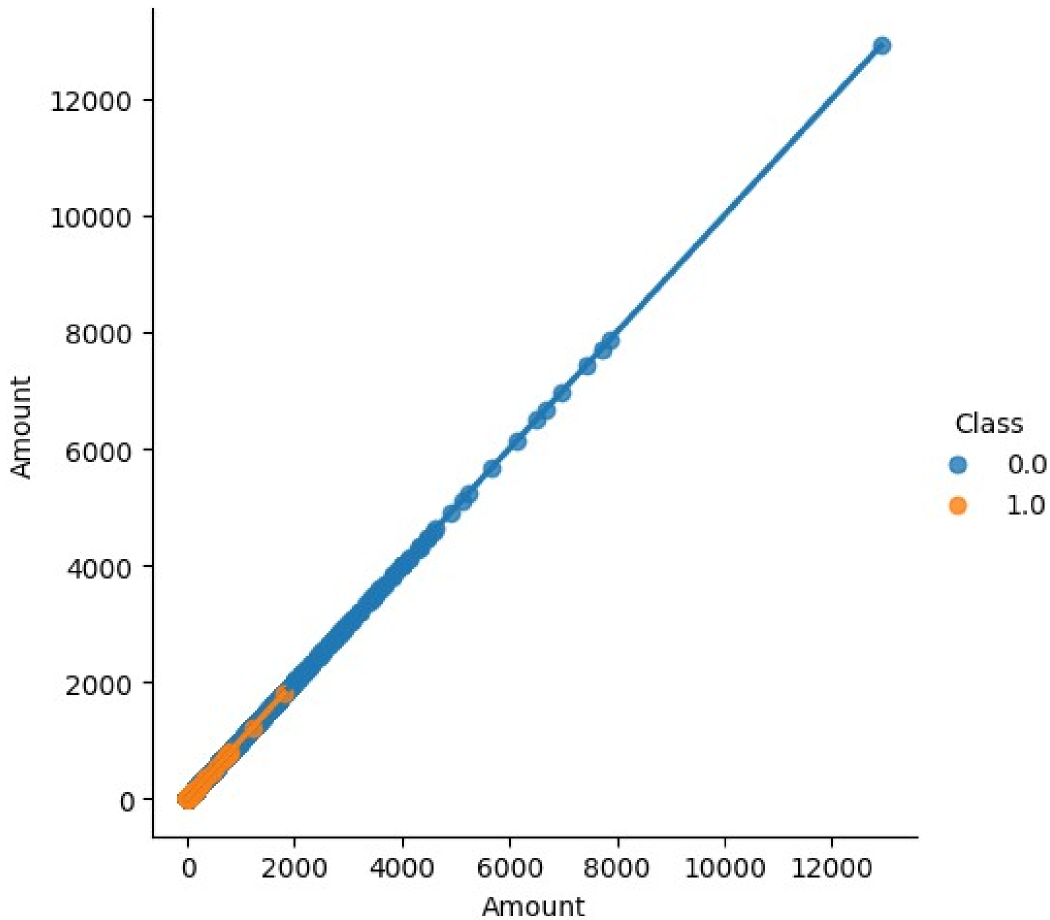

12000
10000
8000
6000
4000
2000
0
Amount
−1
0
1
2
3
V26
Class
0.0
1.0



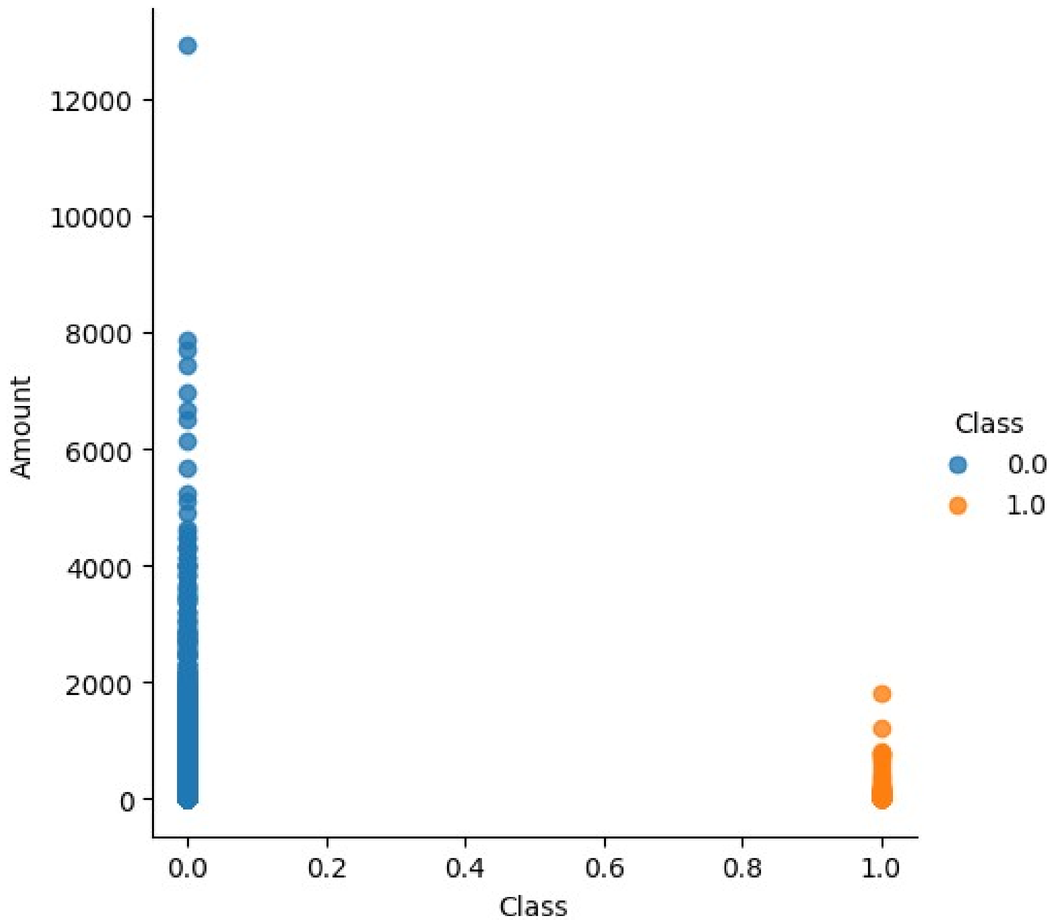
12000
10000
8000
6000
4000
2000
0
Amount
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
V27
Class
0.0
1.0



12000
10000
8000
6000
4000
2000
0
Amount
−10
0
10
20
30
V28
Class
0.0
1.0



12000
10000
8000
6000
4000
2000
0
Amount
0
2000
4000
6000
8000
10000
12000
Amount
Class
0.0
1.0




```
data.columns

Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9',
'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19',
'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28',
'Amount',
       'Class'],
      dtype='object')

X = data.drop(['Class'], axis = 1)
Y = data["Class"]
print(X.shape)
print(Y.shape)
xData = X.values
yData = Y.values

(49609, 30)
(49609,)
```

```python
from sklearn.model_selection import train_test_split
xTrain, xTest, yTrain, yTest = train_test_split(xData, yData,
test_size = 0.2, random_state = 42)

from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(xTrain, yTrain)
yPred = rfc.predict(xTest)

# Evaluating the classifier
# printing every score of the classifier
# scoring in anything
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix

n_outliers = len(fraud)
print("The total outliers are{}".format(n_outliers))

n_errors = (yPred != yTest).sum()
print("The taotal error is {}".format(n_errors))
print("The model used is Random Forest classifier")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))

rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is{}".format(MCC))
```

```
The total outliers are148
The taotal error is 3
The model used is Random Forest classifier
The accuracy is 0.9996976416045152
The precision is 1.0
The recall is 0.9090909090909091
The F1-Score is 0.9523809523809523
The Matthews correlation coefficient is0.9533179974202829
```

```python
LABELS = ['Normal', 'Fraud']
conf_matrix = confusion_matrix(yTest, yPred)
plt.figure(figsize =(12, 12))
```

```
sns.heatmap(conf_matrix, xticklabels = LABELS,
            yticklabels = LABELS, annot = True, fmt ="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```