HW8 due 05/11/2023 in class

1. Projection of a dataset on a two dimension plane using PCA (50%)

(a) (25%)First, Sketch the noiseless deterministic version of the trajectory in 3D space:

0<t<2*pi

X1(t)=(t/6.28)*cos t

X2(t)=(t/6.28)*sin t X3(t)

= 0

You can simply sketch it in a paper or using graphic software. No need for python coding. Now using google colab, modify the code to plot the 2 principal components in 2D plane for a parametrized trajectory defined as

θ~ U(0,2pi), i.e., from a uniform distribution from 0 and 2pi.

X1=(θ/6.28)*cos θ + 0.1* N(0,1)

X2=(θ/6.28)*sin θ+0.1*N(0,1)

X3= 0.1*N(0,1)

N(0,1) is the Gaussian noise with zero mean and standard deviation 1. Use total N= 1000 in your plot.

(b) (25%) First, sketch the noiseless deterministic version of the trajectory in 3D space:

0<t<2*pi

X1(t)=(t/6.28)*cos t

X2(t)=(t/6.28)*sin t X3(t)

= t

You can simply sketch it in a paper or using graphic software. No need for python coding. Using this sketch to determine the proper scale for the principal component plot. Use 2 principal component and plot it in 2D as in the example code.

Now switch to the trajectory with noise

θ~ U(0,2pi),

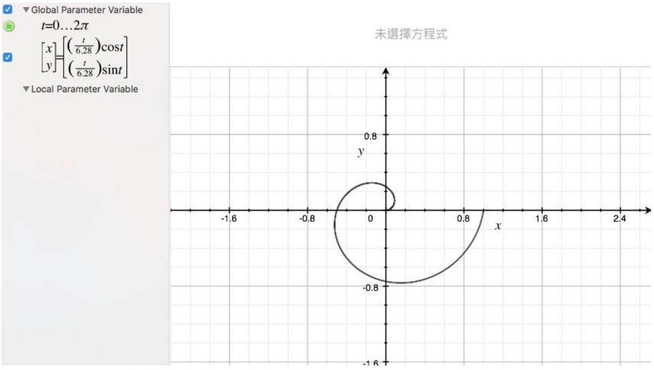
X1=(θ/6.28)*cos θ + 0.1* N(0,1)

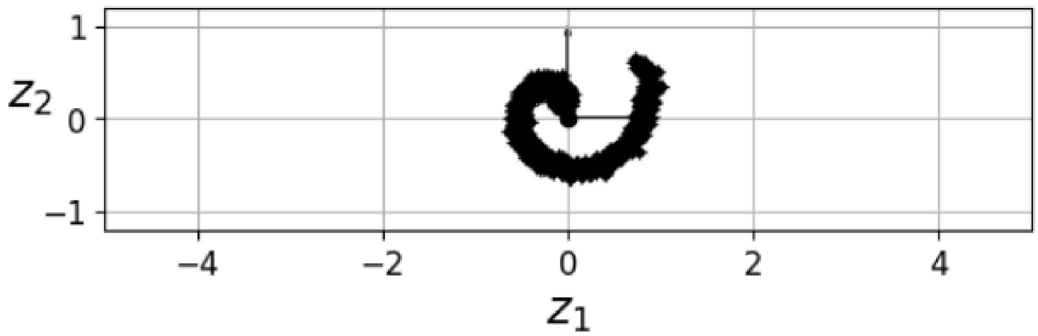
X2=(θ/6.28)*sin θ+0.1*N(0,1)

X3= θ*1 + 0.1*N(0,1)

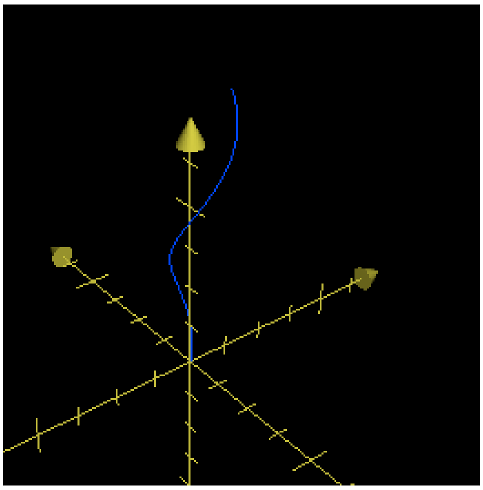
Solution:The plot looks like:

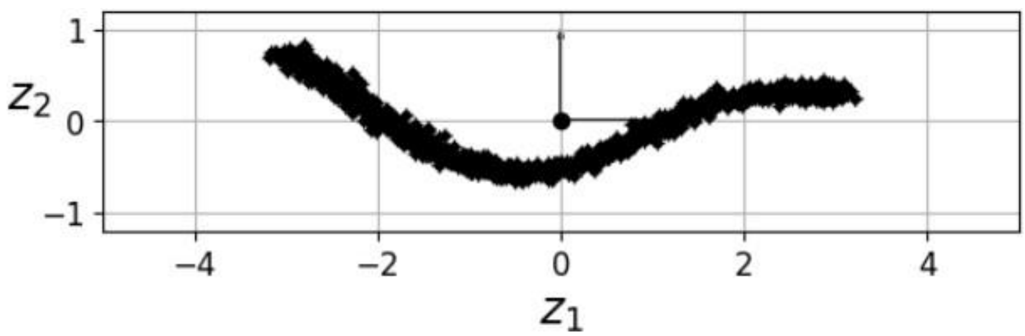
（a）the curve lies in X1 X2 plane and there is 2D curve like




(b) It is 3D spiral curve. The height is roughly 2*pi






The code is

For (a) replace the data generation code with

```
angles = np.random.rand(m) * 2 * np.pi
X = np.empty((m, 3))
X[:, 0] = (angles/6.28)*np.cos(angles) + noise * np.random.randn(m) / 2
X[:, 1] = (angles/6.28)*np.sin(angles) + noise * np.random.randn(m) / 2
X[:, 2] = noise * np.random.randn(m)
```

For (b) replace the data generation code with

```
angles = np.random.rand(m) * 2 * np.pi
X = np.empty((m, 3))
X[:, 0] = (angles/6.28)*np.cos(angles) + noise * np.random.randn(m) / 2
X[:, 1] = (angles/6.28)*np.sin(angles) + noise * np.random.randn(m) / 2
X[:, 2] = angles*1 + noise * np.random.randn(m)
```

k means toy python code

2. (30%) In this homework, I give you the example code for a two cluster python.

The initial setting is two well-separated Gaussian.

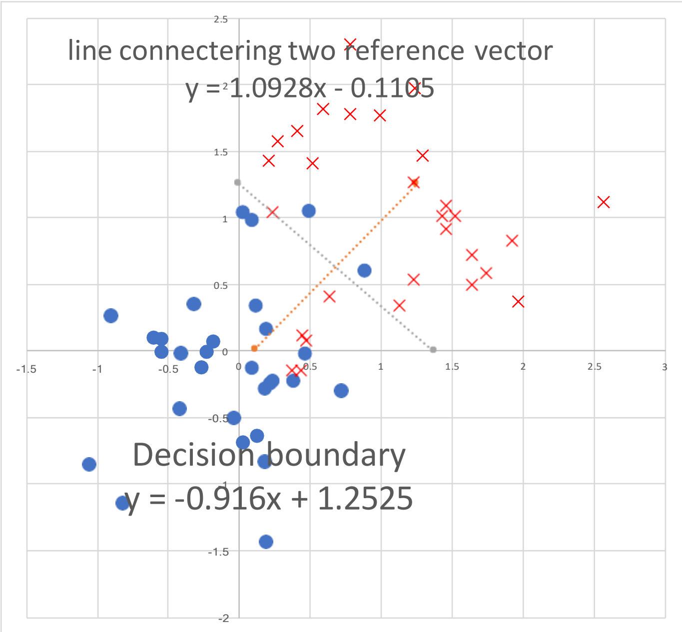
Please set the two clusters to be two Gaussian

```
x1=random.gauss(0,0.6)
x2=random.gauss(0,0.6)
y1=1.0+random.gauss(0,0.6)
y2=1.0+random.gauss(0,0.6)
```

(a)(20%) Modify the code. Use N=60 sample (30 sample for each Gaussian cluster) and run for 10 iteration. Plot the sample point in excel using scatter plot along with final estimated cluster center.

(b) (10%) Write down the equation for decision boundary using the estimated cluster center. (it is a line.)

solution: Insert a graph




Decision boundary is determined by

$$\|x - m_1\| = \|x - m_2\|$$

$m_1$ $m_2$ is the final estimated cluster centers.

You can first fit the line connecting m1 and m2. Got the slope of the equation.

Y= (slope)x+ constant

Slope = tan(theta)

Then add pi/2 to theta to the new slope

(y- ym) =tan(theta+pi/2)(x-xm)

(xm, ym) is the middle point of m1 and m2.

Xm=(1/2)(m1x+m2x)

Ym=(1/2)(m1y+m2y)

Typically, you will obtain y-0.631=-0.916* (x-0.6785)

Or y= -0.916*x +1.2525

3. (20%) Lagrange multiplier

Suppose we want to minimize the function

$$f(x, y) = x^2 + 2y$$

subject to the constraint

$$g(x, y) = 3x + 2y + 1 = 0$$

Using the method of Lagrange multiplier to find the solution $(\hat{x}, \hat{y})$ and corresponding Lagrange multiplier $\lambda$ such that f$(\hat{x}, \hat{y})$ is minimized. (Hint: you need to introduce langrange multiplier $\lambda$ and take partial derivatives.)

Solution:

$$f = f + \lambda g$$

Solve

$\frac{3f}{3x} = 0, \frac{3f}{3y} = 0 \frac{3f}{36} = 0$

The final answer is

$(\hat{x}, \hat{y})$ = (3/2, -11/4)

λ= -1

f($x$, $y$, λ) = -13/4