In [75]:
```
import numpy as np
```

In [76]:
```
import pandas as pd
```

In [77]:
```
import matplotlib.pyplot as plt
```

In [78]:
```
import seaborn as sns
```

In [79]:
```
ds=pd.read_csv("Downloads\creditcard.csv")
```

In [80]:
```
ds
```

Out[80]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	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	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	0.409542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.77	0
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053825	24.79	0
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.88	0
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.00	0
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.00	0

284807 rows × 31 columns

In [81]:
```
ds.isnull()
```

Out[81]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	False	False	False	False	False	False	False	False	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	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	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	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284803	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284804	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284805	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
284806	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

284807 rows × 31 columns

In [82]:
```
ds.isnull().sum()
```

Out[82]:
```
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
```

In [83]:
```
ds.columns
```

Out[83]:
```
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')
```

In [84]:
```
x=ds.drop(columns=["Class"])
```

In [85]:
```
x
```

Out[85]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	0.251412	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.069083	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.524980	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.208038	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	0.408542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	2.606837	-4.918215	7.305334	1.914428	...	1.475829	0.213454	0.111864	1.014480	-0.509348	1.436807	0.250034	0.943651	0.823731	0.77
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.059616	0.214205	0.924384	0.012463	-1.016226	-0.606624	-0.395255	0.068472	-0.053825	24.79
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.001396	0.232045	0.578229	-0.037501	0.640134	0.265745	-0.087371	0.004455	-0.026561	67.88
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.127434	0.265245	0.800049	-0.163298	0.123205	-0.569159	0.546668	0.108821	0.104533	10.00
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.382948	0.261057	0.643078	0.376777	0.008797	-0.473649	-0.818267	-0.002415	0.013649	217.00

284807 rows × 30 columns

In [86]:
```
y=ds['Class']
```

In [87]:
```
y
```

Out[87]:
```
0         0
1         0
2         0
3         0
4         0
         ..
284802    0
284803    0
284804    0
284805    0
284806    0
Name: Class, Length: 284807, dtype: int64
```

In [88]:
```
y.count()
```

Out[88]:
```
284807
```

In [89]:
```
x.shape
```

Out[89]:
```
(284807, 30)
```

In [90]:
```
y.shape
```

Out[90]:
```
(284807,)
```

In [91]:
```
from sklearn.model_selection import train_test_split
```

In [92]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [93]:
```
x_train,x_test,y_train,y_test
```

Out[93]:
```
(        Time        V1        V2        V3        V4        V5        V6  \
 68806   53150.0 -1.115047  1.935583  0.800712 -1.060398  0.032621  0.853422
 40018   40000.0  1.228473 -0.138826  0.473795 -0.166381 -0.453564 -0.199235
 256360  154865.0 -1.926148 -0.186628 -0.286727  1.596852 -0.488963 -0.311495
 234820  148130.0 -2.040232  0.196183 -1.784050 -0.530027  0.177496 -1.443051
 10001    10014.0 -0.837077  1.034710  2.333129  3.103776  0.229143  0.501123
 ...          ...       ...       ...       ...       ...       ...       ...
 211543  134659.0 -1.321976  1.138686 -0.940851  0.154160  0.109802 -0.538822
 86293    61167.0 -0.927810  0.918729  1.478453  0.213171  0.933695  1.261486
 122579   76016.0  1.512592 -0.949435 -0.210962 -1.638850 -0.854348 -0.465996
 152315   97253.0  1.798863 -1.699791 -0.142182 -0.619533 -1.570248  0.083268
 117952   74887.0 -0.589400  0.747928  1.784781  0.899612  0.257067 -0.891301

                V7        V8        V9  ...       V20       V21       V22  \
 68806   -0.614243 -3.231161  1.539948  ... -0.644896  3.920385 -0.539618
 40018   -0.355309  0.048598  0.128943  ...  0.032481 -0.172147 -0.574738
 256360  -0.303207  0.035513  1.301312  ... -0.355216 -0.528432 -1.251300
 234820  0.255483 -0.365759  0.602945  ... -0.216247  0.201184  0.752745
 10001    0.314283 -0.161583  0.409634  ... -0.134542 -0.363839 -0.458101
 ...           ...       ...       ...  ...       ...       ...       ...
 211543  0.490058  0.513762 -0.493834  ... -0.436962 -0.012778 -0.237503
 86293   0.594752  0.404286 -0.939740  ...  0.888281 -0.051356 -0.004245
 122579 -0.669193 -0.135566 -2.284345  ... -0.279028 -0.558803 -1.377240
 152315 -1.501980  0.176287  1.755507  ...  0.146998  0.181914  0.351358
 117952  0.122334  0.034736 -0.283998  ...  0.047357 -0.000910  0.000367

                V23       V24       V25       V26       V27       V28  Amount
 68806   0.033156 -0.774946  0.105868 -0.430853  0.229737 -0.070591   12.95
 40018   0.036834 -0.903782  0.073315  0.810356 -0.069178  0.001890   25.57
 256360  0.455697 -0.120539 -0.381515 -1.099295  0.052747 -0.032622    6.90
 234820 -0.013690 -0.079318  0.234805 -0.098151  0.000028 -0.031675    4.55
 10001   0.164097  0.234167 -0.418734 -0.074078 -0.355105 -0.109616    3.79
 ...          ...       ...       ...       ...       ...       ...     ...
 211543  0.008713 -0.767844 -0.397162  0.316379 -0.463125 -0.010589   49.89
 86293   0.096535 -0.964599 -0.522294  0.296733  0.145939  0.118400   24.99
 122579  0.808444 -0.579511  0.297851 -0.485367 -0.001415  0.003665   34.90
 152315  0.115638 -0.566188 -0.596200 -0.295152 -0.033616 -0.032471  171.31
 117952 -0.238139 -0.463529 -0.243573 -0.370928  0.086592  0.118084   15.99

 [227845 rows x 30 columns],
             Time        V1        V2        V3        V4        V5        V6  \
 183484  125821.0 -0.323334 -1.057445 -0.048341 -0.607284  1.259821 -0.091761
 255448  157235.0 -0.349718  0.932619  0.142992 -0.657071  1.169784 -0.733369
 244749  152471.0 -1.614711 -2.406529  0.326194  0.665520  2.369268 -1.775367
 63919    50927.0 -2.477184  0.860613  1.441850  1.051819 -1.856621  2.078384
 11475    15899.0  1.338831 -0.547264  0.737389 -0.212383 -1.113039 -0.525744
 ...          ...       ...       ...       ...       ...       ...       ...
 236778  148949.0 -1.227033  0.987207  0.654979 -2.559724  0.346834 -0.634095
 127073   78200.0  1.250596  0.159552  0.147621  0.472220 -0.023937 -0.287444
 208502  137149.0  1.125402 -2.288998 -3.123785 -0.103566 -0.311680 -1.151728
 263323  160893.0  2.064857  0.285198 -2.487311  0.357674  0.965436 -0.971181
 246221  153986.0  2.351382 -1.325229 -1.339220 -1.741109 -0.726172 -0.191973

                V7        V8        V9  ...       V20       V21       V22  \
 183484  1.155181 -0.124335 -0.174640  ...  0.186409 -0.207998 -0.433898
 255448  1.009985 -0.071069 -0.302083  ... -0.096502 -0.271537 -0.833209
 244749 -1.139946  0.329904  0.903513  ... -0.419835  0.701199  1.134489
 63919   0.510828 -0.243399 -0.260691  ... -0.987790  0.810468  0.692245
 11475  -0.801403 -0.063672  0.997276  ... -0.126871 -0.139436 -0.074719
 ...          ...       ...       ...  ...       ...       ...       ...
 236778  0.738285  0.405477  0.070475  ... -0.097164 -0.070342 -0.293542
 127073  0.021041 -0.041342  0.102588  ... -0.139836 -0.210803 -0.560817
 208502 -1.184290 -0.635862 -1.314091  ...  0.484541  0.263259  0.020985
 263323  0.622246 -0.313264 -0.190080  ... -0.161876  0.090425  0.422986
 246221 -1.059594 -0.157321 -1.225573  ... -0.399231 -0.213217 -0.189354

                V23       V24       V25       V26       V27       V28  Amount
 183484 -0.261613 -0.046651  0.211512  0.008297  0.108494  0.161139   40.00
 255448 -0.030360  0.490035 -0.404810  0.134350  0.076830  0.175562    1.98
 244749  0.965054  0.640981 -1.801998 -1.041114  0.286285  0.437322   96.00
 63919   0.150121 -0.260777  0.005183 -0.177847 -0.510966 -0.668533  308.00
 11475   0.067055  0.333122  0.379087 -0.268706 -0.002769  0.003272    5.00
 ...          ...       ...       ...       ...       ...       ...     ...
 236778 -0.371721  0.518298  0.858450  0.749546  0.011475  0.027138   20.30
 127073  0.049119 -0.429060  0.299399  0.234883 -0.022217  0.006667    6.15
 208502 -0.636845 -0.259110  0.272149  0.961966 -0.240867 -0.007494  583.21
 263323  0.019409  0.691878  0.354899  0.682896 -0.103162 -0.061743    2.95
 246221  0.183131  0.011192 -0.064682 -0.148726  0.007269 -0.051634   15.00

 [56962 rows x 30 columns],
 68806     0
 40018     0
 256360    0
 234820    0
 10001     0
          ..
 211543    0
 86293     0
 122579    0
 152315    0
 117952    0
 Name: Class, Length: 227845, dtype: int64,
 183484    0
 255448    0
 244749    0
 63919     0
 11475     0
          ..
 236778    0
 127073    0
 208502    0
 263323    0
 246221    0
 Name: Class, Length: 56962, dtype: int64)
```

In [94]:
```
x_train.shape,x_test.shape
```

Out[94]:
```
((227845, 30), (56962, 30))
```

In [95]:
```
y_train.shape,y_test.shape
```

Out[95]:
```
((227845,), (56962,))
```

In [96]:
```
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
lg.fit(x_train,y_train)
```

Out[96]:
```
C:\Users\Nitesh\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

In [97]:
```
y_lg=lg.predict(x_test)
print(y_lg)
```

Out[97]:
```
[0 0 0 ... 0 0 0]
```

In [98]:
```
from sklearn.metrics import accuracy_score
a=accuracy_score(y_test,y_lg)*100
print(a)
```

Out[98]:
```
99.89817773253344
```

In [99]:
```
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score
def metrics(actuals,predictions):
    print("accuracy:{:.5f}".format(accuracy_score(actuals,predictions)))
    print("precision:{:.5f}".format(precision_score(actuals,predictions)))
    print("recall:{:.5f}".format(recall_score(actuals,predictions)))
    print("f1:{:.5f}".format(f1_score(actuals,predictions)))
```

In [100]:
```
metrics(y_test,y_lg.round())
```

Out[100]:
```
accuracy:0.99898
precision:0.73118
recall:0.67327
f1:0.70103
```

In [101]:
```
from sklearn.metrics import confusion_matrix
```

In [102]:
```
confusion_matrix(y_test,y_lg)
```

Out[102]:
```
array([[56836,    25],
       [   33,    68]], dtype=int64)
```

In [106]:
```
from sklearn.ensemble import RandomForestClassifier
random_forest=RandomForestClassifier(n_estimators=100)
```

In [108]:
```
random_forest.fit(x_train,y_train)
```

Out[108]:
```
RandomForestClassifier()
```

In [109]:
```
predictions_rf=random_forest.predict(x_test)
```

In [110]:
```
random_forest_score=random_forest.score(x_test,y_test)*100
```

In [111]:
```
random_forest_score
```

Out[111]:
```
99.94733300992591
```

In [113]:
```
from sklearn.tree import DecisionTreeClassifier
decision_tree=DecisionTreeClassifier()
```

In [116]:
```
decision_tree.fit(x_train,y_train)
prediction_dt=decision_tree.predict(x_test)
decision_tree_score=decision_tree.score(x_test,y_test)*100
decision_tree_score
```

Out[116]:
```
99.91924448805307
```

In [117]:
```
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score
def metrics(actuals,predictions):
    print("accuracy:{:.5f}".format(accuracy_score(actuals,predictions)))
    print("precision:{:.5f}".format(precision_score(actuals,predictions)))
    print("recall:{:.5f}".format(recall_score(actuals,predictions)))
    print("f1:{:.5f}".format(f1_score(actuals,predictions)))
```

In [119]:
```
metrics(y_test,prediction_dt)
```

Out[119]:
```
accuracy:0.99919
precision:0.75701
recall:0.80198
f1:0.77885
```

In [ ]:
```

```