

```
In [1]: 1 import pandas as pd
        2 from matplotlib import pyplot as plt
        3 %matplotlib inline
```

```
In [2]: 1 df=pd.read_csv(r"C:\Users\Niranjan\Downloads\BreastCancerPrediction.csv")
        2 df
        3
```

Out[2]:

	rst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concav
80		17.33	184.60	2019.0	0.16220	0.66560	
90		23.41	158.80	1956.0	0.12380	0.18660	
70		25.53	152.50	1709.0	0.14440	0.42450	
10		26.50	98.87	567.7	0.20980	0.86630	
40		16.67	152.20	1575.0	0.13740	0.20500	
...	
50		26.40	166.10	2027.0	0.14100	0.21130	
90		38.25	155.00	1731.0	0.11660	0.19220	
80		34.12	126.70	1124.0	0.11390	0.30940	
40		39.42	184.60	1821.0	0.16500	0.86810	
56		30.37	59.16	268.6	0.08996	0.06444	

```
In [3]: 1 df.head()
```

Out[3]:

	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.
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.

5 rows × 32 columns

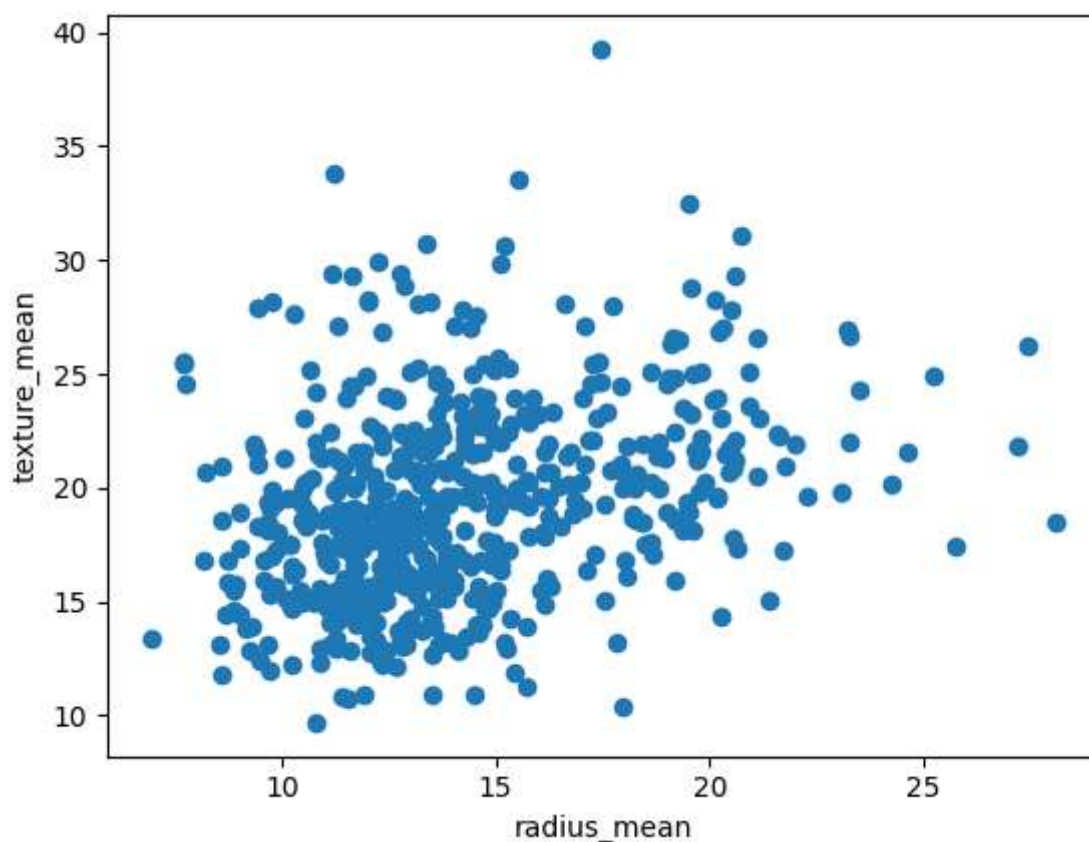
In [4]: 1 df.tail()

Out[4]:

	rst	texture_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavit
150		26.40	166.10	2027.0	0.14100	0.21130	
190		38.25	155.00	1731.0	0.11660	0.19220	
180		34.12	126.70	1124.0	0.11390	0.30940	
140		39.42	184.60	1821.0	0.16500	0.86810	
156		30.37	59.16	268.6	0.08996	0.06444	

In [7]: 1 plt.scatter(df["radius_mean"],df["texture_mean"])
 2 plt.xlabel("radius_mean")
 3 plt.ylabel("texture_mean")
 4

Out[7]: Text(0, 0.5, 'texture_mean')



```
In [8]: 1 from sklearn.cluster import KMeans
        2 km=KMeans()
        3 km
        4
```

```
Out[8]: ▾ KMeans
        KMeans()
```

```
In [9]: 1 y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
        2 y_predicted
```

C:\Users\Niranjana\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(

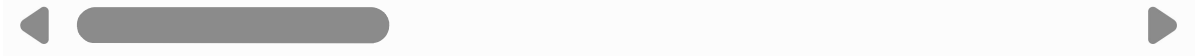
```
Out[9]: array([2, 4, 4, 1, 4, 2, 4, 0, 6, 6, 0, 0, 5, 6, 6, 7, 0, 0, 4, 2, 2, 3,
                2, 5, 0, 2, 0, 4, 6, 2, 5, 1, 5, 5, 0, 0, 0, 1, 6, 0, 6, 6, 5, 0,
                6, 4, 1, 1, 3, 6, 6, 2, 1, 4, 0, 1, 4, 0, 1, 3, 3, 1, 6, 3, 6, 6,
                1, 1, 1, 2, 4, 3, 5, 2, 1, 0, 3, 2, 5, 1, 6, 2, 5, 5, 3, 4, 0, 5,
                6, 2, 6, 0, 2, 1, 0, 5, 1, 1, 3, 0, 6, 3, 1, 1, 1, 2, 1, 1, 4, 6,
                1, 6, 0, 1, 3, 6, 3, 2, 0, 4, 3, 4, 4, 3, 2, 2, 6, 4, 2, 5, 3, 0,
                0, 2, 4, 6, 1, 3, 2, 3, 3, 0, 1, 2, 3, 3, 1, 0, 2, 1, 6, 1, 3, 3,
                2, 1, 0, 0, 3, 3, 1, 4, 4, 6, 4, 0, 3, 0, 5, 2, 3, 0, 2, 3, 3, 3,
                1, 0, 6, 3, 4, 5, 0, 3, 0, 3, 4, 1, 1, 2, 6, 6, 1, 7, 6, 2, 6, 4,
                4, 0, 1, 0, 5, 6, 1, 2, 1, 0, 6, 2, 4, 1, 4, 5, 6, 2, 1, 1, 4, 5,
                2, 2, 1, 0, 2, 2, 3, 2, 6, 6, 0, 7, 7, 5, 3, 0, 5, 4, 7, 7, 2, 3,
                1, 6, 5, 1, 1, 3, 6, 3, 5, 1, 4, 2, 4, 2, 5, 2, 0, 7, 5, 0, 0, 0,
                0, 5, 1, 6, 2, 1, 2, 3, 4, 3, 5, 1, 3, 4, 1, 2, 5, 3, 4, 0, 2, 1,
                6, 3, 1, 1, 0, 0, 2, 1, 3, 2, 3, 1, 0, 6, 4, 1, 5, 1, 1, 6, 2, 3,
                3, 3, 1, 2, 3, 3, 1, 1, 3, 4, 1, 1, 3, 4, 3, 4, 3, 1, 2, 1, 0, 0,
                2, 1, 1, 3, 1, 0, 2, 4, 1, 5, 2, 1, 3, 4, 3, 3, 1, 2, 3, 3, 1, 0,
                4, 6, 3, 1, 1, 2, 3, 1, 1, 6, 1, 0, 2, 4, 5, 1, 4, 4, 0, 2, 4, 4,
                2, 2, 1, 7, 2, 1, 3, 3, 6, 1, 2, 6, 3, 2, 3, 5, 3, 1, 0, 4, 1, 2,
                1, 1, 3, 1, 4, 3, 1, 2, 3, 1, 2, 6, 4, 1, 1, 1, 6, 0, 7, 6, 6, 0,
                3, 6, 1, 2, 3, 0, 1, 6, 3, 6, 1, 1, 0, 1, 4, 4, 2, 0, 1, 2, 0, 2,
                1, 5, 2, 1, 4, 6, 5, 2, 0, 4, 6, 5, 7, 2, 1, 7, 7, 6, 6, 7, 5, 5,
                7, 1, 1, 0, 0, 1, 5, 1, 1, 7, 2, 7, 3, 2, 0, 2, 3, 0, 1, 0, 2, 2,
                2, 2, 2, 4, 1, 0, 6, 2, 4, 3, 0, 0, 1, 1, 4, 4, 2, 6, 2, 4, 3, 3,
                1, 1, 2, 6, 3, 2, 0, 2, 0, 1, 4, 4, 1, 2, 3, 4, 1, 1, 3, 3, 1, 3,
                2, 3, 1, 1, 2, 4, 1, 4, 6, 6, 6, 6, 3, 6, 6, 7, 0, 6, 1, 1, 1, 6,
                6, 6, 7, 6, 7, 7, 1, 7, 6, 6, 7, 7, 7, 5, 4, 5, 7, 5, 6])
```

```
In [10]: 1 df["cluster"]=y_predicted
         2 df.head()
         3
```

Out[10]:

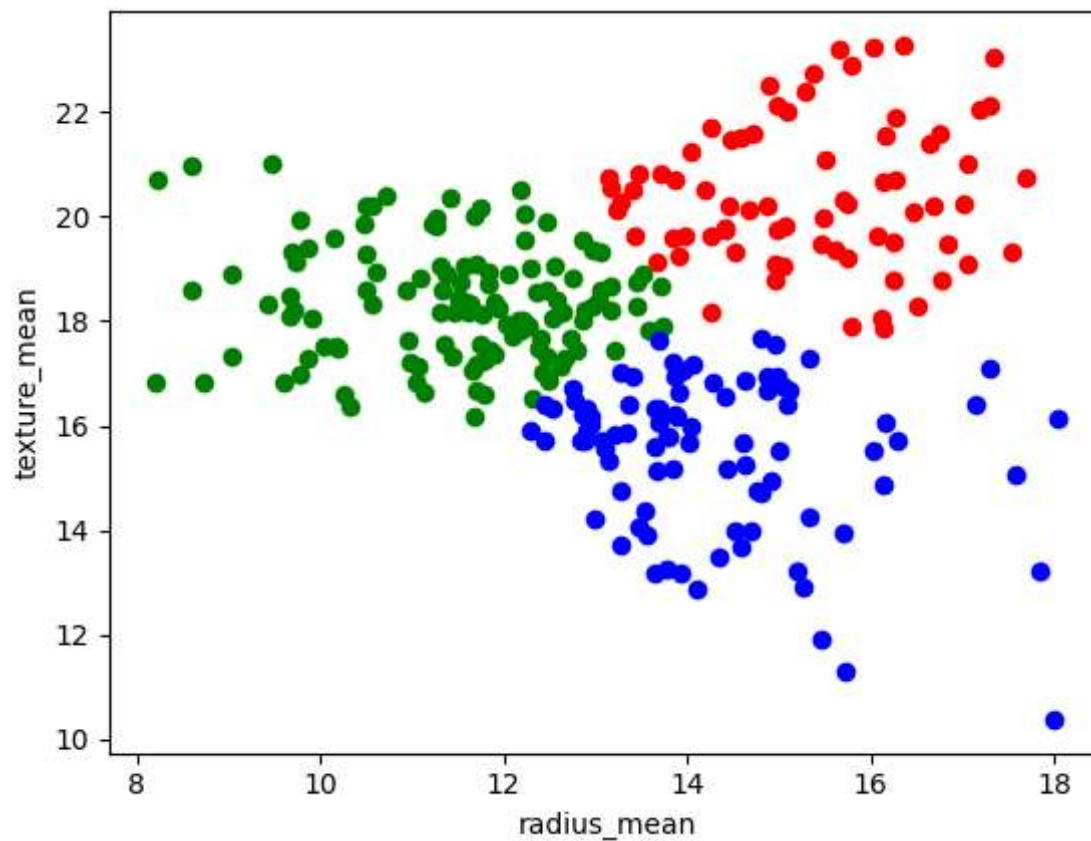
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
0	842302	M	17.99	10.38	122.80	1001.0	0.1
1	842517	M	20.57	17.77	132.90	1326.0	0.0
2	84300903	M	19.69	21.25	130.00	1203.0	0.1
3	84348301	M	11.42	20.38	77.58	386.1	0.1
4	84358402	M	20.29	14.34	135.10	1297.0	0.1

5 rows × 33 columns



```
In [11]: 1 df1=df[df.cluster==0]
2         df2=df[df.cluster==1]
3         df3=df[df.cluster==2]
4         plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
5         plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
6         plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
7         plt.xlabel("radius_mean")
8         plt.ylabel("texture_mean")
9
```

Out[11]: Text(0, 0.5, 'texture_mean')



```
In [12]: 1 from sklearn.preprocessing import MinMaxScaler
2 scaler=MinMaxScaler()
3 scaler.fit(df[["texture_mean"]])
4 df["texture_mean"]=scaler.transform(df[["texture_mean"]])
5 df.head()
6
```

Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	842302	M	17.99	0.022658	122.80	1001.0	0.
1	842517	M	20.57	0.272574	132.90	1326.0	0.
2	84300903	M	19.69	0.390260	130.00	1203.0	0.
3	84348301	M	11.42	0.360839	77.58	386.1	0.
4	84358402	M	20.29	0.156578	135.10	1297.0	0.

5 rows × 33 columns



```
In [13]: 1 scaler.fit(df[["radius_mean"]])
2 df["radius_mean"]=scaler.transform(df[["radius_mean"]])
3 df.head()
4
```

Out[13]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	842302	M	0.521037	0.022658	122.80	1001.0	0.
1	842517	M	0.643144	0.272574	132.90	1326.0	0.
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.
3	84348301	M	0.210090	0.360839	77.58	386.1	0.
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.

5 rows × 33 columns



```
In [14]: 1 y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
          2 y_predicted
          3
```

C:\Users\Niranjan\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[14]: array([2, 7, 7, 1, 7, 2, 7, 3, 3, 3, 3, 2, 0, 3, 3, 4, 3, 3, 7, 2, 2, 5,
                2, 0, 3, 7, 3, 7, 3, 7, 0, 1, 0, 0, 7, 3, 3, 1, 3, 3, 3, 1, 0, 3,
                3, 7, 5, 1, 5, 3, 1, 2, 1, 7, 3, 1, 7, 3, 1, 5, 5, 1, 3, 5, 3, 3,
                1, 1, 5, 2, 7, 5, 0, 2, 2, 3, 2, 7, 0, 1, 1, 2, 6, 0, 5, 7, 3, 0,
                3, 2, 3, 3, 2, 1, 3, 0, 1, 1, 5, 3, 3, 5, 1, 1, 1, 2, 1, 1, 6, 1,
                5, 1, 3, 1, 5, 1, 5, 2, 3, 7, 5, 7, 6, 2, 2, 2, 3, 7, 2, 0, 5, 3,
                3, 2, 7, 3, 1, 5, 2, 5, 5, 7, 1, 2, 5, 5, 1, 3, 2, 2, 3, 1, 5, 5,
                2, 1, 7, 7, 5, 5, 1, 7, 7, 3, 6, 3, 5, 7, 0, 2, 5, 3, 2, 5, 5, 5,
                1, 7, 3, 2, 6, 0, 3, 5, 3, 5, 7, 1, 1, 2, 3, 3, 1, 4, 3, 2, 3, 7,
                7, 3, 1, 7, 6, 3, 1, 2, 1, 7, 3, 2, 7, 1, 6, 0, 3, 2, 1, 1, 7, 0,
                2, 2, 1, 3, 2, 2, 5, 2, 3, 3, 7, 4, 4, 0, 5, 3, 6, 7, 4, 4, 2, 2,
                1, 3, 0, 1, 2, 2, 4, 5, 0, 1, 7, 7, 7, 2, 0, 2, 3, 4, 0, 0, 7, 3,
                7, 0, 1, 3, 2, 1, 2, 5, 6, 5, 0, 1, 5, 7, 2, 2, 0, 5, 7, 7, 2, 1,
                1, 2, 1, 1, 3, 3, 2, 1, 2, 2, 5, 1, 2, 1, 7, 1, 0, 1, 1, 4, 2, 5,
                2, 2, 1, 2, 2, 5, 1, 1, 5, 7, 1, 1, 5, 7, 2, 7, 5, 1, 2, 1, 3, 3,
                2, 1, 1, 5, 1, 7, 2, 7, 1, 6, 2, 5, 5, 7, 5, 5, 1, 2, 5, 5, 1, 3,
                6, 3, 5, 1, 1, 2, 5, 1, 1, 3, 1, 7, 2, 7, 0, 1, 7, 6, 3, 2, 7, 7,
                2, 2, 1, 4, 2, 1, 5, 5, 3, 1, 2, 3, 5, 2, 5, 0, 5, 5, 3, 6, 1, 2,
                1, 1, 5, 1, 7, 5, 1, 2, 2, 1, 2, 3, 7, 1, 1, 1, 1, 3, 4, 1, 1, 3,
                2, 1, 1, 2, 5, 3, 1, 1, 5, 1, 5, 1, 3, 1, 7, 7, 2, 3, 1, 2, 3, 2,
                1, 0, 2, 1, 7, 4, 0, 2, 3, 7, 1, 0, 4, 2, 1, 4, 4, 4, 4, 0, 6,
                4, 1, 1, 3, 3, 1, 0, 1, 1, 4, 2, 4, 5, 2, 3, 2, 5, 7, 1, 3, 2, 2,
                2, 2, 2, 7, 5, 7, 3, 2, 7, 5, 3, 3, 1, 1, 7, 7, 2, 3, 2, 6, 5, 5,
                1, 1, 2, 3, 5, 2, 3, 2, 3, 1, 7, 7, 1, 2, 5, 6, 1, 1, 5, 5, 1, 5,
                2, 5, 1, 1, 2, 7, 1, 7, 3, 4, 4, 4, 5, 3, 3, 4, 3, 3, 5, 5, 1, 4,
                1, 1, 4, 1, 4, 4, 1, 4, 3, 4, 4, 4, 4, 0, 6, 0, 0, 0, 4])
```

```
In [15]: 1 df["New Cluster"]=y_predicted
         2 df.head()
         3
```

Out[15]:

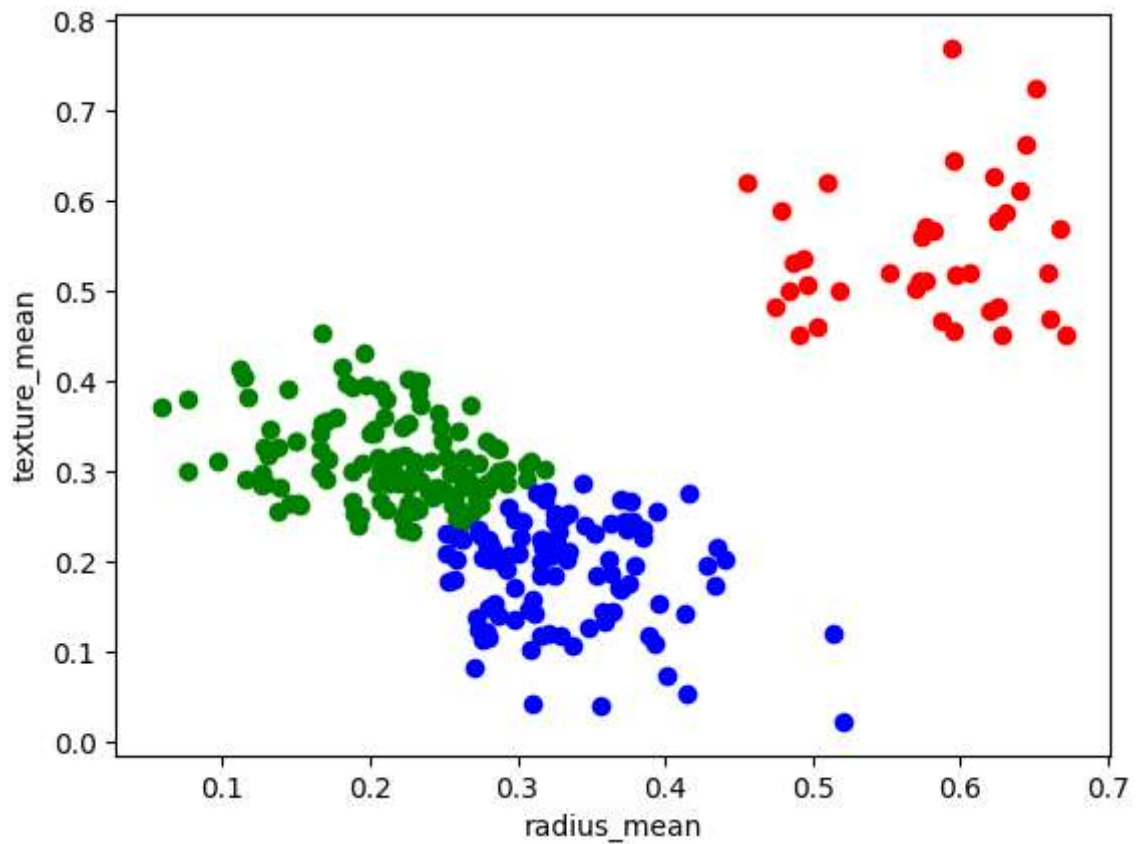
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	842302	M	0.521037	0.022658	122.80	1001.0	0.
1	842517	M	0.643144	0.272574	132.90	1326.0	0.
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.
3	84348301	M	0.210090	0.360839	77.58	386.1	0.
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.

5 rows × 34 columns




```
In [16]: 1 df1=df[df["New Cluster"]==0]
2 df2=df[df["New Cluster"]==1]
3 df3=df[df["New Cluster"]==2]
4 plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
5 plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
6 plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
7 plt.xlabel("radius_mean")
8 plt.ylabel("texture_mean")
```

Out[16]: Text(0, 0.5, 'texture_mean')

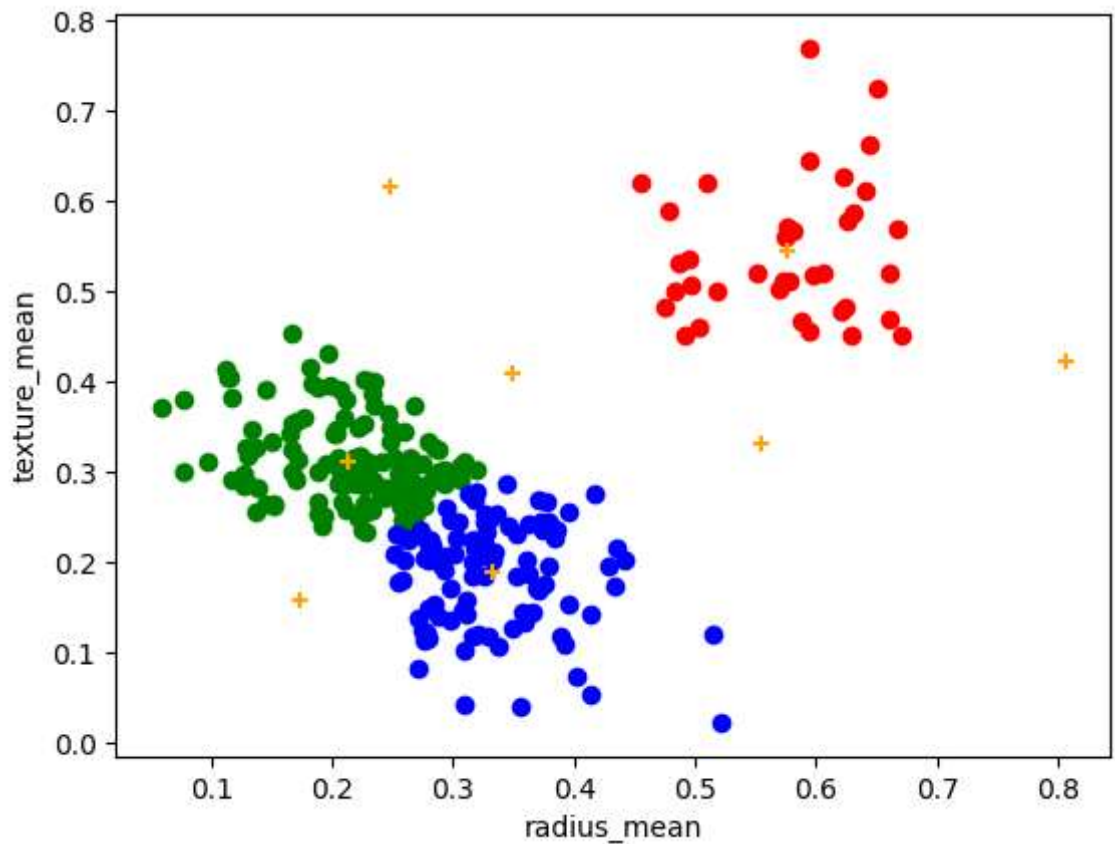


```
In [17]: 1 km.cluster_centers_
```

Out[17]: array([[0.57605341, 0.54408687],
 [0.21276186, 0.31200594],
 [0.33158632, 0.18812362],
 [0.34814903, 0.40844623],
 [0.24753115, 0.61622301],
 [0.17405343, 0.15815861],
 [0.80589822, 0.42316338],
 [0.55409393, 0.33027383]])

```
In [18]: 1 df1=df[df["New Cluster"]==0]
2 df2=df[df["New Cluster"]==1]
3 df3=df[df["New Cluster"]==2]
4 plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
5 plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
6 plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
7 plt.scatter(km.cluster_centers[:,0],km.cluster_centers[:,1],color="orange")
8 plt.xlabel("radius_mean")
9 plt.ylabel("texture_mean")
10
```

Out[18]: Text(0, 0.5, 'texture_mean')



```
In [19]: 1 k_rng=range(1,10)
2 sse=[]
3
```

```
In [20]: 1 for k in k_rng:
2         km=KMeans(n_clusters=k)
3         km.fit(df[["radius_mean","texture_mean"]])
4         sse.append(km.inertia_)
5         #km.inertia_ will give you the value of sum of square error
6         print(sse)
7         plt.plot(k_rng,sse)
8         plt.xlabel("K")
9         plt.ylabel("Sum of Squared Error")
10
```

```
C:\Users\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` 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\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning
```

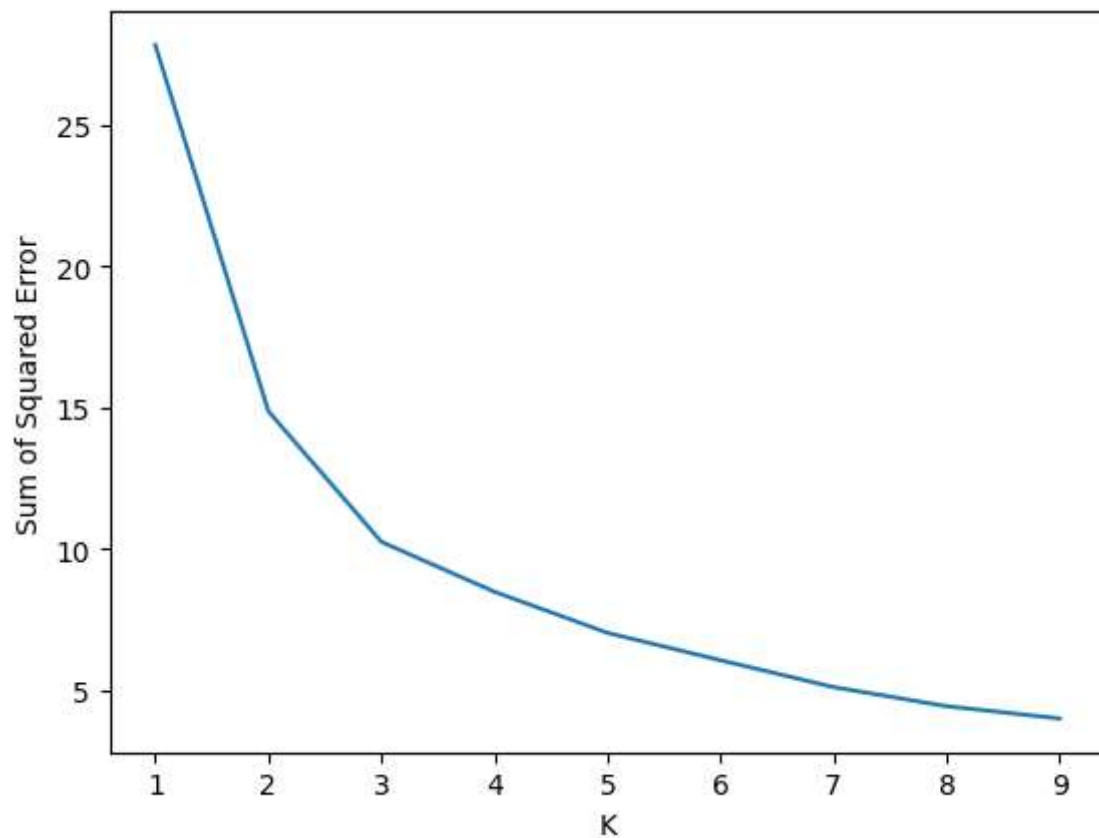
```
warnings.warn(
```

```
C:\Users\Niranjan\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_ini
t` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning
```

```
warnings.warn(
```

[27.817507595043075, 14.87203295827117, 10.252751496105198, 8.49005022151144, 7.035328314875909, 6.067992241432947, 5.117380844808091, 4.44301570025843, 4.0073949917195915]

```
Out[20]: Text(0, 0.5, 'Sum of Squared Error')
```



CONCLUSION

for the given dataset we can use multiple models, for that models we get different types of accuracies but that accuracy is not good so, that's why we will take it as a clustering and done with K-Means Clustering

In []:

1