

# Breast Cancer Prediction

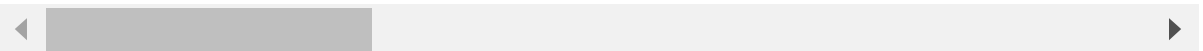
```
In [1]: import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

```
In [2]: df=pd.read_csv(r"C:\Users\venka\OneDrive\Documents\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	(
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

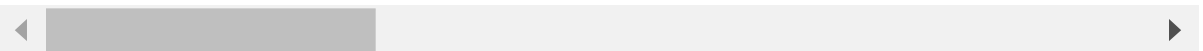


```
In [3]: df.head()
```

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	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

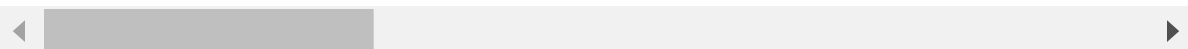


In [4]: `df.tail()`

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
<b>564</b>	926424	M	21.56	22.39	142.00	1479.0	0.1
<b>565</b>	926682	M	20.13	28.25	131.20	1261.0	0.0
<b>566</b>	926954	M	16.60	28.08	108.30	858.1	0.0
<b>567</b>	927241	M	20.60	29.33	140.10	1265.0	0.1
<b>568</b>	92751	B	7.76	24.54	47.92	181.0	0.0

5 rows × 33 columns

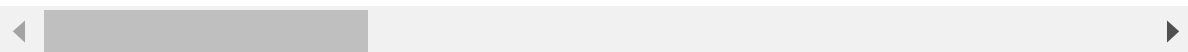


In [5]: `df.drop(['Unnamed: 32'],axis=1)`

Out[5]:

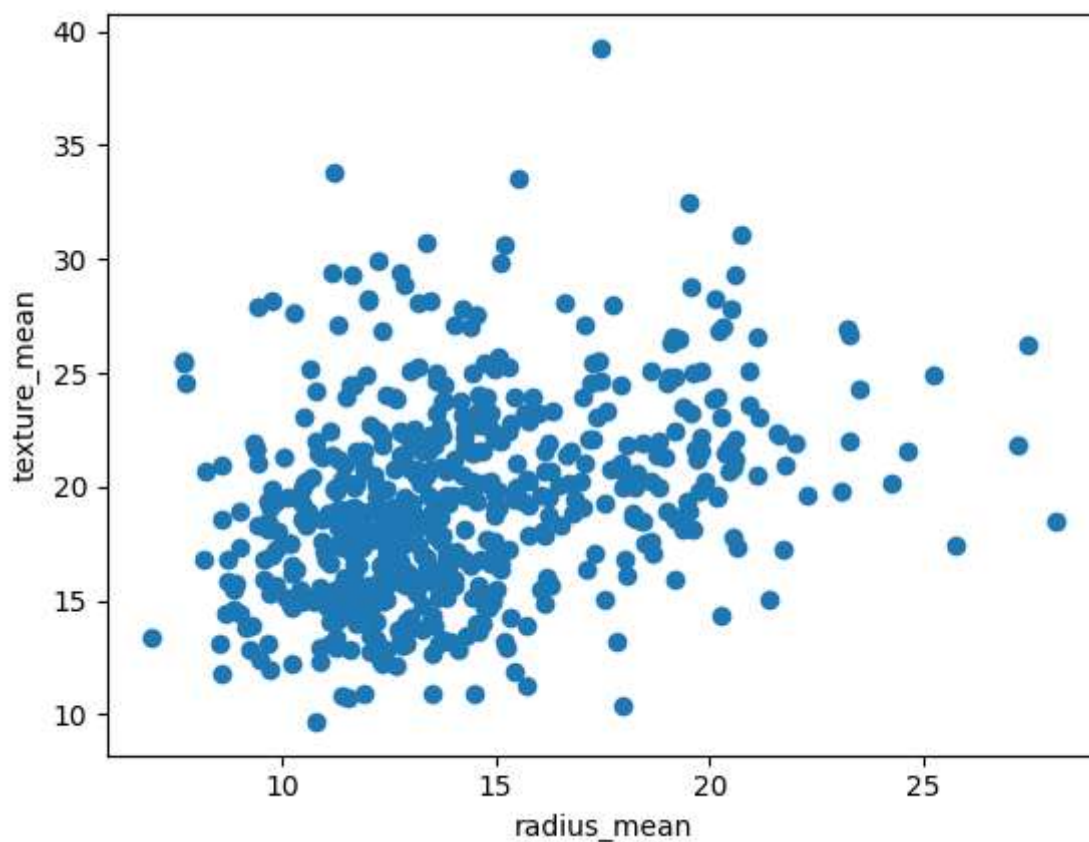
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
<b>0</b>	842302	M	17.99	10.38	122.80	1001.0	(
<b>1</b>	842517	M	20.57	17.77	132.90	1326.0	(
<b>2</b>	84300903	M	19.69	21.25	130.00	1203.0	(
<b>3</b>	84348301	M	11.42	20.38	77.58	386.1	(
<b>4</b>	84358402	M	20.29	14.34	135.10	1297.0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



KMeans()


```

```
In [8]: y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

C:\Users\venka\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(
```

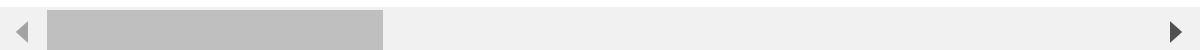
```
Out[8]: array([1, 4, 4, 7, 4, 1, 4, 3, 0, 0, 3, 3, 2, 3, 0, 5, 3, 3, 4, 1, 1, 6,
1, 2, 3, 1, 3, 4, 0, 1, 2, 7, 3, 2, 3, 3, 3, 7, 0, 3, 0, 0, 2, 3,
0, 4, 7, 7, 6, 0, 0, 1, 7, 4, 3, 7, 4, 3, 7, 6, 6, 7, 0, 6, 0, 0,
7, 7, 7, 1, 4, 6, 2, 1, 7, 3, 6, 1, 2, 7, 0, 1, 2, 2, 6, 4, 3, 2,
0, 1, 0, 3, 1, 7, 3, 2, 7, 7, 6, 3, 0, 6, 7, 7, 7, 1, 7, 7, 4, 0,
7, 0, 3, 7, 6, 0, 6, 1, 3, 3, 6, 4, 4, 1, 1, 1, 0, 4, 1, 2, 6, 3,
3, 1, 4, 0, 7, 6, 1, 6, 6, 3, 7, 1, 6, 6, 7, 3, 1, 7, 0, 7, 6, 6,
1, 7, 3, 3, 6, 6, 7, 4, 4, 0, 4, 3, 6, 3, 2, 1, 6, 7, 1, 6, 6, 6,
7, 3, 0, 6, 4, 2, 3, 6, 3, 6, 4, 7, 7, 1, 0, 0, 7, 5, 0, 1, 0, 3,
4, 3, 7, 3, 2, 0, 7, 1, 7, 3, 0, 1, 4, 7, 4, 2, 0, 1, 7, 7, 4, 2,
1, 1, 7, 3, 1, 1, 6, 1, 0, 0, 3, 5, 5, 2, 6, 3, 2, 4, 5, 5, 1, 6,
7, 0, 2, 7, 7, 6, 0, 6, 2, 7, 4, 1, 4, 1, 2, 1, 3, 5, 2, 3, 3, 3,
3, 2, 7, 0, 1, 7, 1, 6, 4, 6, 2, 7, 6, 4, 7, 1, 2, 6, 4, 3, 1, 7,
0, 6, 7, 7, 3, 3, 1, 7, 6, 1, 6, 7, 7, 0, 4, 7, 2, 7, 7, 0, 1, 6,
6, 6, 7, 1, 6, 6, 7, 7, 6, 4, 7, 7, 6, 4, 6, 4, 6, 7, 1, 7, 3, 3,
1, 7, 7, 6, 7, 3, 1, 4, 7, 2, 1, 7, 6, 4, 6, 6, 7, 1, 6, 6, 7, 3,
4, 0, 6, 7, 7, 1, 6, 7, 7, 0, 7, 3, 1, 4, 2, 7, 4, 4, 3, 1, 4, 4,
1, 1, 7, 5, 1, 7, 6, 6, 0, 7, 1, 0, 6, 1, 6, 2, 6, 7, 3, 4, 7, 1,
7, 7, 6, 7, 3, 6, 7, 1, 6, 7, 1, 0, 3, 7, 7, 7, 0, 3, 5, 0, 0, 3,
6, 0, 7, 1, 6, 7, 7, 0, 6, 0, 7, 7, 3, 7, 4, 4, 1, 3, 7, 1, 3, 1,
7, 2, 1, 7, 4, 0, 2, 1, 3, 4, 0, 2, 5, 1, 7, 5, 5, 0, 0, 5, 2, 2,
5, 7, 7, 7, 0, 7, 3, 7, 7, 5, 1, 5, 6, 1, 3, 1, 6, 3, 7, 3, 1, 7,
1, 7, 1, 4, 7, 3, 0, 1, 3, 6, 0, 3, 7, 7, 4, 4, 1, 0, 1, 4, 6, 6,
7, 7, 1, 0, 6, 1, 3, 1, 3, 7, 4, 4, 7, 7, 6, 4, 7, 7, 6, 6, 7, 6,
1, 6, 7, 7, 1, 4, 7, 4, 0, 0, 0, 0, 6, 0, 0, 5, 3, 0, 7, 7, 7, 0,
0, 0, 5, 0, 5, 5, 7, 5, 0, 0, 5, 5, 5, 2, 4, 2, 5, 2, 0])
```

```
In [9]: df["cluster"]=y_predicted
df.head()
```

```
Out[9]:
```

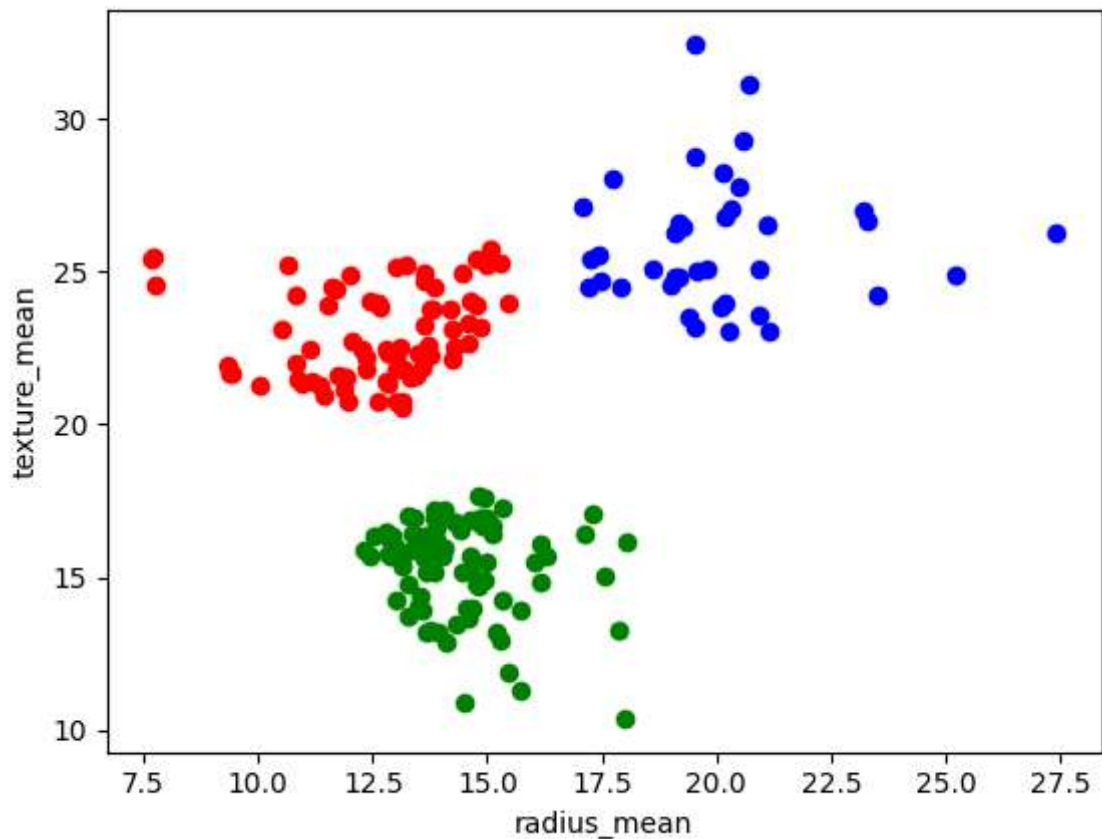
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
0	842302	M	17.99	10.38	122.80	1001.0	0.1
1	842517	M	20.57	17.77	132.90	1326.0	0.0
2	84300903	M	19.69	21.25	130.00	1203.0	0.1
3	84348301	M	11.42	20.38	77.58	386.1	0.1
4	84358402	M	20.29	14.34	135.10	1297.0	0.1

5 rows × 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')

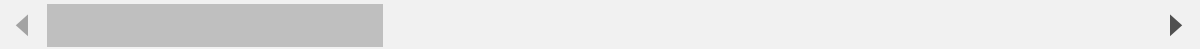


```
In [11]: from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["texture_mean"]])
df["texture_mean"]=scaler.transform(df[["texture_mean"]])
df.head()
```

Out[11]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
0	842302	M	17.99	0.022658	122.80	1001.0	0.1
1	842517	M	20.57	0.272574	132.90	1326.0	0.0
2	84300903	M	19.69	0.390260	130.00	1203.0	0.1
3	84348301	M	11.42	0.360839	77.58	386.1	0.1
4	84358402	M	20.29	0.156578	135.10	1297.0	0.1

5 rows × 34 columns

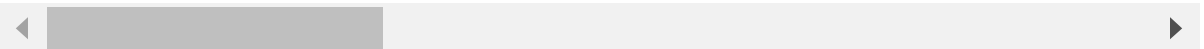


```
In [12]: scaler.fit(df[["radius_mean"]])
df["radius_mean"]=scaler.transform(df[["radius_mean"]])
df.head()
```

Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
0	842302	M	0.521037	0.022658	122.80	1001.0	0.1
1	842517	M	0.643144	0.272574	132.90	1326.0	0.0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.1
3	84348301	M	0.210090	0.360839	77.58	386.1	0.1
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.1

5 rows × 34 columns



```
In [13]: y_predicted=km.fit_predict(df[["radius_mean", "texture_mean"]])
y_predicted
```

C:\Users\venka\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(

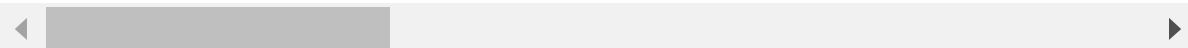
```
Out[13]: array([4, 0, 0, 1, 0, 4, 0, 6, 6, 6, 6, 4, 7, 6, 6, 2, 6, 6, 0, 4, 4, 3,
4, 5, 0, 0, 6, 0, 6, 0, 7, 1, 7, 7, 0, 0, 6, 1, 6, 6, 6, 1, 7, 6,
6, 0, 3, 1, 3, 6, 1, 4, 1, 0, 6, 1, 0, 6, 1, 3, 3, 1, 6, 3, 6, 6,
1, 1, 3, 4, 0, 3, 7, 4, 4, 0, 4, 0, 7, 1, 1, 4, 5, 7, 3, 0, 6, 7,
6, 4, 6, 6, 4, 1, 6, 7, 1, 1, 3, 6, 6, 3, 1, 1, 1, 4, 1, 1, 5, 1,
3, 1, 6, 1, 3, 1, 3, 4, 6, 0, 3, 0, 5, 4, 4, 4, 6, 0, 4, 7, 3, 6,
6, 4, 0, 6, 1, 3, 4, 3, 3, 0, 1, 4, 3, 3, 1, 6, 4, 4, 6, 1, 3, 3,
4, 1, 0, 0, 3, 3, 1, 0, 0, 6, 5, 6, 3, 0, 7, 4, 3, 6, 4, 3, 3, 3,
1, 0, 6, 4, 5, 7, 6, 3, 6, 3, 0, 1, 1, 4, 6, 6, 1, 2, 6, 4, 6, 0,
7, 6, 1, 0, 5, 6, 1, 4, 1, 0, 6, 4, 5, 1, 5, 7, 6, 4, 1, 1, 0, 7,
4, 4, 1, 6, 4, 4, 3, 4, 6, 6, 0, 2, 2, 7, 3, 6, 5, 5, 2, 2, 4, 4,
1, 6, 7, 1, 4, 4, 2, 3, 5, 1, 0, 0, 0, 4, 7, 4, 6, 2, 7, 7, 0, 6,
0, 7, 1, 6, 4, 1, 4, 3, 5, 3, 7, 1, 3, 0, 4, 4, 7, 3, 0, 0, 4, 1,
1, 4, 1, 1, 6, 6, 4, 1, 4, 4, 3, 1, 4, 1, 0, 1, 7, 1, 1, 2, 4, 3,
4, 4, 1, 4, 4, 3, 1, 1, 3, 0, 1, 1, 3, 0, 4, 5, 3, 1, 4, 1, 6, 6,
4, 1, 1, 3, 1, 0, 4, 0, 1, 5, 4, 3, 3, 0, 3, 3, 1, 4, 3, 3, 1, 6,
5, 6, 3, 1, 1, 4, 3, 1, 1, 6, 1, 0, 4, 5, 7, 1, 5, 5, 6, 4, 0, 0,
4, 4, 1, 2, 4, 1, 3, 3, 6, 1, 4, 6, 3, 4, 3, 7, 3, 3, 6, 5, 1, 4,
1, 1, 3, 1, 0, 3, 1, 4, 4, 1, 4, 6, 0, 1, 1, 1, 1, 6, 2, 1, 1, 6,
4, 1, 1, 4, 3, 6, 1, 1, 3, 1, 3, 1, 6, 1, 0, 0, 4, 6, 1, 4, 6, 4,
1, 7, 4, 1, 0, 2, 7, 4, 6, 5, 1, 7, 2, 4, 1, 2, 2, 2, 2, 2, 7, 5,
2, 1, 1, 6, 6, 1, 7, 1, 1, 2, 4, 2, 3, 4, 6, 4, 3, 0, 1, 6, 4, 4,
4, 4, 4, 0, 3, 0, 6, 4, 0, 3, 6, 6, 1, 1, 0, 5, 4, 6, 4, 5, 3, 3,
1, 1, 4, 6, 3, 4, 6, 4, 6, 1, 0, 0, 1, 4, 3, 5, 1, 1, 3, 3, 1, 3,
4, 3, 1, 1, 4, 0, 1, 5, 6, 2, 2, 2, 3, 6, 6, 2, 6, 6, 3, 3, 1, 2,
1, 1, 2, 1, 2, 2, 1, 2, 6, 2, 2, 2, 2, 7, 5, 7, 7, 7, 2])
```

```
In [14]: df["New Cluster"]=y_predicted
df.head()
```

```
Out[14]:
```

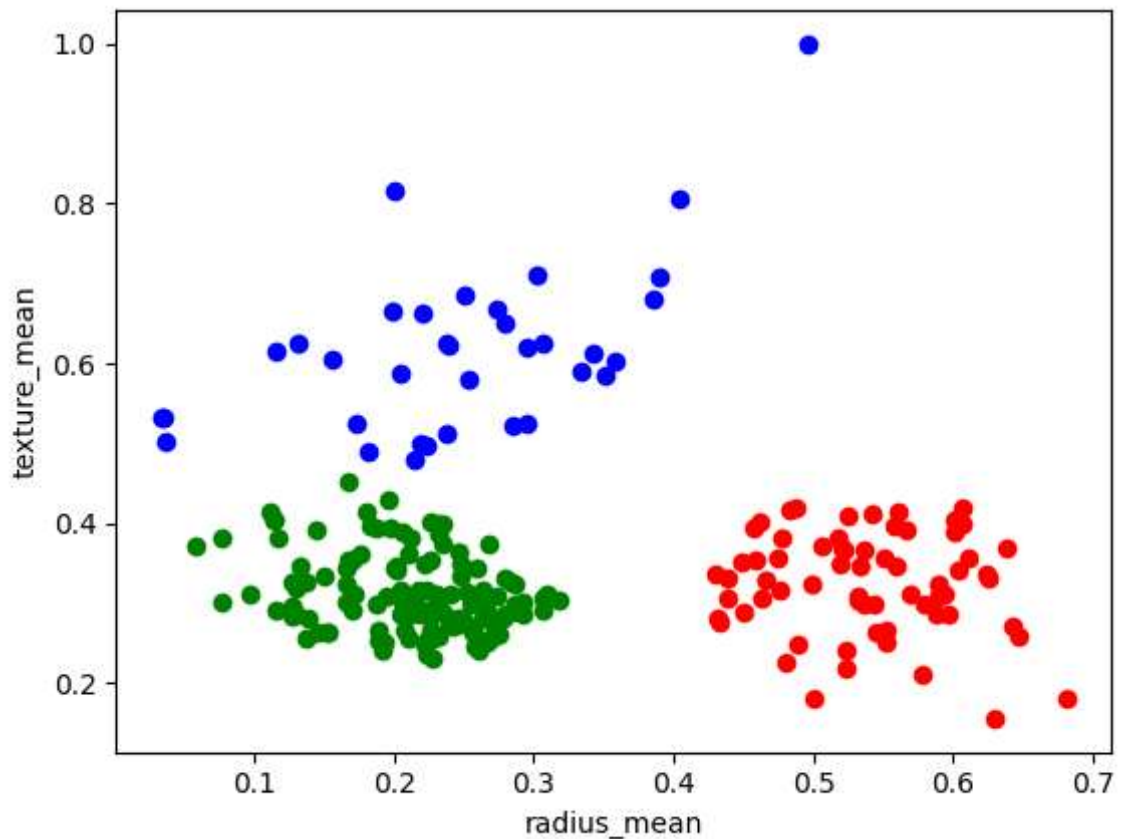
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_r
0	842302	M	0.521037	0.022658	122.80	1001.0	0.1
1	842517	M	0.643144	0.272574	132.90	1326.0	0.0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.1
3	84348301	M	0.210090	0.360839	77.58	386.1	0.1
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.1

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')



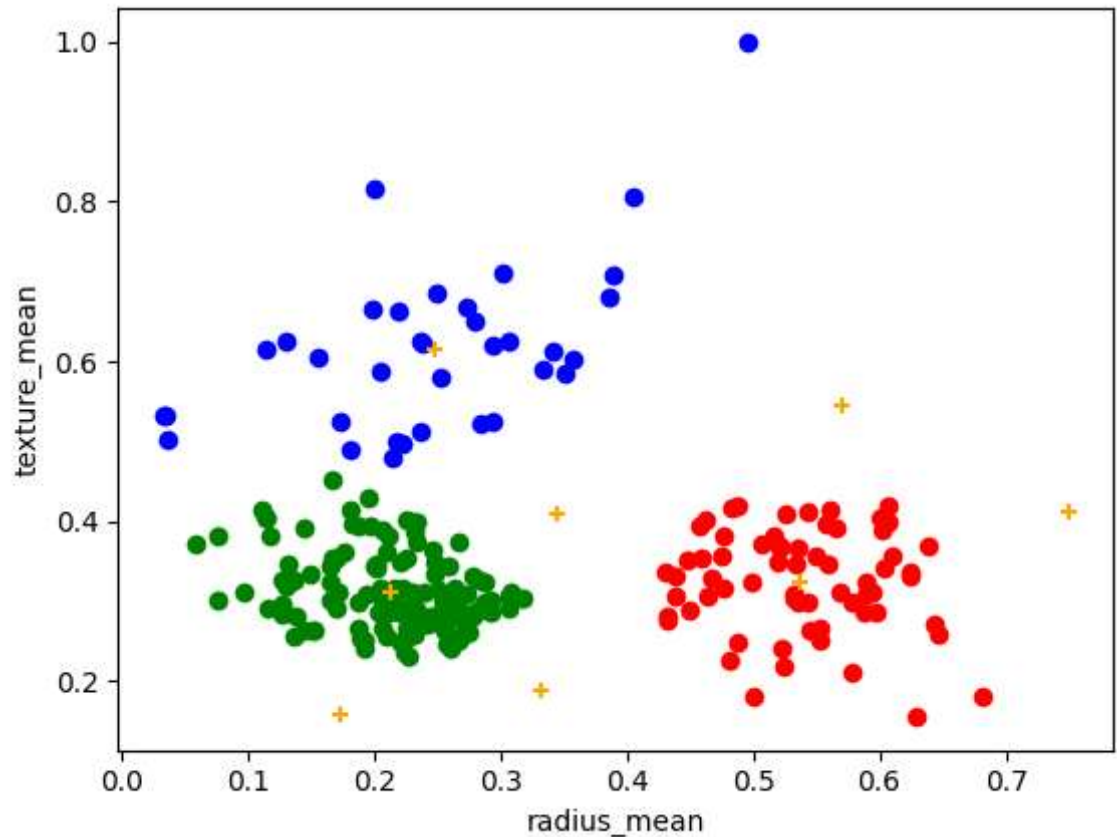
```
In [16]: km.cluster_centers_
```

Out[16]: array([[0.53707081, 0.32518925],  
 [0.21276186, 0.31200594],  
 [0.24753115, 0.61622301],  
 [0.17405343, 0.15815861],  
 [0.33158632, 0.18812362],  
 [0.74945792, 0.41205426],  
 [0.34493624, 0.40942278],  
 [0.57110028, 0.54567317]])



```
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",)
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_)
          #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\venka\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\venka\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\venka\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\venka\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\venka\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\venka\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\venka\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(

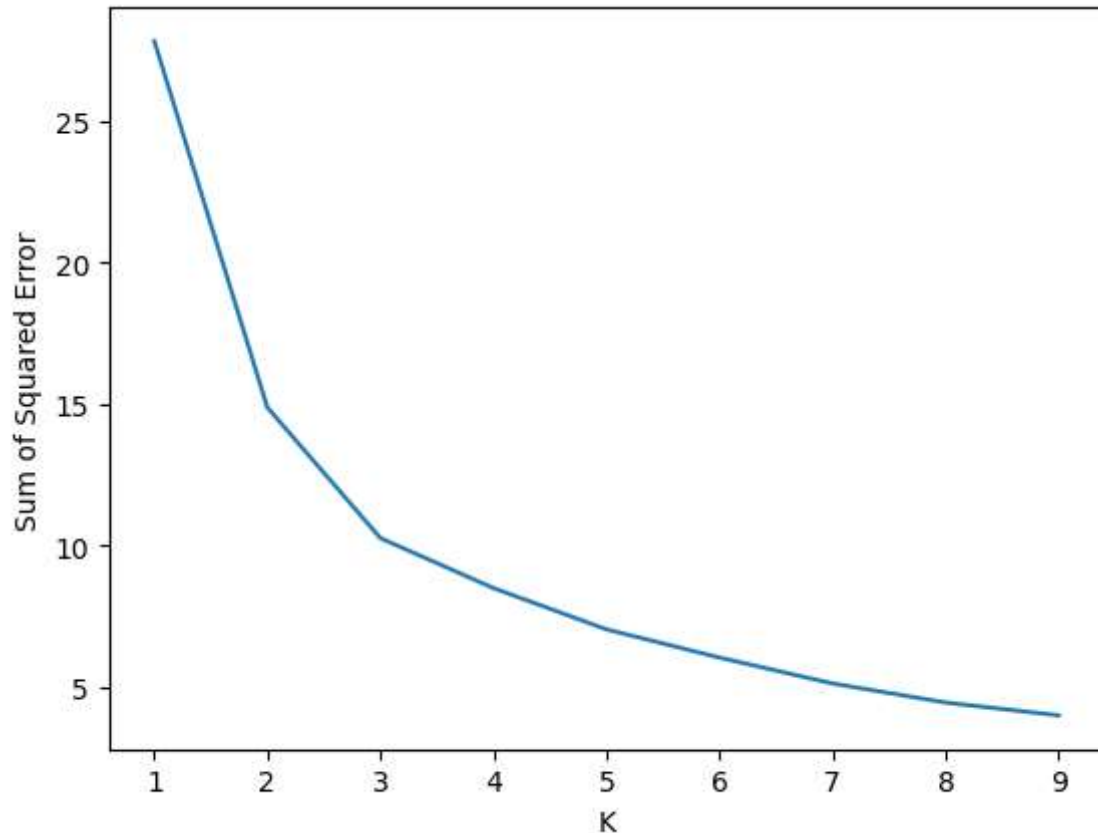
C:\Users\venka\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\\_kmeans.py:870: FutureWarning: The default value of `n\_init` will change from 10 to 'auto' in 1.4. Set the value of `n\_init` explicitly to suppress the warning

warnings.warn(

```
[27.81750759504308, 14.87203295827117, 10.25325555898541, 8.489103553996125,
7.030202097311372, 6.032638028620887, 5.117927753802227, 4.444537230930339,
3.9916276477713906]
```

```
C:\Users\venka\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[19]: 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 accuracies is not good so, that's why we will take it as a clustering and done with K-Means Clustering**

In [ ]: