

PROJECT-4

```
In [1]: import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

```
In [2]: df=pd.read_csv(r"C:\Users\user\Downloads\BreastCancerPrediction.csv")
df
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	842302	M	17.99	10.38	122.80	1001.0	0.
1	842517	M	20.57	17.77	132.90	1326.0	0.0
2	84300903	M	19.69	21.25	130.00	1203.0	0.
3	84348301	M	11.42	20.38	77.58	386.1	0.
4	84358402	M	20.29	14.34	135.10	1297.0	0.
...
564	926424	M	21.56	22.39	142.00	1479.0	0.
565	926682	M	20.13	28.25	131.20	1261.0	0.0
566	926954	M	16.60	28.08	108.30	858.1	0.0
567	927241	M	20.60	29.33	140.10	1265.0	0.
568	92751	B	7.76	24.54	47.92	181.0	0.0

569 rows × 33 columns



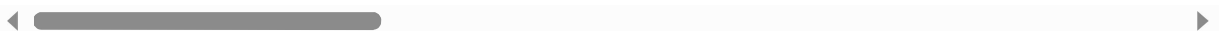
DATA CLEANING AND PREPROCESSING

```
In [3]: df.head()
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 33 columns




```
In [4]: df.tail()
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
564	926424	M	21.56	22.39	142.00	1479.0	0.117
565	926682	M	20.13	28.25	131.20	1261.0	0.097
566	926954	M	16.60	28.08	108.30	858.1	0.084
567	927241	M	20.60	29.33	140.10	1265.0	0.117
568	92751	B	7.76	24.54	47.92	181.0	0.052

5 rows × 33 columns



```
In [5]: df.shape
```

Out[5]: (569, 33)

In [6]: 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 [7]: df.describe()
```

Out[7]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mea
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.09636
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.01406
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.05263
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.08637
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.09587
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.10530
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.16340

8 rows × 32 columns



```
In [8]: df.isnull().sum()
```

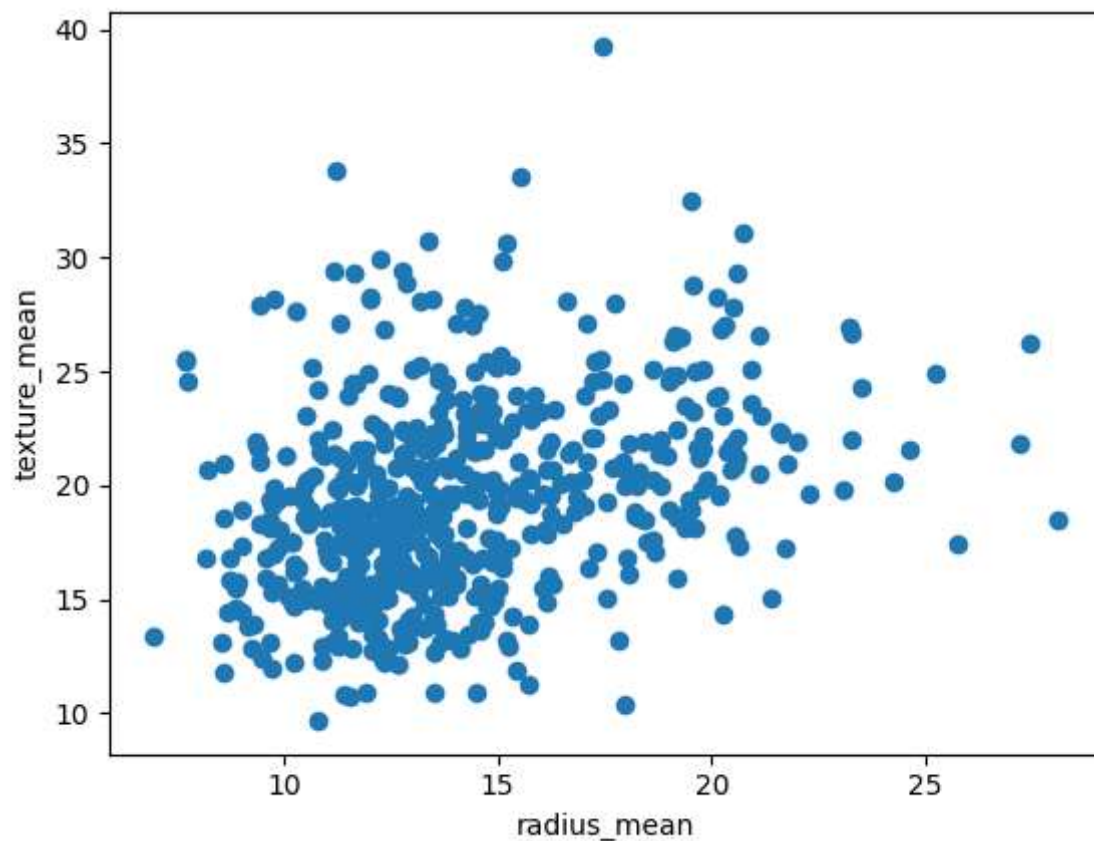
```
Out[8]: id                                0
diagnosis                                0
radius_mean                             0
texture_mean                             0
perimeter_mean                           0
area_mean                                0
smoothness_mean                           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
Unnamed: 32                                569
dtype: int64
```

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

```
Out[9]: 0      False
1      False
2      False
3      False
4      False
...
564    False
565    False
566    False
567    False
568    False
Length: 569, dtype: bool
```

```
In [13]: plt.scatter(df["radius_mean"],df["texture_mean"])  
plt.xlabel("radius_mean")  
plt.ylabel("texture_mean")
```

```
Out[13]: Text(0, 0.5, 'texture_mean')
```



KMeans Clustering

```
In [14]: from sklearn.cluster import KMeans
```

```
In [15]: km=KMeans()  
km
```

```
Out[15]: 

▼ KMeans



KMeans()


```

```
In [16]: y_predicted=km.fit_predict(df[["radius_mean", "area_mean"]])
y_predicted
```

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning
 warnings.warn(

```
Out[16]: array([3, 1, 1, 7, 1, 7, 3, 0, 0, 7, 6, 6, 1, 6, 0, 6, 6, 6, 1, 0, 0, 2,
 6, 1, 3, 3, 6, 3, 6, 3, 3, 7, 3, 1, 6, 3, 6, 0, 6, 0, 0, 7, 3, 0,
 0, 3, 2, 0, 7, 0, 7, 0, 7, 3, 6, 7, 1, 6, 0, 2, 2, 2, 6, 2, 0, 6,
 2, 7, 2, 0, 1, 2, 3, 0, 7, 6, 0, 3, 1, 0, 7, 0, 4, 1, 7, 3, 6, 3,
 7, 6, 6, 6, 0, 0, 6, 1, 7, 2, 7, 6, 0, 2, 7, 2, 2, 0, 7, 7, 4, 7,
 2, 7, 0, 2, 2, 7, 2, 6, 6, 3, 7, 3, 4, 6, 0, 0, 0, 1, 6, 1, 7, 6,
 6, 6, 3, 0, 7, 7, 6, 7, 2, 6, 7, 0, 7, 7, 7, 6, 6, 0, 0, 2, 2, 7,
 0, 7, 3, 3, 7, 7, 7, 1, 1, 7, 4, 6, 7, 3, 3, 6, 7, 0, 6, 7, 2, 2,
 2, 6, 0, 0, 5, 1, 6, 7, 6, 2, 3, 7, 7, 7, 0, 0, 2, 7, 6, 0, 0, 3,
 1, 6, 7, 3, 4, 0, 7, 6, 2, 3, 0, 6, 1, 7, 5, 3, 0, 0, 7, 2, 1, 1,
 0, 0, 2, 6, 0, 6, 2, 6, 0, 0, 3, 7, 7, 1, 2, 0, 4, 1, 0, 3, 0, 7,
 7, 0, 1, 2, 0, 0, 2, 7, 1, 7, 1, 3, 1, 0, 1, 6, 6, 6, 1, 3, 3, 6,
 3, 1, 2, 0, 0, 2, 6, 7, 4, 2, 3, 7, 7, 3, 0, 0, 1, 7, 1, 6, 0, 0,
 7, 0, 7, 7, 6, 6, 0, 7, 0, 0, 7, 7, 6, 2, 1, 7, 1, 2, 7, 7, 0, 2,
 0, 0, 7, 6, 0, 7, 2, 7, 7, 3, 2, 7, 2, 1, 0, 1, 7, 0, 0, 7, 6, 6,
 6, 0, 7, 7, 7, 3, 0, 3, 2, 4, 6, 2, 7, 1, 7, 2, 7, 6, 7, 7, 7, 6,
 4, 6, 7, 0, 0, 0, 2, 2, 0, 0, 0, 6, 0, 1, 1, 7, 4, 4, 6, 6, 1, 1,
 0, 6, 2, 0, 0, 7, 7, 7, 7, 7, 0, 6, 7, 0, 7, 1, 2, 2, 6, 1, 7, 0,
 0, 0, 7, 7, 3, 7, 0, 0, 7, 7, 6, 0, 3, 7, 7, 7, 2, 6, 6, 7, 2, 6,
 0, 7, 7, 6, 7, 0, 2, 2, 2, 7, 7, 0, 6, 7, 1, 3, 6, 0, 0, 0, 0, 0,
 7, 3, 0, 2, 3, 7, 3, 6, 6, 1, 7, 1, 7, 6, 0, 0, 7, 0, 0, 2, 3, 5,
 6, 7, 0, 0, 0, 2, 3, 7, 2, 7, 6, 7, 7, 0, 0, 0, 7, 6, 7, 0, 0, 0,
 6, 7, 6, 1, 7, 3, 7, 3, 3, 7, 0, 6, 0, 7, 3, 1, 6, 0, 7, 4, 2, 2,
 7, 7, 6, 6, 7, 6, 0, 6, 6, 7, 3, 1, 0, 0, 2, 4, 7, 0, 2, 2, 0, 7,
 0, 7, 7, 7, 0, 1, 7, 1, 0, 7, 2, 2, 7, 6, 6, 0, 0, 0, 2, 2, 2, 7,
 7, 7, 0, 2, 0, 2, 2, 2, 6, 7, 0, 7, 6, 1, 4, 1, 3, 1, 2])
```

```
In [17]: df["Cluster"]=y_predicted
df.head()
```

```
Out[17]:
```

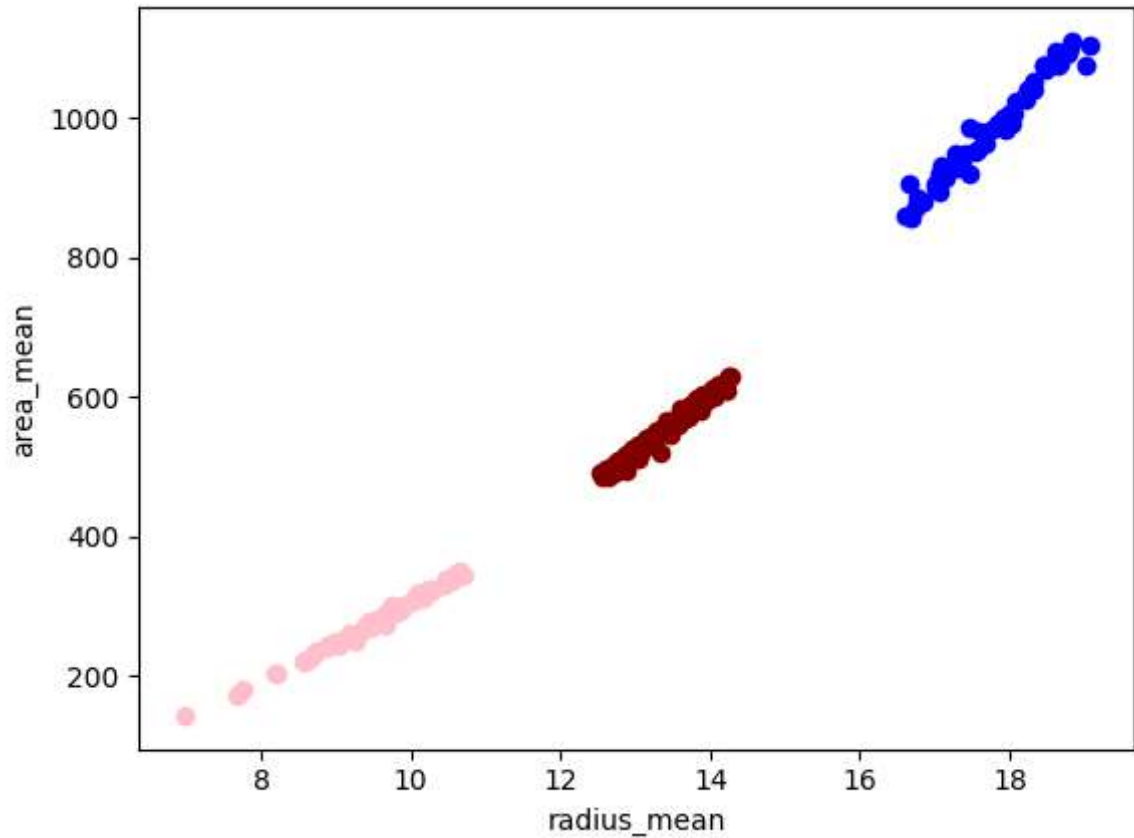
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 34 columns



```
In [19]: df1=df[df.Cluster==0]
df2=df[df.Cluster==2]
df3=df[df.Cluster==3]
plt.scatter(df1["radius_mean"],df1["area_mean"],color="maroon")
plt.scatter(df2["radius_mean"],df2["area_mean"],color="pink")
plt.scatter(df3["radius_mean"],df3["area_mean"],color="blue")
plt.xlabel("radius_mean")
plt.ylabel("area_mean")
```

Out[19]: Text(0, 0.5, 'area_mean')



```
In [20]: from sklearn.preprocessing import MinMaxScaler
```

```
In [21]: scaler=MinMaxScaler()
```



```
In [22]: scaler.fit(df[["area_mean"]])
df["area_mean"]=scaler.transform(df[["area_mean"]])
df.head()
```

Out[22]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	0.363733	0.118
1	842517	M	20.57	17.77	132.90	0.501591	0.084
2	84300903	M	19.69	21.25	130.00	0.449417	0.109
3	84348301	M	11.42	20.38	77.58	0.102906	0.142
4	84358402	M	20.29	14.34	135.10	0.489290	0.100

5 rows × 34 columns



```
In [23]: km=KMeans()
```

```
In [24]: y_predicted=km.fit_predict(df[["radius_mean", "area_mean"]])
y_predicted
```

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning
warnings.warn(

```
Out[24]: array([6, 1, 6, 5, 1, 0, 6, 0, 0, 0, 3, 7, 6, 3, 0, 7, 7, 3, 1, 0, 0, 2,
7, 1, 3, 3, 7, 6, 7, 3, 6, 5, 3, 6, 3, 3, 7, 0, 7, 0, 0, 5, 6, 0,
0, 6, 2, 0, 5, 0, 5, 0, 5, 6, 7, 5, 6, 7, 0, 2, 2, 2, 7, 2, 0, 7,
2, 5, 2, 0, 6, 2, 3, 0, 5, 3, 0, 6, 1, 0, 5, 0, 4, 6, 5, 6, 7, 6,
0, 7, 7, 7, 0, 0, 7, 1, 5, 2, 5, 7, 0, 2, 5, 2, 5, 0, 5, 0, 1, 5,
2, 0, 7, 5, 2, 5, 2, 7, 7, 6, 5, 6, 4, 7, 0, 0, 0, 6, 7, 1, 5, 7,
3, 7, 6, 0, 5, 5, 7, 5, 2, 3, 5, 0, 5, 5, 5, 7, 7, 0, 0, 2, 2, 5,
0, 5, 3, 3, 5, 5, 5, 6, 6, 5, 4, 7, 5, 3, 3, 7, 5, 0, 7, 5, 5, 2,
2, 3, 0, 0, 4, 1, 7, 5, 7, 2, 6, 5, 5, 5, 7, 0, 2, 5, 7, 0, 0, 6,
6, 7, 5, 3, 4, 0, 0, 7, 2, 3, 0, 7, 1, 5, 4, 3, 7, 0, 5, 2, 1, 6,
0, 0, 2, 7, 0, 7, 2, 7, 0, 0, 3, 5, 5, 1, 2, 7, 4, 1, 7, 3, 0, 0,
5, 0, 6, 5, 0, 0, 5, 5, 1, 5, 1, 3, 6, 0, 6, 7, 7, 7, 1, 3, 3, 7,
3, 1, 5, 0, 0, 5, 7, 5, 1, 2, 6, 5, 5, 6, 0, 0, 6, 5, 6, 3, 0, 0,
5, 0, 5, 5, 7, 7, 0, 5, 0, 0, 5, 5, 7, 5, 6, 0, 1, 5, 5, 5, 0, 2,
0, 0, 5, 7, 0, 5, 2, 0, 5, 6, 2, 0, 2, 1, 0, 1, 5, 0, 7, 5, 3, 3,
3, 0, 5, 5, 5, 3, 0, 6, 2, 4, 7, 2, 5, 6, 5, 2, 5, 7, 5, 5, 5, 7,
4, 7, 5, 0, 0, 0, 2, 2, 0, 0, 0, 3, 0, 1, 1, 5, 1, 1, 3, 7, 1, 1,
0, 3, 5, 0, 0, 5, 5, 5, 5, 0, 0, 7, 5, 0, 5, 6, 2, 2, 7, 1, 5, 7,
0, 0, 5, 5, 6, 5, 0, 0, 5, 5, 3, 0, 6, 5, 5, 5, 2, 7, 7, 5, 2, 7,
0, 5, 5, 7, 5, 0, 2, 2, 5, 5, 5, 0, 7, 0, 1, 6, 7, 0, 0, 7, 0, 7,
5, 3, 0, 5, 6, 5, 3, 7, 7, 1, 5, 6, 5, 7, 0, 0, 5, 0, 0, 2, 3, 4,
7, 5, 0, 0, 0, 2, 3, 5, 2, 5, 7, 5, 5, 0, 7, 0, 5, 3, 5, 0, 0, 0,
7, 0, 7, 6, 5, 3, 5, 6, 6, 0, 0, 7, 0, 0, 6, 1, 7, 0, 0, 4, 2, 2,
5, 5, 3, 7, 5, 7, 0, 7, 7, 5, 6, 1, 0, 0, 2, 4, 5, 0, 2, 2, 0, 5,
0, 5, 5, 5, 0, 1, 5, 1, 7, 5, 2, 2, 5, 7, 7, 0, 0, 0, 2, 2, 2, 5,
5, 5, 0, 2, 0, 2, 2, 2, 7, 5, 7, 5, 7, 1, 1, 1, 3, 1, 2])
```

```
In [25]: df["New cluster"]=y_predicted
df.head()
```

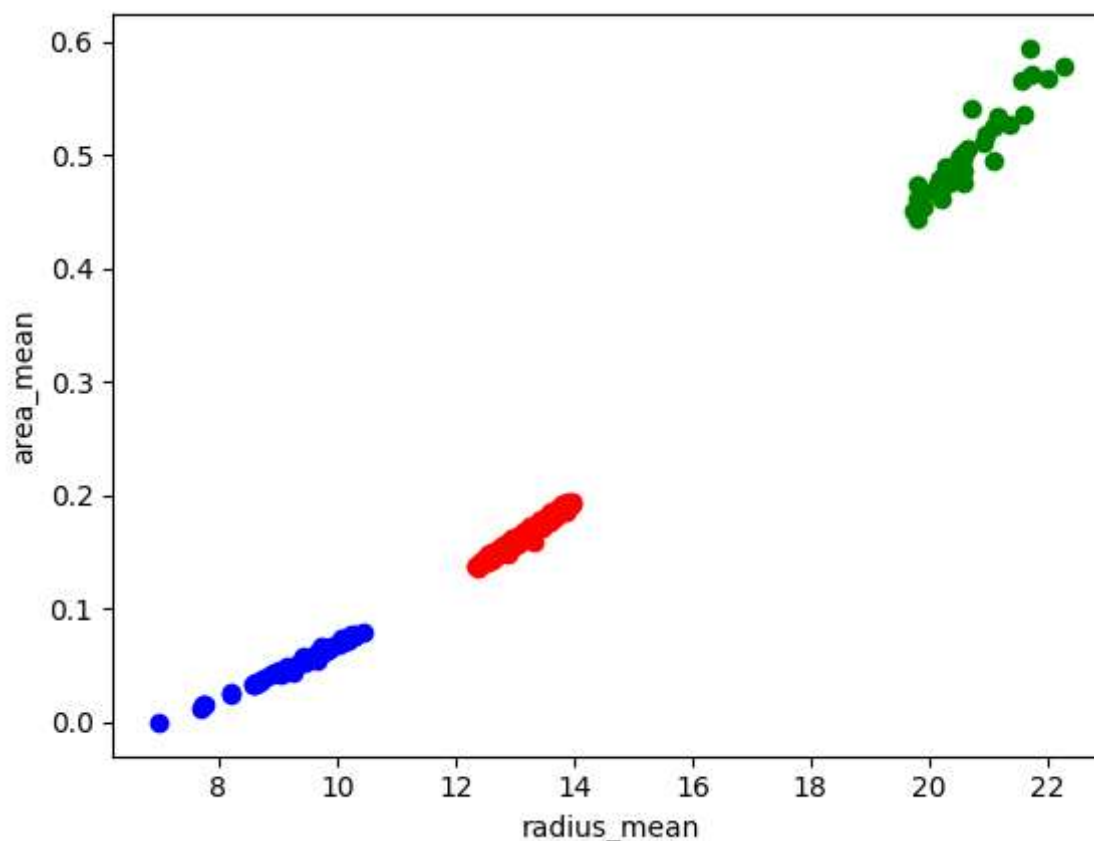
Out[25]:

id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	co
842302	M	17.99	10.38	122.80	0.363733	0.11840	
842517	M	20.57	17.77	132.90	0.501591	0.08474	
300903	M	19.69	21.25	130.00	0.449417	0.10960	
348301	M	11.42	20.38	77.58	0.102906	0.14250	
358402	M	20.29	14.34	135.10	0.489290	0.10030	

× 35 columns

```
In [26]: df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["radius_mean"],df1["area_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["area_mean"],color="green")
plt.scatter(df3["radius_mean"],df3["area_mean"],color="blue")
plt.xlabel("radius_mean")
plt.ylabel("area_mean")
```

Out[26]: Text(0, 0.5, 'area_mean')

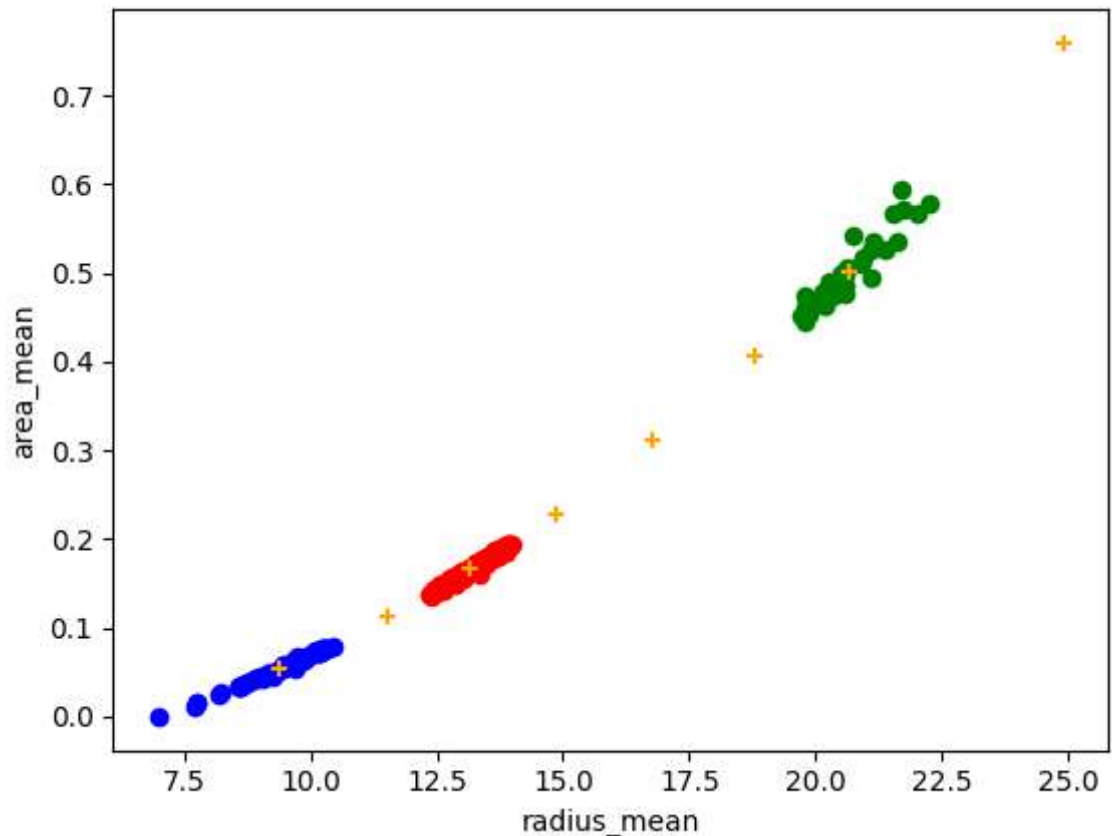


In [27]: `km.cluster_centers_`

Out[27]: `array([[13.16808824, 0.16588298],
 [20.65289474, 0.50078696],
 [9.39162295, 0.05365436],
 [16.77704545, 0.31124747],
 [24.9125 , 0.75956168],
 [11.5335 , 0.11260112],
 [18.795 , 0.40533492],
 [14.84311111, 0.22816402]])`

In [28]: `df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["radius_mean"],df1["area_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["area_mean"],color="green")
plt.scatter(df3["radius_mean"],df3["area_mean"],color="blue")
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color="orange",marker="x")
plt.xlabel("radius_mean")
plt.ylabel("area_mean")`

Out[28]: `Text(0, 0.5, 'area_mean')`



In [29]: `k_rng=range(1,10)
sse=[]`

```
In [30]: for k in k_rng:
          km=KMeans(n_clusters=k)
          km.fit(df[["radius_mean","area_mean"]])
          sse.append(km.inertia_)
          #km.inertia_ will give you the value of sum of square error
          print(sse)
          plt.plot(k_rng,sse)
          plt.xlabel("K")
          plt.ylabel("Sum of Squared Error")
```

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

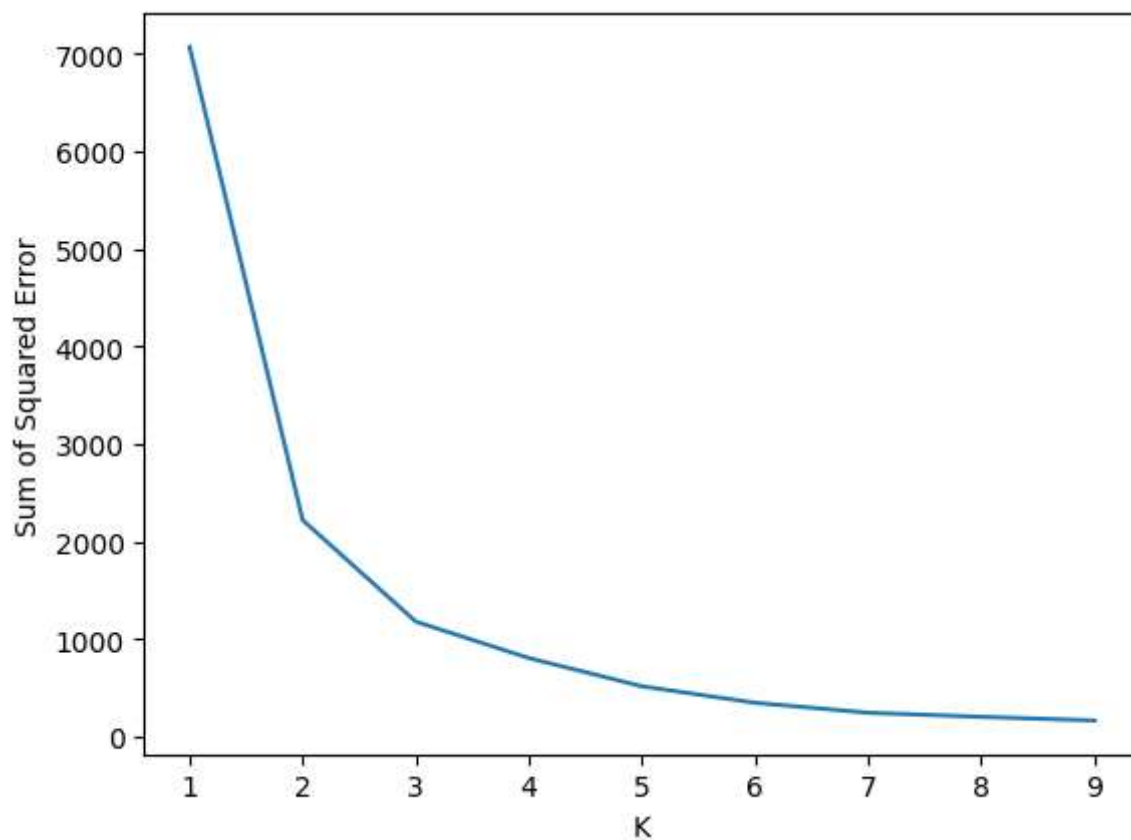
C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

warnings.warn(

C:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\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` explicitly to suppress the warning

warnings.warn(

Out[30]: Text(0, 0.5, 'Sum of Squared Error')



Conclusion:

In this dataset we are doing clustering on Radius_mean and Area_mean. This is the best model for this dataset. When k value is high error rate is low, or k value is low error rate is high.

In []: