6521603795 นายปฏิพัทธ์ เอี่ยมรัมย์

lab8

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [8]:
```
#import cancer data from kaggle
df = pd.read_csv('./Cancer_Data.csv')
df
```

Out[8]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840
1	842517	M	20.57	17.77	132.90	1326.0	0.08474
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960
3	84348301	M	11.42	20.38	77.58	386.1	0.14250
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030
...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100
565	926682	M	20.13	28.25	131.20	1261.0	0.09780
566	926954	M	16.60	28.08	108.30	858.1	0.08455
567	927241	M	20.60	29.33	140.10	1265.0	0.11780
568	92751	B	7.76	24.54	47.92	181.0	0.05263

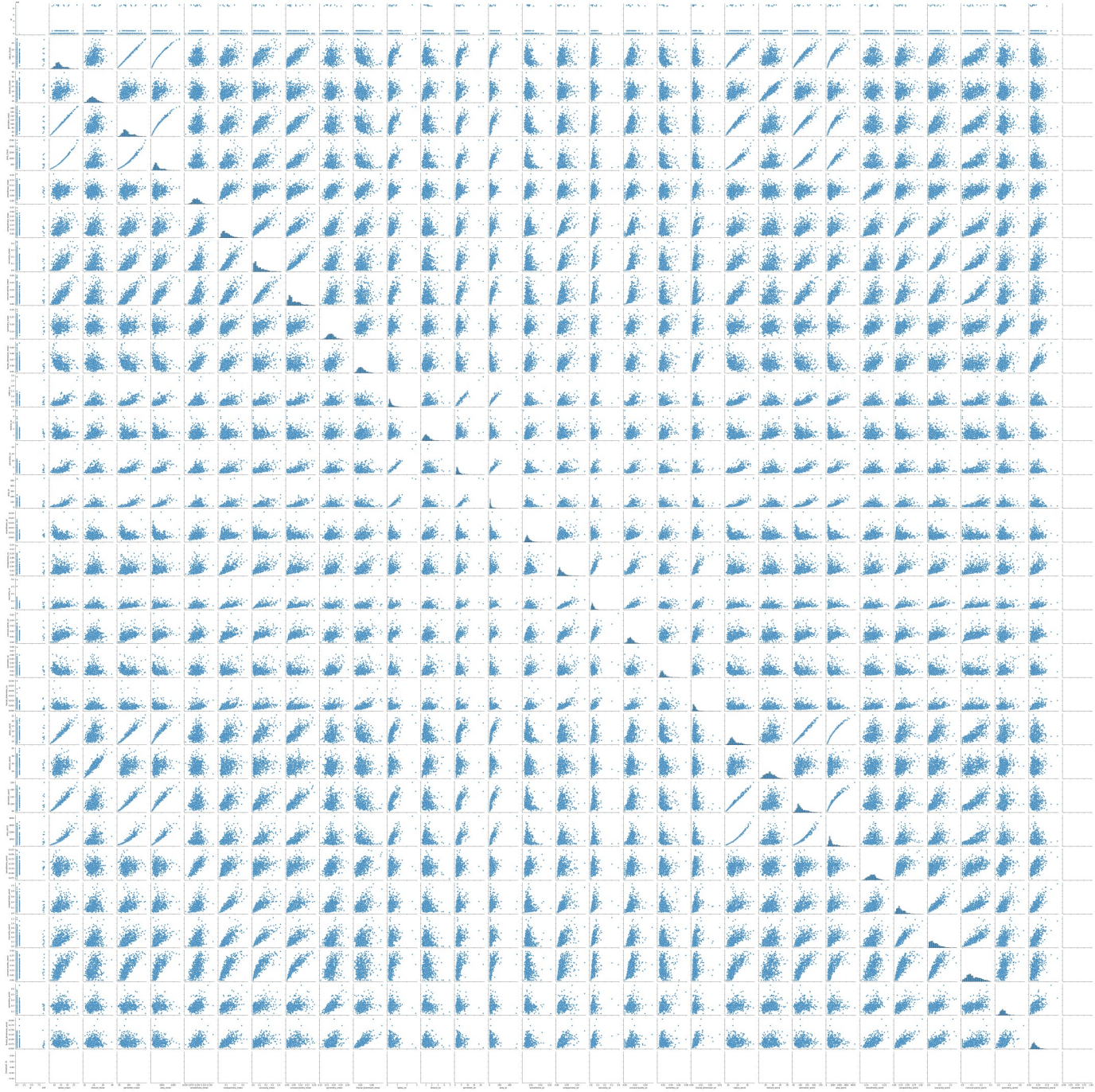
569 rows × 33 columns

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

In [14]:
```
sns.pairplot(df)
```

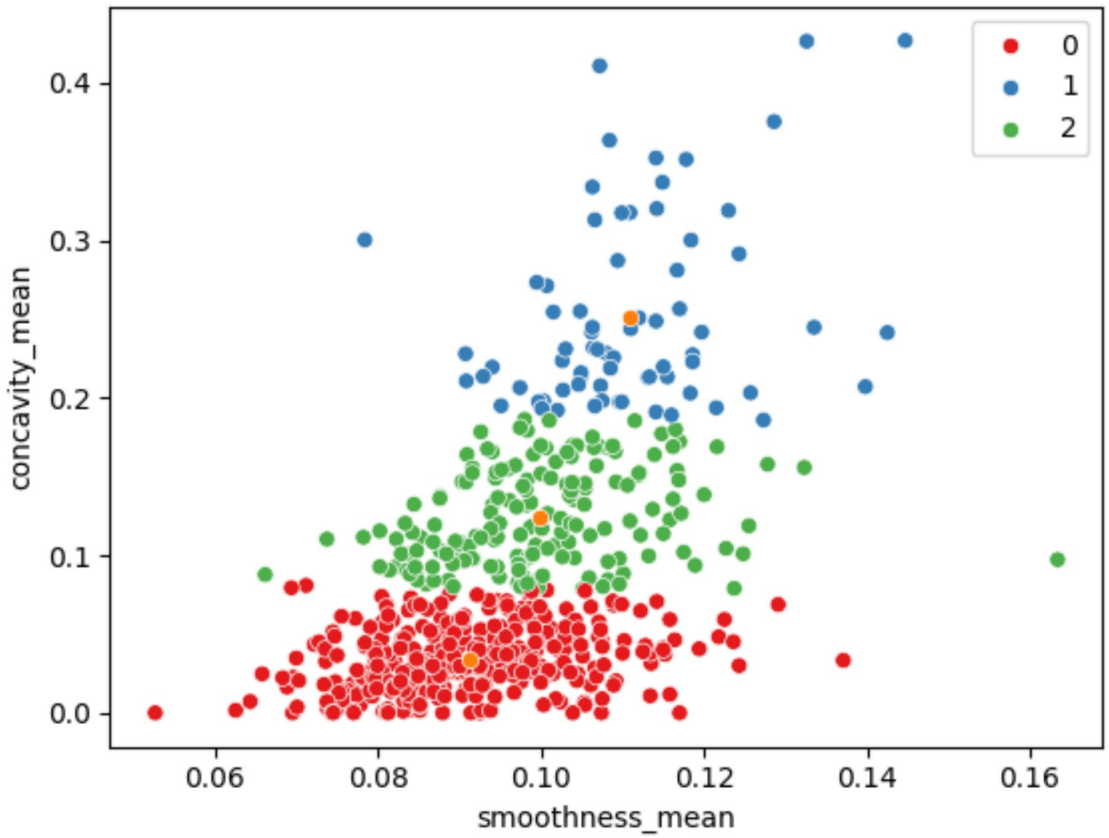
Out[14]:
```
<seaborn.axisgrid.PairGrid at 0x1f13bee2bd0>
```



In [13]:
```
# 1.สร้าง K-mean จาก data โดยใช้ 2 col พร้อมวาดกราฟ เลือก col ที่แบ่งชัดเจน
from sklearn.cluster import KMeans

df2 = df[ ['smoothness_mean','concavity_mean']]
model = KMeans(n_clusters=3, random_state=0)
model.fit(df2)
sns.scatterplot(data = df2, x='smoothness_mean', y = 'concavity_mean', hue = model.labels_, pa
sns.scatterplot(x = model.cluster_centers_[:,0], y = model.cluster_centers_[:,1])
```

Out[13]: <Axes: xlabel='smoothness_mean', ylabel='concavity_mean'>



In [37]:
```
# 2 สร้าง k-mean จาก  data โดยใช้ 3 col ขึ้นไป โชว์ค่า score มากกว่า 0.5
from sklearn import metrics
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

df3 = df[ ['radius_mean','texture_mean', 'area_mean']]
df3[['radius_mean','texture_mean', 'area_mean']] = scaler.fit_transform(df3)



for k in range(2,11):
    model = KMeans(n_clusters=k, random_state=0)
    model.fit(df3)
    score = metrics.silhouette_score(df3, model.labels_)

    print('K = ',k, 'score = ',score )
```

```
C:\Users\francis\AppData\Local\Temp\ipykernel_7856\3137854071.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/
indexing.html#returning-a-view-versus-a-copy
  df3[['radius_mean','texture_mean', 'area_mean']] = scaler.fit_transform(df3)
K =  2 score =  0.5024713434090203
K =  3 score =  0.3952665846368692
K =  4 score =  0.32501298857422034
K =  5 score =  0.32751981699636545
K =  6 score =  0.3304174046540058
K =  7 score =  0.34647780964608027
K =  8 score =  0.34449168557026155
K =  9 score =  0.34878471257420196
K =  10 score =  0.3275921459275659
```

In [ ]: