```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score,recall_score,ConfusionMatrixDisplay,
data=pd.read_csv('creditcard.csv')
```

data.head()

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	• • •	V21	V22	V2:
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
1	0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425		-0.225775	-0.638672	0.10128
2	1	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654		0.247998	0.771679	0.909412
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024		-0.108300	0.005274	-0.19032 ⁻
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739		-0.009431	0.798278	-0.137458
5 ro	ws × :	31 columns												

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9965 entries, 0 to 9964
Data columns (total 31 columns):
    Column Non-Null Count Dtype
---
    -----
            -----
            9965 non-null int64
9965 non-null float64
0
    Time
1
    V1
2
            9965 non-null float64
3
            9964 non-null
                            float64
    V3
            9964 non-null float64
    ٧4
            9964 non-null
                            float64
5
    V5
            9964 non-null
                           float64
6
    V6
7
            9964 non-null
    V7
                           float64
8
            9964 non-null
                            float64
    V8
9
    V9
            9964 non-null
                           float64
10 V10
            9964 non-null
                            float64
    V11
            9964 non-null
                            float64
12 V12
            9964 non-null
                            float64
13 V13
            9964 non-null
                           float64
14 V14
            9964 non-null
                            float64
15 V15
            9964 non-null
                            float64
            9964 non-null
                            float64
16 V16
            9964 non-null
17
    V17
                            float64
            9964 non-null
18 V18
                            float64
19 V19
            9964 non-null
                           float64
20 V20
            9964 non-null
                            float64
21 V21
            9964 non-null
                            float64
22 V22
            9964 non-null
                            float64
23 V23
            9964 non-null
                            float64
 24 V24
            9964 non-null
                            float64
            9964 non-null
25 V25
                            float64
            9964 non-null
26 V26
                            float64
27
    V27
            9964 non-null
                            float64
            9964 non-null
                            float64
28 V28
29 Amount 9964 non-null
                            float64
30 Class
            9964 non-null
                            float64
dtypes: float64(30), int64(1)
memory usage: 2.4 MB
```

data.isnull()

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	•••	V21	V22
0	False	False	False	False	False	False	False	False	False	False		False	False
1	False	False	False	False	False	False	False	False	False	False		False	False
2	False	False	False	False	False	False	False	False	False	False		False	False
3	False	False	False	False	False	False	False	False	False	False		False	False
4	False	False	False	False	False	False	False	False	False	False		False	False
9960	False	False	False	False	False	False	False	False	False	False		False	False
9961	False	False	False	False	False	False	False	False	False	False		False	False
9962	False	False	False	False	False	False	False	False	False	False		False	False
9963	False	False	False	False	False	False	False	False	False	False		False	False
9964	False	False	False	True		True	True						
0005 70	v 94	اماليما	20										•

data.tail()

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9		V21	V22	
9960	14837	1.286884	-0.124610	0.148283	-0.259343	0.248357	0.896718	-0.626627	0.227693	1.618678		-0.381864	-0.904515	-0.02
9961	14854	1.318742	0.496408	0.114876	0.695262	0.170133	-0.537180	0.025492	-0.272931	1.267298		-0.484943	-1.111176	0.02
9962	14857	1.241757	0.419587	0.806183	0.894811	-0.507886	-1.118126	0.018908	-0.343335	1.210781		-0.379396	-0.817785	0.18
9963	14861	1.304800	-0.052885	0.415235	-0.081725	-0.223525	0.097752	-0.561240	0.067228	1.617203		-0.379597	-0.929204	0.02
9964	14864	-1.747939	3.712444	NaN	NaN	NaN	NaN	NaN	NaN	NaN		NaN	NaN	
5 rows × 31 columns														

Handling Null Values

credit_card_data=data.fillna(value=0)
credit_card_data

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	 V21	V22	
0	0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	 -0.018307	0.277838	-0.1
1	0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	 -0.225775	-0.638672	0.10
2	1	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	 0.247998	0.771679	0.90
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	 -0.108300	0.005274	-0.19
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	 -0.009431	0.798278	-0.10
9960	14837	1.286884	-0.124610	0.148283	-0.259343	0.248357	0.896718	-0.626627	0.227693	1.618678	 -0.381864	-0.904515	-0.02
9961	14854	1.318742	0.496408	0.114876	0.695262	0.170133	-0.537180	0.025492	-0.272931	1.267298	 -0.484943	-1.111176	0.02
9962	14857	1.241757	0.419587	0.806183	0.894811	-0.507886	-1.118126	0.018908	-0.343335	1.210781	 -0.379396	-0.817785	0.18
9963	14861	1.304800	-0.052885	0.415235	-0.081725	-0.223525	0.097752	-0.561240	0.067228	1.617203	 -0.379597	-0.929204	0.02
9964	14864	-1.747939	3.712444	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	 0.000000	0.000000	0.00
9965 rows × 31 columns													

credit_card_data.isnull().sum()

Time	0
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
V8	0
V9	0

```
3/27/24, 6:31 AM
```

```
V10
V11
          0
V12
          0
V13
V14
V15
V16
V17
V18
V19
V20
          0
V21
          0
V22
V23
V24
V25
          0
V26
V27
V28
Amount
          0
Class
dtype: int64
```

credit_card_data.duplicated().sum()
credit_card_data.drop_duplicates()

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	•••	V21	V22	
0	0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787		-0.018307	0.277838	-0.1
1	0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425		-0.225775	-0.638672	0.10
2	1	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654		0.247998	0.771679	0.90
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024		-0.108300	0.005274	-0.19
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739		-0.009431	0.798278	-0.10
9960	14837	1.286884	-0.124610	0.148283	-0.259343	0.248357	0.896718	-0.626627	0.227693	1.618678		-0.381864	-0.904515	-0.02
9961	14854	1.318742	0.496408	0.114876	0.695262	0.170133	-0.537180	0.025492	-0.272931	1.267298		-0.484943	-1.111176	0.02
9962	14857	1.241757	0.419587	0.806183	0.894811	-0.507886	-1.118126	0.018908	-0.343335	1.210781		-0.379396	-0.817785	0.18
9963	14861	1.304800	-0.052885	0.415235	-0.081725	-0.223525	0.097752	-0.561240	0.067228	1.617203		-0.379597	-0.929204	0.0
9964	14864	-1.747939	3.712444	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		0.000000	0.000000	0.00
9923 ro	ws × 31	columns												

#checking the distribution of data among the legitimate and fraud transactions
credit_card_data['Class'].value_counts()

```
0.0 9927
1.0 38
```

Name: Class, dtype: int64

#THIS IS HIGHLY UNBALANCED DATA #0-NORMAL TRANSACTION #1-FRAUD TRANSACTION

Exploratory Data Analysis

```
25% 5.155000
50% 15.950000
75% 51.045000
max 7712.430000
Name: Amount, dtype: float64
```

fraud.Amount.describe()

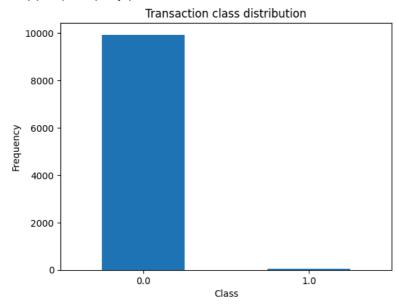
```
count
           38.000000
mean
           75.730526
std
          304.521215
            0.000000
min
25%
            1.000000
50%
            1.000000
75%
            1.000000
         1809.680000
max
Name: Amount, dtype: float64
```

credit_card_data.groupby('Class').mean()

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	• • •	V20	V21	
Class														
0.0	5922.508512	-0.235729	0.267180	0.934294	0.239711	-0.041739	0.140911	-0.057655	-0.069964	0.812921		0.0261	-0.055133	-
1.0	9063.157895	-1.796662	3.810809	-6.415255	5.618146	-1.247563	-2.111328	-3.777261	1.150469	-2.276505		0.4677	0.741934	-
2 rows ×	30 columns													

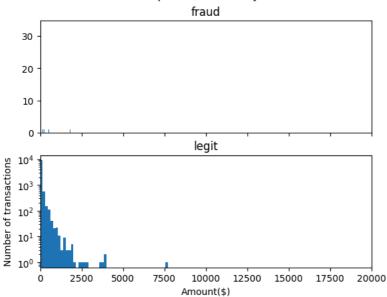
```
count_classes=pd.value_counts(credit_card_data['Class'],sort=True)
count_classes.plot(kind='bar',rot=0)
plt.title("Transaction class distribution")
plt.xlabel('Class')
plt.ylabel('Frequency')
```

Text(0, 0.5, 'Frequency')



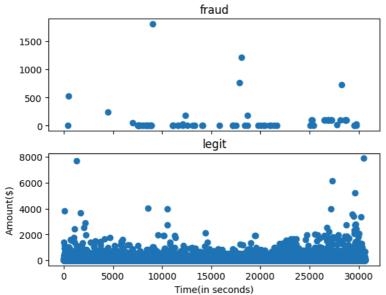
```
f,(ax1,ax2)=plt.subplots(2,1,sharex=True)
f.suptitle('Amount per transaction by class')
bins=50
ax1.hist(fraud.Amount,bins=bins)
ax1.set_title('fraud')
ax2.hist(legit.Amount,bins=bins)
ax2.set_title('legit')
plt.xlabel('Amount($)')
plt.ylabel('Number of transactions')
plt.xlim(0,20000)
plt.yscale('log')
plt.show()
```

Amount per transaction by class



```
#we will be checking the time frame from the data
f,(ax1,ax2)=plt.subplots(2,1,sharex=True)
f.suptitle('Time of tansaction vs Amount per transaction by class')
ax1.scatter(fraud.Time,fraud.Amount)
ax1.set_title('fraud')
ax2.scatter(legit.Time,legit.Amount)
ax2.scatter(legit')
plt.ylabel('Amount($)')
plt.xlabel('Time(in seconds)')
```

Time of tansaction vs Amount per transaction by class



```
#Correlation
import seaborn as sns
cor=credit_card_data.corr()
top_corr_features=cor.index
plt.figure(figsize=(20,20))
sns.heatmap(credit_card_data[top_corr_features].corr(),annot=True)
```

```
<Axes: >
                0.29 0.39 0.13 0.17 0.14 0.31 -0.17 0.0230.0640.087 0.1 0.026 0.23 0.05 0.2 0.19 0.0540.00410.16 -0.13 0.045-0.0620.00140.15 0.02 -0.14 0.038-0.19 -0.3
                         -0.39 <mark>0.16 -</mark>0.25-0.046-0.13 0.11-0.0980.057 0.13 -0.14 0.031 0.19 0.061 -0.12 -0.15-0.0370.0110.0580.062 -0.14-0.0230.0230.0810.067 0.16 -0.058 0.46 0.3
                             -0.21 0.38 0.065 0.5 -0.37 0.22 0.29 -0.21 0.2 0.0088 0.35 -0.16 0.14 0.29 0.1 -0.048 -0.14 -0.042 0.25 0.047 0.036 -0.18 0.063 -0.2 -0.015 -0.12 -0.5
                                                                                                                                                                               0.8
                                   9 -0.082 0.17 -0.25 0.38 -0.14 1
                                       0.16 0.23 -0.21 0.049 0.23 -0.12 0.097 0.039 0.13 0.085 0.2 0.15 0.15 -0.0130.0370.0790.0590.0350.0036-0.07-0.049-0.12-0.077-0.31 -0.28
                                            0.1 -0.0980.057 0.1 -0.12 0.011 0.026 0.12 -0.11 0.052 0.07 0.06 0.093-0.0220.058.00048.00810.0260.0540.00450.0690.024 0.19 -0.13
       S -0.056 0.31 -0.13 0.5 -0.19 0.23 0.1 1
                                                 -0.2 0.079 0.32 -0.21 0.26 -0.03 0.17 0.075 0.21 0.27 0.17 -0.0620.011 -0.13 0.049 0.058 0.009 40.14 -0.041 -0.21 -0.065 0.25 -0.41
                                                                                                                                                                               0.6
           0.06 -0.17 0.11 -0.37 0.12 -0.21-0.098 -0.2 1
                                                     -0.11 -0.21 0.062-0.11-0.016-0.11-0.0140.087-0.15-0.0710.0310.088 -0.2 0.061-0.092 0.01 0.05-0.00990.11 0.059-0.054 0.22
       9 -0.14 0.0230.098 0.22 -0.15 0.0490.057 0.079 0.11 1 0.0420.091 0.25 0.28 0.32 0.22-0.041 0.28 0.15 -0.0670.00880.0320.022-0.0020.00530.11 0.0420.0750.0780.012 0.19
       8 - 0.02 0.0640.057 0.29 -0.097 0.23 0.1 0.32 0.21 0.042 1 0.19 0.28 0.095 0.12 0.066 0.28 0.26 0.0970.00560.0390.00620.04 0.0420.0220.0240.031 -0.14 -0.11 -0.087 0.38
       5 -0.0490.087 0.13 -0.21 0.099 -0.12 -0.12 -0.21 0.062 0.091 -0.19 1 -0.37 0.18 -0.025 -0.12 -0.18 -0.15 -0.0760.0690.036 0.02 0.0360.032 0.015 -0.0810.0460.0660.00460.023 0.3
       G - 0.11 0.1 -0.14 0.2 -0.17 0.0970.011 0.26 0.11 0.25 0.28 0.37 1 -0.39-0.078 0.18 0.13 0.0890.0580.00760.00950.0530.00270.0340.00260.018-0.06 0.0150.033 0.33
          -0.13 0.026 0.0310.00880.063 0.039 0.026 -0.03-0.016 0.28 -0.095 0.18 -0.39 1 0.28 -0.17 0.05 0.16 0.07 -0.030.00720.0350.0450.0020.00310.043 0.034-0.029.00
           -0.1 0.23 -0.19 0.35 -0.15 0.13 0.12 0.17 -0.11 0.32 0.12 -0.0250.078 0.28 1 -0.1 0.32 0.45 0.2 0.063 -0.15 -0.0350.0650.0470.049 0.0460.029 -0.14 0.041 0.013 -0.46
           0.083 0.05 0.061 -0.16 -0.14 0.085 -0.11 0.075 0.014 -0.22 0.066 -0.12 0.18 -0.17 -0.1 1 0.047 0.098 0.0040 0.010 0.010 0.019 0.019 0.014 0.008 0.098 0.07 0.021 -0.050 0.008
           0.038 0.2 -0.12 0.14 -0.2 0.2 0.052 0.21 -0.0870.041 0.28 -0.18 0.13 0.05 0.32 0.047 1
                                                                                                                                                                               0.2
          0.029 0.19 -0.15 0.29 -0.032 0.15 0.07 0.27 0.15 0.28 0.26 -0.15 0.089 0.16 0.45 -0.098 0.3 1 0.22 -0.11 -0.0970.0510.0130.0650.00570.0770.066-0.16 -0.073 0.01 -0.44
           0.0150.0540.037 0.1 -0.064 0.15 0.06 0.17 -0.071 0.15 0.097-0.0760.058 0.07 0.20.000470.28 0.22 1 0.0480.0360.0270.0590.0150.0050.0170.022-0.0620.0350.027 -0.21
           0.0050.00410.0110.0480.0220.013<mark>0.093</mark>0.0620.0310.0670.0550.0690.0076-0.03-0.0630.0030.072-0.11 0.048 1 0.0260.0240.0030.0280.0280.0230.0210.0320.0130.00930.0370.033
            .013 -0.16-0.058-0.14 0.026-0.0370.0220.0110.0880.00880.0390.0360.0098.0072-0.150.0064-0.03-0.0970.0360.026 1 -0.19-0.043-0.0440.00160.0160.0160.0160.0590.063 0.37 0.073
                                                                                                                                                                               0.0
          0.011 0.13 0.062 0.0420.00730.0790.055 0.13 0.2 0.0320.00620.02 0.0530.0350.0350.0190.0450.0510.0270.024 0.19 1 0.23 0.0850.0170.0230.0110.0450.0540.0860.04
            0.0190.045 - 0.14 0.25 - 0.0210.059.0004 0.0490.0610.022 - 0.04 0.0360.0830.0450.0650.0190.0560.0130.0590.0030.043 - 0.23 1 0.0240.0130.0510.0590.0030.0420.023
            .00680.0620.0230.0470.00230.0350.00810.0580.0920.0020.0420.0320.00270.0470.0140.0230.0650.0150.0280.0440.0850.024 1 0.0350.058 0.03 -0.0230.045 -0.15-0.04
            .0199.00140.0230.0360.0360.0360.0260.000940.010.000530.0220.0150.0340.00310.0490.00860.0040.00570.0050.00239.00160.0170.0130.035 \\ 1
       724
                                                                                                                                                                                -0.2
            .062 0.15 -0.081-0.18-0.029-0.07 0.054 -0.14 0.05 0.11 -0.0240.0810.00260.043-0.0460.0980.066-0.0770.017-0.0210.0160.023-0.0510.058-0.027
            0021-0.14 0.16 -0.2 0.055-0.12-0.069-0.21 0.11-0.075-0.14 0.066-0.06-0.029-0.14 0.021-0.12-0.16-0.0620.0130.0590.045-0.0630.0230.00280.07-0.018
       8 -0.0230.0380.0580.0130.00820.0770.0240.0650.0590.078-0.110.00460.0150.00970.041-0.0210.0150.0730.0330.00970.0630.0540.0330.0450.0320.053 0.03 0.24 1
Model Training
columns=credit_card_data.columns.tolist()
columns=[c for c in columns if c not in ['Class']]
target='Class'
state=np.random.RandomState(2)
x=credit_card_data[columns]
y=credit_card_data[target]
print(x.shape)
print(y.shape)
      (9965, 30)
      (9965,)
```

```
print(x)
                                V2
                      V1
                                          V3
                                                   V4
                                                             V5
    0
              0 -1.359807 -0.072781 2.536347
                                             1.378155 -0.338321 0.462388
              0 1.191857 0.266151 0.166480 0.448154 0.060018 -0.082361
    1
    2
              1 -1.358354 -1.340163 1.773209 0.379780 -0.503198 1.800499
              1 -0.966272 -0.185226 1.792993 -0.863291 -0.010309 1.247203
    3
              2 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921
    4
    9960 14837 1.286884 -0.124610 0.148283 -0.259343 0.248357 0.896718
    9961
          14854 1.318742 0.496408 0.114876 0.695262 0.170133 -0.537180
          14857 1.241757 0.419587 0.806183 0.894811 -0.507886 -1.118126
          14861 1.304800 -0.052885 0.415235 -0.081725 -0.223525 0.097752
```

```
V7
                         V8
                                   V9
                                                V20
                                                         V21
                                                                   V22
                                      ... 0.251412 -0.018307
     0
          0.239599 0.098698 0.363787
                                                             0.277838
                                      ... -0.069083 -0.225775 -0.638672
     1
         -0.078803 0.085102 -0.255425
                                      ... 0.524980 0.247998 0.771679
          0.791461 0.247676 -1.514654
                                      ... -0.208038 -0.108300
          0.237609 0.377436 -1.387024
     3
          0.592941 -0.270533  0.817739  ...  0.408542 -0.009431  0.798278
                                      . . .
     9960 -0.626627 0.227693 1.618678
                                      ... -0.093459 -0.381864 -0.904515
                                      ... -0.051795 -0.484943 -1.111176
     9961 0.025492 -0.272931 1.267298
                                      ... -0.107163 -0.379396 -0.817785
    9962 0.018908 -0.343335 1.210781
     9963 -0.561240 0.067228 1.617203
                                      ... -0.108758 -0.379597 -0.929204
    9964 0.000000 0.000000 0.000000
                                      ... 0.000000 0.000000 0.000000
               V23
                        V24
                                 V25
                                           V26
                                                    V27
                                                              V28 Amount
    0
         -0.110474 0.066928 0.128539 -0.189115 0.133558 -0.021053 149.62
          0.101288 -0.339846  0.167170  0.125895 -0.008983  0.014724
          0.909412 -0.689281 -0.327642 -0.139097 -0.055353 -0.059752
         -0.190321 -1.175575  0.647376 -0.221929  0.062723  0.061458  123.50
         -0.137458   0.141267   -0.206010   0.502292   0.219422   0.215153
     9960 -0.027985 -1.743540 0.090885 0.870425 -0.084116 -0.022744
                                                                    12.18
     9961 0.028259 -0.549934 0.328634 0.106061 -0.046154 0.017304
                                                                     1.78
     9962
          0.181425 0.662879
                            0.172535 0.033636 -0.051084 0.017208
                                                                     1.29
     9963
          0.020955 -0.877006 0.084384 0.807465 -0.099851 -0.015404
                                                                     4.72
     9964
          0.000000 \quad 0.000000 \quad 0.000000 \quad 0.000000 \quad 0.000000 \quad 0.0000000
                                                                     0.00
     [9965 rows x 30 columns]
print(y)
     0
            0.0
            0.0
     3
            0.0
    4
            0.0
     9960
            9.9
     9961
            0.0
     9962
            0.0
     9963
            0.0
     9964
            0.0
     Name: Class, Length: 9965, dtype: float64
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,stratify=y,random_state=2)
print(x.shape,x train.shape,x test.shape)
     (9965, 30) (7972, 30) (1993, 30)
#model fitting
rf=RandomForestClassifier()
rf.fit(x_train,y_train)
     RandomForestClassifier
     RandomForestClassifier()
x_train_pred=rf.predict(x_train)
model = LogisticRegression(solver='saga')
# training the Logistic Regression Model with Training Data
model.fit(x train, y train)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='12',
random_state=None, solver='saga', tol=0.0001, verbose=0,
warm_start=False)
```

9964 14864 -1.747939 3.712444 0.000000 0.000000 0.000000 0.000000

```
Model Evaluation
```

```
#train data accuracy
accuracy=accuracy_score(x_train_pred,y_train)
print("train_Accuracy:",accuracy)
→ train_Accuracy: 1.0
#accuracy on test datatest
x_test_pred=rf.predict(x_test)
test_data_accuracy=accuracy_score(x_test_pred,y_test)
print("Test_data_accuracy",test_data_accuracy)
    Test_data_accuracy 0.9989949748743718
precision=precision_score(y_test,x_test_pred)
print("Precision:",precision)
    Precision: 0.93333333333333333
f1score=f1_score(y_test,x_test_pred)
print("f1_score:",f1score)
    f1_score: 0.874999999999999
recall=recall_score(x_test_pred,x_test_pred)
print('recall:',recall)
    recall: 1.0
confusion_matrix = metrics.confusion_matrix(y_test, x_test_pred)
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])
cm_display.plot()
plt.show()
```

