In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv("G:\iNurture_Rathinam\Machine Learning - M.Sc DSBA\Lab\Credit Card Fraud D

df.describe()
```

Out[2]:

| | Time | V1 | V2 | V3 | V4 | V |
|---|---|---|---|---|---|---|
| **count** | 284807.000000 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+05 | 2.848070e+0 |
| **mean** | 94813.859575 | 3.919560e-15 | 5.688174e-16 | -8.769071e-15 | 2.782312e-15 | -1.552563e-1 |
| **std** | 47488.145955 | 1.958696e+00 | 1.651309e+00 | 1.516255e+00 | 1.415869e+00 | 1.380247e+0 |
| **min** | 0.000000 | -5.640751e+01 | -7.271573e+01 | -4.832559e+01 | -5.683171e+00 | -1.137433e+0 |
| **25%** | 54201.500000 | -9.203734e-01 | -5.985499e-01 | -8.903648e-01 | -8.486401e-01 | -6.915971e-0 |
| **50%** | 84692.000000 | 1.810880e-02 | 6.548556e-02 | 1.798463e-01 | -1.984653e-02 | -5.433583e-0 |
| **75%** | 139320.500000 | 1.315642e+00 | 8.037239e-01 | 1.027196e+00 | 7.433413e-01 | 6.119264e-0 |
| **max** | 172792.000000 | 2.454930e+00 | 2.205773e+01 | 9.382558e+00 | 1.687534e+01 | 3.480167e+0 |

8 rows × 31 columns

In [3]:

```
df.head()
```

Out[3]:

| | 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

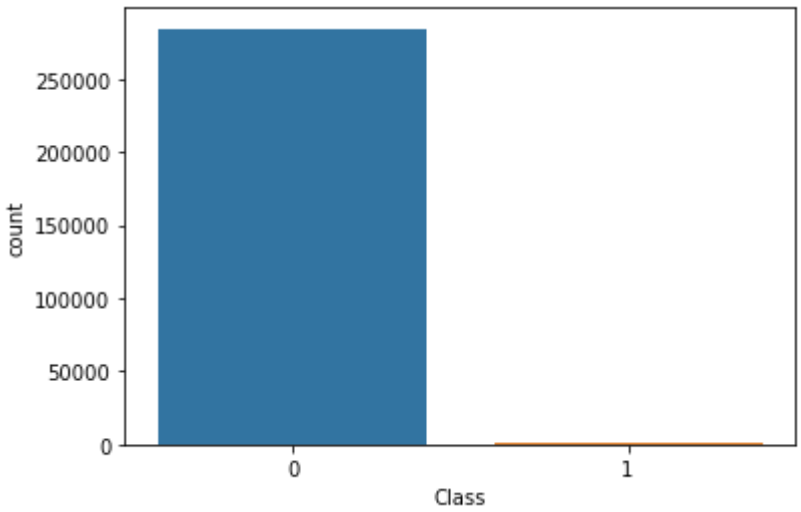
In [5]:

```
import seaborn as sns

sns.countplot(df['Class'])
print(df.Class.value_counts())
```

```
0    284315
1       492
Name: Class, dtype: int64
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: Future
Warning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other argumen
ts without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

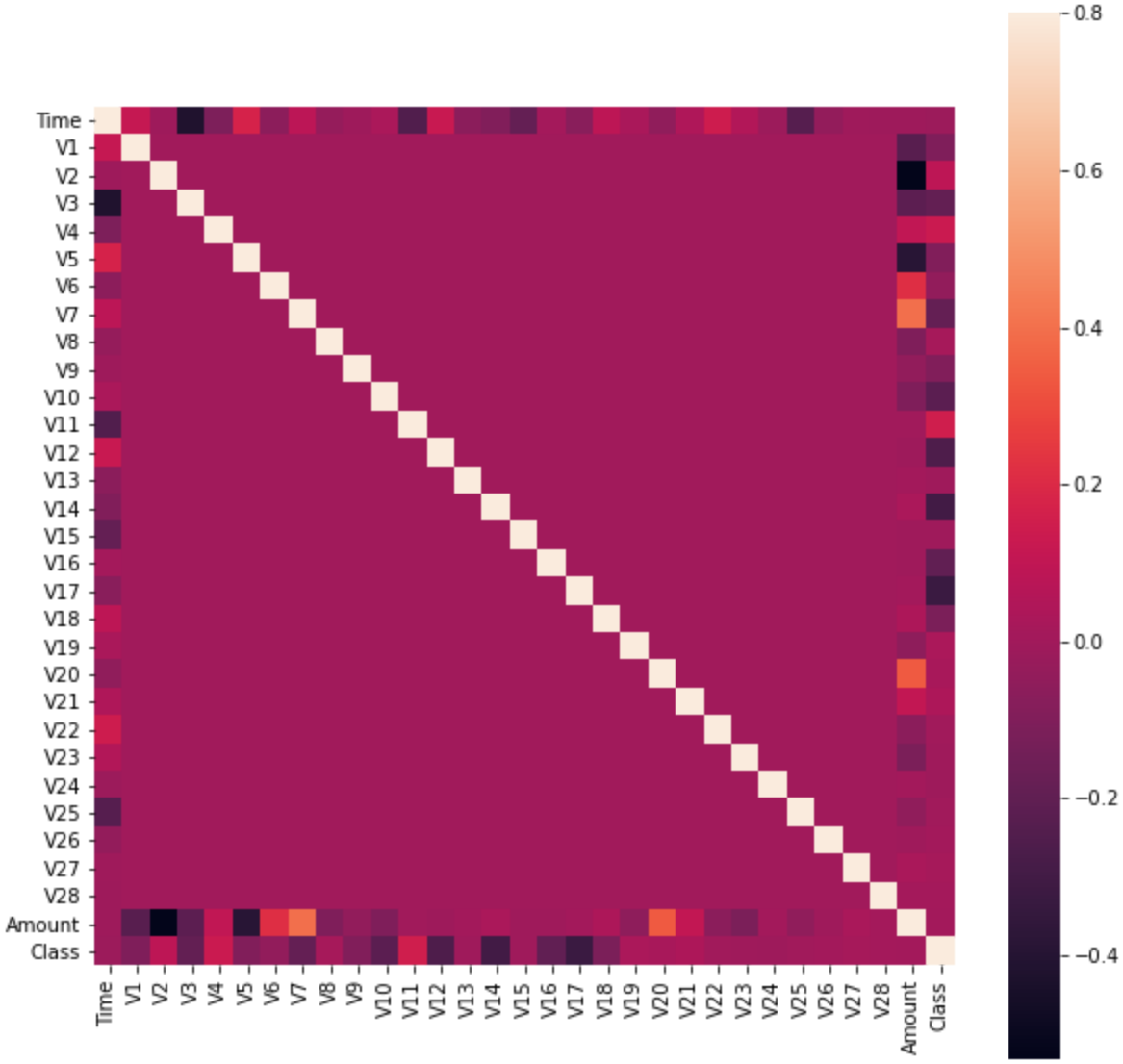


In [15]:

```
import matplotlib.pyplot as plt

corr_data = df.corr()

fig = plt.figure(figsize = (10, 10))
sns.heatmap(corr_data, vmax = .8, square = True)
plt.show()
```



In [17]:

```
from sklearn.model_selection import train_test_split

target = df["Class"].values
train = df.drop(["Class"], axis=1).values

x_train, x_test, y_train, y_test = train_test_split(train, target, test_size=0.3, random_st

print("No. of training samples : ",len(x_train))
print("No. of test samples : ",len(x_test))
```

```
No. of training samples :  199364
No. of test samples :  85443
```

In [21]:

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.naive_bayes import GaussianNB

classifierNB = GaussianNB()
classifierNB.fit(x_train, y_train)
classifierNB.score(x_test, y_test)

y_preds = classifierNB.predict(x_test)
print("Naive Bayes accuracy score: ",accuracy_score(y_test, y_preds))

# Confusion Matrix
cmx=confusion_matrix(y_test,y_preds)
print("\nNo. of test samples : ",len(x_test))
print("\n Confustion Matrix : \n",cmx)

print("\nPerfomance measures are: \n",classification_report(y_test, y_preds))
```

```
Naive Bayes accuracy score:  0.9925681448451014

No. of test samples :  85443

 Confustion Matrix :
 [[84717   585]
 [   50    91]]

Perfomance measures are:
               precision    recall  f1-score   support

           0       1.00      0.99      1.00     85302
           1       0.13      0.65      0.22       141

    accuracy                           0.99     85443
   macro avg       0.57      0.82      0.61     85443
weighted avg       1.00      0.99      0.99     85443
```

In [ ]:

```

```