```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix,accuracy_score,classification_report,ConfusionMatrixDisplay
df=pd.read_csv('/content/drive/MyDrive/creditcard.csv')
df
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	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
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	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
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0
...	...	...	...	...	...	...	...	...	...	...	...	...	
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	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
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	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
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0

284807 rows × 31 columns

```
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	
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.
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.
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.
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.
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.

5 rows × 31 columns

```
df.tail()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.1
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.9
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.5
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.8
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.6

5 rows × 31 columns

```
df.columns
```

```
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')
```

```
df.shape
```

```
(284807, 31)
```

```
df.isna().sum()
```

	0
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

```
df.dtypes
```

	0
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

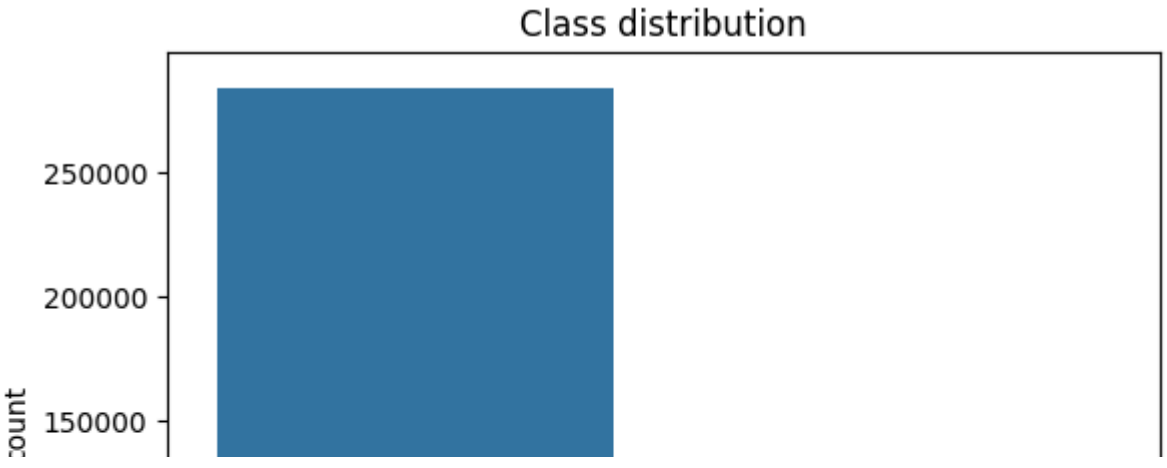
```
df['Class'].value_counts()
```

Class	count
0	284315
1	492

dtype: int64

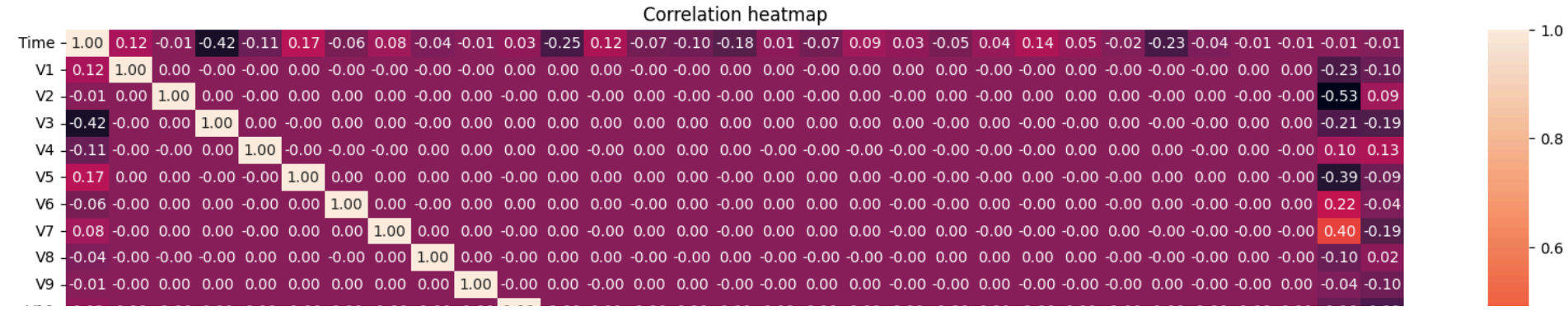
```
sns.countplot(x='Class',data=df)
plt.title("Class distribution")
```

```
Text(0.5, 1.0, 'Class distribution')
```




```
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True,fmt='0.2f')
plt.title("Correlation heatmap")
```

```
Text(0.5, 1.0, 'Correlation heatmap')
```




```
x=df.drop(columns=['Class'])
y=df['Class']
```

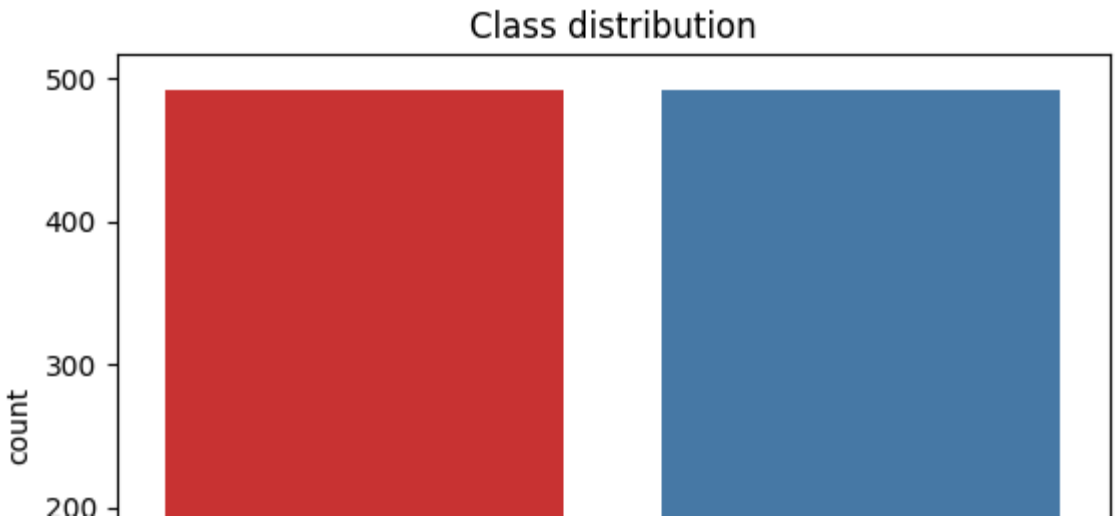
```
from imblearn.under_sampling import RandomUnderSampler
sampler=RandomUnderSampler()
x_res,y_res=sampler.fit_resample(x,y)
df1=pd.DataFrame(x_res)
df2=pd.DataFrame(y_res)
result=pd.concat([df1,df2],axis=1,join='inner')
#resampled dataset
df=result;
```

```
sns.countplot(x='Class',data=df,palette='Set1')
plt.title("Class distribution")
```

```
<ipython-input-159-2738ae6ca0f4>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `

  sns.countplot(x='Class',data=df,palette='Set1')
Text(0.5, 1.0, 'Class distribution')
```




```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
print(x_train)
```

```
             Time        V1        V2        V3        V4        V5        V6  \
2557       2105.0 -2.289565 -0.480260  0.818685 -1.706423  0.822102 -1.660326
247823   153702.0 -0.313717 -4.064342 -3.398445  0.704011  0.101662  1.529848
152342    97283.0 -1.809763 -0.567439  2.265186 -0.960318 -1.212537  1.516493
103385    68628.0  1.192319  0.178575  0.141491  0.459628 -0.049959 -0.112122
8771      11951.0 -0.963451  0.700311  1.097333 -1.547626  0.669966  0.513533
...           ...       ...       ...       ...       ...       ...       ...
119879    75618.0  1.173488  0.100792  0.490512  0.461596 -0.296377 -0.213165
259178   159000.0 -0.775981  0.144023 -1.142399 -1.241113  1.940358  3.912076
131932    79795.0 -0.146609  0.992946  1.524591  0.485774  0.349308 -0.815198
146867    87931.0 -2.948638  2.354849 -2.521201 -3.798905  1.866302  2.727695
121958    76381.0  1.233174 -0.784851  0.386784 -0.698559 -1.034018 -0.637028

              V7        V8        V9  ...       V20       V21       V22  \
2557    0.944047 -0.541765  1.323156  ... -0.831985 -0.210837  0.914737
247823  1.551670 -0.036774  0.015829  ...  2.142593  0.853186 -0.091941
152342 -1.417176  0.903421  1.961027  ... -0.554004 -0.509915 -0.424978
103385 -0.163883  0.155740 -0.067566  ... -0.149985 -0.240464 -0.739862
8771    0.333683  0.270900  1.381880  ...  0.122458 -0.279519 -0.470181
...          ...       ...       ...  ...       ...       ...       ...
119879 -0.165254  0.119221 -0.114199  ... -0.157534 -0.186027 -0.574283
259178 -0.466107  1.360620  0.400697  ... -0.295730  0.037078 -0.019575
131932  1.076640 -0.395316 -0.491303  ...  0.007155  0.052649  0.354089
146867 -0.471769  2.217537  0.580199  ...  0.417396 -0.332759 -1.047514
121958 -0.502369 -0.188057 -0.749637  ...  0.337732  0.027634 -0.234522

             V23       V24       V25       V26       V27       V28    Amount
2557    0.867888  0.422969  0.310584 -0.781488  0.392241 -0.147757      1.00
247823 -0.936215 -0.833081 -0.498728  0.651183 -0.290331  0.110360   1194.28
152342 -0.268621  0.010121  0.466862  0.835540 -0.062385  0.088079     75.00
103385  0.116799 -0.373837  0.125470  0.130126 -0.016956  0.011937      1.98
8771   -0.124037 -1.388839 -0.237453  0.785347  0.349708  0.216207     37.31
...          ...       ...       ...       ...       ...       ...       ...
```

```
119879  0.161405 -0.006140  0.091444  0.109235 -0.020922  0.003967     1.98
259178  0.241830  0.682820 -1.635109 -0.770941  0.066006  0.137056    89.23
131932 -0.291198  0.402849  0.237383 -0.398467 -0.121139 -0.196195     3.94
146867  0.143326  0.678869  0.319710  0.426309  0.496912  0.335822     1.00
121958 -0.059544 -0.109073  0.290326 -0.393074  0.001217  0.038588   113.00

[199364 rows x 30 columns]
```

```
print(y_train)
```

```
2557      0
247823    0
152342    0
103385    0
8771      0
         ..
119879    0
259178    0
131932    0
146867    0
121958    0
Name: Class, Length: 199364, dtype: int64
```

```
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
scaler.fit(x_train)
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
print(x_train)
```

```
[[-1.95144063 -1.16681856 -0.28654908 ...  0.96676    -0.4486209
  -0.33974783]
 [ 1.24181171 -0.15922175 -2.43543667 ... -0.71311931  0.33519183
   4.27773998]
 [ 0.05339701 -0.92214029 -0.33881865 ... -0.15212107  0.26753092
  -0.0533992 ]
 ...
 [-0.31497174 -0.07400401  0.59673261 ... -0.29672056 -0.59571272
  -0.32837128]
 [-0.14359433 -1.50291696  1.41328113 ...  1.2243654   1.01984233
  -0.33974783]
 [-0.38688453  0.62962541 -0.46917078 ...  0.0044092   0.11724428
   0.09364469]]
```

```
print(x_test)
```

```
[[-1.12151562 -8.42703903  5.1486402  ... -4.96650013 -3.16657662
   1.06564352]
 [-1.06346299  0.17404952 -1.64365414 ...  0.10230985  0.3099202
   1.66902653]
 [-1.24834248  0.71449029 -0.35276585 ...  0.02949291  0.01413826
  -0.22366055]
 ...
 [-1.32360446  0.38159558 -0.61498614 ...  0.1450263   0.177531
   0.41327168]
 [ 0.97516144  1.06174851  0.1019923  ... -0.15089508 -0.09379868
  -0.34017349]
 [-0.70684801  0.55744777 -0.03940111 ...  0.19863631  0.10769086
  -0.30492165]]
```

```
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
y_pred=dt.predict(x_test)
y_pred
```

```
array([1, 0, 0, ..., 0, 0, 0])
```

```
print(y_test)
```

```
43428     1
49906     0
29474     0
276481    0
278846    0
         ..
180795    0
259979    0
```

```
21885     0
217812    0
86348     0
Name: Class, Length: 85443, dtype: int64
```

```
accuracy_score(y_test,y_pred)
```

```
0.9991222218320986
```

```
cm=confusion_matrix(y_test,y_pred)
cm
```

```
array([[85261,    46],
       [   29,   107]])
```

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85307
           1       0.70      0.79      0.74       136

    accuracy                           1.00     85443
   macro avg       0.85      0.89      0.87     85443
weighted avg       1.00      1.00      1.00     85443
```

```
cmd=ConfusionMatrixDisplay(cm,display_labels=[0,1])
cmd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f71f3c7add0>
```