In [1]:

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

In [2]:

```
data=pd.read_csv('creditcard.csv')
data
```

Out[2]:

	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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078

284807 rows × 31 columns

In [3]:

```
data.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
```

In [4]:

```
data.head()
```

Out[4]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	
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	-0.110474	
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	0.101288	
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	0.909412	
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	-0.190321	
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	-0.137458	

5 rows × 31 columns

In [5]:

```
data.describe()
```

Out[5]:

	Time	V1	V2	V3	V4	V5	V6	V7	
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927028
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	7.321672e+
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+

8 rows × 31 columns

In [6]:

```
print(data.shape)
```

```
(284807, 31)
```

In [7]:

```
data=data.sample(frac=0.1, random_state=1)
data
```

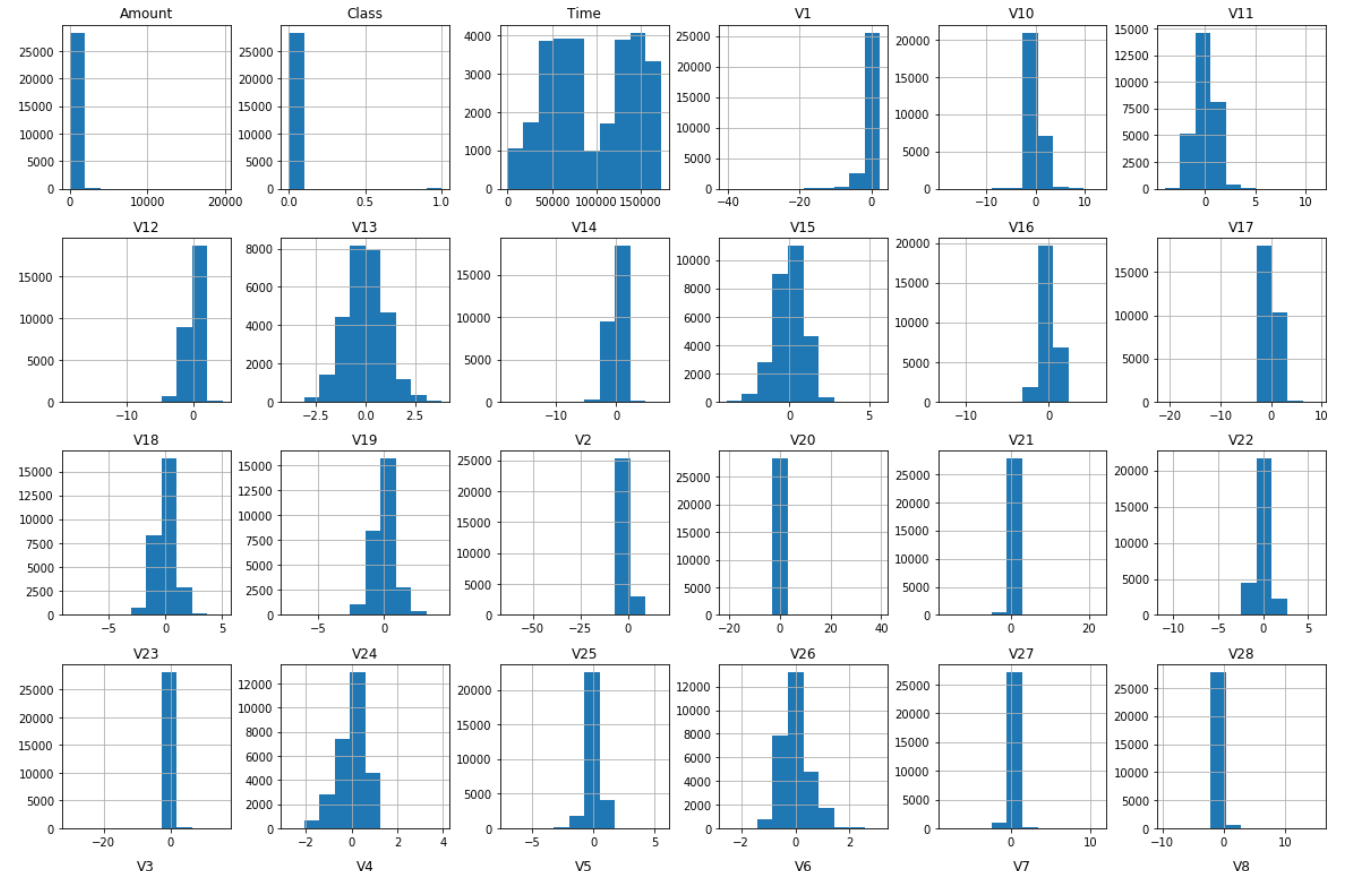
Out[7]:

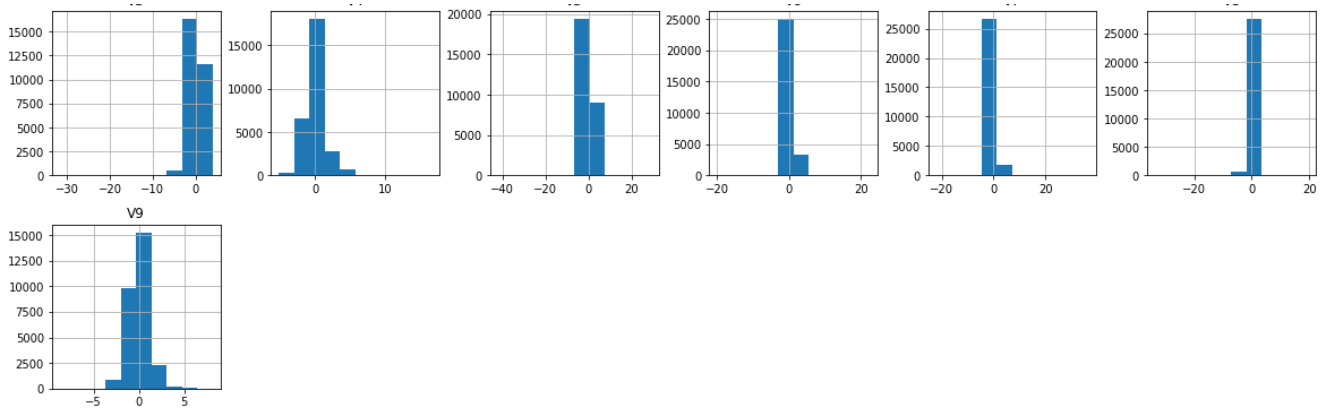
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	
169876	119907.0	-0.611712	-0.769705	-0.149759	-0.224877	2.028577	-2.019887	0.292491	-0.523020	0.358468	...	-0.075208	0.045536	0.
127467	78340.0	-0.814682	1.319219	1.329415	0.027273	-0.284871	-0.653985	0.321552	0.435975	-0.704298	...	-0.128619	-0.368565	0.
137900	82382.0	-0.318193	1.118618	0.969864	-0.127052	0.569563	-0.532484	0.706252	-0.064966	-0.463271	...	-0.305402	-0.774704	0.
21513	31717.0	-1.328271	1.018378	1.775426	-1.574193	-0.117696	-0.457733	0.681867	-0.031641	-0.383872	...	-0.220815	-0.419013	0.
134700	80923.0	1.276712	0.617120	-0.578014	0.879173	0.061706	-1.472002	0.373692	-0.287204	-0.084482	...	-0.160161	-0.430404	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2032	1574.0	-0.615776	0.654356	2.618793	0.857434	-0.487340	0.593957	-0.095191	0.426786	0.011607	...	0.010440	0.113631	0.
240932	150813.0	-3.517229	3.326821	-3.590262	0.674769	-0.679266	-0.469516	-1.135362	2.778095	-2.404956	...	0.455767	0.388102	0.
3701	3169.0	-0.315540	1.054303	1.484711	1.138262	0.394713	-0.168883	0.737923	-0.061284	-0.952381	...	0.005626	0.094740	0.
153365	98752.0	-3.580417	4.100916	-2.577720	-1.476718	-0.006201	-2.008418	0.887262	0.304192	-2.879710	...	-0.194866	0.571678	0.
97365	66187.0	1.213349	0.227172	-0.886860	1.345683	2.254592	3.788565	-0.521816	0.891366	-0.776104	...	-0.102366	0.116553	0.

28481 rows × 31 columns

In [8]:

```
#plot histogram of each parameter
data.hist(figsize=(20,20))
plt.show()
```





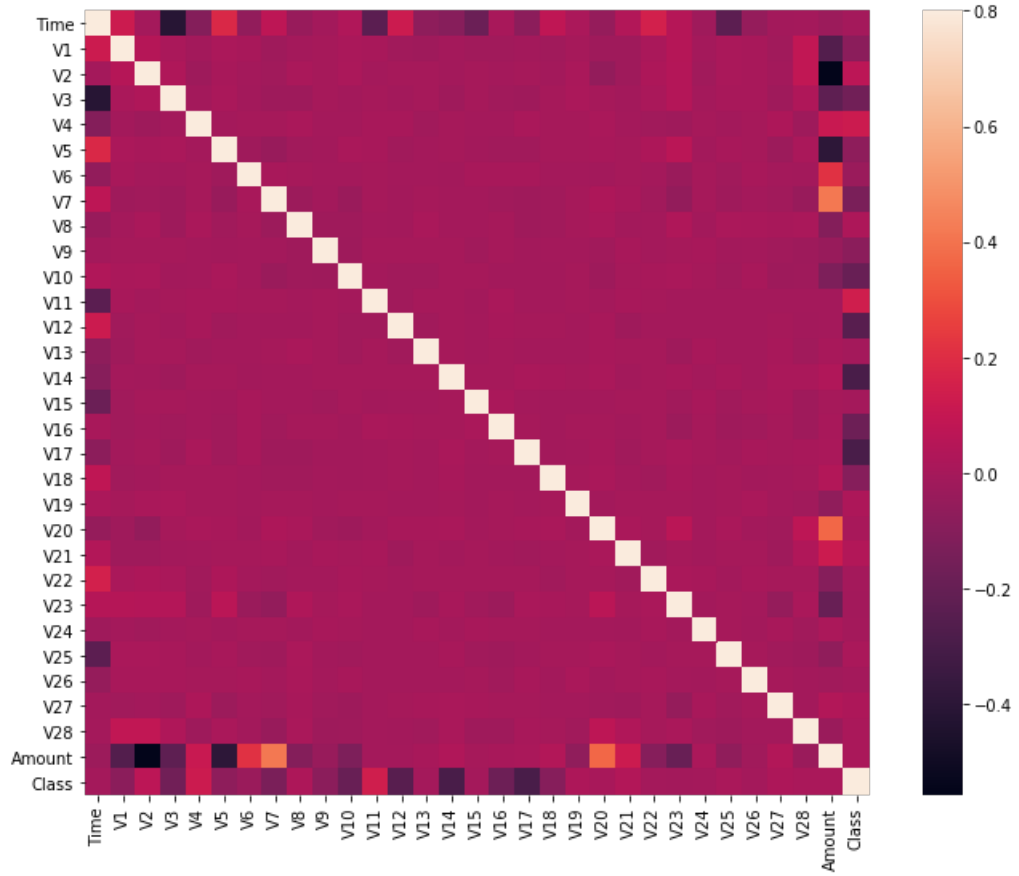
In [9]:

```
#deteremining the fraud cases in dataset
fr=data[data['Class']==1]
valid=data[data['Class']==0]

outlier_fraction=len(fr)/float(len(valid))
print(outlier_fraction)
print('fraud cases:{}'.format(len(fr)))
print('valid cases:{}'.format(len(valid)))
```

```
0.0017234102419808666
fraud cases:49
valid cases:28432
```

In [10]:

```
#corelation matrix
cr=data.corr()
fig=plt.figure(figsize=(12,9))
sns.heatmap(cr,vmax=.8,square=True)
plt.show()
```



In [11]:

```
data.isnull().any()
```

Out[11]:

```
Time      False
V1        False
V2        False
V3        False
V4        False
V5        False
V6        False
V7        False
V8        False
V9        False
V10       False
V11       False
V12       False
V13       False
V14       False
V15       False
V16       False
V17       False
V18       False
V19       False
V20       False
V21       False
V22       False
V23       False
V24       False
V25       False
V26       False
V27       False
V28       False
Amount    False
Class     False
dtype: bool
```

feature scaling

In [12]:

```
from sklearn.preprocessing import StandardScaler
data['normalizedAmount']=StandardScaler().fit_transform(data['Amount'].values.reshape(-1,1))
data=data.drop(['Amount'],axis=1)
data=data.drop(['Time'],axis=1)
data.head()
```

Out[12]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V21	V22	
169876	-0.611712	-0.769705	-0.149759	-0.224877	2.028577	-2.019887	0.292491	-0.523020	0.358468	0.070050	...	-0.075208	0.045536	0.
127467	-0.814682	1.319219	1.329415	0.027273	-0.284871	-0.653985	0.321552	0.435975	-0.704298	-0.600684	...	-0.128619	-0.368565	0.
137900	-0.318193	1.118618	0.969864	-0.127052	0.569563	-0.532484	0.706252	-0.064966	0.463271	-0.528357	...	-0.305402	-0.774704	0.
21513	-1.328271	1.018378	1.775426	-1.574193	-0.117696	-0.457733	0.681867	-0.031641	0.383872	0.334853	...	-0.220815	-0.419013	0.
134700	1.276712	0.617120	-0.578014	0.879173	0.061706	-1.472002	0.373692	-0.287204	-0.084482	-0.696578	...	-0.160161	-0.430404	0.

5 rows × 30 columns

In [13]:

```
X = data.iloc[:, data.columns != 'Class']
y = data.iloc[:, data.columns == 'Class']
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28481 entries, 169876 to 97365
Data columns (total 29 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   V1                28481 non-null  float64
 1   V2                28481 non-null  float64
 2   V3                28481 non-null  float64
 3   V4                28481 non-null  float64
 4   V5                28481 non-null  float64
 5   V6                28481 non-null  float64
 6   V7                28481 non-null  float64
 7   V8                28481 non-null  float64
 8   V9                28481 non-null  float64
 9   V10               28481 non-null  float64
 10  V11               28481 non-null  float64
 11  V12               28481 non-null  float64
 12  V13               28481 non-null  float64
 13  V14               28481 non-null  float64
 14  V15               28481 non-null  float64
 15  V16               28481 non-null  float64
 16  V17               28481 non-null  float64
 17  V18               28481 non-null  float64
 18  V19               28481 non-null  float64
 19  V20               28481 non-null  float64
 20  V21               28481 non-null  float64
 21  V22               28481 non-null  float64
 22  V23               28481 non-null  float64
 23  V24               28481 non-null  float64
 24  V25               28481 non-null  float64
 25  V26               28481 non-null  float64
 26  V27               28481 non-null  float64
 27  V28               28481 non-null  float64
 28  normalizedAmount  28481 non-null  float64
dtypes: float64(29)
memory usage: 6.5 MB
```

In [14]:

```
y.head()
```

Out[14]:

	Class
169876	0
127467	0
137900	0
21513	0
134700	0

Model training

In [15]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_state=0)
```

In [16]:

```
# Importing the Keras libraries and packages
import keras
from keras.models import Sequential
from keras.layers import Dense

# Initialising the ANN
classifier = Sequential()
```

```
# Adding the input layer and the first hidden layer
classifier.add(Dense(units =15 , kernel_initializer = 'uniform', activation = 'relu', input_dim = 2
9))

# Adding the second hidden layer
classifier.add(Dense(units = 15, kernel_initializer = 'uniform', activation = 'relu'))

# Adding the output layer
classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))

# Compiling the ANN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Fitting the ANN to the Training set
classifier.fit(X_train, y_train, batch_size = 32, epochs = 100)
```

```
Using TensorFlow backend.
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
H:\phython\swaroop\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Pas
sing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it wil
l be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

```
WARNING:tensorflow:From H:\phython\swaroop\lib\site-
packages\keras\backend\tensorflow_backend.py:74: The name tf.get_default_graph is deprecated. Plea
se use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From H:\phython\swaroop\lib\site-
packages\keras\backend\tensorflow_backend.py:517: The name tf.placeholder is deprecated. Please us
e tf.compat.v1.placeholder instead.

WARNING:tensorflow:From H:\phython\swaroop\lib\site-
packages\keras\backend\tensorflow_backend.py:4138: The name tf.random_uniform is deprecated. Pleas
e use tf.random.uniform instead.
```

```
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:543:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:544:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:545:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
```

```
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
H:\phython\swaroop\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:550:
FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future
version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])

WARNING:tensorflow:From H:\phython\swaroop\lib\site-packages\keras\optimizers.py:790: The name tf.
train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From H:\phython\swaroop\lib\site-
packages\keras\backend\tensorflow_backend.py:3376: The name tf.log is deprecated. Please use tf.ma
th.log instead.

WARNING:tensorflow:From H:\phython\swaroop\lib\site-packages\tensorflow\python\ops\nn_impl.py:180:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and
will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From H:\phython\swaroop\lib\site-
packages\keras\backend\tensorflow_backend.py:986: The name tf.assign_add is deprecated. Please use
tf.compat.v1.assign_add instead.

Epoch 1/100
19936/19936 [==============================] - 1s 40us/step - loss: 0.1190 - acc: 0.9977
Epoch 2/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0056 - acc: 0.9981
Epoch 3/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0049 - acc: 0.9981
Epoch 4/100
19936/19936 [==============================] - 1s 30us/step - loss: 0.0045 - acc: 0.9981
Epoch 5/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0042 - acc: 0.9981
Epoch 6/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0040 - acc: 0.9981
Epoch 7/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0036 - acc: 0.9985
Epoch 8/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0032 - acc: 0.9992
Epoch 9/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0030 - acc: 0.9993
Epoch 10/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0025 - acc: 0.9993
Epoch 11/100
19936/19936 [==============================] - ETA: 0s - loss: 0.0025 - acc: 0.9993000 - ETA: 0s -
loss: 0.0011 - acc: 0.999 - ETA: 0s - loss: 0.0031 - acc:  - 1s 30us/step - loss: 0.0024 - acc: 0.
9993
Epoch 12/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0023 - acc: 0.9993
Epoch 13/100
19936/19936 [==============================] - 1s 30us/step - loss: 0.0020 - acc: 0.9994
Epoch 14/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0025 - acc: 0.9994 ETA: 0s -
loss: 0.0021 - acc:
Epoch 15/100
19936/19936 [==============================] - 1s 29us/step - loss: 0.0019 - acc: 0.9993
Epoch 16/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0019 - acc: 0.9994
Epoch 17/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0017 - acc: 0.9994 ETA: 0s -
loss: 0.0015 - acc
Epoch 18/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0015 - acc: 0.9995
Epoch 19/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0019 - acc: 0.9995
Epoch 20/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0015 - acc: 0.9994: 0s - los
s: 0.0017 - acc: 0.
Epoch 21/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0019 - acc: 0.9993
Epoch 22/100
19936/19936 [==============================] - ETA: 0s - loss: 0.0015 - acc: 0.9994 - 1s 27us/step
- loss: 0.0014 - acc: 0.9994
Epoch 23/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0016 - acc: 0.9996
Epoch 24/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0016 - acc: 0.9995
Epoch 25/100
```

```
Epoch 25/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0012 - acc: 0.9995
Epoch 26/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.9373e-04 - acc: 0.9996
Epoch 27/100
19936/19936 [==============================] - 1s 30us/step - loss: 0.0011 - acc: 0.9994
Epoch 28/100
19936/19936 [==============================] - 1s 26us/step - loss: 8.4527e-04 - acc: 0.9997
Epoch 29/100
19936/19936 [==============================] - 1s 27us/step - loss: 6.8399e-04 - acc: 0.9998
Epoch 30/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0013 - acc: 0.9996
Epoch 31/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.4279e-04 - acc: 0.9996
Epoch 32/100
19936/19936 [==============================] - 1s 27us/step - loss: 6.0997e-04 - acc: 0.9998
Epoch 33/100
19936/19936 [==============================] - 1s 27us/step - loss: 6.9306e-04 - acc: 0.9998
Epoch 34/100
19936/19936 [==============================] - 1s 27us/step - loss: 5.3092e-04 - acc: 0.9999: 0s -
loss: 5.6752e-04 - acc: 0.99
Epoch 35/100
19936/19936 [==============================] - 1s 26us/step - loss: 5.4192e-04 - acc: 0.9999
Epoch 36/100
19936/19936 [==============================] - 1s 27us/step - loss: 4.7222e-04 - acc: 0.9998
Epoch 37/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.4212e-04 - acc: 0.9998: 0s -
loss: 0.0015 - acc: 0 - ETA: 0s - loss: 7.5615e-04 - acc: 0.9
Epoch 38/100
19936/19936 [==============================] - ETA: 0s - loss: 0.0014 - acc: 0.9997    - ETA: 0s -
loss: 0.0028 - acc:  - 1s 27us/step - loss: 0.0013 - acc: 0.9997
Epoch 39/100
19936/19936 [==============================] - 1s 27us/step - loss: 4.1321e-04 - acc: 0.9998: 0s -
loss: 4.5331e-04 - a
Epoch 40/100
19936/19936 [==============================] - 1s 27us/step - loss: 6.0333e-04 - acc: 0.9997
Epoch 41/100
19936/19936 [==============================] - 1s 27us/step - loss: 4.4012e-04 - acc: 0.9999: 0s -
loss: 4.9897e-04 - acc:
Epoch 42/100
19936/19936 [==============================] - 1s 26us/step - loss: 5.7819e-04 - acc: 0.9998
Epoch 43/100
19936/19936 [==============================] - 1s 27us/step - loss: 7.1007e-04 - acc: 0.9997: 0s -
loss: 4.5832e-04 - acc: 0.999
Epoch 44/100
19936/19936 [==============================] - 1s 26us/step - loss: 0.0012 - acc: 0.9997 ETA: 0s -
loss: 2.0620e-04 - ac
Epoch 45/100
19936/19936 [==============================] - 1s 27us/step - loss: 3.8218e-04 - acc: 0.9998
Epoch 46/100
19936/19936 [==============================] - ETA: 0s - loss: 3.5019e-04 - acc: 0.999 - 1s
27us/step - loss: 3.3518e-04 - acc: 0.9999
Epoch 47/100
19936/19936 [==============================] - 1s 27us/step - loss: 4.7584e-04 - acc: 0.9998: 0s -
loss: 3.9476e-04 - acc: 0.
Epoch 48/100
19936/19936 [==============================] - 1s 28us/step - loss: 9.9795e-04 - acc: 0.9999
Epoch 49/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.4782e-04 - acc: 0.9999
Epoch 50/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0012 - acc: 0.9998
Epoch 51/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.2754e-04 - acc: 0.9999
Epoch 52/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.4829e-04 - acc: 0.9999
Epoch 53/100
19936/19936 [==============================] - 1s 26us/step - loss: 8.9054e-04 - acc: 0.9999
Epoch 54/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.7154e-04 - acc: 0.9999
Epoch 55/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.5663e-04 - acc: 0.9999
Epoch 56/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.4131e-04 - acc: 0.9999
Epoch 57/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.5717e-04 - acc: 0.9999
Epoch 58/100
19936/19936 [==============================] - ETA: 0s - loss: 8.6594e-04 - acc: 0.999 - 1s
27us/step - loss: 8.3300e-04 - acc: 0.9999
```

```
27us/step - loss: 8.9906e-04 - acc: 0.9999
Epoch 59/100
19936/19936 [==============================] - 1s 29us/step - loss: 8.6031e-04 - acc: 0.9999: 0s -
loss: 4.3307e-05 - ac
Epoch 60/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0013 - acc: 0.9998
Epoch 61/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0011 - acc: 0.9998A: 0s - lo
ss: 0.0014 - acc: 0.9
Epoch 62/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.2051e-04 - acc: 0.9999
Epoch 63/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.1886e-04 - acc: 0.9999
Epoch 64/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.1776e-04 - acc: 0.9999
Epoch 65/100
19936/19936 [==============================] - 1s 26us/step - loss: 8.1723e-04 - acc: 0.9999
Epoch 66/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.1647e-04 - acc: 0.9999: 0s -
loss: 5.2478e-06 - acc: 1.0 - ETA: 0s - loss: 0.0015 - acc: 0
Epoch 67/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.1636e-04 - acc: 0.9999: 0s -
loss: 6.5559e-06 - ac
Epoch 68/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.1562e-04 - acc: 0.9999
Epoch 69/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.8529e-04 - acc: 0.9999
Epoch 70/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.2870e-04 - acc: 0.9998
Epoch 71/100
19936/19936 [==============================] - 1s 27us/step - loss: 9.5749e-04 - acc: 0.9999
Epoch 72/100
19936/19936 [==============================] - 1s 27us/step - loss: 0.0012 - acc: 0.9999
Epoch 73/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.1342e-04 - acc: 0.9999: 0s -
loss: 7.1695e-06 - acc: 1.000 - ETA: 0s - loss: 4.8132e-06 - ac
Epoch 74/100
19936/19936 [==============================] - ETA: 0s - loss: 8.5273e-04 - acc: 0.999 - 1s
28us/step - loss: 8.1309e-04 - acc: 0.9999
Epoch 75/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.1281e-04 - acc: 0.9999
Epoch 76/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1195e-04 - acc: 0.9999
Epoch 77/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1238e-04 - acc: 0.9999
Epoch 78/100
19936/19936 [==============================] - 1s 25us/step - loss: 8.2440e-04 - acc: 0.9999: 0s -
loss: 7.7188e-06 - ac
Epoch 79/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.8718e-04 - acc: 0.9999
Epoch 80/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1958e-04 - acc: 0.9999
Epoch 81/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1181e-04 - acc: 0.9999
Epoch 82/100
19936/19936 [==============================] - 1s 25us/step - loss: 8.6002e-04 - acc: 0.9999: 0s -
loss: 2.4128e-06 - acc: 1.000 - ETA: 0s - loss: 4.8697e-06 - acc
Epoch 83/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1105e-04 - acc: 0.9999
Epoch 84/100
19936/19936 [==============================] - 1s 25us/step - loss: 8.1050e-04 - acc: 0.9999: 0s -
loss: 0.0040 - acc
Epoch 85/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.1026e-04 - acc: 0.9999
Epoch 86/100
19936/19936 [==============================] - 1s 25us/step - loss: 8.1007e-04 - acc: 0.9999
Epoch 87/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.0992e-04 - acc: 0.9999
Epoch 88/100
19936/19936 [==============================] - 0s 25us/step - loss: 8.0979e-04 - acc: 0.9999
Epoch 89/100
19936/19936 [==============================] - 1s 26us/step - loss: 8.0967e-04 - acc: 0.99990s - l
oss: 8.9734e-04 - acc: 0.99
Epoch 90/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0954e-04 - acc: 0.9999
Epoch 91/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0941e-04 - acc: 0.9999: 0s -
loss: 9.2202e-07 - acc:
```

```
loss: 9.2202e-07 - acc:
Epoch 92/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0935e-04 - acc: 0.9999
Epoch 93/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0915e-04 - acc: 0.9999
Epoch 94/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0902e-04 - acc: 0.9999
Epoch 95/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0900e-04 - acc: 0.9999
Epoch 96/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0889e-04 - acc: 0.9999
Epoch 97/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0982e-04 - acc: 0.9999
Epoch 98/100
19936/19936 [==============================] - 1s 28us/step - loss: 0.0018 - acc: 0.9998
Epoch 99/100
19936/19936 [==============================] - 1s 28us/step - loss: 8.0886e-04 - acc: 0.9999
Epoch 100/100
19936/19936 [==============================] - 1s 27us/step - loss: 8.0883e-04 - acc: 0.9999: 0s -
loss: 1.4849e-07 - ac
```

Out[16]:

```
<keras.callbacks.History at 0x1b6acae3f88>
```

In [17]:

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)
```

In [18]:

```
score=classifier.evaluate(X_test,y_test)
score
```

```
8545/8545 [==============================] - 0s 16us/step
```

Out[18]:

```
[0.004387019985230976, 0.9995318899941487]
```

In [19]:

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

Out[19]:

```
array([[8532,    1],
       [   3,    9]], dtype=int64)
```

In [20]:

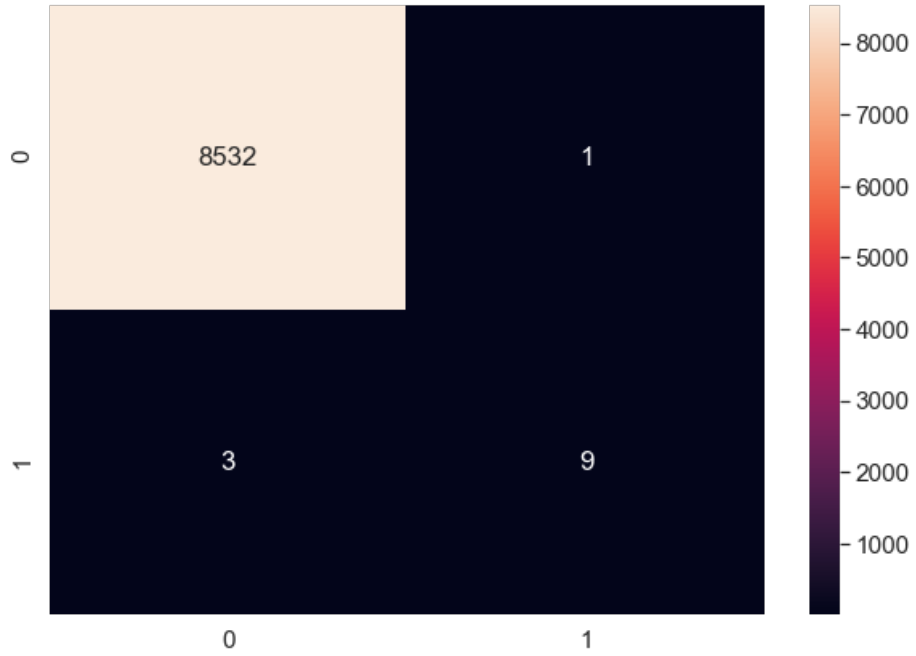
```
#Let's see how our model performed
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      8533
           1       0.90      0.75      0.82        12

    accuracy                           1.00      8545
   macro avg       0.95      0.87      0.91      8545
weighted avg       1.00      1.00      1.00      8545
```

In [21]:

```
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test, y_pred) # rows = truth, cols = prediction
df_cm = pd.DataFrame(cm, index = (0, 1), columns = (0, 1))
plt.figure(figsize = (10,7))
sns.set(font_scale=1.4)
sns.heatmap(df_cm, annot=True, fmt='g')
print("Test Data Accuracy: %0.4f" % accuracy_score(y_test, y_pred))
```

```
Test Data Accuracy: 0.9995
```



In [ ]:

```

```

In [ ]:

```

```