In [121…
```
import pandas as pd
df=pd.read_csv('C:\\Users\\Atif\\Desktop\\python learning\\creditcard2.csv')
df.head(10)
```

Out[121]:

	Time	V1	V2	V3	V4	V6	V8	V10	V12	V14	V16	V18	V20	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	0.462388	0.098698	0.090794	-0.617801	-0.311169	-0.470401	0.025791	0.251412	0.
1	0.0	1.191857	0.266151	0.166480	0.448154	-0.082361	0.085102	-0.166974	1.065235	-0.143772	0.463917	-0.183361	-0.069083	-0.
2	1.0	-1.358354	-1.340163	1.773209	0.379780	1.800499	0.247676	0.207643	0.066084	-0.165946	-2.890083	-0.121359	0.524980	0.
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	1.247203	0.377436	-0.054952	0.178228	-0.287924	-1.059647	1.965775	-0.208038	0.
4	2.0	-1.158233	0.877737	1.548718	0.403034	0.095921	-0.270533	0.753074	0.538196	-1.119670	-0.451449	-0.038195	0.408542	0.
5	2.0	-0.425966	0.960523	1.141109	-0.168252	-0.029728	0.260314	-0.371407	0.359894	-0.137134	0.401726	0.068653	0.084968	-0.
6	4.0	1.229658	0.141004	0.045371	1.202613	0.272708	0.081213	-0.099254	-0.153826	0.167372	-0.443587	-0.611987	-0.219633	-0.
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.428118	-3.807864	1.249376	0.291474	-1.323865	-0.076127	-0.358222	-0.156742	-1.
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	3.721818	0.851084	-0.410430	-0.110452	0.074355	-0.210077	0.118765	0.052736	-0.
9	9.0	-0.338262	1.119593	1.044367	-0.222187	-0.246761	0.069539	-0.366846	0.836390	-0.443523	0.739453	0.476677	0.203711	-0.

In [122…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 17 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   V6      284807 non-null  float64
 6   V8      284807 non-null  float64
 7   V10     284807 non-null  float64
 8   V12     284807 non-null  float64
 9   V14     284807 non-null  float64
 10  V16     284807 non-null  float64
 11  V18     284807 non-null  float64
 12  V20     284807 non-null  float64
 13  V22     284807 non-null  float64
 14  V26     284807 non-null  float64
 15  Amount  284807 non-null  float64
 16  Class   284807 non-null  int64  
dtypes: float64(16), int64(1)
memory usage: 36.9 MB
```

In [123…
```
df['Class'].value_counts()
#this data is highly unbalanced.
```

Out[123]:
```
0    284315
1       492
Name: Class, dtype: int64
```

In [124…
```
legit = df[df.Class == 0]
fraud = df[df.Class == 1]
```

In [125…
```
print(legit.shape)
print(fraud.shape)
```

```
(284315, 17)
(492, 17)
```

In [126…
```
legit.Amount.describe()
```

Out[126]:
```
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
```

In [127…
```
fraud.Amount.describe()
```

Out[127]:
```
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
```

In [128…
```
df.groupby('Class').mean()
```

Out[128]:

Class	Time	V1	V2	V3	V4	V6	V8	V10	V12	V14	V16	V18
0	94838.202258	0.008258	-0.006271	0.012171	-0.007860	0.002419	-0.000987	0.009824	0.010832	0.012064	0.007164	0.003887
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-1.397737	0.570636	-5.676883	-6.259393	-6.971723	-4.139946	-2.246308

In [149…
```
legit_sample = legit.sample(n=500)
#Sample is use to contain similar distribution of fraud detection
```

In [150…
```
df2 = pd.concat([legit_sample,fraud],axis=0)
```

In [151…
```
df2.head()
```

Out[151]:

	Time	V1	V2	V3	V4	V6	V8	V10	V12	V14	V16	V18	
120586	75880.0	-0.360940	0.575255	1.231896	-0.018485	0.066020	-0.712466	0.687133	0.219406	-0.015294	-0.579871	-0.533838	0.11
247210	153502.0	1.975834	-1.279472	-1.732163	-0.668546	-1.302051	-0.384095	0.830592	-1.542513	0.298829	0.619769	-1.058380	0.11
205114	135596.0	2.071768	0.005524	-1.399734	0.435342	-0.641073	-0.348161	0.028771	0.975142	-0.052640	-0.191206	-0.879975	-0.11
11450	19897.0	1.342356	-0.579009	0.494596	-0.486911	0.185507	0.096775	0.268431	-2.542256	1.309462	0.577987	-1.259467	-0.06
88529	62179.0	1.353088	-0.522374	-0.038076	-0.670227	-0.800684	-0.306488	0.656556	0.029077	-0.009785	-1.201385	0.969406	-0.31

In [152…
```
df.tail()
```

Out[152]:

	Time	V1	V2	V3	V4	V6	V8	V10	V12	V14	V16	V18	
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-2.606837	7.305334	4.356170	2.711941	4.626942	1.107641	0.510632	1.4
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	1.058415	0.294869	-0.975926	0.915802	-0.675143	-0.711757	-1.221179	0.0
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	3.031260	0.708417	-0.484782	0.063119	-0.510602	0.140716	0.395652	0.0
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	0.623708	0.679145	-0.399126	-0.962886	0.449624	-0.608577	1.113981	0.1
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.649617	-0.414650	-0.915427	-0.031513	-0.084316	-0.302620	0.167430	0.3

In [153…
```
df2['Class'].value_counts()
```

Out[153]:
```
0    500
1    492
Name: Class, dtype: int64
```

In [154…
```
df2.groupby('Class').mean()
```

Out[154]:

Class	Time	V1	V2	V3	V4	V6	V8	V10	V12	V14	V16	V18
0	94928.818000	0.067118	0.020873	0.017716	-0.050567	-0.050908	-0.050201	0.042651	-0.036229	-0.010355	-0.032509	0.013191	-0
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-1.397737	0.570636	-5.676883	-6.259393	-6.971723	-4.139946	-2.246308	0

In [155…
```
X = df2.drop(columns='Class',axis=1)
Y = df2['Class']
```

In [156…
```
print(X)
```

```
            Time        V1        V2        V3        V4        V6        V8  \
120586   75880.0 -0.360940  0.575255  1.231896 -0.018485  0.066020 -0.712466
247210  153502.0  1.975834 -1.279472 -1.732163 -0.668546 -1.302051 -0.384095
205114  135596.0  2.071768  0.005524 -1.399734  0.435342 -0.641073 -0.348161
11450    19897.0  1.342356 -0.579009  0.494596 -0.486911  0.185507  0.096775
88529    62179.0  1.353088 -0.522374 -0.038076 -0.670227 -0.800684 -0.306488
...          ...       ...       ...       ...       ...       ...       ...
279863  169142.0 -1.927883  1.125653 -4.518331  1.749293 -2.010494  0.697211
280143  169347.0  1.378559  1.289381 -5.004247  1.411850 -1.326536  0.248525
280149  169351.0 -0.676143  1.126366 -2.213700  0.468308 -0.003346  1.210158
281144  169966.0 -3.113832  0.585864 -5.399730  1.817092 -2.943548  1.058733
281674  170348.0  1.991976  0.158476 -2.583441  0.408670 -0.096695 -0.068384

              V10       V12       V14       V16       V18       V20       V22  \
120586  0.687133  0.219406 -0.015294 -0.579871 -0.533838  0.115132 -0.531106
247210  0.830592 -1.542513  0.298829  0.619769 -1.058380  0.118725  1.505175
205114  0.028771  0.975142 -0.052640 -0.191206 -0.879975 -0.110054 -0.519126
11450   0.268431 -2.542256  1.309462  0.577987 -1.259467 -0.062370 -0.041527
88529   0.656556  0.029077 -0.009785 -1.201385  0.969406 -0.318645 -0.693206
...          ...       ...       ...       ...       ...       ...       ...
279863 -5.587794 -5.417424 -6.665177 -2.897825 -1.315147  1.252967 -0.319189
280143 -3.232153 -3.096915 -5.210141 -2.155297 -0.688505  0.226138  0.028234
280149 -3.463891 -2.775022 -4.057162 -1.603015 -0.507000  0.247968  0.834108
281144 -5.245984 -5.030465 -6.416628 -2.549498 -1.478138  0.306271 -0.269209
281674 -0.888722  0.728903 -1.948883  0.519436  1.197315 -0.017652 -0.295135

              V26  Amount
120586  0.059193   59.95
247210  0.289274  150.00
205114  0.135621   15.00
11450  -0.247289    5.00
88529   1.065988   54.97
...          ...     ...
279863  0.788395  390.00
280143  0.739467    0.76
280149  0.471111   77.89
281144  0.606116  245.00
281674 -0.289617   42.53

[992 rows x 16 columns]
```

In [157…
```
X.shape
```

Out[157]:
```
(992, 16)
```

In [158…
```
print(Y)
```

```
120586    0
247210    0
205114    0
11450     0
88529     0
         ..
279863    1
280143    1
280149    1
281144    1
281674    1
Name: Class, Length: 992, dtype: int64
```

In [159…
```
from sklearn.model_selection import train_test_split
```

In [160…
```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
```

In [161…
```
print(X.shape,X_train.shape,X_test.shape)
```

```
(992, 16) (793, 16) (199, 16)
```

In [162…
```
print(Y.shape,Y_train.shape,Y_test.shape)
```

```
(992,) (793,) (199,)
```

In [163…
```
from sklearn.linear_model import LogisticRegression
```

In [164…
```
model = LogisticRegression()
```

In [165…
```
model.fit(X_train,Y_train)
```

Out[165]:
```
▾ LogisticRegression
LogisticRegression()
```

In [166…
```
model.score(X_test,Y_test)
```

Out[166]:
```
0.9246231155778895
```

In [167…
```
model.predict([[10,1.449043781,-1.176338825,0.913859833,-1.375666655,-0.629152139,0.048455888,1.626659058,-0.67
]])
```

```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
s not have valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

Out[167]:
```
array([0], dtype=int64)
```

In [168…
```
model.predict([[170348,1.991976096,0.158475887,-2.583440645,0.408669993,-0.096694744,-0.068383878,-0.888721676,
]])
```

```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
s not have valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

Out[168]:
```
array([0], dtype=int64)
```

In [ ]: