In [1]:
```
import numpy as np
import pandas as pd
credit = pd.read_csv('E:\DATA_SCIENCE\Intership\Task-5\creditcard.csv')
credit.head()
```

Out[1]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533

5 rows × 31 columns

In [2]:
```
credit.shape

credit.describe().T
```

Out[2]:

	count	mean	std	min	25%	50%	
Time	284807.0	9.481386e+04	47488.145955	0.000000	54201.500000	84692.000000	139
V1	284807.0	1.168375e-15	1.958696	-56.407510	-0.920373	0.018109	
V2	284807.0	3.416908e-16	1.651309	-72.715728	-0.598550	0.065486	
V3	284807.0	-1.379537e-15	1.516255	-48.325589	-0.890365	0.179846	
V4	284807.0	2.074095e-15	1.415869	-5.683171	-0.848640	-0.019847	
V5	284807.0	9.604066e-16	1.380247	-113.743307	-0.691597	-0.054336	
V6	284807.0	1.487313e-15	1.332271	-26.160506	-0.768296	-0.274187	
V7	284807.0	-5.556467e-16	1.237094	-43.557242	-0.554076	0.040103	
V8	284807.0	1.213481e-16	1.194353	-73.216718	-0.208630	0.022358	
V9	284807.0	-2.406331e-15	1.098632	-13.434066	-0.643098	-0.051429	
V10	284807.0	2.239053e-15	1.088850	-24.588262	-0.535426	-0.092917	
V11	284807.0	1.673327e-15	1.020713	-4.797473	-0.762494	-0.032757	
V12	284807.0	-1.247012e-15	0.999201	-18.683715	-0.405571	0.140033	
V13	284807.0	8.190001e-16	0.995274	-5.791881	-0.648539	-0.013568	
V14	284807.0	1.207294e-15	0.958596	-19.214325	-0.425574	0.050601	
V15	284807.0	4.887456e-15	0.915316	-4.498945	-0.582884	0.048072	
V16	284807.0	1.437716e-15	0.876253	-14.129855	-0.468037	0.066413	
V17	284807.0	-3.772171e-16	0.849337	-25.162799	-0.483748	-0.065676	
V18	284807.0	9.564149e-16	0.838176	-9.498746	-0.498850	-0.003636	
V19	284807.0	1.039917e-15	0.814041	-7.213527	-0.456299	0.003735	
V20	284807.0	6.406204e-16	0.770925	-54.497720	-0.211721	-0.062481	
V21	284807.0	1.654067e-16	0.734524	-34.830382	-0.228395	-0.029450	
V22	284807.0	-3.568593e-16	0.725702	-10.933144	-0.542350	0.006782	
V23	284807.0	2.578648e-16	0.624460	-44.807735	-0.161846	-0.011193	
V24	284807.0	4.473266e-15	0.605647	-2.836627	-0.354586	0.040976	
V25	284807.0	5.340915e-16	0.521278	-10.295397	-0.317145	0.016594	
V26	284807.0	1.683437e-15	0.482227	-2.604551	-0.326984	-0.052139	
V27	284807.0	-3.660091e-16	0.403632	-22.565679	-0.070840	0.001342	
V28	284807.0	-1.227390e-16	0.330083	-15.430084	-0.052960	0.011244	
Amount	284807.0	8.834962e+01	250.120109	0.000000	5.600000	22.000000	
Class	284807.0	1.727486e-03	0.041527	0.000000	0.000000	0.000000	

In [3]:
```
fraud = credit[credit['Class'] == 1]
valid = credit[credit['Class'] == 0]
fraction = len(fraud)/float(len(valid))

print(fraction)
print("Fraud Cases: {}".format(len(credit[credit['Class'] == 1])))
print("Valid Cases: {}".format(len(credit[credit['Class'] == 0])))
```

```
0.0017304750013189597
Fraud Cases: 492
Valid Cases: 284315
```

In [4]:
```
print("Amount of details for the Fraudulent Transaction")
fraud.Amount.describe()



print("Amount of details for Normal Transaction")
valid.Amount.describe()
```

```
Amount of details for the Fraudulent Transaction
Amount of details for Normal Transaction
```

Out[4]:
```
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 [5]:
```
corrmat = credit.corr()

print(corrmat)
```

```
               Time            V1            V2            V3            V4
\
Time    1.000000  1.173963e-01 -1.059333e-02 -4.196182e-01 -1.052602e-01
V1      0.117396  1.000000e+00  4.135835e-16 -1.227819e-15 -9.215150e-16
V2     -0.010593  4.135835e-16  1.000000e+00  3.243764e-16 -1.121065e-15
V3     -0.419618 -1.227819e-15  3.243764e-16  1.000000e+00  4.711293e-16
V4     -0.105260 -9.215150e-16 -1.121065e-15  4.711293e-16  1.000000e+00
V5      0.173072  1.812612e-17  5.157519e-16 -6.539009e-17 -1.719944e-15
V6     -0.063016 -6.506567e-16  2.787346e-16  1.627627e-15 -7.491959e-16
V7      0.084714 -1.005191e-15  2.055934e-16  4.895305e-16 -4.104503e-16
V8     -0.036949 -2.433822e-16 -5.377041e-17 -1.268779e-15  5.697192e-16
V9     -0.008660 -1.513678e-16  1.978488e-17  5.568367e-16  6.923247e-16
V10     0.030617  7.388135e-17 -3.991394e-16  1.156587e-15  2.232685e-16
V11    -0.247689  2.125498e-16  1.975426e-16  1.576830e-15  3.459380e-16
V12     0.124348  2.053457e-16 -9.568710e-17  6.310231e-16 -5.625518e-16
V13    -0.065902 -2.425603e-17  6.295388e-16  2.807652e-16  1.303306e-16
V14    -0.098757 -5.020280e-16 -1.730566e-16  4.739859e-16  2.282280e-16
V15    -0.183453  3.547782e-16 -4.995814e-17  9.068793e-16  1.377649e-16
V16     0.011903  7.212815e-17  1.177316e-17  8.299445e-16 -9.614528e-16
```

In [6]:
```
X = credit.drop(['Class'], axis=1 )
Y = credit['Class']

print(X.shape)
print(Y.shape)
```

```
(284807, 30)
(284807,)
```

In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (classification_report, accuracy_score,precisio

X_train, X_test, y_train, y_test = train_test_split(X,Y , test_size=0.2, ra

model = LogisticRegression(solver='liblinear',max_iter=250)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [8]:
```
print("The model used is  LogisticRegression")
acc = accuracy_score(y_test, y_pred)
print(f"The accuracy is {acc}")
prec = precision_score(y_test, y_pred)
print(f"The precision score is {prec}")
rec = recall_score(y_test, y_pred)
print(f"The recall score is {rec}")
f1 = f1_score(y_test, y_pred)
print(f"The f1 score is {f1}")
MCC = matthews_corrcoef(y_test, y_pred)
print(f"The Matthews correlation coeficient is {MCC}")
```

```
The model used is  LogisticRegression
The accuracy is 0.9991046662687406
The precision score is 0.7647058823529411
The recall score is 0.5977011494252874
The f1 score is 0.6709677419354838
The Matthews correlation coeficient is 0.6756352111956744
```

In [ ]: