

Problem Statement

Predictive Study using the Breast cancer diagnostic dataset

Importing libraries

In [1]:

```
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:

```
df=pd.read_csv(r"C:\Users\shaha\OneDrive\Desktop\Excel\Breast Cancer Prediction.csv")
df
```

Out[2]:

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

569 rows × 32 columns



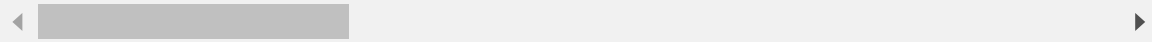
In [3]:

```
df.head()
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 32 columns



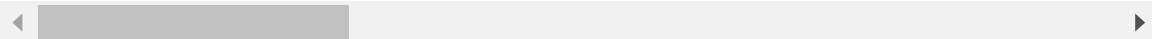
In [4]:

```
df.tail()
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

5 rows × 32 columns



In [5]:

```
df.shape
```

Out[5]:

```
(569, 32)
```

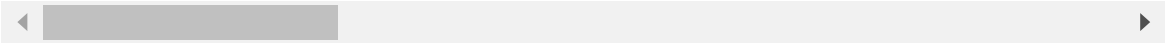
In [6]:

```
df.describe()
```

Out[6]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.0
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.0
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.0
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.0
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.0
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.0
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.1
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.1

8 rows × 31 columns



In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 569 entries, 0 to 568
```

```
Data columns (total 32 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

```
dtypes: float64(30), int64(1), object(1)
```

```
memory usage: 142.4+ KB
```

In [9]:

```
df.columns
```

Out[9]:

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',  
      'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',  
      'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',  
      'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',  
      'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',  
      'fractal_dimension_se', 'radius_worst', 'texture_worst',  
      'perimeter_worst', 'area_worst', 'smoothness_worst',  
      'compactness_worst', 'concavity_worst', 'concave points_worst',  
      'symmetry_worst', 'fractal_dimension_worst'],  
      dtype='object')
```

In [10]:

```
df.isnull().sum()
```

Out[10]:

```
id                0  
diagnosis         0  
radius_mean      0  
texture_mean     0  
perimeter_mean   0  
area_mean        0  
smoothness_mean  0  
compactness_mean 0  
concavity_mean   0  
concave points_mean 0  
symmetry_mean    0  
fractal_dimension_mean 0  
radius_se        0  
texture_se       0  
perimeter_se     0  
area_se          0  
smoothness_se    0  
compactness_se   0  
concavity_se     0  
concave points_se 0  
symmetry_se      0  
fractal_dimension_se 0  
radius_worst     0  
texture_worst    0  
perimeter_worst  0  
area_worst       0  
smoothness_worst 0  
compactness_worst 0  
concavity_worst  0  
concave points_worst 0  
symmetry_worst   0  
fractal_dimension_worst 0  
dtype: int64
```

In [11]:

```
df['id'].value_counts()
```

Out[11]:

```
id
842302    1
90250     1
901315    1
9013579   1
9013594   1
..
873885    1
873843    1
873701    1
873593    1
92751     1
Name: count, Length: 569, dtype: int64
```

In [14]:

```
df['radius_mean'].value_counts()
```

Out[14]:

```
radius_mean
12.34     4
11.71     3
12.46     3
13.05     3
10.26     3
..
12.23     1
14.45     1
19.18     1
18.08     1
7.76      1
Name: count, Length: 456, dtype: int64
```

In [15]:

```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[15]:

```
▼ KMeans
KMeans()
```

In [17]:

```
y_predicted=km.fit_predict(df[["id","radius_mean"]])
y_predicted
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning
warnings.warn(

Out[17]:

```
array([1, 1, 5, 5, 5, 1, 1, 5, 1, 5, 1, 5, 1, 1, 5, 5, 1, 5, 1, 3, 3, 3,  
       3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       5, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3, 5, 5,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 5, 1, 4, 4, 3, 1,  
       3, 3, 3, 3, 1, 1, 3, 3, 1, 3, 3, 3, 1, 3, 3, 3, 5, 1, 1, 1, 1, 1,  
       1, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 7, 3, 3, 7, 3, 3, 3, 3, 3, 0,  
       3, 3, 3, 3, 3, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1,  
       1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 3, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,  
       1, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 3, 3, 3,  
       3, 1, 3, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,  
       1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 3, 1,  
       1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,  
       1, 3, 1, 3, 3, 1, 3, 2, 2, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 1,  
       3, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,  
       0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 0, 1, 3, 3, 3, 3, 1, 6, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 6, 6,  
       3, 6, 6, 3, 3, 3, 3, 1, 3, 3, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0,  
       1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

In [19]:

```
df["cluster"]=y_predicted
df.head()
```

Out[19]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 33 columns

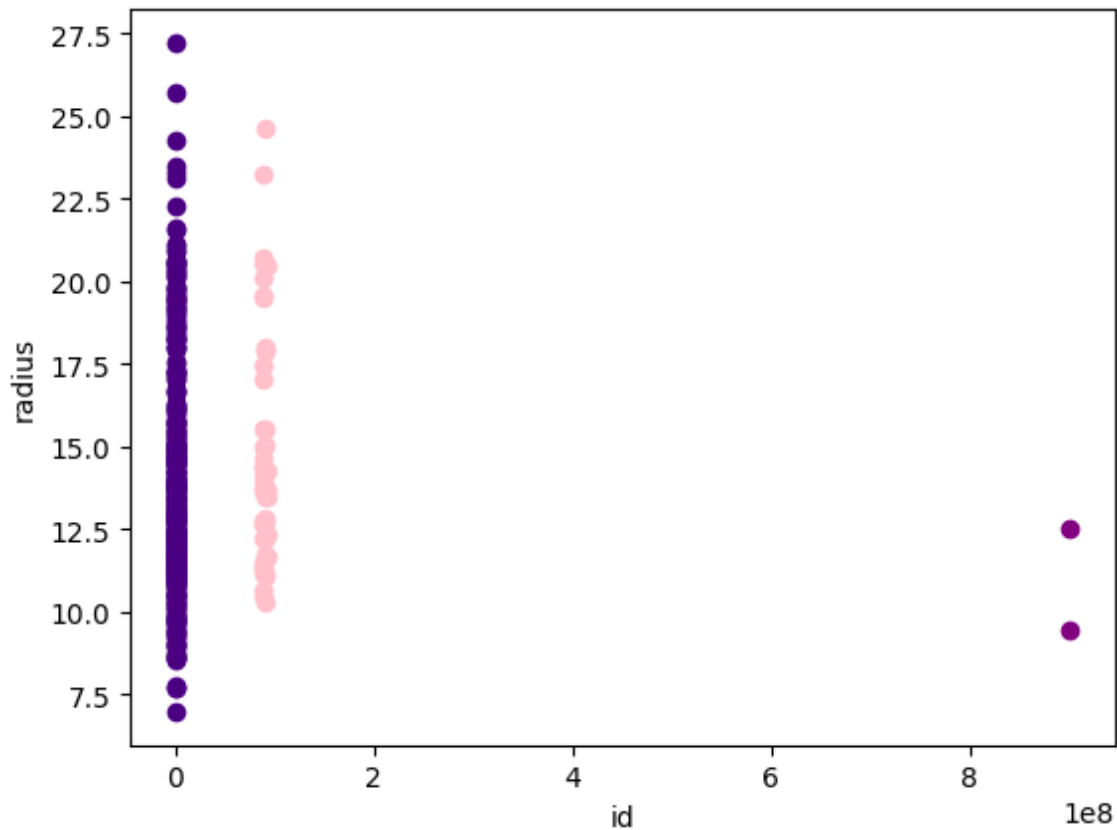


In [20]:

```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["id"],df1["radius_mean"],color="pink")
plt.scatter(df2["id"],df2["radius_mean"],color="indigo")
plt.scatter(df3["id"],df3["radius_mean"],color="purple")
plt.xlabel("id")
plt.ylabel("radius")
```

Out[20]:

Text(0, 0.5, 'radius')



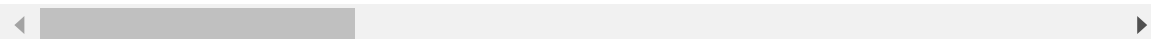
In [22]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["radius_mean"]])
df["radius_mean"]=scaler.transform(df[["radius_mean"]])
df.head()
```

Out[22]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	842302	M	0.521037	10.38	122.80	1001.0	
1	842517	M	0.643144	17.77	132.90	1326.0	
2	84300903	M	0.601496	21.25	130.00	1203.0	
3	84348301	M	0.210090	20.38	77.58	386.1	
4	84358402	M	0.629893	14.34	135.10	1297.0	

5 rows × 33 columns



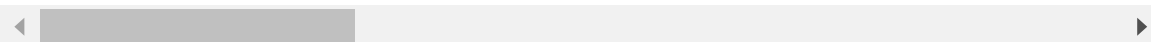
In [26]:

```
scaler.fit(df[["id"]])
df["id"]=scaler.transform(df[["id"]])
df.head()
```

Out[26]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	0.000915	M	0.521037	10.38	122.80	1001.0	
1	0.000915	M	0.643144	17.77	132.90	1326.0	
2	0.092495	M	0.601496	21.25	130.00	1203.0	
3	0.092547	M	0.210090	20.38	77.58	386.1	
4	0.092559	M	0.629893	14.34	135.10	1297.0	

5 rows × 33 columns



In [27]:

```
km=KMeans()
```

In [29]:

```
y_predicted=km.fit_predict(df[["id","radius_mean"]])
y_predicted
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning
warnings.warn(

Out[29]:

```
array([3, 3, 3, 1, 3, 1, 3, 7, 7, 1, 0, 0, 3, 0, 7, 7, 7, 0, 3, 7, 7, 4,  
       0, 3, 0, 0, 7, 3, 0, 0, 3, 1, 0, 3, 0, 0, 7, 7, 0, 7, 7, 1, 3, 7,  
       7, 3, 4, 7, 1, 7, 1, 7, 1, 3, 0, 1, 3, 0, 7, 4, 4, 4, 7, 4, 7, 0,  
       4, 1, 4, 7, 3, 4, 0, 7, 1, 0, 7, 3, 3, 7, 1, 7, 6, 3, 1, 3, 7, 3,  
       1, 7, 7, 0, 7, 7, 0, 3, 1, 4, 1, 7, 7, 4, 1, 4, 4, 7, 1, 1, 6, 1,  
       4, 1, 7, 4, 4, 1, 4, 0, 0, 3, 1, 3, 6, 7, 7, 7, 7, 3, 0, 3, 1, 0,  
       0, 0, 3, 7, 1, 1, 0, 1, 4, 0, 1, 7, 1, 1, 1, 0, 7, 7, 2, 2, 4, 1,  
       7, 1, 3, 0, 1, 1, 1, 3, 3, 1, 6, 0, 1, 0, 0, 0, 1, 7, 0, 1, 1, 4,  
       4, 0, 7, 7, 6, 3, 0, 1, 0, 4, 3, 1, 1, 1, 7, 7, 4, 1, 0, 7, 7, 3,  
       3, 7, 1, 0, 6, 7, 1, 0, 4, 0, 7, 0, 5, 1, 6, 5, 7, 7, 1, 4, 3, 3,  
       7, 7, 4, 0, 7, 7, 4, 0, 7, 7, 0, 1, 1, 3, 4, 7, 6, 3, 7, 0, 7, 1,  
       1, 7, 3, 4, 7, 7, 1, 1, 3, 1, 3, 0, 3, 7, 3, 0, 0, 0, 3, 0, 0, 0,  
       0, 3, 4, 7, 7, 1, 7, 1, 3, 4, 3, 1, 1, 3, 7, 7, 3, 1, 3, 0, 7, 1,  
       1, 7, 1, 1, 7, 0, 7, 1, 7, 7, 1, 1, 7, 4, 3, 1, 3, 4, 1, 1, 7, 4,  
       7, 7, 1, 7, 7, 1, 4, 1, 1, 3, 4, 1, 4, 3, 7, 3, 1, 7, 7, 1, 0, 0,  
       0, 7, 1, 1, 1, 0, 7, 3, 4, 6, 7, 4, 1, 3, 1, 4, 1, 0, 1, 1, 1, 0,  
       6, 0, 1, 1, 7, 7, 4, 2, 2, 7, 7, 0, 7, 3, 3, 1, 3, 6, 0, 0, 3, 3,  
       7, 0, 4, 7, 7, 1, 1, 1, 1, 1, 7, 7, 1, 7, 1, 3, 4, 4, 0, 3, 1, 7,  
       7, 7, 1, 1, 3, 1, 7, 7, 1, 1, 0, 7, 3, 1, 1, 1, 4, 0, 0, 1, 4, 0,  
       7, 1, 1, 7, 1, 7, 4, 4, 4, 1, 1, 7, 0, 1, 3, 3, 0, 7, 7, 7, 7, 7,  
       1, 0, 7, 4, 3, 1, 3, 0, 7, 5, 1, 3, 1, 7, 1, 7, 1, 7, 7, 4, 5, 5,  
       7, 2, 2, 7, 7, 4, 0, 1, 4, 1, 0, 1, 1, 7, 7, 7, 1, 0, 1, 7, 7, 7,  
       0, 1, 7, 3, 1, 0, 1, 3, 3, 1, 7, 0, 1, 1, 3, 3, 0, 7, 1, 6, 4, 4,  
       1, 1, 0, 0, 1, 0, 7, 7, 0, 1, 3, 3, 7, 7, 4, 6, 1, 7, 4, 4, 7, 1,  
       7, 1, 1, 1, 7, 3, 1, 3, 7, 1, 4, 4, 1, 7, 0, 7, 7, 7, 4, 4, 4, 1,  
       1, 1, 7, 4, 7, 4, 4, 4, 7, 1, 7, 1, 0, 3, 3, 3, 0, 3, 4])
```

In [31]:

```
df["New Cluster"]=y_predicted
df.head()
```

Out[31]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
0	0.000915	M	0.521037	10.38	122.80	1001.0	
1	0.000915	M	0.643144	17.77	132.90	1326.0	
2	0.092495	M	0.601496	21.25	130.00	1203.0	
3	0.092547	M	0.210090	20.38	77.58	386.1	
4	0.092559	M	0.629893	14.34	135.10	1297.0	

5 rows × 34 columns

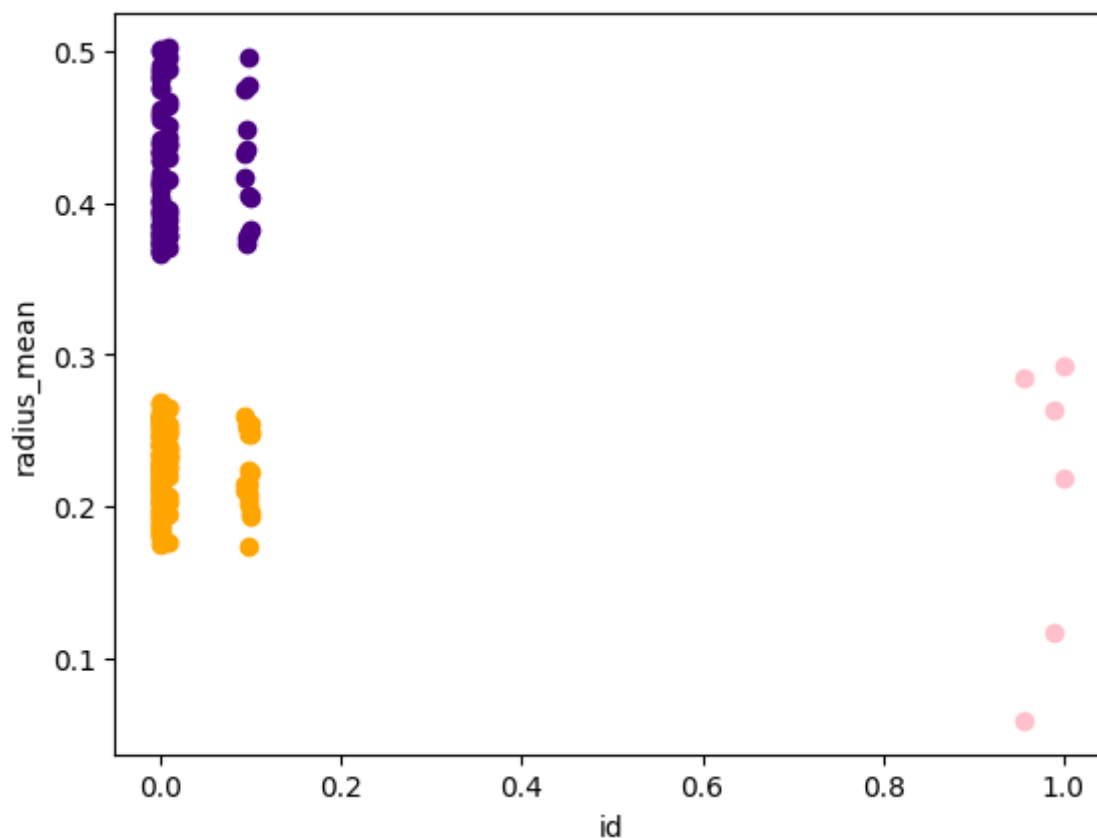


In [33]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["id"],df1["radius_mean"],color="indigo")
plt.scatter(df2["id"],df2["radius_mean"],color="orange")
plt.scatter(df3["id"],df3["radius_mean"],color="pink")
plt.xlabel("id")
plt.ylabel("radius_mean")
```

Out[33]:

Text(0, 0.5, 'radius_mean')



In [34]:

km.cluster_centers_

Out[34]:

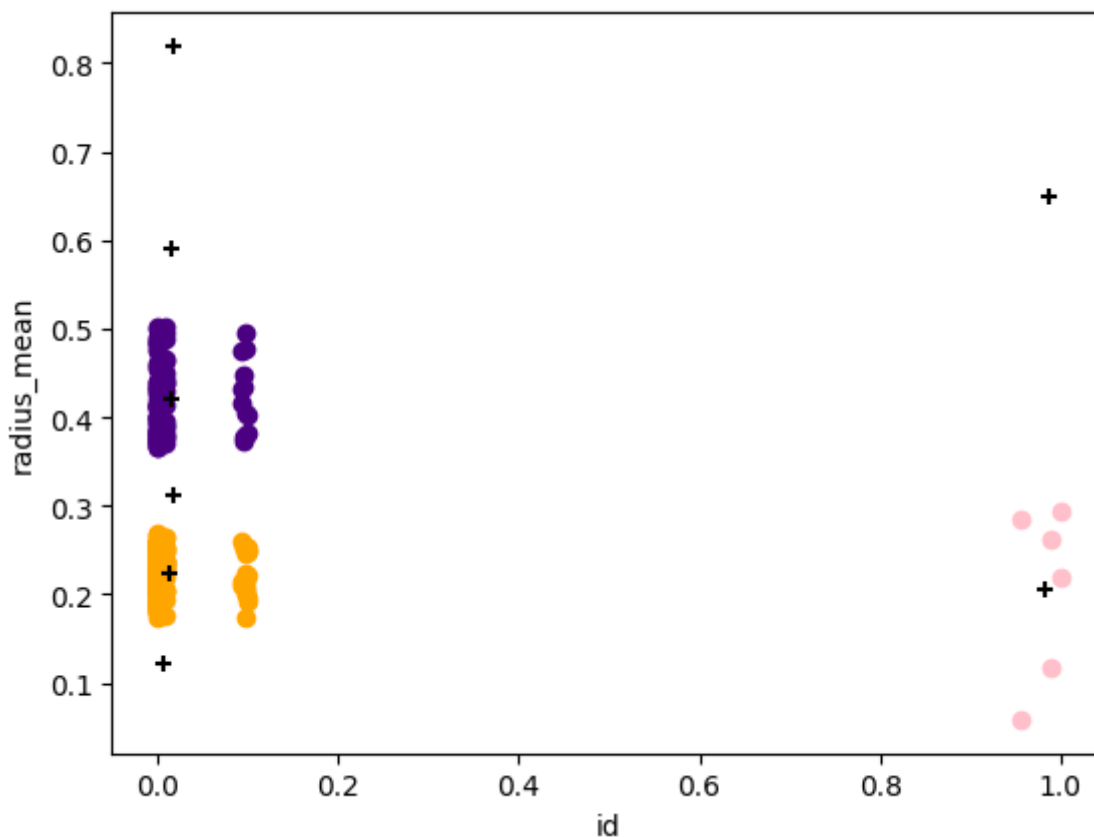
```
array([[0.01609849, 0.41962623],
       [0.01333193, 0.22400619],
       [0.98148998, 0.20571253],
       [0.01641903, 0.59162424],
       [0.0064419 , 0.12195811],
       [0.98667602, 0.65024374],
       [0.01854515, 0.81935146],
       [0.01716984, 0.3114415 ]])
```

In [35]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["id"],df1["radius_mean"],color="indigo")
plt.scatter(df2["id"],df2["radius_mean"],color="orange")
plt.scatter(df3["id"],df3["radius_mean"],color="pink")
plt.scatter(km.cluster_centers_[ :,0],km.cluster_centers_[ :,1],color="black",marker="+")
plt.xlabel("id")
plt.ylabel("radius_mean")
```

Out[35]:

Text(0, 0.5, 'radius_mean')



In [36]:

```
k_rng=range(1,10)
sse=[]
```

In [38]:

```

for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["id","radius_mean"]])
    sse.append(km.inertia_)
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")

```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

```
[26.490617520909876, 15.59142504847003, 6.011422276021483, 3.7289426736473  
175, 2.9732459507608073, 2.3793176933559246, 1.8403134359778621, 1.5003376  
921115434, 1.2881933684794757]
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

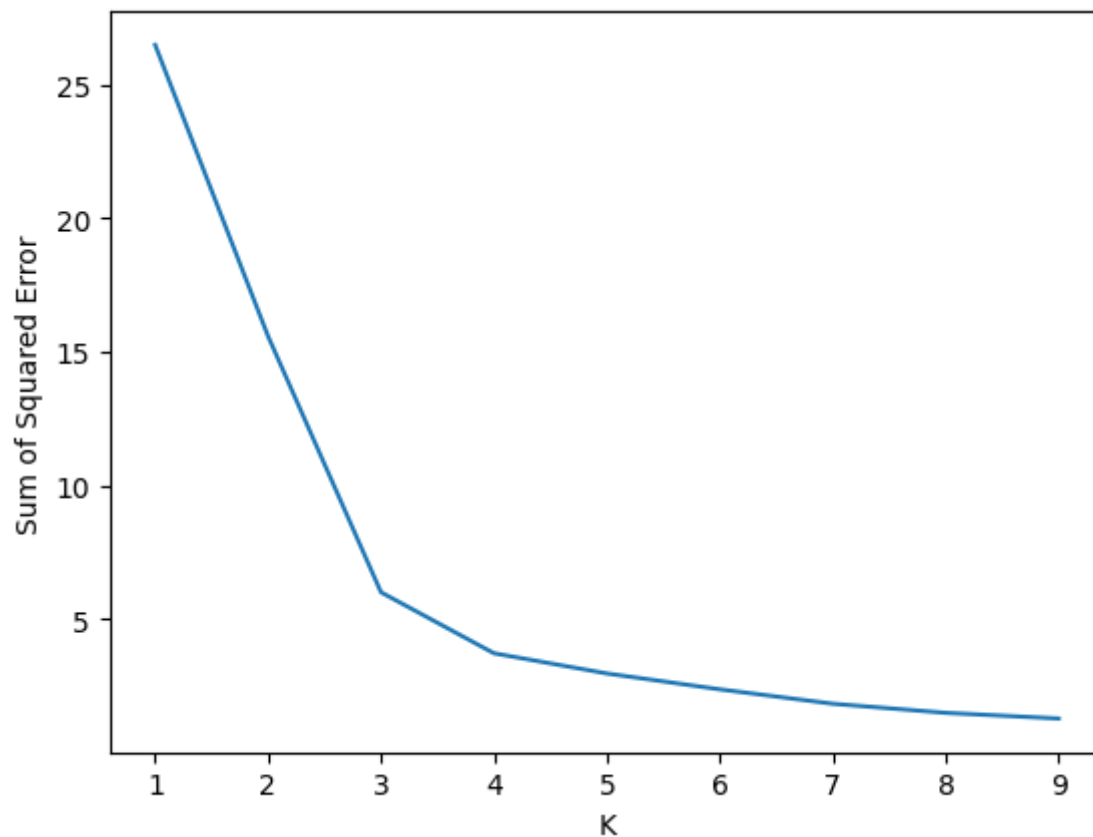
```
warnings.warn(
```

C:\Users\shaha\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

```
warnings.warn(
```

Out[38]:

Text(0, 0.5, 'Sum of Squared Error')



Conclusion ¶

Form the given Breast Canceer dataset we can use multiple models,for that models we get different types of accuracies but that accuracies is not goog so, that is why we will take it as a clustering and done with K-Means clustering

In []: