Chapter 17 - Exercise 2: Bill Authentication

In [1]:
```
# from google.colab import drive
# drive.mount("/content/gdrive", force_remount=True)
# %cd '/content/gdrive/My Drive/LDS6_MachineLearning/practice/Chapter17_LLE/'
```

Cho dữ liệu bill_authentication.csv

Áp dụng thuật toán LLE để trực quan hóa dữ liệu với 2 (và 3 thành phần) thay vì 4 thành phần

Sau khi giảm chiều (còn 2 hoặc 3 chiều) => áp dụng thuật toán ML cơ sở => kiểm tra kết quả => so sánh với khi chưa giảm chiều => quyết định có giảm chiều hay không???

In [2]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn import svm
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
```

In [3]:
```
bankdata = pd.read_csv("bill_authentication.csv")
```

In [4]:
```
bankdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1372 entries, 0 to 1371
Data columns (total 5 columns):
Variance    1372 non-null float64
Skewness    1372 non-null float64
Curtosis    1372 non-null float64
Entropy     1372 non-null float64
Class       1372 non-null int64
dtypes: float64(4), int64(1)
memory usage: 53.7 KB
```

In [5]:
```
# Class: có giá trị là 0 và 1
X = bankdata[["Variance", "Skewness","Curtosis","Entropy"]]
y = bankdata["Class"]
```

In [6]:
```
X.head(3)
```

Out[6]:

	Variance	Skewness	Curtosis	Entropy
0	3.6216	8.6661	-2.8073	-0.44699
1	4.5459	8.1674	-2.4586	-1.46210
2	3.8660	-2.6383	1.9242	0.10645

In [7]:
```
y.head(3)
```

Out[7]:
```
0    0
1    0
2    0
Name: Class, dtype: int64
```

In [8]:
```
X = np.asarray(X)
```

Trực quan hóa dữ liệu với LLE - 2 components

In [9]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
# Fit on training set only.
scaler.fit(X)
# Apply transform to both the training set and the test set.
X = scaler.transform(X)
```

In [10]:
```
from sklearn.manifold import LocallyLinearEmbedding
lle = LocallyLinearEmbedding(n_components=2, n_neighbors=10)
```

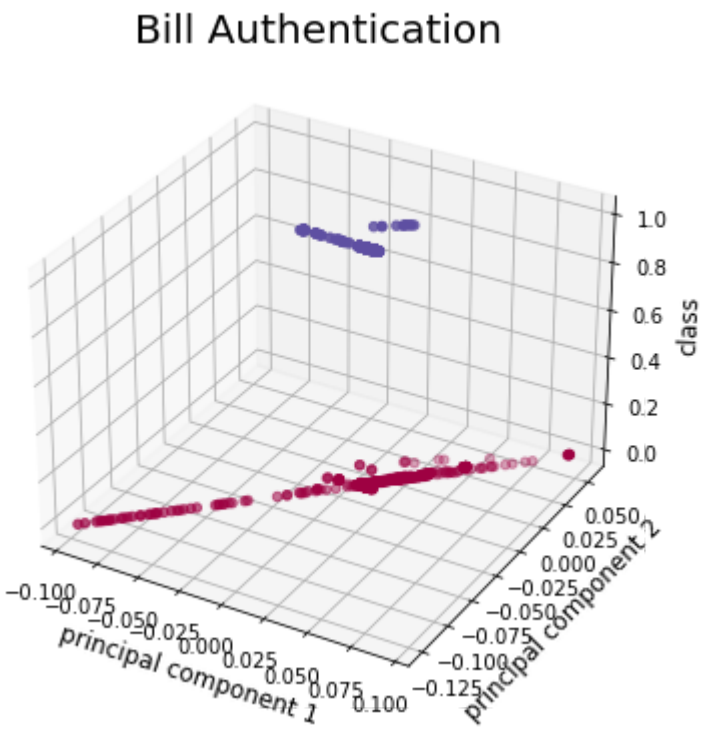
In [11]:
```
X_reduced = lle.fit_transform(X)
```

In [12]:
```
X_reduced[:2]
```

Out[12]:
```
array([[ 0.03513756, -0.00010805],
       [ 0.03471435, -0.00202771]])
```

In [13]:
```
import numpy as np
types = np.reshape(y.values, -1)
```

In [14]:
```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_reduced[:, 0], X_reduced[:, 1], types,
           c=types, cmap=plt.cm.Spectral)
ax.set_xlabel('principal component 1', fontsize = 12)
ax.set_ylabel('principal component 2', fontsize = 12)
ax.set_zlabel('class', fontsize=12)
ax.set_title('Bill Authentication', fontsize =20)
plt.show()
```



Trực quan hóa dữ liệu với LLE - 3 components

In [15]:
```
lle1 = LocallyLinearEmbedding(n_components=3, n_neighbors=10)
```

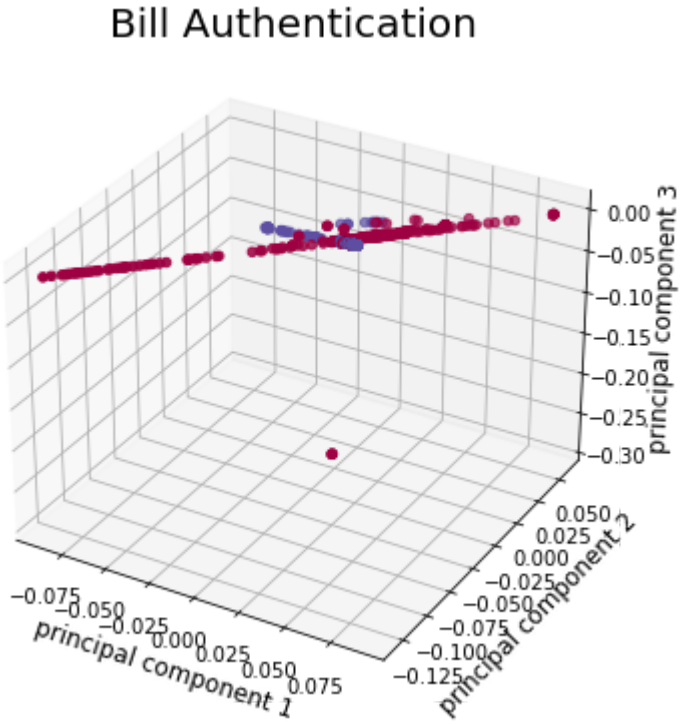
In [16]:
```
X_reduced = lle1.fit_transform(X)
```

In [17]:
```
X_reduced[:2]
```

Out[17]:
```
array([[ 3.45923473e-02,  5.41780552e-04,  4.40495193e-05],
       [ 3.46098040e-02, -6.14370946e-04,  4.03921162e-05]])
```

In [18]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

In [19]:
```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(6,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_reduced[:, 0], X_reduced[:, 1], X_reduced[:, 2],
           c=types, cmap=plt.cm.Spectral)
ax.set_xlabel('principal component 1', fontsize = 12)
ax.set_ylabel('principal component 2', fontsize = 12)
ax.set_zlabel('principal component 3', fontsize=12)
ax.set_title('Bill Authentication', fontsize =20)
plt.show()
```



In [20]:
```
# Áp dụng ML cơ sở cho dữ liệu giảm chiều
# Chọn phương án : 2 components hay 3 components?
# So sánh với trước khi giảm chiều
```