

In [1]:

```
1 import pandas as pd
2 from matplotlib import pyplot as plt
3 %matplotlib inline
```

In [3]:

```
1 df=pd.read_csv(r"C:\Users\91628\Downloads\BreastCancerPrediction.csv")
2 df
```

Out[3]:

	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 × 33 columns

In [4]:

```
1 df.head()
```

Out[4]:

	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 [5]:

```
1 df.tail()
```

Out[5]:

	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 × 33 columns



In [8]:

```
1 df.drop(['Unnamed: 32'],axis=1)
```

Out[8]:

	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

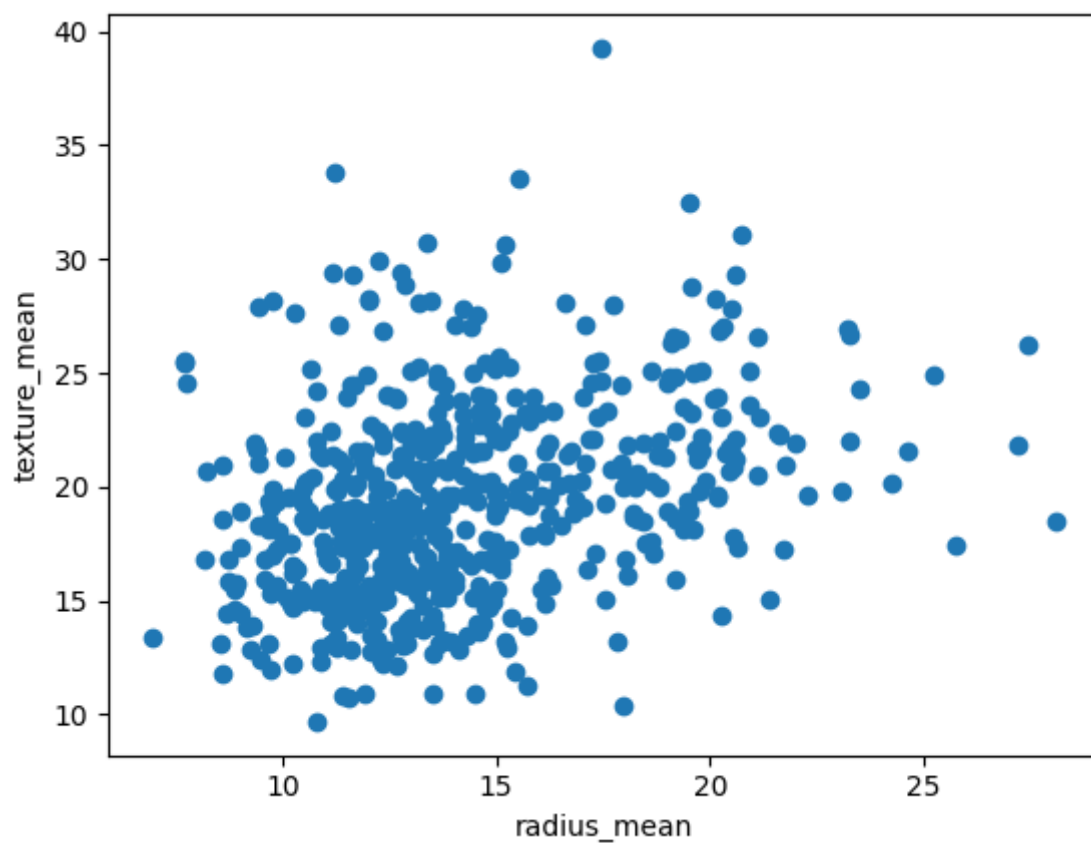


In [9]:

```
1 plt.scatter(df["radius_mean"],df["texture_mean"])
2 plt.xlabel("radius_mean")
3 plt.ylabel("texture_mean")
```

Out[9]:

Text(0, 0.5, 'texture_mean')



In [10]:

```
1 from sklearn.cluster import KMeans
2 km=KMeans()
3 km
```

Out[10]:

▼ KMeans
KMeans()

In [11]:

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

C:\Users\91628\AppData\Local\Programs\Python\Python311\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[11]:

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

In [12]:

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

Out[12]:

	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	

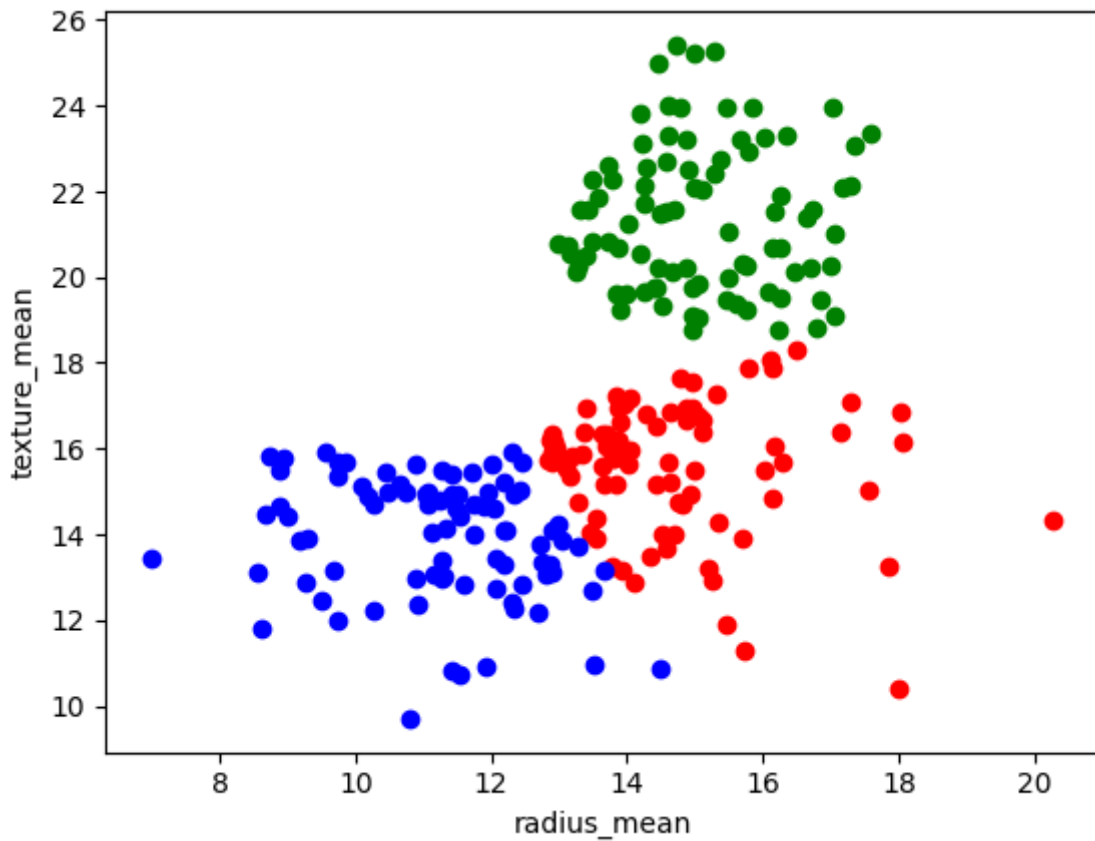
5 rows × 34 columns

In [13]:

```
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")
```

Out[13]:

Text(0, 0.5, 'texture_mean')



In [14]:

```
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()
```

Out[14]:

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

5 rows × 34 columns



In [15]:

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

Out[15]:

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

5 rows × 34 columns



In [16]:

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

C:\Users\91628\AppData\Local\Programs\Python\Python311\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[16]:

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

In [17]:

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

Out[17]:

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

5 rows × 35 columns

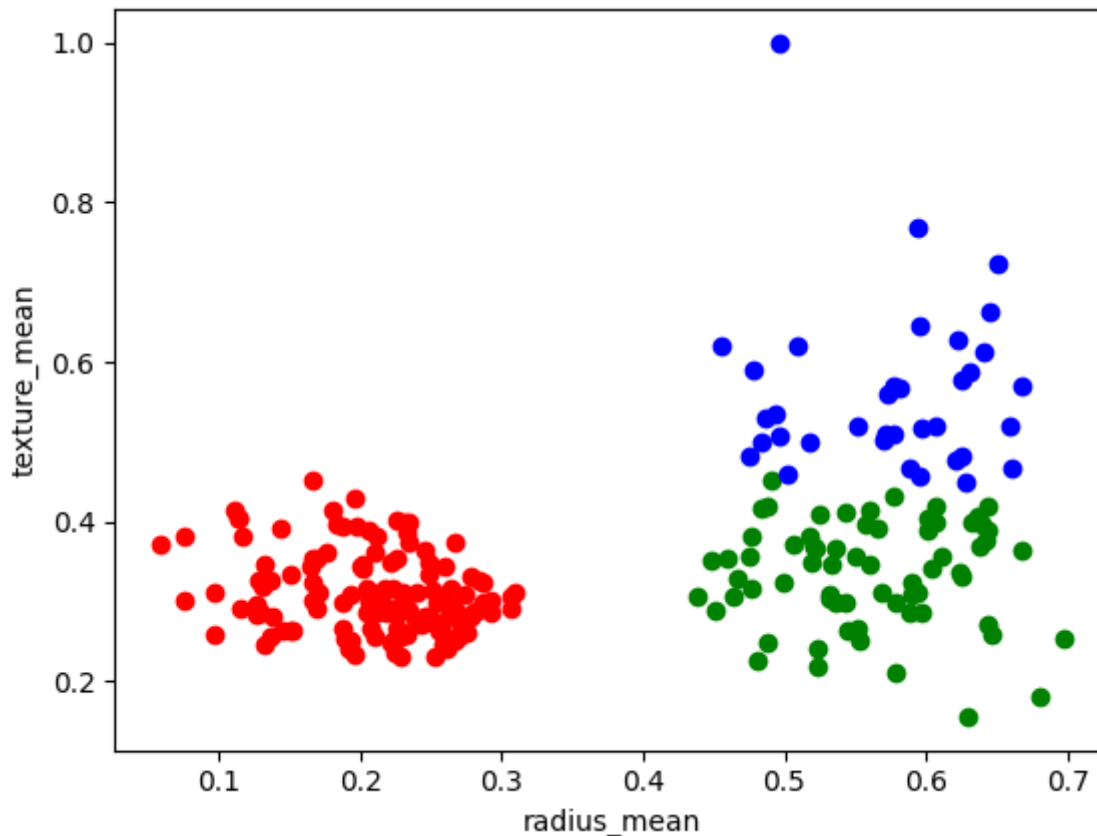


In [18]:

```
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.xlabel("radius_mean")
8 plt.ylabel("texture_mean")
```

Out[18]:

Text(0, 0.5, 'texture_mean')



In [19]:

```
1 km.cluster_centers_
```

Out[19]:

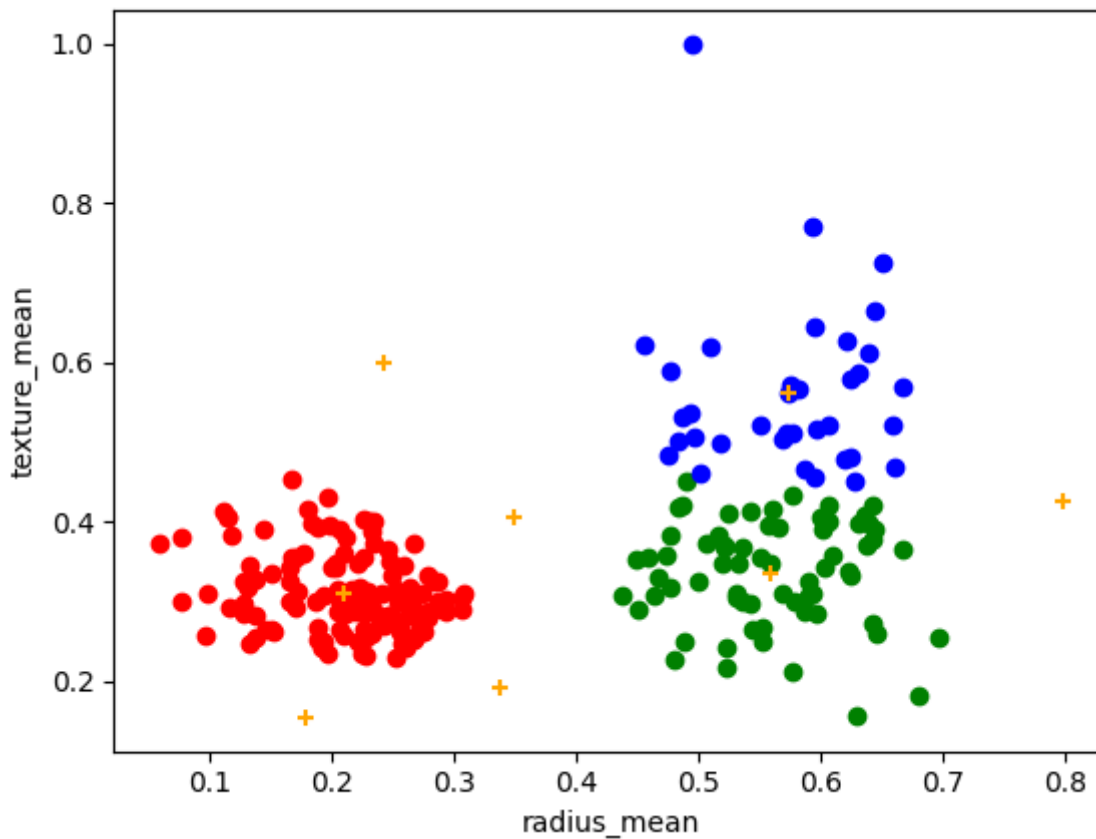
```
array([[0.21063269, 0.30993347],
       [0.55921329, 0.3357437 ],
       [0.57355872, 0.56191523],
       [0.33809493, 0.19063439],
       [0.17850466, 0.15444707],
       [0.24279689, 0.59913388],
       [0.79840767, 0.42469846],
       [0.35002767, 0.40470899]])
```

In [20]:

```
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",marker=
8 plt.xlabel("radius_mean")
9 plt.ylabel("texture_mean")
```

Out[20]:

Text(0, 0.5, 'texture_mean')



In [21]:

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

In [22]:

```
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")
```

C:\Users\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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\91628\AppData\Local\Programs\Python\Python311\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(

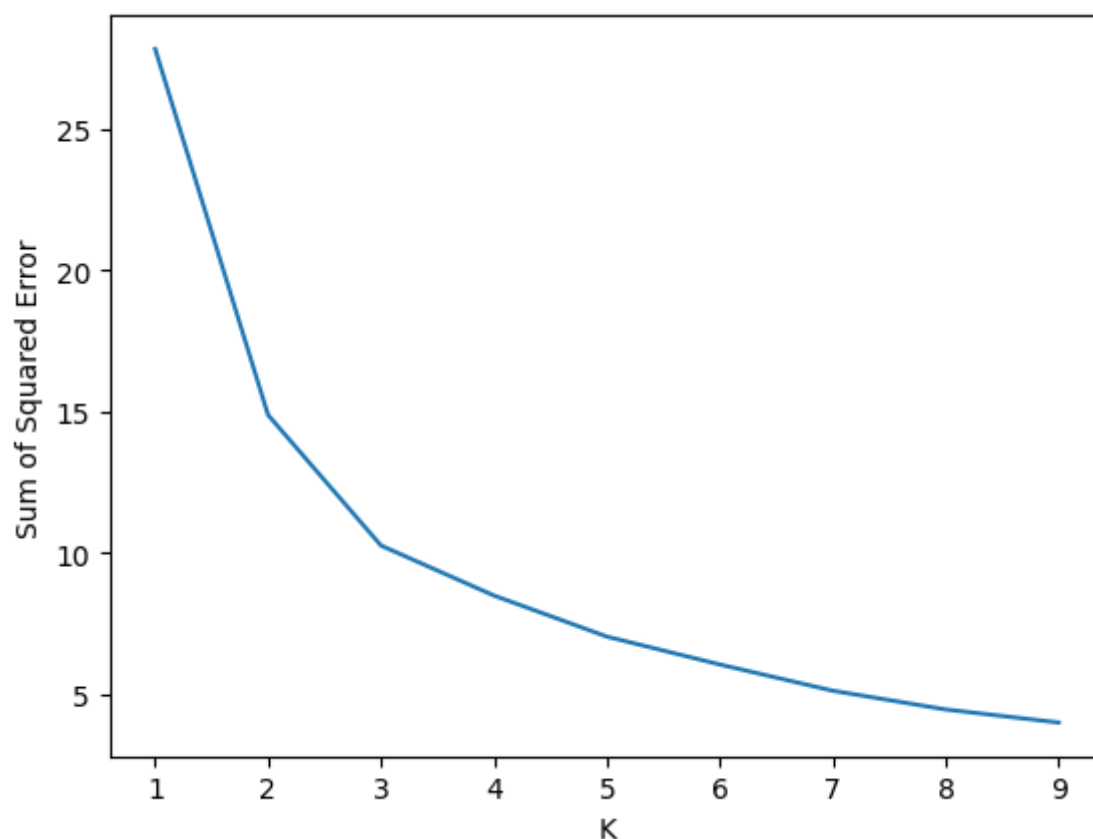
C:\Users\91628\AppData\Local\Programs\Python\Python311\Lib\site-packages\s
klearn\cluster_kmeans.py:870: FutureWarning: The default value of `n_init`
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning

warnings.warn(

```
[27.81750759504308, 14.87203295827117, 10.2527514961052, 8.484725277027605, 7.032096080906509, 6.046311983660964, 5.118144061229121, 4.45499490621673, 3.993976423073571]
```

Out[22]:

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



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

In []:

1