jupyter Bank Note Authentication 1 Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [1]:
```
import pandas as pd
import numpy as np
```

In [3]:
```
df=pd.read_csv(r"C:\Users\pc\Downloads\BankNote_Authentication.csv")
```

In [4]:
```
df
```

Out[4]:

	variance	skewness	curtosis	entropy	class
0	3.62160	8.66610	-2.8073	-0.44699	0
1	4.54590	8.16740	-2.4586	-1.46210	0
2	3.86600	-2.63830	1.9242	0.10645	0
3	3.45660	9.52280	-4.0112	-3.59440	0
4	0.32924	-4.45520	4.5718	-0.98880	0
...	...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949	1
1368	-1.38870	-4.87730	6.4774	0.34179	1
1369	-3.75030	-13.45860	17.5932	-2.77710	1
1370	-3.56370	-8.38270	12.3930	-1.28230	1
1371	-2.54190	-0.65804	2.6842	1.19520	1

1372 rows × 5 columns

In [5]:
```
from sklearn.model_selection import train_test_split
```

In [6]:
```
df.columns
```

File Edit View Insert Cell Kernel Widgets Help Trusted

Run Code

1372 rows × 5 columns

In [5]:
```
from sklearn.model_selection import train_test_split
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['variance', 'skewness', 'curtosis', 'entropy', 'class'], dtype='object')
```

In [8]:
```
D=df.drop(['variance', 'skewness', 'curtosis', 'entropy'],axis=1)
```

In [9]:
```
D
```

Out[9]:

	class
0	0
1	0
2	0
3	0
4	0
...	...
1367	1
1368	1
1369	1
1370	1
1371	1

1372 rows × 1 columns

In [10]:
```
D=D.values
```

jupyter Bank Note Authentication 1 Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [10]:
```
D=D.values
```

In [11]:
```
len(D)
```

Out[11]: 1372

In [13]:
```
I=df.drop(['class'],axis=1)
```

In [14]:
```
I
```

Out[14]:

	variance	skewness	curtosis	entropy
0	3.62160	8.66610	-2.8073	-0.44699
1	4.54590	8.16740	-2.4586	-1.46210
2	3.86600	-2.63830	1.9242	0.10645
3	3.45660	9.52280	-4.0112	-3.59440
4	0.32924	-4.45520	4.5718	-0.98880
...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949
1368	-1.38870	-4.87730	6.4774	0.34179
1369	-3.75030	-13.45860	17.5932	-2.77710
1370	-3.56370	-8.38270	12.3930	-1.28230
1371	-2.54190	-0.65804	2.6842	1.19520

1372 rows × 4 columns

In [15]:
```
I=I.values
```

In [16]:
```
I_train,I_test,D_train,D_test=train_test_split(I,D,test_size=0.2,random_state=0)
```

jupyter Bank Note Authentication 1 Last Checkpoint: a day ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [16]:
```
I_train,I_test,D_train,D_test=train_test_split(I,D,test_size=0.2,random_state=0)
```

In [17]:
```
I_train
```

Out[17]:
```
array([[-1.2943  ,  2.6735  , -0.84085 , -2.0323  ],
       [-0.40857 ,  3.0977  , -2.9607  , -2.6892  ],
       [-1.5228  , -6.4789  ,  5.7568  ,  0.87325 ],
       ...,
       [ 0.6005  ,  0.99945 , -2.2126  ,  0.097399],
       [ 2.0165  , -0.25246 ,  5.1707  ,  1.0763  ],
       [-2.0759  , 10.8223  ,  2.6439  , -4.837   ]])
```

In [18]:
```
I_test
```

Out[18]:
```
array([[ -1.7713  , -10.7665  ,  10.2184  ,  -1.0043  ],
       [  5.1321  ,  -0.031048,   0.32616 ,   1.1151  ],
       [ -2.0149  ,   3.6874  ,  -1.9385  ,  -3.8918  ],
       ...,
       [ -2.1652  ,   3.0211  ,  -2.4132  ,  -2.4241  ],
       [  0.48797 ,   3.5674  ,  -4.3882  ,  -3.8116  ],
       [ -4.3667  ,   6.0692  ,   0.57208 ,  -5.4668  ]])
```

In [19]:
```
D_train
```

Out[19]:
```
array([[1],
       [1],
       [1],
       ...,
       [1],
       [0],
       [0]], dtype=int64)
```

In [20]:
```
D_test
```

jupyter Bank Note Authentication 1 Last Checkpoint: a day ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
[1],
[0],
[0],
```

In [21]:
```
from sklearn.svm import SVC
s=SVC(kernel='linear')
s.fit(I_train,D_train)
```

```
C:\Users\pc\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(**kwargs)
```

Out[21]:
```
SVC(kernel='linear')
```

In [22]:
```
p=s.predict(I_test)
```

In [23]:
```
df1=pd.DataFrame({'ACTUAL':D_test.flatten(),'PREDICTED':p})
df1
```

Out[23]:

	ACTUAL	PREDICTED
0	1	1
1	0	0
2	1	1
3	0	0
4	0	0
...	...	...
270	1	1
271	0	0
272	1	1
273	1	1

localhost:8888/notebooks/Bank%20Note%20Authentication%201.ipynb

jupyter Bank Note Authentication 1 Last Checkpoint: a day ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

275 rows × 2 columns

In [24]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
confusion_matrix(D_test,p)
```

Out[24]:
```
array([[155,   2],
       [  0, 118]], dtype=int64)
```

In [25]:
```
accuracy_score(D_test,p)
```

Out[25]:
```
0.9927272727272727
```

In [26]:
```
x=df1.ACTUAL
y=df1.PREDICTED
import matplotlib.pyplot as plt
plt.scatter(np.arange(275),x)
plt.scatter(np.arange(275),y)
```

Out[26]:
```
<matplotlib.collections.PathCollection at 0x24cabe28d90>
```