Fraud Detection

September 4, 2023

[10]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

[ ]:

```
credit_data = pd.read_csv('creditcard.csv')
```

[13]:

```
credit_data
```

[13]:

```
            Time         V1         V2        V3        V4        V5  \
0            0.0  -1.359807  -0.072781  2.536347  1.378155 -0.338321
1            0.0   1.191857   0.266151  0.166480  0.448154  0.060018
2            1.0  -1.358354  -1.340163  1.773209  0.379780 -0.503198
3            1.0  -0.966272  -0.185226  1.792993 -0.863291 -0.010309
4            2.0  -1.158233   0.877737  1.548718  0.403034 -0.407193
…              …          …          …         …         …         …
284802  172786.0 -11.881118  10.071785 -9.834783 -2.066656 -5.364473
284803  172787.0  -0.732789  -0.055080  2.035030 -0.738589  0.868229
284804  172788.0   1.919565  -0.301254 -3.249640 -0.557828  2.630515
284805  172788.0  -0.240440   0.530483  0.702510  0.689799 -0.377961
284806  172792.0  -0.533413  -0.189733  0.703337 -0.506271 -0.012546

              V6        V7        V8        V9  …       V21       V22  \
0       0.462388  0.239599  0.098698  0.363787  … -0.018307  0.277838
1      -0.082361 -0.078803  0.085102 -0.255425  … -0.225775 -0.638672
2       1.800499  0.791461  0.247676 -1.514654  …  0.247998  0.771679
3       1.247203  0.237609  0.377436 -1.387024  … -0.108300  0.005274
4       0.095921  0.592941 -0.270533  0.817739  … -0.009431  0.798278
…              …         …         …         …  …         …         …
284802 -2.606837 -4.918215  7.305334  1.914428  …  0.213454  0.111864
284803  1.058415  0.024330  0.294869  0.584800  …  0.214205  0.924384
284804  3.031260 -0.296827  0.708417  0.432454  …  0.232045  0.578229
284805  0.623708 -0.686180  0.679145  0.392087  …  0.265245  0.800049
284806 -0.649617  1.577006 -0.414650  0.486180  …  0.261057  0.643078

             V23       V24       V25       V26       V27       V28  Amount  \
0       -0.110474  0.066928  0.128539 -0.189115  0.133558 -0.021053  149.62
1        0.101288 -0.339846  0.167170  0.125895 -0.008983  0.014724    2.69
2        0.909412 -0.689281 -0.327642 -0.139097 -0.055353 -0.059752  378.66
3       -0.190321 -1.175575  0.647376 -0.221929  0.062723  0.061458  123.50
4       -0.137458  0.141267 -0.206010  0.502292  0.219422  0.215153   69.99
…             …         …         …         …         …         …      …
284802  1.014480 -0.509348  1.436807  0.250034  0.943651  0.823731    0.77
284803  0.012463 -1.016226 -0.606624 -0.395255  0.068472 -0.053527   24.79
284804 -0.037501  0.640134  0.265745 -0.087371  0.004455 -0.026561   67.88
284805 -0.163298  0.123205 -0.569159  0.546668  0.108821  0.104533   10.00
284806  0.376777  0.008797 -0.473649 -0.818267 -0.002415  0.013649  217.00

        Class
0           0
1           0
2           0
3           0
4           0
…         …
284802      0
284803      0
284804      0
284805      0
284806      0

[284807 rows x 31 columns]
```

```
[16]: credit_data.head()
```

```
[16]:    Time        V1        V2        V3        V4        V5        V6        V7  \
      0   0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388  0.239599
      1   0.0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361 -0.078803
      2   1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499  0.791461
      3   1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203  0.237609
      4   2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921  0.592941

              V8        V9  …       V21       V22       V23       V24       V25  \
      0  0.098698  0.363787  … -0.018307  0.277838 -0.110474  0.066928  0.128539
      1  0.085102 -0.255425  … -0.225775 -0.638672  0.101288 -0.339846  0.167170
      2  0.247676 -1.514654  …  0.247998  0.771679  0.909412 -0.689281 -0.327642
      3  0.377436 -1.387024  … -0.108300  0.005274 -0.190321 -1.175575  0.647376
      4 -0.270533  0.817739  … -0.009431  0.798278 -0.137458  0.141267 -0.206010

              V26       V27       V28  Amount  Class
      0 -0.189115  0.133558 -0.021053  149.62      0
      1  0.125895 -0.008983  0.014724    2.69      0
      2 -0.139097 -0.055353 -0.059752  378.66      0
```

```
3 -0.221929  0.062723  0.061458  123.50      0
4  0.502292  0.219422  0.215153   69.99      0

[5 rows x 31 columns]
```

[26]:
```
credit_data.isnull().sum()
```

[26]:
```
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
```

[18]:
```
credit_data['Class'].value_counts()
```

[18]:
```
0    284315
1       492
Name: Class, dtype: int64
```

[24]:
```
legit = credit_data[credit_data.Class == 0]
fraud = credit_data[credit_data.Class == 1]
```

[26]:
```
print(legit.shape)
print(fraud.shape)
```

```
(284315, 31)
(492, 31)
```

[28]:
```
legit.Amount.describe()
```

[28]:
```
count    284315.000000
mean         88.291022
std         250.105092
min           0.000000
25%           5.650000
50%          22.000000
75%          77.050000
max       25691.160000
Name: Amount, dtype: float64
```

[30]:
```
fraud.Amount.describe()
```

[30]:
```
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

[32]:
```
credit_data.groupby('Class').mean()
```

[32]:
```
               Time        V1        V2        V3        V4        V5  \
Class
0      94838.202258  0.008258 -0.006271  0.012171 -0.007860  0.005453
1      80746.806911 -4.771948  3.623778 -7.033281  4.542029 -3.151225

             V6        V7        V8        V9  …       V20       V21  \
Class                                          …
0      0.002419  0.009637 -0.000987  0.004467  … -0.000644 -0.001235
1     -1.397737 -5.568731  0.570636 -2.581123  …  0.372319  0.713588

            V22       V23       V24       V25       V26       V27       V28  \
Class
```

```
0     -0.000024  0.000070  0.000182 -0.000072 -0.000089 -0.000295 -0.000131
1      0.014049 -0.040308 -0.105130  0.041449  0.051648  0.170575  0.075667

           Amount
Class
0       88.291022
1      122.211321

[2 rows x 30 columns]
```

```
[34]: legit_sample = legit.sample(n=492)
```

```
[36]: new_Dataset = pd.concat([legit_sample,fraud], axis = 0)
```

```
[38]: new_Dataset.head()
```

```
[38]:             Time        V1        V2        V3        V4        V5        V6  \
      202142  134191.0  2.105699 -0.096734 -1.646378 -0.071226  0.753233  0.162771
      191015  129072.0 -1.042917  0.383237  0.996325 -3.493985  0.789021  0.062523
      108550   70960.0  1.290044  0.557055 -1.009430  0.336459  1.075805  0.191822
      169298  119626.0  0.025423  0.875333  0.134680 -0.832928  0.841171 -0.414141
      175399  122319.0  2.066871  0.057168 -1.292515  0.352487  0.070528 -1.265175

                    V7        V8        V9  …       V21       V22       V23  \
      202142  0.041102 -0.055834  0.254505  … -0.307434 -0.794361  0.223205
      191015  0.822466  0.324396  1.123297  …  0.029282  0.325216 -0.423414
      108550  0.272615 -0.016897 -0.521548  … -0.133573 -0.261358 -0.245430
      169298  0.958980 -0.069149 -0.448027  … -0.221406 -0.429478 -0.010304
      175399  0.301441 -0.429491  0.465441  …  0.260473  0.970039 -0.003167

                   V24       V25       V26       V27       V28  Amount  Class
      202142 -0.335629 -0.146090  0.209132 -0.077160 -0.072713    1.98      0
      191015 -1.040133  0.782130 -0.189057  0.276484  0.082480    7.59      0
      108550 -1.393994  0.692878  0.461040 -0.014796  0.011991    0.76      0
      169298 -0.527207 -0.430356  0.152144  0.253524  0.084307    2.69      0
      175399  0.047588  0.282419 -0.108327 -0.010702 -0.057457    1.00      0

      [5 rows x 31 columns]
```

```
[42]: new_Dataset.groupby('Class').mean()
```

```
[42]:                Time        V1        V2        V3        V4        V5  \
      Class
      0      95656.193089  0.043065 -0.052197 -0.021868  0.002373 -0.003808
      1      80746.806911 -4.771948  3.623778 -7.033281  4.542029 -3.151225

                   V6        V7        V8        V9  …       V20       V21  \
      Class                                          …
      0     -0.080430 -0.056483 -0.049978  0.003580  … -0.033660 -0.005883
      1     -1.397737 -5.568731  0.570636 -2.581123  …  0.372319  0.713588

                  V22       V23       V24       V25       V26       V27       V28  \
      Class
      0     -0.016922  0.011402  0.061825  0.045292 -0.016363  0.014996 -0.030499
      1      0.014049 -0.040308 -0.105130  0.041449  0.051648  0.170575  0.075667

                 Amount
      Class
      0       79.776585
      1      122.211321

      [2 rows x 30 columns]
```

[44]:
```
X= new_Dataset.drop(columns='Class', axis=1)
```

[46]:
```
Y = new_Dataset['Class']
```

[48]:
```
print(X)
```

```
            Time        V1        V2        V3        V4        V5        V6  \
202142  134191.0  2.105699 -0.096734 -1.646378 -0.071226  0.753233  0.162771
191015  129072.0 -1.042917  0.383237  0.996325 -3.493985  0.789021  0.062523
108550   70960.0  1.290044  0.557055 -1.009430  0.336459  1.075805  0.191822
169298  119626.0  0.025423  0.875333  0.134680 -0.832928  0.841171 -0.414141
175399  122319.0  2.066871  0.057168 -1.292515  0.352487  0.070528 -1.265175
…            …       …       …       …       …       …       …       …
279863  169142.0 -1.927883  1.125653 -4.518331  1.749293 -1.566487 -2.010494
280143  169347.0  1.378559  1.289381 -5.004247  1.411850  0.442581 -1.326536
280149  169351.0 -0.676143  1.126366 -2.213700  0.468308 -1.120541 -0.003346
281144  169966.0 -3.113832  0.585864 -5.399730  1.817092 -0.840618 -2.943548
281674  170348.0  1.991976  0.158476 -2.583441  0.408670  1.151147 -0.096695

              V7        V8        V9  …       V20       V21       V22  \
202142  0.041102 -0.055834  0.254505  … -0.109866 -0.307434 -0.794361
191015  0.822466  0.324396  1.123297  … -0.232890  0.029282  0.325216
108550  0.272615 -0.016897 -0.521548  …  0.027504 -0.133573 -0.261358
169298  0.958980 -0.069149 -0.448027  …  0.056372 -0.221406 -0.429478
175399  0.301441 -0.429491  0.465441  … -0.198317  0.260473  0.970039
…            …       …       …  …       …       …       …
279863 -0.882850  0.697211 -2.064945  …  1.252967  0.778584 -0.319189
280143 -1.413170  0.248525 -1.127396  …  0.226138  0.370612  0.028234
280149 -2.234739  1.210158 -0.652250  …  0.247968  0.751826  0.834108
281144 -2.208002  1.058733 -1.632333  …  0.306271  0.583276 -0.269209
281674  0.223050 -0.068384  0.577829  … -0.017652 -0.164350 -0.295135
```

```
             V23       V24       V25       V26       V27       V28  Amount
202142  0.223205 -0.335629 -0.146090  0.209132 -0.077160 -0.072713    1.98
191015 -0.423414 -1.040133  0.782130 -0.189057  0.276484  0.082480    7.59
108550 -0.245430 -1.393994  0.692878  0.461040 -0.014796  0.011991    0.76
169298 -0.010304 -0.527207 -0.430356  0.152144  0.253524  0.084307    2.69
175399 -0.003167  0.047588  0.282419 -0.108327 -0.010702 -0.057457    1.00
…          …       …       …       …       …       …     …
279863  0.639419 -0.294885  0.537503  0.788395  0.292680  0.147968  390.00
280143 -0.145640 -0.081049  0.521875  0.739467  0.389152  0.186637    0.76
280149  0.190944  0.032070 -0.739695  0.471111  0.385107  0.194361   77.89
281144 -0.456108 -0.183659 -0.328168  0.606116  0.884876 -0.253700  245.00
281674 -0.072173 -0.450261  0.313267 -0.289617  0.002988 -0.015309   42.53

[984 rows x 30 columns]
```

```
[53]: print(Y)
```

```
269531    0
157452    0
17795     0
134892    0
59339     0
         ..
279863    1
280143    1
280149    1
281144    1
281674    1
Name: Class, Length: 984, dtype: int64
```

```
[50]: X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2,␣
        ↪stratify = Y, random_state=2)
```

```
[52]: print(X.shape, X_train.shape,X_test.shape )
```

```
(984, 30) (787, 30) (197, 30)
```

```
[58]: model = LogisticRegression()
```

```
[60]: model.fit(X_train, Y_train)
```

```
[60]: LogisticRegression()
```

```
[62]: X_train_prediction = model.predict(X_train)
```

```
[66]: training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

[68]:
```
print('Accuracy Score of Training Data:',training_data_accuracy )
```

```
Accuracy Score of Training Data: 0.9453621346886912
```

[72]:
```
X_test_prediction = model.predict(X_test)
```

[74]:
```
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

[76]:
```
print('Accuracy Score of Traning Data:', test_data_accuracy)
```

```
Accuracy Score of Traning Data: 0.9238578680203046
```

[ ]:
```

```