1-credit-card-fraudulent-detection

June 24, 2024

[1]:
```python
#Mount drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[2]:
```python
#Import necessary libraries
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```python
#Create datframe
df=pd.read_csv('/content/drive/MyDrive/creditcard.csv')
df
```

[3]:
```
            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]
```

[4]:
```python
#Printing head and tail of dataset to get the overview of the dataset
df.head()
```

[4]:
```
   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]
```

[5]:
```
df.tail()
```

[5]:
```
            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]
```

[6]:
```
#Check for missing values
df.isna().sum()
```

[6]:
```
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
```

[7]:
```
#Check for datatypes
df.dtypes
```

[7]:
```
Time      float64
V1        float64
V2        float64
V3        float64
V4        float64
V5        float64
V6        float64
V7        float64
V8        float64
V9        float64
V10       float64
```

```
V11       float64
V12       float64
V13       float64
V14       float64
V15       float64
V16       float64
V17       float64
V18       float64
V19       float64
V20       float64
V21       float64
V22       float64
V23       float64
V24       float64
V25       float64
V26       float64
V27       float64
V28       float64
Amount    float64
Class       int64
dtype: object
```

[8]:
```
#Display information about each columns
df.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
```

[9]:
```
#Print description of the dataset
df.describe()
```

[9]:
```
                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]
```

[10]:
```
df.drop(['Time'],axis=1,inplace=True)
```

[11]:
```
#Print all 31 columns
df.columns
```

[11]:
```
Index(['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')
```

[12]:
```
#Check the class count for the target column
df['Class'].value_counts()
```

[12]:
```
Class
0    284315
1       492
Name: count, dtype: int64
```

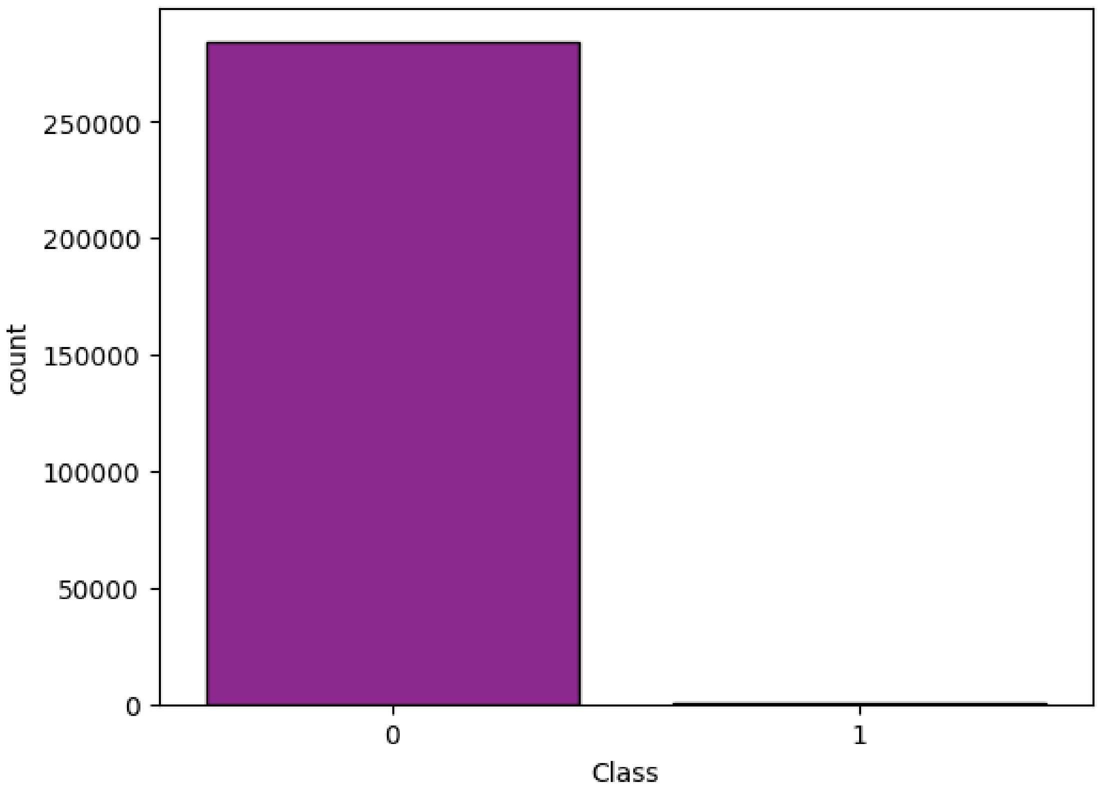
Observation:- Imbalanced set of data

[13]:
```
sns.countplot(x='Class',data=df,palette='plasma',edgecolor='k')
```

```
<ipython-input-13-57a1288133d3>:1: FutureWarning:
```

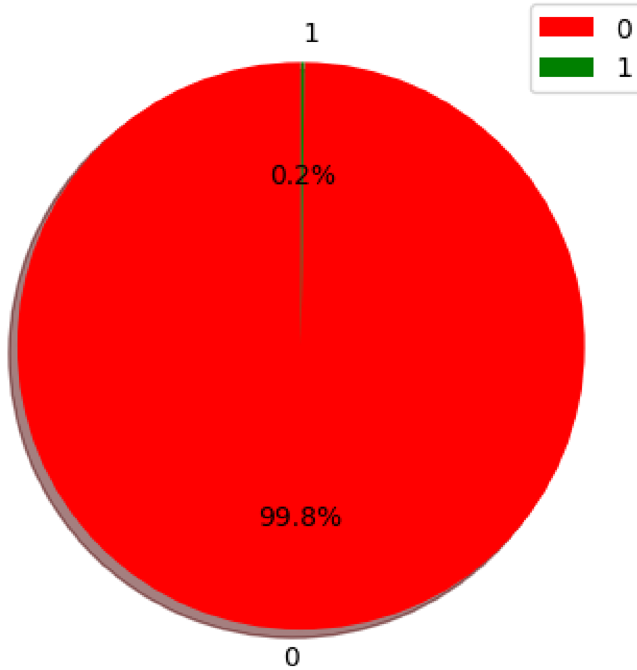
```
Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x='Class',data=df,palette='plasma',edgecolor='k')
```

[13]: <Axes: xlabel='Class', ylabel='count'>



[14]:
```
plt.pie(df['Class'].value_counts(),labels=[0,1],autopct='%1.1f%%',␣
↪startangle=90,
        colors=['red','green'], shadow=True)
plt.legend(loc='upper right')
```

[14]: <matplotlib.legend.Legend at 0x7ef6138c2e90>



[15]:
```
corre=df.corr()
corre
```

[15]:
```
               V1            V2            V3            V4            V5  \
V1   1.000000e+00  4.135835e-16 -1.227819e-15 -9.215150e-16  1.812612e-17
V2   4.135835e-16  1.000000e+00  3.243764e-16 -1.121065e-15  5.157519e-16
V3  -1.227819e-15  3.243764e-16  1.000000e+00  4.711293e-16 -6.539009e-17
V4  -9.215150e-16 -1.121065e-15  4.711293e-16  1.000000e+00 -1.719944e-15
V5   1.812612e-17  5.157519e-16 -6.539009e-17 -1.719944e-15  1.000000e+00
V6  -6.506567e-16  2.787346e-16  1.627627e-15 -7.491959e-16  2.408382e-16
V7  -1.005191e-15  2.055934e-16  4.895305e-16 -4.104503e-16  2.715541e-16
V8  -2.433822e-16 -5.377041e-17 -1.268779e-15  5.697192e-16  7.437229e-16
V9  -1.513678e-16  1.978488e-17  5.568367e-16  6.923247e-16  7.391702e-16
V10  7.388135e-17 -3.991394e-16  1.156587e-15  2.232685e-16 -5.202306e-16
V11  2.125498e-16  1.975426e-16  1.576830e-15  3.459380e-16  7.203963e-16
V12  2.053457e-16 -9.568710e-17  6.310231e-16 -5.625518e-16  7.412552e-16
V13 -2.425603e-17  6.295388e-16  2.807652e-16  1.303306e-16  5.886991e-16
V14 -5.020280e-16 -1.730566e-16  4.739859e-16  2.282280e-16  6.565143e-16
V15  3.547782e-16 -4.995814e-17  9.068793e-16  1.377649e-16 -8.720275e-16
V16  7.212815e-17  1.177316e-17  8.299445e-16 -9.614528e-16  2.246261e-15
V17 -3.879840e-16 -2.685296e-16  7.614712e-16 -2.699612e-16  1.281914e-16
V18  3.230206e-17  3.284605e-16  1.509897e-16 -5.103644e-16  5.308590e-16
V19  1.502024e-16 -7.118719e-18  3.463522e-16 -3.980557e-16 -1.450421e-16
```

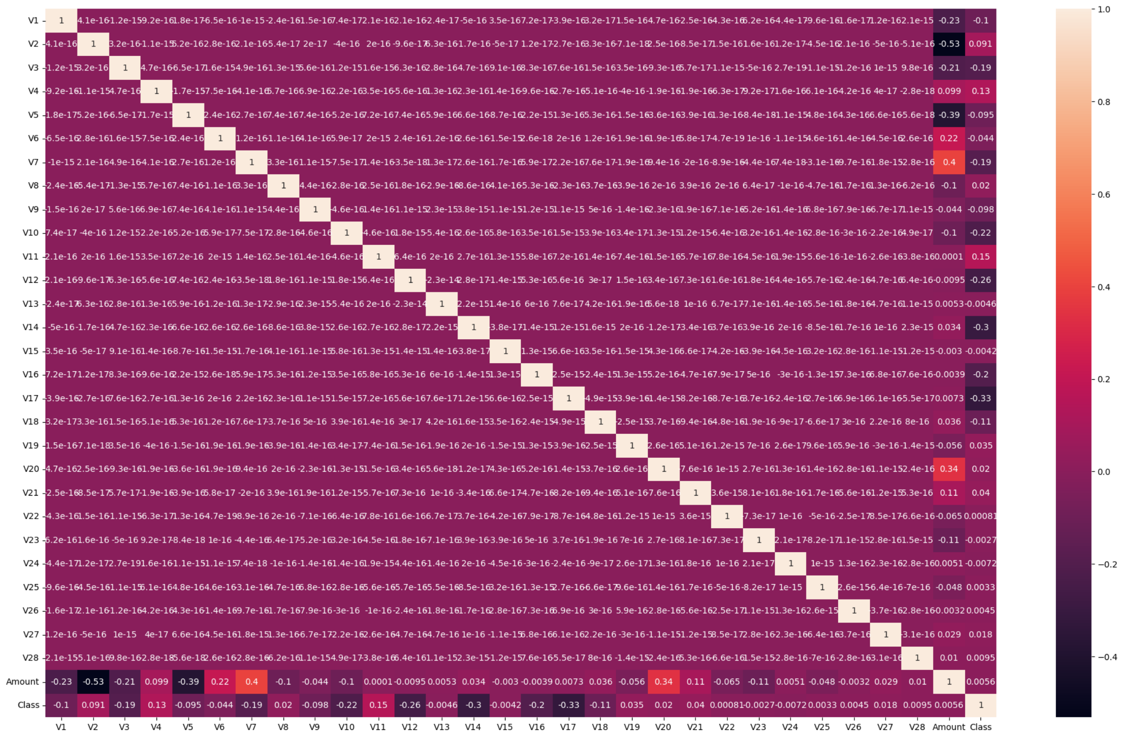
```
V20     4.654551e-16  2.506675e-16 -9.316409e-16 -1.857247e-16 -3.554057e-16
V21    -2.457409e-16 -8.480447e-17  5.706192e-17 -1.949553e-16 -3.920976e-16
V22    -4.290944e-16  1.526333e-16 -1.133902e-15 -6.276051e-17  1.253751e-16
V23     6.168652e-16  1.634231e-16 -4.983035e-16  9.164206e-17 -8.428683e-18
V24    -4.425156e-17  1.247925e-17  2.686834e-19  1.584638e-16 -1.149255e-15
V25    -9.605737e-16 -4.478846e-16 -1.104734e-15  6.070716e-16  4.808532e-16
V26    -1.581290e-17  2.057310e-16 -1.238062e-16 -4.247268e-16  4.319541e-16
V27     1.198124e-16 -4.966953e-16  1.045747e-15  3.977061e-17  6.590482e-16
V28     2.083082e-15 -5.093836e-16  9.775546e-16 -2.761403e-18 -5.613951e-18
Amount -2.277087e-01 -5.314089e-01 -2.108805e-01  9.873167e-02 -3.863563e-01
Class  -1.013473e-01  9.128865e-02 -1.929608e-01  1.334475e-01 -9.497430e-02

                  V6            V7            V8            V9           V10  \
V1     -6.506567e-16 -1.005191e-15 -2.433822e-16 -1.513678e-16  7.388135e-17
V2      2.787346e-16  2.055934e-16 -5.377041e-17  1.978488e-17 -3.991394e-16
V3      1.627627e-15  4.895305e-16 -1.268779e-15  5.568367e-16  1.156587e-15
V4     -7.491959e-16 -4.104503e-16  5.697192e-16  6.923247e-16  2.232685e-16
V5      2.408382e-16  2.715541e-16  7.437229e-16  7.391702e-16 -5.202306e-16
V6      1.000000e+00  1.191668e-16 -1.104219e-16  4.131207e-16  5.932243e-17
V7      1.191668e-16  1.000000e+00  3.344412e-16  1.122501e-15 -7.492834e-17
V8     -1.104219e-16  3.344412e-16  1.000000e+00  4.356078e-16 -2.801370e-16
V9      4.131207e-16  1.122501e-15  4.356078e-16  1.000000e+00 -4.642274e-16
V10     5.932243e-17 -7.492834e-17 -2.801370e-16 -4.642274e-16  1.000000e+00
V11     1.980503e-15  1.425248e-16  2.487043e-16  1.354680e-16 -4.622103e-16
V12     2.375468e-16 -3.536655e-18  1.839891e-16 -1.079314e-15  1.771869e-15
V13    -1.211182e-16  1.266462e-17 -2.921856e-16  2.251072e-15 -5.418460e-16
V14     2.621312e-16  2.607772e-16 -8.599156e-16  3.784757e-15  2.635936e-16
V15    -1.531188e-15 -1.690540e-16  4.127777e-16 -1.051167e-15  5.786332e-16
V16     2.623672e-18  5.869302e-17 -5.254741e-16 -1.214086e-15  3.545450e-16
V17     2.015618e-16  2.177192e-16 -2.269549e-16  1.113695e-15  1.542955e-15
V18     1.223814e-16  7.604126e-17 -3.667974e-16  4.993240e-16  3.902423e-16
V19    -1.865597e-16 -1.881008e-16 -3.875186e-16 -1.376135e-16  3.437633e-17
V20    -1.858755e-16  9.379684e-16  2.033737e-16 -2.343720e-16 -1.331556e-15
V21     5.833316e-17 -2.027779e-16  3.892798e-16  1.936953e-16  1.177547e-15
V22    -4.705235e-19 -8.898922e-16  2.026927e-16 -7.071869e-16 -6.418202e-16
V23     1.046712e-16 -4.387401e-16  6.377260e-17 -5.214137e-16  3.214491e-16
V24    -1.071589e-15  7.434913e-18 -1.047097e-16 -1.430343e-16 -1.355885e-16
V25     4.562861e-16 -3.094082e-16 -4.653279e-16  6.757763e-16 -2.846052e-16
V26    -1.357067e-16 -9.657637e-16 -1.727276e-16 -7.888853e-16 -3.028119e-16
V27    -4.452461e-16 -1.782106e-15  1.299943e-16 -6.709655e-17 -2.197977e-16
V28     2.594754e-16 -2.776530e-16 -6.200930e-16  1.110541e-15  4.864782e-17
Amount  2.159812e-01  3.973113e-01 -1.030791e-01 -4.424560e-02 -1.015021e-01
Class  -4.364316e-02 -1.872566e-01  1.987512e-02 -9.773269e-02 -2.168829e-01

        …           V21           V22           V23           V24  \
V1      … -2.457409e-16 -4.290944e-16  6.168652e-16 -4.425156e-17
V2      … -8.480447e-17  1.526333e-16  1.634231e-16  1.247925e-17
```

```
V3      …  5.706192e-17 -1.133902e-15 -4.983035e-16  2.686834e-19
V4      … -1.949553e-16 -6.276051e-17  9.164206e-17  1.584638e-16
V5      … -3.920976e-16  1.253751e-16 -8.428683e-18 -1.149255e-15
V6      …  5.833316e-17 -4.705235e-19  1.046712e-16 -1.071589e-15
V7      … -2.027779e-16 -8.898922e-16 -4.387401e-16  7.434913e-18
V8      …  3.892798e-16  2.026927e-16  6.377260e-17 -1.047097e-16
V9      …  1.936953e-16 -7.071869e-16 -5.214137e-16 -1.430343e-16
V10     …  1.177547e-15 -6.418202e-16  3.214491e-16 -1.355885e-16
V11     … -5.658364e-16  7.772895e-16 -4.505332e-16  1.933267e-15
V12     …  7.300527e-16  1.644699e-16  1.800885e-16  4.436512e-16
V13     …  1.008461e-16  6.747721e-17 -7.132064e-16 -1.397470e-16
V14     … -3.356561e-16  3.740383e-16  3.883204e-16  2.003482e-16
V15     …  6.605263e-17 -4.208921e-16 -3.912243e-16 -4.478263e-16
V16     … -4.715090e-16 -7.923387e-17  5.020770e-16 -3.005985e-16
V17     … -8.230527e-16 -8.743398e-16  3.706214e-16 -2.403828e-16
V18     … -9.408680e-16 -4.819365e-16 -1.912006e-16 -8.986916e-17
V19     …  5.115885e-16 -1.163768e-15  7.032035e-16  2.587708e-17
V20     … -7.614597e-16  1.009285e-15  2.712885e-16  1.277215e-16
V21     …  1.000000e+00  3.649908e-15  8.119580e-16  1.761054e-16
V22     …  3.649908e-15  1.000000e+00 -7.303916e-17  9.970809e-17
V23     …  8.119580e-16 -7.303916e-17  1.000000e+00  2.130519e-17
V24     …  1.761054e-16  9.970809e-17  2.130519e-17  1.000000e+00
V25     … -1.686082e-16 -5.018575e-16 -8.232727e-17  1.015391e-15
V26     … -5.557329e-16 -2.503187e-17  1.114524e-15  1.343722e-16
V27     … -1.211281e-15  8.461337e-17  2.839721e-16 -2.274142e-16
V28     …  5.278775e-16 -6.627203e-16  1.481903e-15 -2.819805e-16
Amount  …  1.059989e-01 -6.480065e-02 -1.126326e-01  5.146217e-03
Class   …  4.041338e-02  8.053175e-04 -2.685156e-03 -7.220907e-03

                 V25           V26           V27           V28    Amount  \
V1     -9.605737e-16 -1.581290e-17  1.198124e-16  2.083082e-15 -0.227709
V2     -4.478846e-16  2.057310e-16 -4.966953e-16 -5.093836e-16 -0.531409
V3     -1.104734e-15 -1.238062e-16  1.045747e-15  9.775546e-16 -0.210880
V4      6.070716e-16 -4.247268e-16  3.977061e-17 -2.761403e-18  0.098732
V5      4.808532e-16  4.319541e-16  6.590482e-16 -5.613951e-18 -0.386356
V6      4.562861e-16 -1.357067e-16 -4.452461e-16  2.594754e-16  0.215981
V7     -3.094082e-16 -9.657637e-16 -1.782106e-15 -2.776530e-16  0.397311
V8     -4.653279e-16 -1.727276e-16  1.299943e-16 -6.200930e-16 -0.103079
V9      6.757763e-16 -7.888853e-16 -6.709655e-17  1.110541e-15 -0.044246
V10    -2.846052e-16 -3.028119e-16 -2.197977e-16  4.864782e-17 -0.101502
V11    -5.600475e-16 -1.003221e-16 -2.640281e-16 -3.792314e-16  0.000104
V12    -5.712973e-16 -2.359969e-16 -4.672391e-16  6.415167e-16 -0.009542
V13    -5.497612e-16 -1.769255e-16 -4.720898e-16  1.144372e-15  0.005293
V14    -8.547932e-16 -1.660327e-16  1.044274e-16  2.289427e-15  0.033751
V15     3.206423e-16  2.817791e-16 -1.143519e-15 -1.194130e-15 -0.002986
V16    -1.345418e-15 -7.290010e-16  6.789513e-16  7.588849e-16 -0.003910
V17     2.666806e-16  6.932833e-16  6.148525e-16 -5.534540e-17  0.007309
```

```
V18    -6.629212e-17  2.990167e-16  2.242791e-16  7.976796e-16  0.035650
V19     9.577163e-16  5.898033e-16 -2.959370e-16 -1.405379e-15 -0.056151
V20     1.410054e-16 -2.803504e-16 -1.138829e-15 -2.436795e-16  0.339403
V21    -1.686082e-16 -5.557329e-16 -1.211281e-15  5.278775e-16  0.105999
V22    -5.018575e-16 -2.503187e-17  8.461337e-17 -6.627203e-16 -0.064801
V23    -8.232727e-17  1.114524e-15  2.839721e-16  1.481903e-15 -0.112633
V24     1.015391e-15  1.343722e-16 -2.274142e-16 -2.819805e-16  0.005146
V25     1.000000e+00  2.646517e-15 -6.406679e-16 -7.008939e-16 -0.047837
V26     2.646517e-15  1.000000e+00 -3.667715e-16 -2.782204e-16 -0.003208
V27    -6.406679e-16 -3.667715e-16  1.000000e+00 -3.061287e-16  0.028825
V28    -7.008939e-16 -2.782204e-16 -3.061287e-16  1.000000e+00  0.010258
Amount -4.783686e-02 -3.208037e-03  2.882546e-02  1.025822e-02  1.000000
Class   3.307706e-03  4.455398e-03  1.757973e-02  9.536041e-03  0.005632

           Class
V1     -0.101347
V2      0.091289
V3     -0.192961
V4      0.133447
V5     -0.094974
V6     -0.043643
V7     -0.187257
V8      0.019875
V9     -0.097733
V10    -0.216883
V11     0.154876
V12    -0.260593
V13    -0.004570
V14    -0.302544
V15    -0.004223
V16    -0.196539
V17    -0.326481
V18    -0.111485
V19     0.034783
V20     0.020090
V21     0.040413
V22     0.000805
V23    -0.002685
V24    -0.007221
V25     0.003308
V26     0.004455
V27     0.017580
V28     0.009536
Amount  0.005632
Class   1.000000

[30 rows x 30 columns]
```

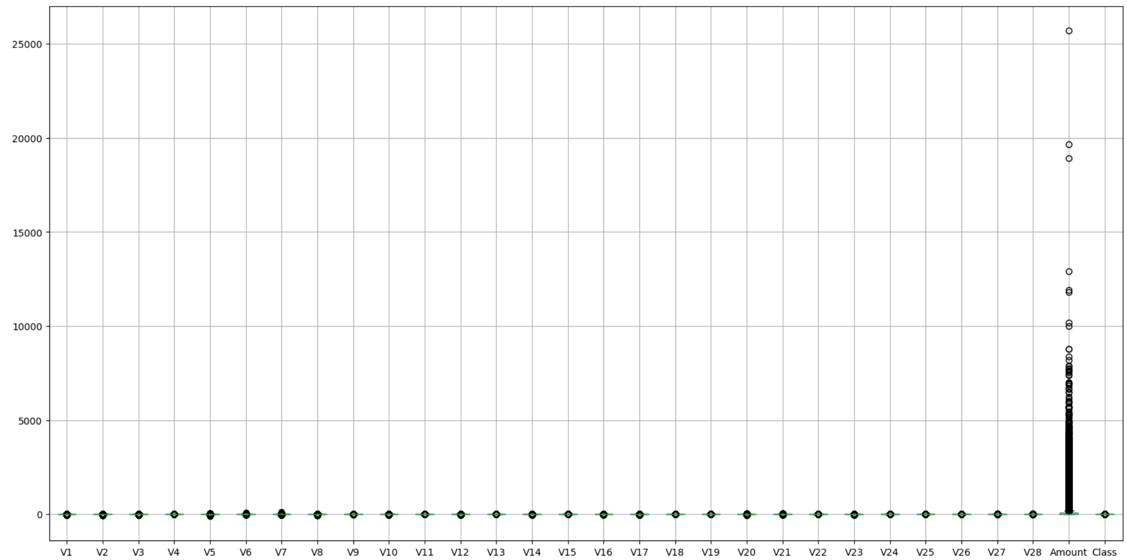
[16]:
```python
plt.figure(figsize=(25,15))
sns.heatmap(corre,annot=True)
```

[16]: <Axes: >



[17]:
```python
plt.figure(figsize=(20,10))
df.boxplot()
```

[17]: <Axes: >



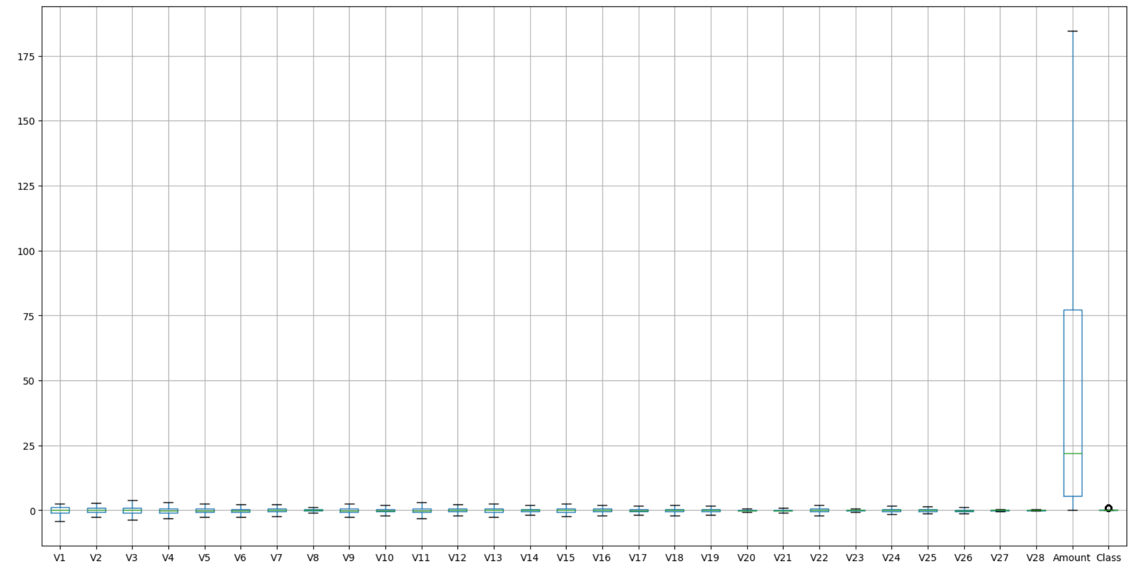
[18]:
```
columns=df.columns[df.columns != 'Class']
columns
```

[18]:
```
Index(['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'],
      dtype='object')
```

[19]:
```
#method2 to remove outliers
def iqr_rem(dfe,cols):
  for col in cols:
    q1=dfe[col].quantile(0.25)
    q3=dfe[col].quantile(0.75)
    IQR=q3-q1
    upper_bound=q3+(1.5*IQR)
    lower_bound=q1-(1.5*IQR)
    dfe[col]=dfe[col].clip(lower_bound, upper_bound)

iqr_rem(df,columns)
```

[20]:
```
plt.figure(figsize=(20,10))
df.boxplot()
```

[20]: <Axes: >



[21]:
```
x=df.drop(['Class'],axis=1).values
x
```

[21]:
```
array([[-1.35980713e+00, -7.27811733e-02,  2.53634674e+00, …,
         1.33558377e-01, -2.10530535e-02,  1.49620000e+02],
       [ 1.19185711e+00,  2.66150712e-01,  1.66480113e-01, …,
        -8.98309914e-03,  1.47241692e-02,  2.69000000e+00],
       [-1.35835406e+00, -1.34016307e+00,  1.77320934e+00, …,
        -5.53527940e-02, -5.97518406e-02,  3.78660000e+02],
       …,
       [ 1.91956501e+00, -3.01253846e-01, -3.24963981e+00, …,
         4.45477214e-03, -2.65608286e-02,  6.78800000e+01],
       [-2.40440050e-01,  5.30482513e-01,  7.02510230e-01, …,
         1.08820735e-01,  1.04532821e-01,  1.00000000e+01],
       [-5.33412522e-01, -1.89733337e-01,  7.03337367e-01, …,
        -2.41530880e-03,  1.36489143e-02,  1.84512500e+02]])
```

[22]:
```
y=df['Class'].values
y
```

[22]:
```
array([0, 0, 0, …, 0, 0, 0])
```

[23]:
```
from imblearn.over_sampling import SMOTE
sam=SMOTE(random_state=42)
x_sm,y_sm=sam.fit_resample(x,y)
y_sm
```

[23]:
```
array([0, 0, 0, …, 1, 1, 1])
```

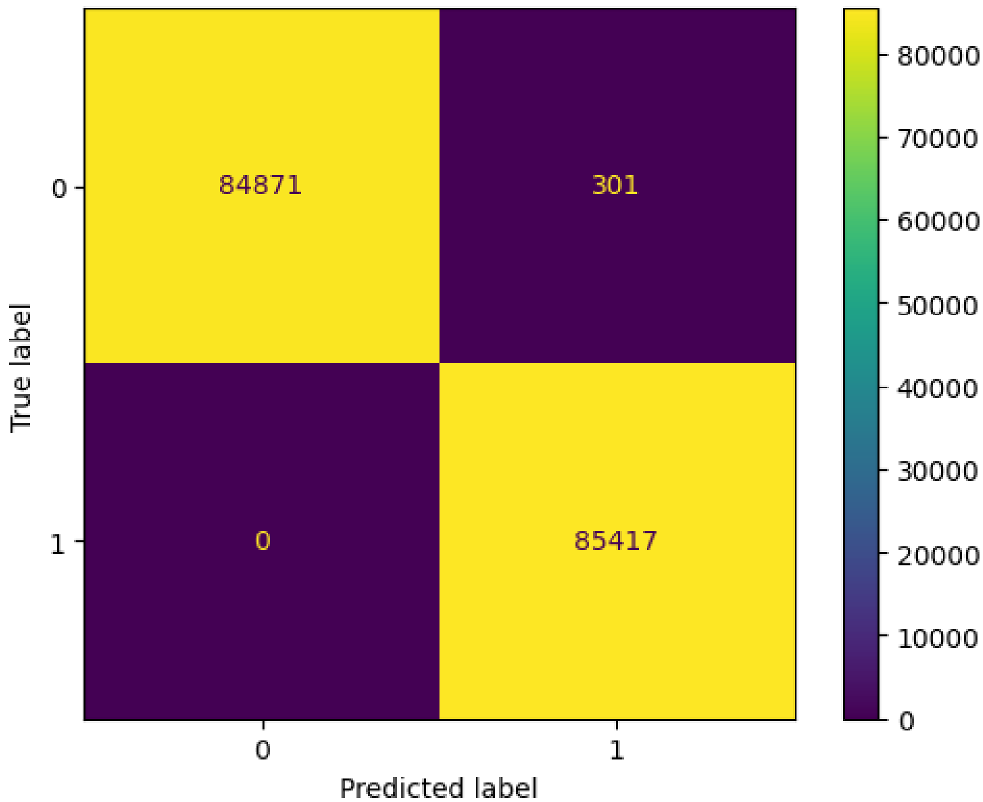
[24]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_sm,y_sm,test_size=0.
↪30,random_state=0)
```

[25]:
```
from sklearn.preprocessing import StandardScaler
norm=StandardScaler()
norm.fit(x_train)
x_train=norm.transform(x_train)
x_test=norm.transform(x_test)
```

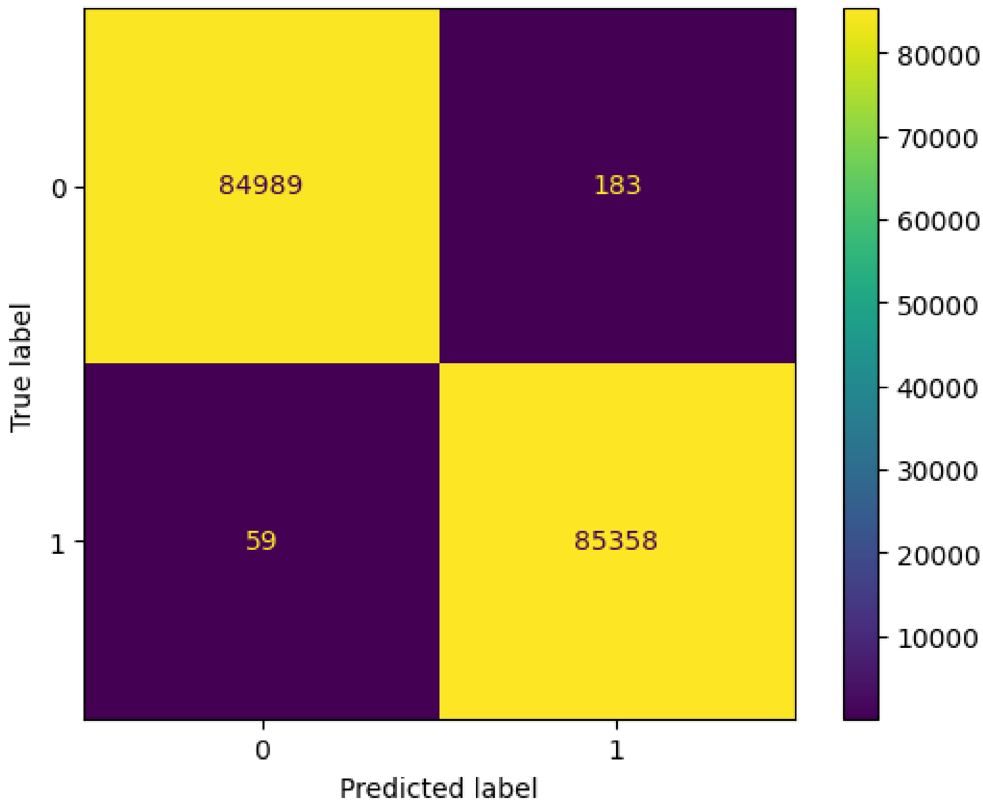
[26]:
```
#Model creation
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import␣
↪accuracy_score,confusion_matrix,classification_report,ConfusionMatrixDisplay
model1=KNeighborsClassifier(n_neighbors=9)
model2=DecisionTreeClassifier(criterion='entropy')
model3=RandomForestClassifier(n_estimators=100,criterion='entropy',random_state=42)
lst=[model1,model2,model3]
```

[27]:
```
for i in lst:
  print("model is",i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  cm=confusion_matrix(y_test,y_pred)
  print("Accuracy score is",accuracy_score(y_test,y_pred))
  print(cm)
  labels=['0','1']
  print(classification_report(y_test,y_pred))
  cmd=ConfusionMatrixDisplay(cm,display_labels=labels)
  cmd.plot()
  plt.show()
```

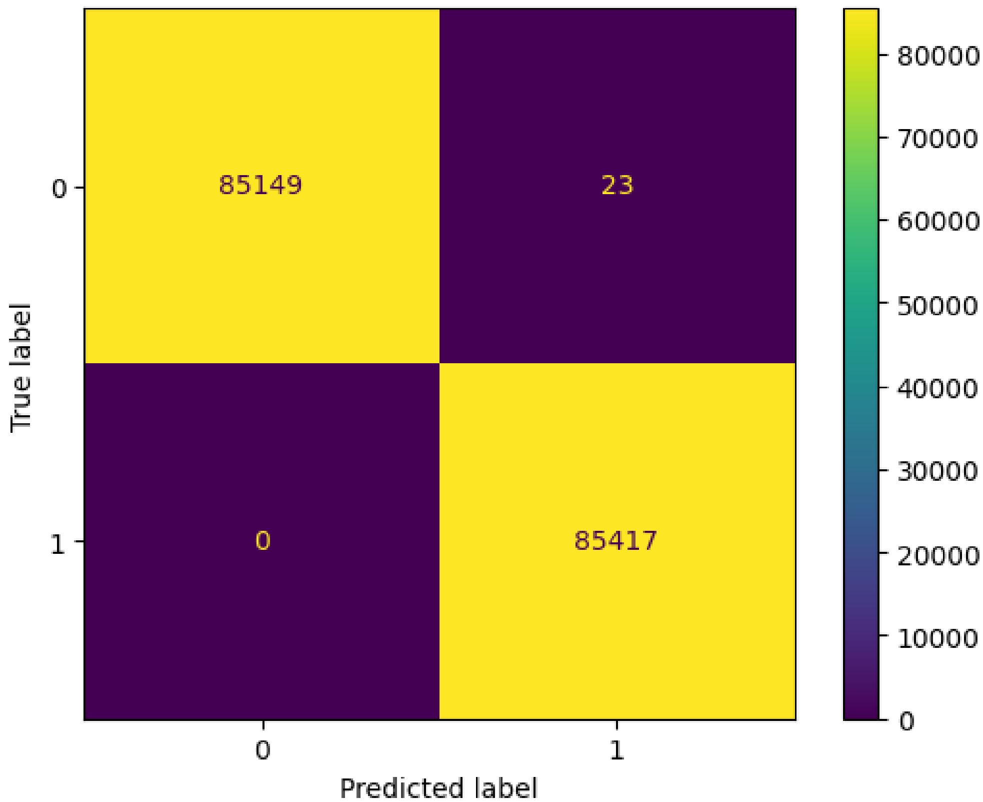
```
model is KNeighborsClassifier(n_neighbors=9)
Accuracy score is 0.9982355251510941
[[84871   301]
 [    0 85417]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85172
           1       1.00      1.00      1.00     85417

    accuracy                           1.00    170589
   macro avg       1.00      1.00      1.00    170589
weighted avg       1.00      1.00      1.00    170589
```




```
model is DecisionTreeClassifier(criterion='entropy')
Accuracy score is 0.9985813856696505
[[84989   183]
 [   59 85358]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85172
           1       1.00      1.00      1.00     85417

    accuracy                           1.00    170589
   macro avg       1.00      1.00      1.00    170589
weighted avg       1.00      1.00      1.00    170589
```




```
model is RandomForestClassifier(criterion='entropy', random_state=42)
Accuracy score is 0.9998651730181899
[[85149    23]
 [    0 85417]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85172
           1       1.00      1.00      1.00     85417

    accuracy                           1.00    170589
   macro avg       1.00      1.00      1.00    170589
weighted avg       1.00      1.00      1.00    170589
```


85149
23
0
85417
True label
Predicted label
0
1
0
1
80000
70000
60000
50000
40000
30000
20000
10000
0