Breast Cancer Prediction

In [1]:

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

In [2]:

```
df=pd.read_csv(r"C:\Users\lenovo\Downloads\BreastCancerPrediction.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 × 33 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 × 33 columns

In [4]:

```
df.tail()
```

Out[4]:

_worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst
26.40	166.10	2027.0	0.14100	0.21130	0.4107
38.25	155.00	1731.0	0.11660	0.19220	0.3215
34.12	126.70	1124.0	0.11390	0.30940	0.3403
39.42	184.60	1821.0	0.16500	0.86810	0.9387
30.37	59.16	268.6	0.08996	0.06444	0.0000

In [5]:

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

Out[5]:

	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	

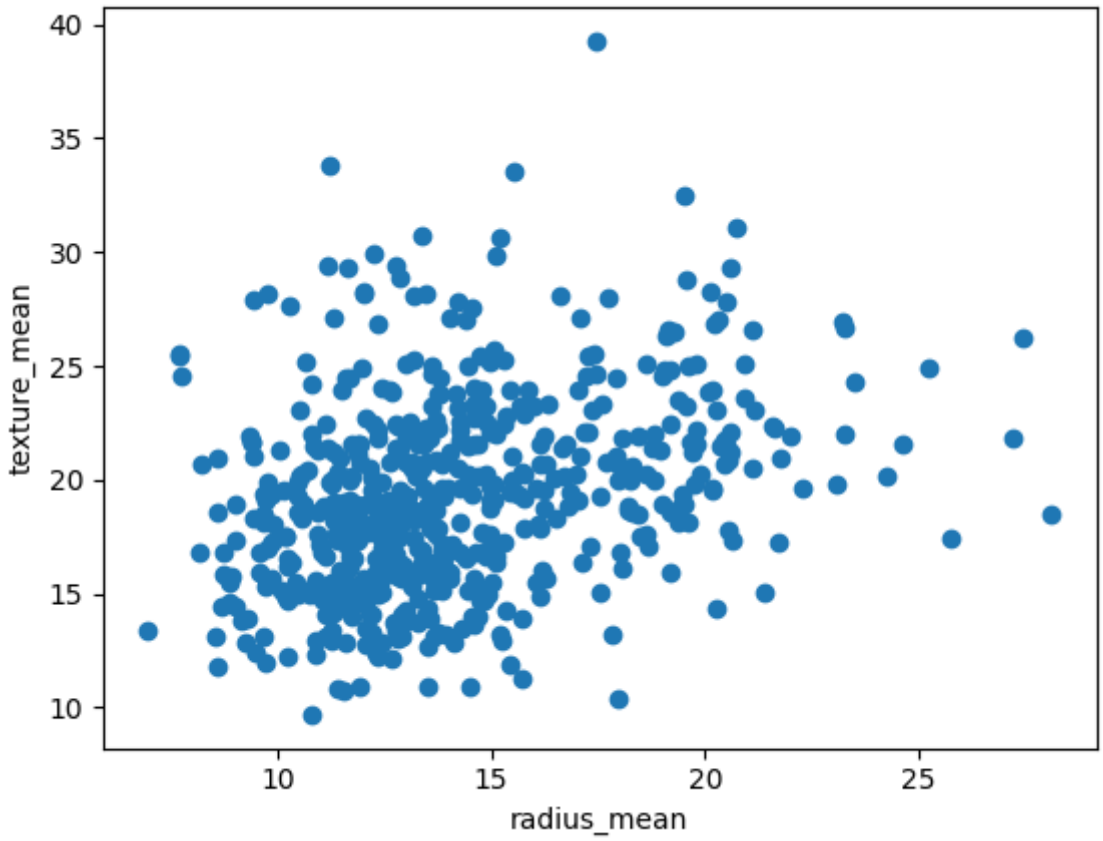
569 rows × 32 columns

In [6]:

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

Out[6]:

```
Text(0, 0.5, 'texture_mean')
```



In [7]:

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

Out[7]:

```
KMeans()
```

In [8]:

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

```
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
```

Out[8]:

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

In [9]:

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

Out[9]:

	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	

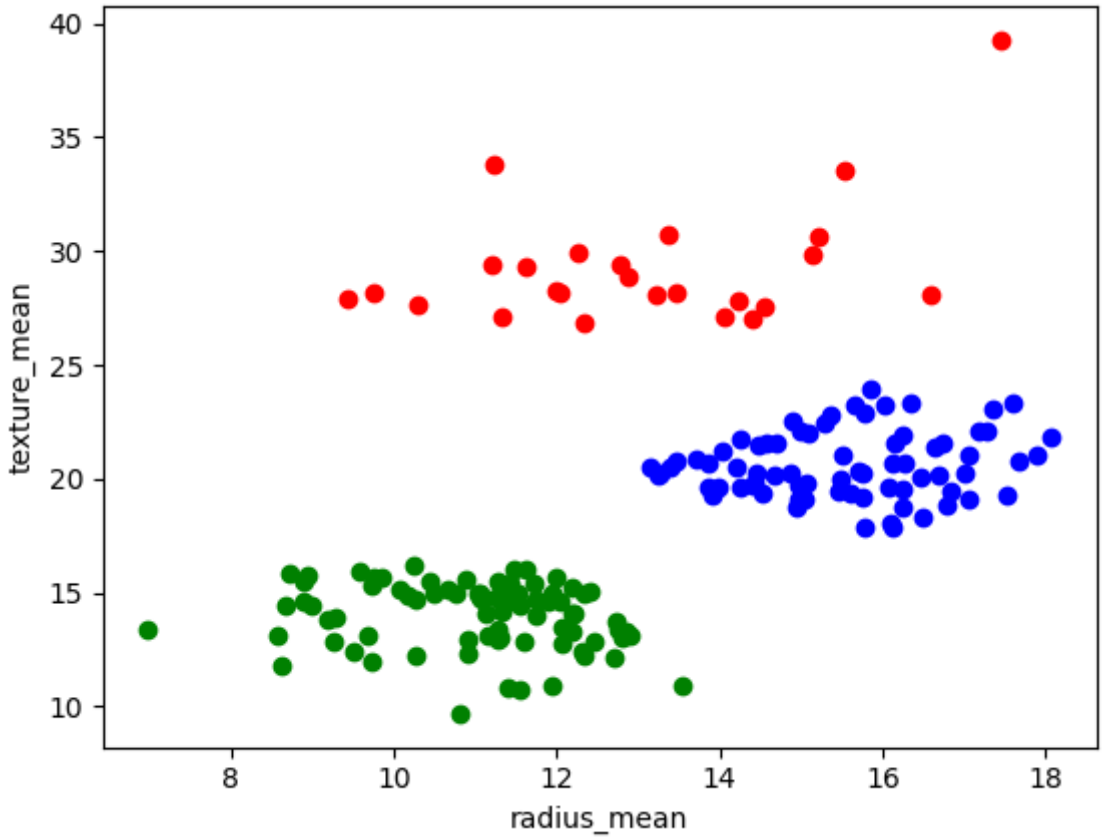
5 rows × 34 columns

In [10]:

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

Out[10]:

```
Text(0, 0.5, 'texture_mean')
```



In [11]:

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

Out[11]:

	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 [12]:

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

Out[12]:

	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 [13]:

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

```
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
```

Out[13]:

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

In [14]:

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

Out[14]:

	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	

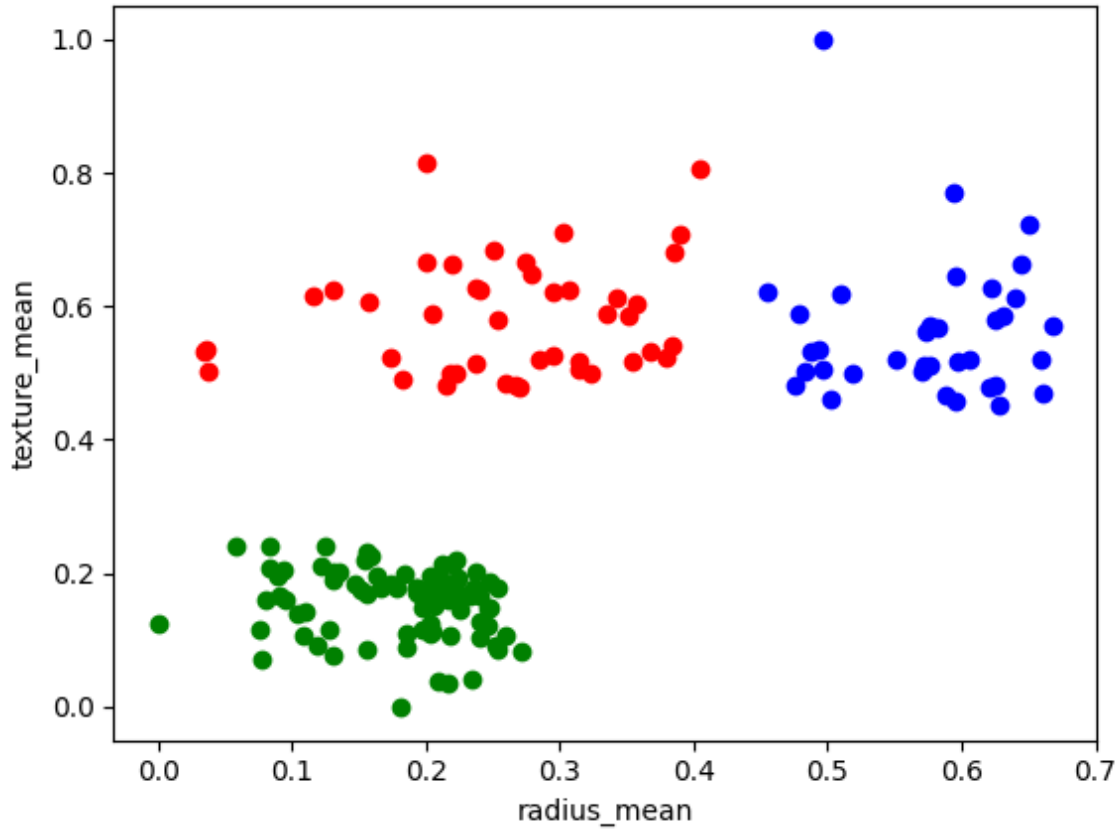
5 rows × 35 columns

In [15]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[15]:

```
Text(0, 0.5, 'texture_mean')
```



In [16]:

```
km.cluster_centers_
```

Out[16]:

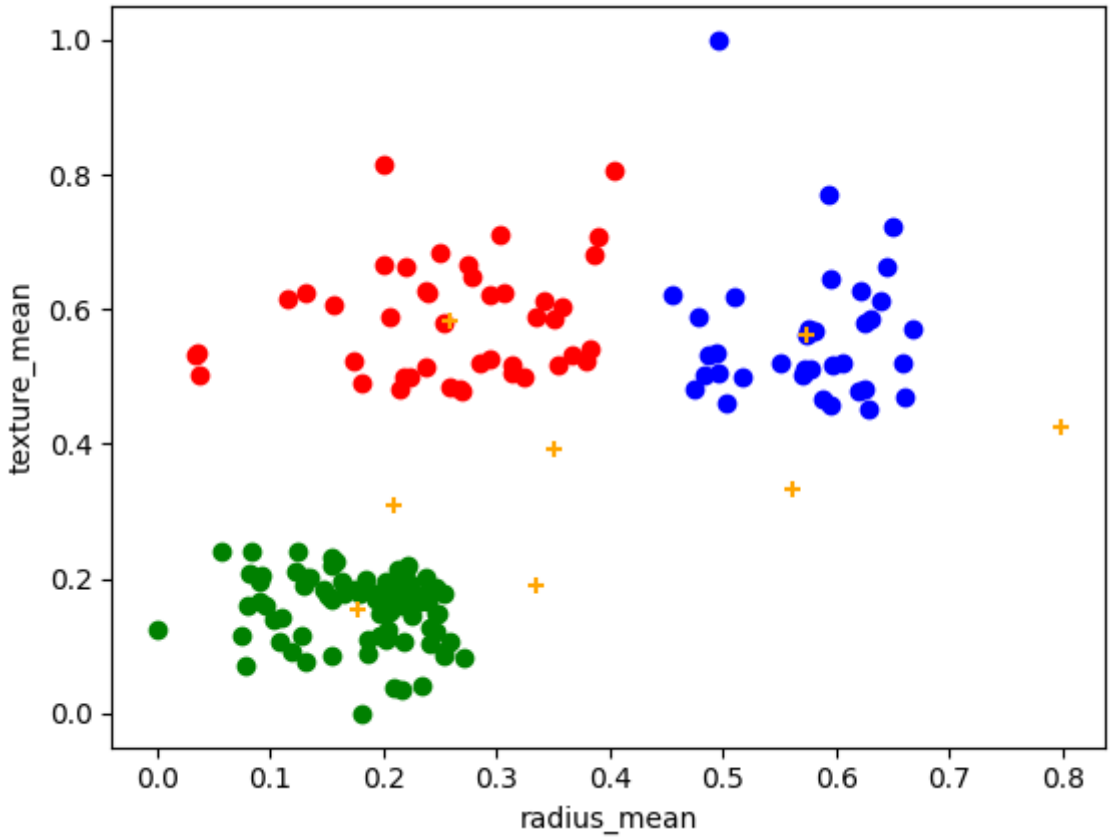
```
array([[0.2590623 , 0.58293879],
       [0.17750575, 0.15412045],
       [0.57355872, 0.56191523],
       [0.33570532, 0.19063107],
       [0.56180336, 0.33362777],
       [0.20867092, 0.3094643 ],
       [0.79840767, 0.42469846],
       [0.35173159, 0.39188367]])
```

In [17]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker="+")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[17]:

```
Text(0, 0.5, 'texture_mean')
```



In [18]:

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

In [19]:

```
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["radius_mean","texture_mean"]])
 sse.append(km.inertia_)
```

```
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
```

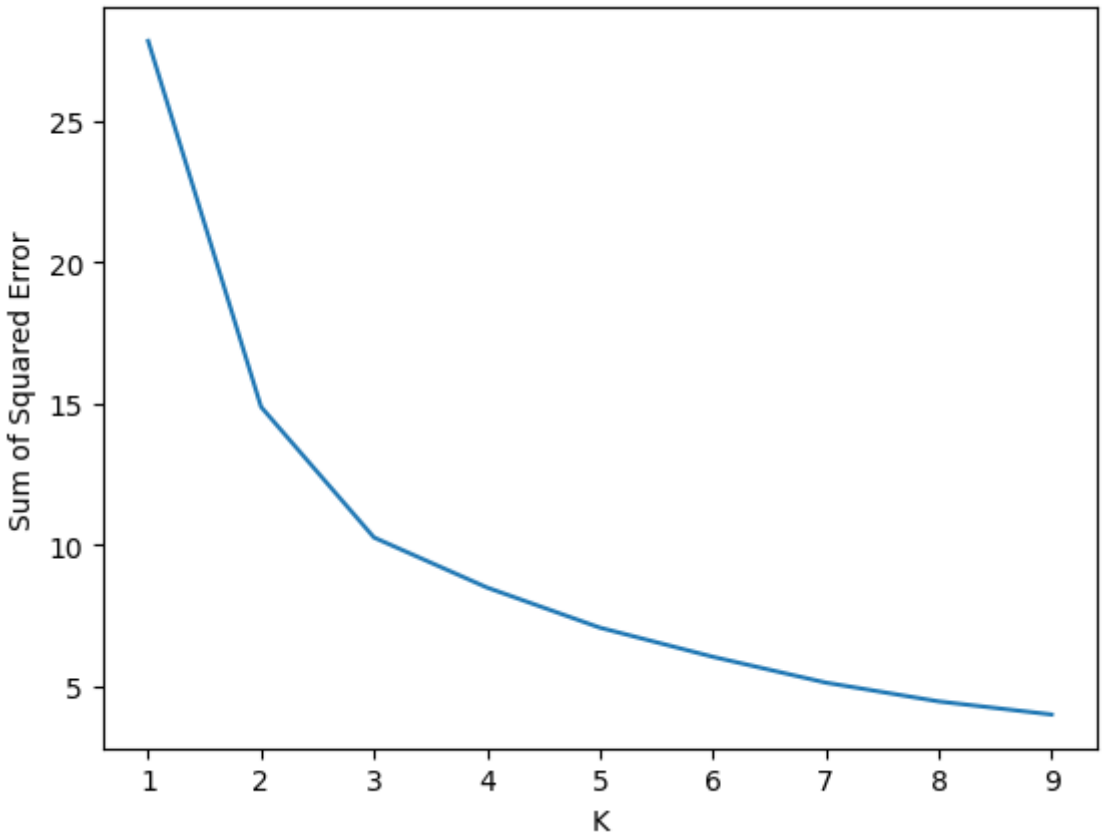
In [20]:

```
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
[27.817507595043075, 14.872296449956036, 10.252751496105198, 8.48435723386
47, 7.063384537187425, 6.032652635605176, 5.119533799348032, 4.45475370481
9518, 3.9912941365012125]
```

Out[20]:

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



Conclusion:- In Above DataSet we can use any models to get different accuracies.But by using clustering technique we can get best accuracy for the Dataset. Therefore we can conclude that breast Cancer prediction DataSet is best fit for "k-Means clustering Model.

In [ ]: