credit-card-fraud-detection

July 19, 2024

[1]:
```python
# Import Library's.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[4]:
```python
# Import Data Set.
credit_card_data = pd.read_csv("C:\\Users\\Shreyas\\OneDrive\\Desktop\\Machine␣
 ↪Learning Project\\creditcard.csv")
credit_card_data
```

[4]:
```
            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]
```

[5]:

```python
# First and last 5 rows of dataset
credit_card_data.head()
```

[5]:

```
   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]
```

[6]:
```python
credit_card_data.tail()
```

[6]:
```
            Time         V1         V2        V3        V4        V5  \
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  \
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  \
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
284802      0
284803      0
284804      0
284805      0
284806      0

[5 rows x 31 columns]
```

[7]:
```python
# Data set Information.
credit_card_data.isnull().sum()
```

[7]:
```
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
```

[8]:
```
credit_card_data.describe()
```

[8]:
```
                Time            V1            V2            V3            V4  \
count  284807.000000  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean    94813.859575  1.168375e-15  3.416908e-16 -1.379537e-15  2.074095e-15
std     47488.145955  1.958696e+00  1.651309e+00  1.516255e+00  1.415869e+00
min         0.000000 -5.640751e+01 -7.271573e+01 -4.832559e+01 -5.683171e+00
25%     54201.500000 -9.203734e-01 -5.985499e-01 -8.903648e-01 -8.486401e-01
50%     84692.000000  1.810880e-02  6.548556e-02  1.798463e-01 -1.984653e-02
75%    139320.500000  1.315642e+00  8.037239e-01  1.027196e+00  7.433413e-01
max    172792.000000  2.454930e+00  2.205773e+01  9.382558e+00  1.687534e+01

                 V5            V6            V7            V8            V9  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   9.604066e-16  1.487313e-15 -5.556467e-16  1.213481e-16 -2.406331e-15
std    1.380247e+00  1.332271e+00  1.237094e+00  1.194353e+00  1.098632e+00
min   -1.137433e+02 -2.616051e+01 -4.355724e+01 -7.321672e+01 -1.343407e+01
25%   -6.915971e-01 -7.682956e-01 -5.540759e-01 -2.086297e-01 -6.430976e-01
50%   -5.433583e-02 -2.741871e-01  4.010308e-02  2.235804e-02 -5.142873e-02
75%    6.119264e-01  3.985649e-01  5.704361e-01  3.273459e-01  5.971390e-01
```

```
max    3.480167e+01  7.330163e+01  1.205895e+02  2.000721e+01  1.559499e+01

       …          V21           V22           V23           V24  \
count  …  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   …  1.654067e-16 -3.568593e-16  2.578648e-16  4.473266e-15
std    …  7.345240e-01  7.257016e-01  6.244603e-01  6.056471e-01
min    … -3.483038e+01 -1.093314e+01 -4.480774e+01 -2.836627e+00
25%    … -2.283949e-01 -5.423504e-01 -1.618463e-01 -3.545861e-01
50%    … -2.945017e-02  6.781943e-03 -1.119293e-02  4.097606e-02
75%    …  1.863772e-01  5.285536e-01  1.476421e-01  4.395266e-01
max    …  2.720284e+01  1.050309e+01  2.252841e+01  4.584549e+00

                V25           V26           V27           V28         Amount  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  284807.000000
mean   5.340915e-16  1.683437e-15 -3.660091e-16 -1.227390e-16      88.349619
std    5.212781e-01  4.822270e-01  4.036325e-01  3.300833e-01     250.120109
min   -1.029540e+01 -2.604551e+00 -2.256568e+01 -1.543008e+01       0.000000
25%   -3.171451e-01 -3.269839e-01 -7.083953e-02 -5.295979e-02       5.600000
50%    1.659350e-02 -5.213911e-02  1.342146e-03  1.124383e-02      22.000000
75%    3.507156e-01  2.409522e-01  9.104512e-02  7.827995e-02      77.165000
max    7.519589e+00  3.517346e+00  3.161220e+01  3.384781e+01   25691.160000

               Class
count  284807.000000
mean        0.001727
std         0.041527
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000

[8 rows x 31 columns]
```

[9]:

```
credit_card_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
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
```

[10]:
```
# distribution of legit transation & fraudulent transation
credit_card_data['Class'].value_counts()
```

[10]:
```
Class
0    284315
1       492
Name: count, dtype: int64
```

[11]:
```
# sepration dataset for analysis.
legit = credit_card_data[credit_card_data.Class ==0]
fraud = credit_card_data[credit_card_data.Class ==1]
```

[12]:
```
print(legit.shape)
print(fraud.shape)
```

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

[13]:
```
# StASTICAL MEASUREMENT OF DATA SET
legit.Amount.describe()
```

[13]:
```
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
```

[14]:
```
fraud.Amount.describe()
```

[14]:
```
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
```

[15]:
```
# COMPARE THE VALUE FOR BOTH TRANSATION.
credit_card_data.groupby('Class').mean()
```

[15]:
```
               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]
```

[16]:
```
legit_sample = legit.sample(n=492)
```

[17]:
```
new_dataset = pd.concat([legit_sample, fraud],axis=0)
```

[18]:
```
new_dataset.head()
```

[18]:
```
            Time        V1        V2        V3        V4        V5        V6  \
37629    39043.0  1.066100 -0.161069  1.118310  0.620694 -0.931990 -0.346915
43389    41489.0  0.698122 -1.094558  0.412464  0.482848 -1.237461 -0.669810
13580    24078.0  1.250127  0.060157 -0.752441  0.285687  2.030749  3.609633
265846  162070.0 -0.486404 -0.138413  2.306777 -2.515815 -0.282440  0.231716
281721  170378.0 -0.126513  1.176895 -0.326781 -0.492242  0.659229 -1.047616

              V7        V8        V9  …       V21       V22       V23  \
37629  -0.450947  0.044601  0.381159  …  0.079557  0.203153  0.084861
43389  -0.006946 -0.126056  0.721621  …  0.093262 -0.279385 -0.238308
13580  -0.788806  0.801707  1.332631  … -0.027853  0.052992 -0.159477
265846 -0.007968 -0.438984  0.440198  …  0.278358  1.510426 -0.267723
281721  0.922631 -0.082445  0.167155  … -0.341945 -0.780615  0.167378

             V24       V25       V26       V27       V28  Amount  Class
37629   0.454484  0.037353  0.394027  0.005238  0.035698   52.15      0
43389   0.457393  0.134987  0.986115 -0.116470  0.056533  292.00      0
13580   0.956027  0.753319 -0.246269  0.008765  0.016394   16.35      0
265846  0.715280 -0.358132 -0.324237 -0.080343 -0.364515    8.99      0
281721  0.970323 -0.458010  0.095416  0.332530  0.147018    2.69      0

[5 rows x 31 columns]
```

[20]:
```
new_dataset['Class'].value_counts()
```

[20]:
```
Class
0    492
1    492
Name: count, dtype: int64
```

[21]:
```
new_dataset.groupby('Class').mean()
```

[21]:
```
               Time        V1        V2        V3        V4        V5  \
Class
0      97136.491870 -0.123785  0.130570  0.017336 -0.010601 -0.065660
1      80746.806911 -4.771948  3.623778 -7.033281  4.542029 -3.151225

             V6        V7        V8        V9  …       V20       V21  \
```

```
Class                                             …
0      -0.030280  0.012729  0.015314  0.104237  …  0.020087 -0.035464
1      -1.397737 -5.568731  0.570636 -2.581123  …  0.372319  0.713588

            V22       V23       V24       V25       V26       V27       V28  \
Class
0      -0.000281  0.005424  0.015587  0.002499 -0.021452 -0.002072 -0.001191
1       0.014049 -0.040308 -0.105130  0.041449  0.051648  0.170575  0.075667

           Amount
Class
0       84.932297
1      122.211321

[2 rows x 30 columns]
```

[28]:
```python
# SEPRATING THE DATASET INTO FEATURES & TARGETS.
x = new_dataset.iloc[:
 ↪,[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29]]
```

[29]:
```python
x
```

[29]:
```
            Time        V1        V2        V3        V4        V5        V6  \
37629    39043.0  1.066100 -0.161069  1.118310  0.620694 -0.931990 -0.346915
43389    41489.0  0.698122 -1.094558  0.412464  0.482848 -1.237461 -0.669810
13580    24078.0  1.250127  0.060157 -0.752441  0.285687  2.030749  3.609633
265846  162070.0 -0.486404 -0.138413  2.306777 -2.515815 -0.282440  0.231716
281721  170378.0 -0.126513  1.176895 -0.326781 -0.492242  0.659229 -1.047616
…            …         …         …         …         …         …         …
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  \
37629  -0.450947  0.044601  0.381159  …  0.009667  0.079557  0.203153
43389  -0.006946 -0.126056  0.721621  …  0.459849  0.093262 -0.279385
13580  -0.788806  0.801707  1.332631  … -0.026359 -0.027853  0.052992
265846 -0.007968 -0.438984  0.440198  …  0.540062  0.278358  1.510426
281721  0.922631 -0.082445  0.167155  …  0.097622 -0.341945 -0.780615
…            …         …         …  …       …         …         …
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
37629   0.084861  0.454484  0.037353  0.394027  0.005238  0.035698   52.15
43389  -0.238308  0.457393  0.134987  0.986115 -0.116470  0.056533  292.00
13580  -0.159477  0.956027  0.753319 -0.246269  0.008765  0.016394   16.35
265846 -0.267723  0.715280 -0.358132 -0.324237 -0.080343 -0.364515    8.99
281721  0.167378  0.970323 -0.458010  0.095416  0.332530  0.147018    2.69
…            …         …         …         …         …         …       …
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]
```

[30]:
```python
y = new_dataset.iloc[:,[30]]
```

[31]:
```python
y
```

[31]:
```
        Class
37629       0
43389       0
13580       0
265846      0
281721      0
…         …
279863      1
280143      1
280149      1
281144      1
281674      1

[984 rows x 1 columns]
```

[32]:
```python
# TRAINING AND TESTING THE DATASET
from sklearn.model_selection import train_test_split
```

[34]:
```python
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
 ↪2,stratify=y,random_state=2)
```

[36]:
```python
print(x.shape,x_train.shape,x_test.shape)
```

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

[37]:
```python
# LOGISTIC REGRESSION ALGORITHUM
from sklearn.linear_model import LogisticRegression
```

[40]:
```
model = LogisticRegression()
```

[41]:
```
model.fit(x_train,y_train)
```

```
C:\Users\Shreyas\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[41]:
```
LogisticRegression()
```

[45]:
```
# REDICTICTION
x_train_prediction = model.predict(x_train)
training_data_accuracy = accuracy_score(x_train_prediction, y_train)
```

[46]:
```
print("Accuracy on training data : ", training_data_accuracy)
```

```
Accuracy on training data :  0.9453621346886912
```

[51]:
```
# FIND THE ACCURACY
lr = LogisticRegression()
lr.fit(x_train,y_train)
lr.score(x_train,y_train)*100,lr.score(x_test,y_test)*100
```

```
C:\Users\Shreyas\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[51]:
```
(94.53621346886912, 89.84771573604061)
```

[47]:
```
x_test_prediction = model.predict(x_test)
test_data_accuracy = accuracy_score(x_test_prediction, y_test)
```

[48]:
```
print("Accuracy on test data : ", test_data_accuracy)
```

```
Accuracy on test data :  0.8984771573604061
```

[49]:
```
from sklearn.metrics import mean_absolute_error
```

[53]:
```
lr.predict([[0, -1.359807134, -0.072781173, 2.536346738, 1.378155224, -0.
 ↪33832077, 0.462387778, 0.239598554, 0.098697901, 0.36378697, 0.090794172, -0.
 ↪551599533, -0.617800856, -0.991389847, -0.311169354, 1.468176972, -0.
 ↪470400525, 0.207971242, 0.02579058, 0.40399296, 0.251412098, -0.018306778, 0.
 ↪277837576, -0.11047391, 0.066928075, 0.128539358, -0.189114844, 0.133558377,␣
 ↪-0.021053053, 149.62]])
```

```
C:\Users\Shreyas\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X
does not have valid feature names, but LogisticRegression was fitted with
feature names
  warnings.warn(
```

[53]:
```
array([0], dtype=int64)
```

[ ]:
```
```