```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df=pd.read_csv("/content/drive/MyDrive/Colab
Notebooks/creditcard.csv")
```

```
df.shape
```

```
(284807, 31)
```

```
df.head()
```

```
   Time        V1        V2        V3  ...       V27       V28  Amount
Class
0   0.0 -1.359807 -0.072781  2.536347  ...  0.133558 -0.021053  149.62
0
1   0.0  1.191857  0.266151  0.166480  ... -0.008983  0.014724    2.69
0
2   1.0 -1.358354 -1.340163  1.773209  ... -0.055353 -0.059752  378.66
0
3   1.0 -0.966272 -0.185226  1.792993  ...  0.062723  0.061458  123.50
0
4   2.0 -1.158233  0.877737  1.548718  ...  0.219422  0.215153   69.99
0

[5 rows x 31 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 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   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

```
df.describe()
```

```
                Time            V1  ...         Amount          Class
count  284807.000000  2.848070e+05  ...  284807.000000  284807.000000
mean    94813.859575  3.919560e-15  ...      88.349619       0.001727
std     47488.145955  1.958696e+00  ...     250.120109       0.041527
min         0.000000 -5.640751e+01  ...       0.000000       0.000000
25%     54201.500000 -9.203734e-01  ...       5.600000       0.000000
50%     84692.000000  1.810880e-02  ...      22.000000       0.000000
75%    139320.500000  1.315642e+00  ...      77.165000       0.000000
max    172792.000000  2.454930e+00  ...   25691.160000       1.000000

[8 rows x 31 columns]
```

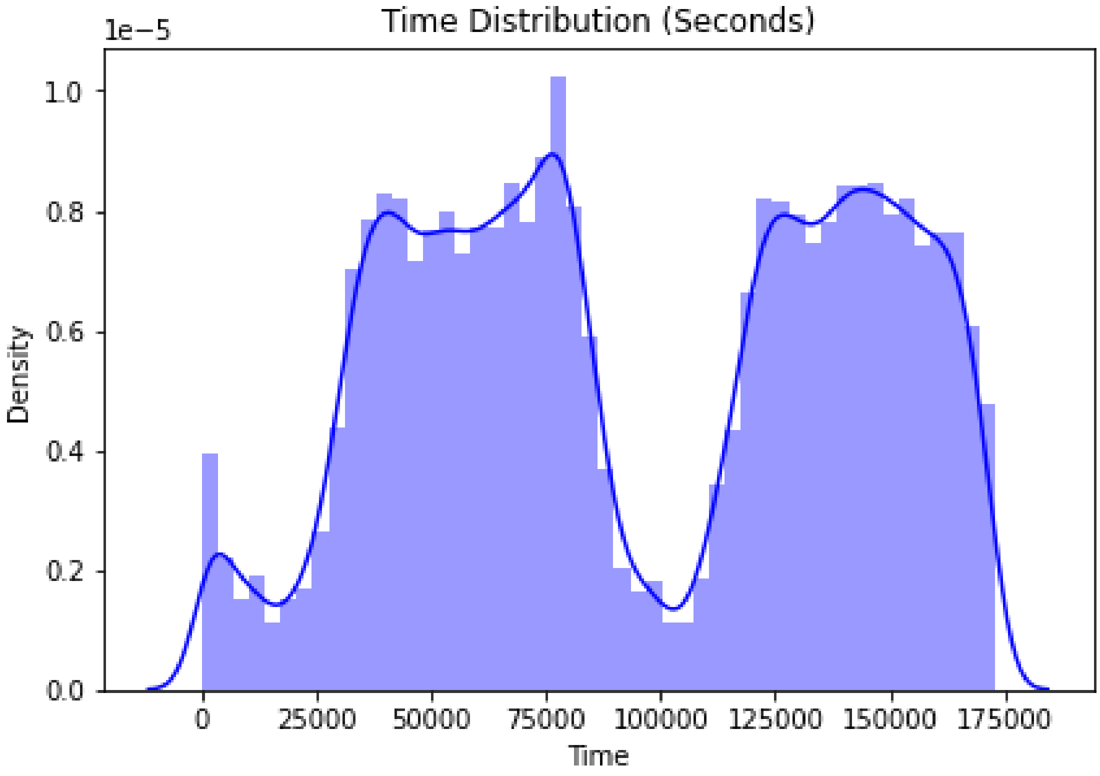
```
df.isnull().sum()
```

```
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
```

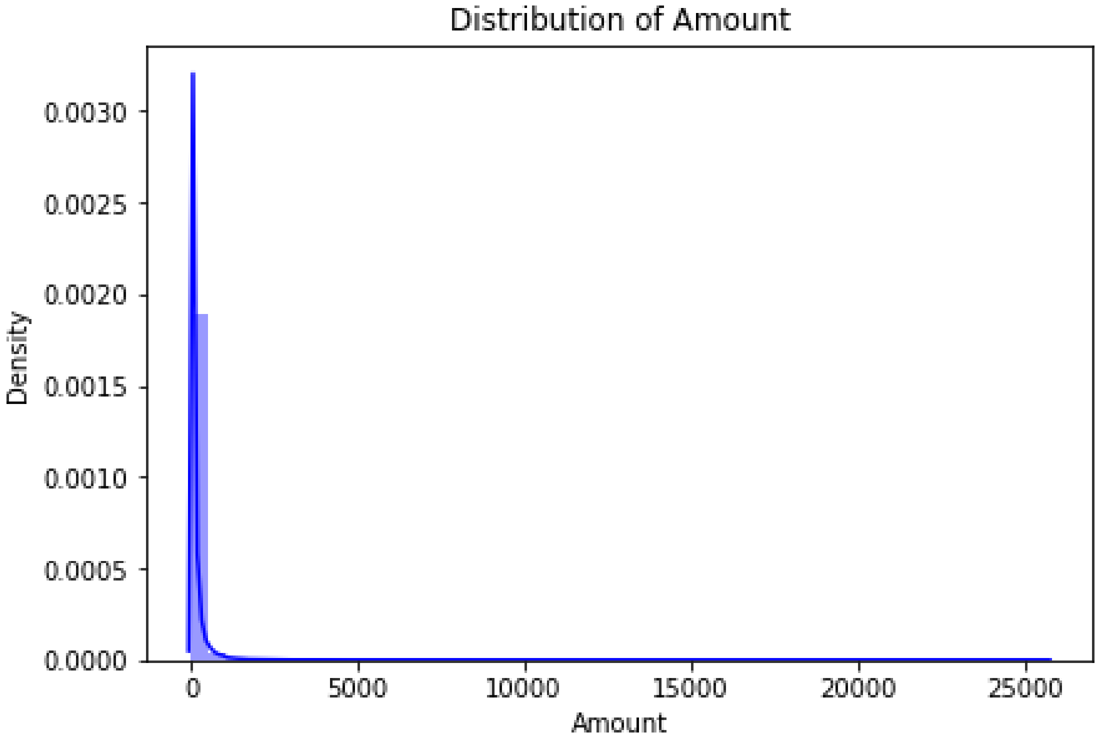
```
plt.figure(figsize=(15,10))
plt.subplot(2, 2, 1)
plt.title('Time Distribution (Seconds)')
sns.distplot(df['Time'], color='blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce0fa0fcd0>
```

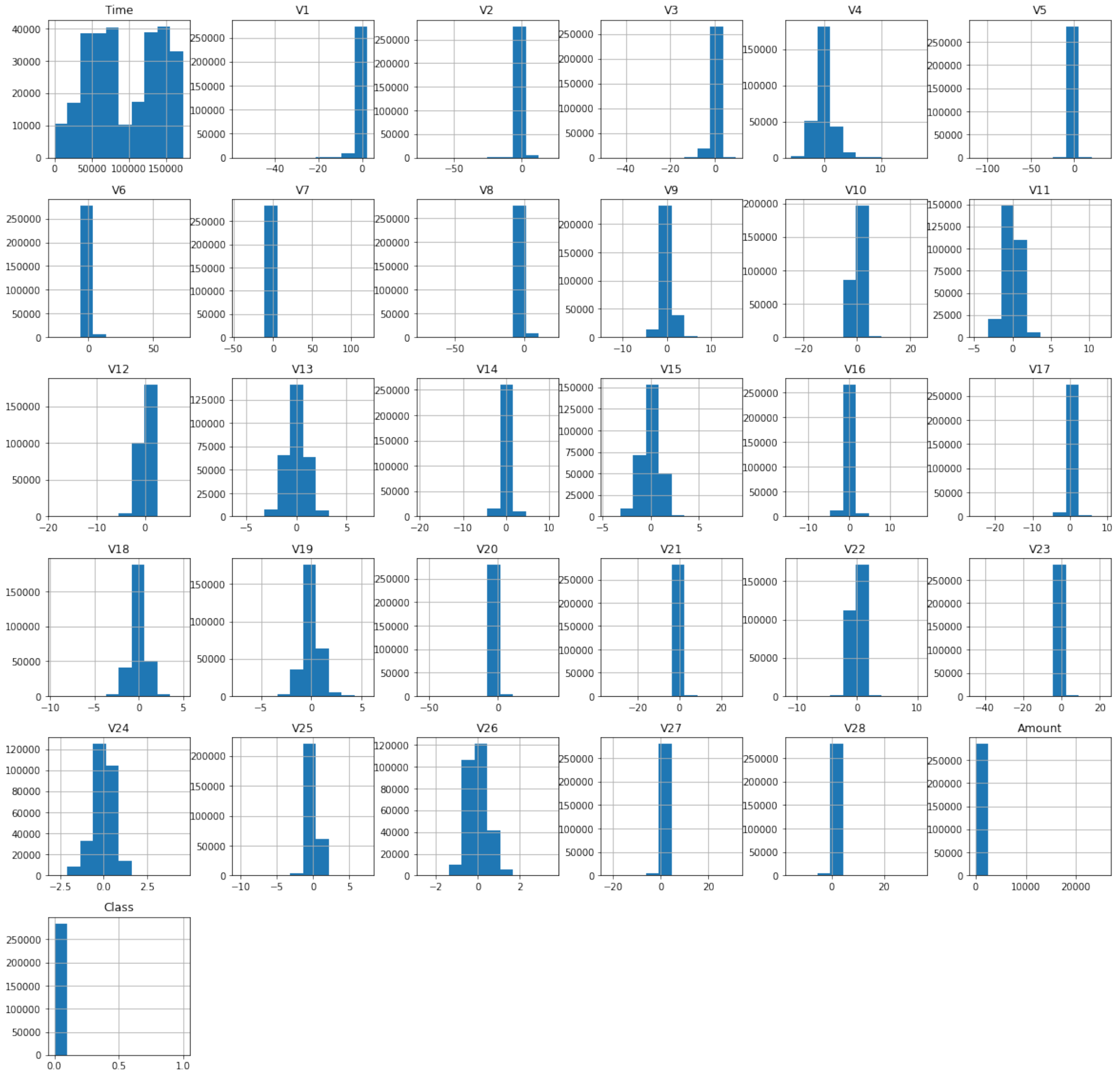


```
plt.figure(figsize=(15,10))
plt.subplot(2, 2, 2)
plt.title('Distribution of Amount')
sns.distplot(df['Amount'],color='blue')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce0662e310>
```

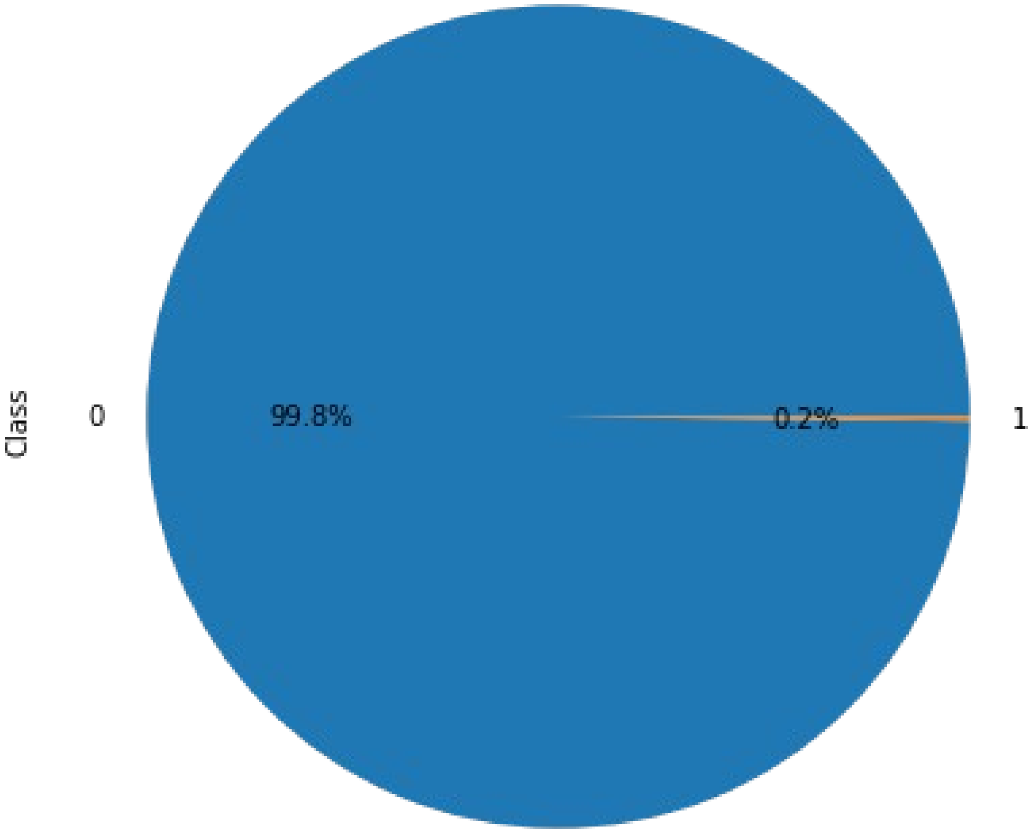


```
df.hist(figsize=(20, 20));
```



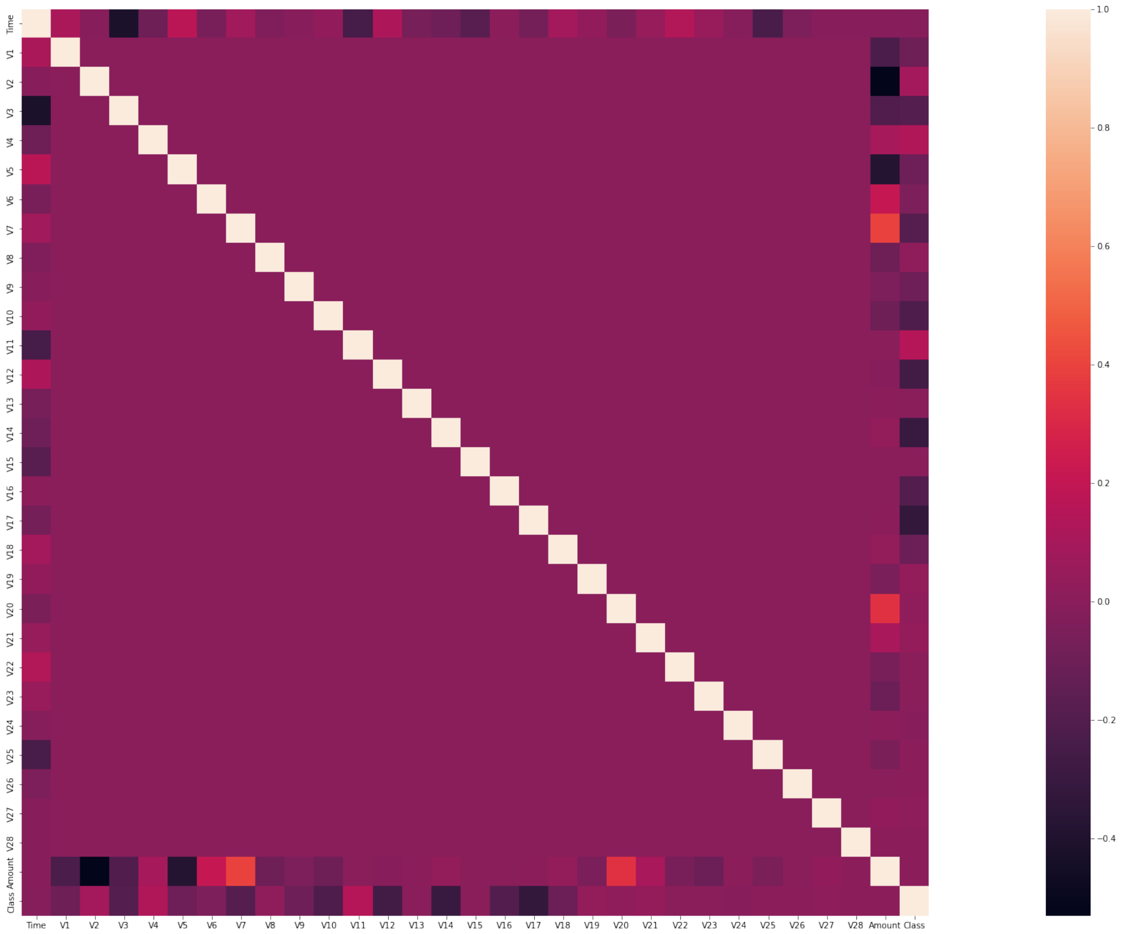
```
print(df["Class"].value_counts())
print("---------------------------------------------------")
plt.figure(figsize=(7,7))
df["Class"].value_counts().plot.pie(autopct="%.1f%%")
plt.show()
```

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



```
 ax = plt.subplots(figsize=(50, 20))
sns.heatmap(df.corr(), square=True,)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fce05095390>
```



```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

y = df["Class"]
X = df.drop("Class",axis=1)
X_train,X_test,y_train,y_test =
train_test_split(X,y,test_size=0.3,random_state=1,stratify=y)

ss = StandardScaler()
X_train_ss = ss.fit_transform(X_train)
X_test_ss = ss.transform(X_test)
```

BaseLine Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential()
model.add(Dense(16,activation="relu", input_dim=30))
model.add(Dense(16,activation="relu"))
model.add(Dense(16,activation="relu"))
model.add(Dense(1, activation="sigmoid"))
model.compile(optimizer="adam", loss="binary_crossentropy")
```

```
model.fit(X_train_ss,y_train,epochs=5, batch_size=8)
```

```
Epoch 1/5
24921/24921 [==============================] - 37s 1ms/step - loss:
0.0071
Epoch 2/5
24921/24921 [==============================] - 37s 1ms/step - loss:
0.0038
Epoch 3/5
24921/24921 [==============================] - 37s 1ms/step - loss:
0.0032
Epoch 4/5
24921/24921 [==============================] - 37s 1ms/step - loss:
0.0031
Epoch 5/5
24921/24921 [==============================] - 37s 1ms/step - loss:
0.0030
```

```
<keras.callbacks.History at 0x7fcdb73c4bd0>
```

```
# testing
y_pred= model.predict(X_test_ss)
```

```
y_pred = y_pred.argmax(axis=1)
```

```
from sklearn.metrics import classification_report
```

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

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.00      0.00      0.00       148

    accuracy                           1.00     85443
   macro avg       0.50      0.50      0.50     85443
weighted avg       1.00      1.00      1.00     85443
```

Using RandomOverSampler to Balanced the training data

```
from imblearn.over_sampling import RandomOverSampler
```

```
ros = RandomOverSampler(random_state=1)
```

```
X_sample1, y_sample1 = ros.fit_sample(X_train_ss,y_train)
```

```
pd.Series(y_sample1).value_counts()
```

```
1    199020
0    199020
dtype: int64
```

```
model1 = Sequential()
model1.add(Dense(16,activation="relu", input_dim=30))
model1.add(Dense(16,activation="relu"))
model1.add(Dense(16,activation="relu"))
model1.add(Dense(1, activation="sigmoid"))
```

```
model.compile(optimizer="adam", loss="binary_crossentropy")
```

```
model.fit(X_sample1, y_sample1,epochs=10, batch_size=8)
```

```
Epoch 1/10
49755/49755 [==============================] - 71s 1ms/step - loss:
0.0197
Epoch 2/10
49755/49755 [==============================] - 71s 1ms/step - loss:
0.0057
Epoch 3/10
49755/49755 [==============================] - 72s 1ms/step - loss:
0.0048
Epoch 4/10
49755/49755 [==============================] - 71s 1ms/step - loss:
0.0039
Epoch 5/10
49755/49755 [==============================] - 71s 1ms/step - loss:
0.0036
Epoch 6/10
49755/49755 [==============================] - 73s 1ms/step - loss:
0.0036
Epoch 7/10
49755/49755 [==============================] - 75s 2ms/step - loss:
0.0034
Epoch 8/10
49755/49755 [==============================] - 73s 1ms/step - loss:
0.0031
Epoch 9/10
49755/49755 [==============================] - 73s 1ms/step - loss:
0.0028
Epoch 10/10
49755/49755 [==============================] - 73s 1ms/step - loss:
0.0027
```

```
<keras.callbacks.History at 0x7fcdb743ab90>
```

```
# testing
y_pred1= model.predict(X_test_ss)
```

```
y_pred = np.where(y_pred1>= 0.5,1,0)
```

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

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85295
           1       0.64      0.89      0.74       148

    accuracy                           1.00     85443
   macro avg       0.82      0.94      0.87     85443
weighted avg       1.00      1.00      1.00     85443
```