Project - 4 (DATASET: 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\smb06\OneDrive\Desktop\BreastCancerPrediction (1).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 × 32 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 × 32 columns

In [4]:

```
df.tail
```

Out[4]:

```
<bound method NDFrame.tail of           id diagnosis  radius_mean  textur
e_mean  perimeter_mean  area_mean
0      842302         M        17.99         10.38          122.80     100
1.0  \
1      842517         M        20.57         17.77          132.90     132
6.0
2    84300903         M        19.69         21.25          130.00     120
3.0
3    84348301         M        11.42         20.38           77.58      38
6.1
4    84358402         M        20.29         14.34          135.10     129
7.0
..        ...       ...          ...           ...             ...
...
564    926424         M        21.56         22.39          142.00     147
9.0
565    926682         M        20.13         28.25          131.20     126
1.0
566    926954         M        16.60         28.08          108.30      85
8.1
567    927241         M        20.60         29.33          140.10     126
5.0
568     92751         B         7.76         24.54           47.92      18
1.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_mea
n
0            0.11840           0.27760         0.30010              0.1471
0  \
1            0.08474           0.07864         0.08690              0.0701
7
2            0.10960           0.15990         0.19740              0.1279
0
3            0.14250           0.28390         0.24140              0.1052
0
4            0.10030           0.13280         0.19800              0.1043
0
..               ...               ...             ...
...
564          0.11100           0.11590         0.24390              0.1389
0
565          0.09780           0.10340         0.14400              0.0979
1
566          0.08455           0.10230         0.09251              0.0530
2
567          0.11780           0.27700         0.35140              0.1520
0
568          0.05263           0.04362         0.00000              0.0000
0

     ...  radius_worst  texture_worst  perimeter_worst  area_worst
0    ...         25.380          17.33           184.60      2019.0  \
1    ...         24.990          23.41           158.80      1956.0
2    ...         23.570          25.53           152.50      1709.0
3    ...         14.910          26.50            98.87       567.7
4    ...         22.540          16.67           152.20      1575.0
..   ...            ...            ...              ...         ...
564  ...         25.450          26.40           166.10      2027.0
565  ...         23.690          38.25           155.00      1731.0
566  ...         18.980          34.12           126.70      1124.0
567  ...         25.740          39.42           184.60      1821.0
```

```
568  ...          9.456          30.37            59.16       268.6
```

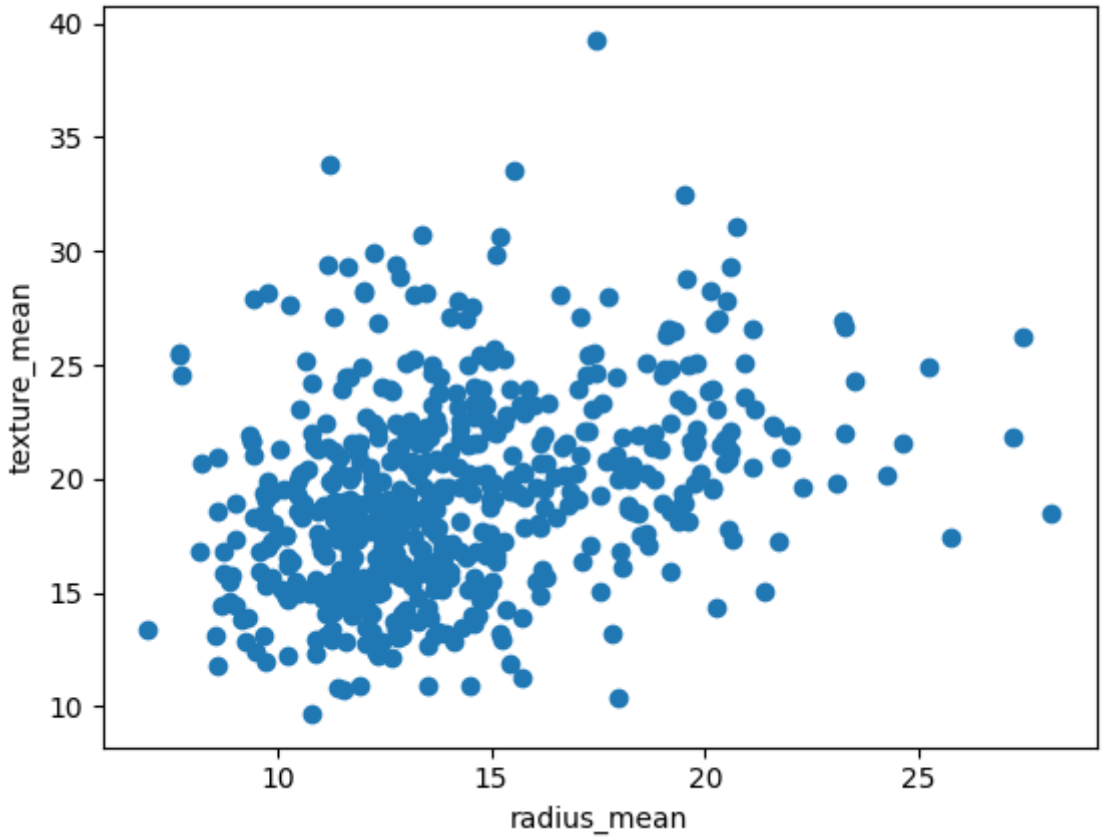
In [6]:

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

```
   smoothness_worst  compactness_worst  concavity_worst
0           0.16220            0.66560           0.7119  \
1           0.12380            0.18660           0.2416
2           0.14440            0.42450           0.4504
3           0.20980            0.86630           0.6869
4           0.13740            0.20500           0.4000

Text(0, 0.5, 'texture_mean')          ...              ...
564         0.14100            0.21130           0.4107
```

Out[6]:



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\smb06\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[8]:

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

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	

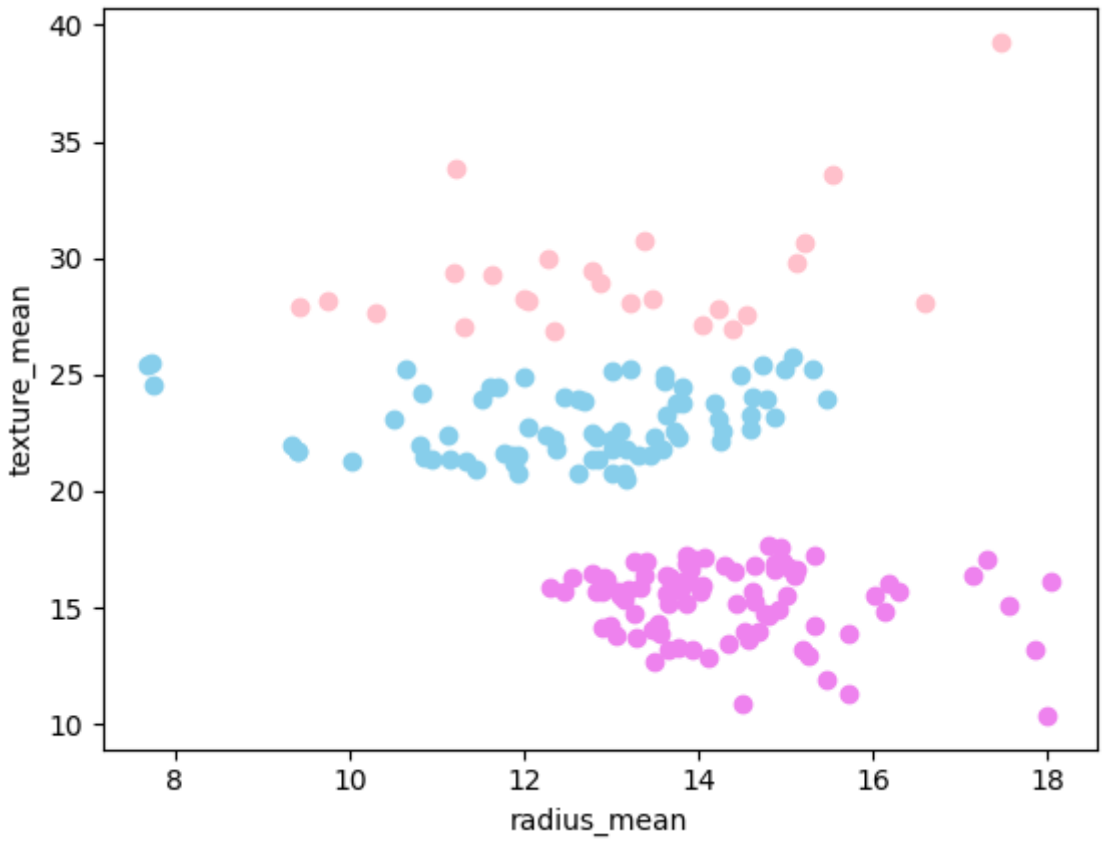
5 rows × 33 columns

In [11]:

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

Out[11]:

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



In [12]:

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

Out[12]:

	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 × 33 columns

In [13]:

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

Out[13]:

	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 × 33 columns

In [14]:

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

```
C:\Users\smb06\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[14]:

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

In [15]:

```
df["New Cluster"]=y_predicted
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	

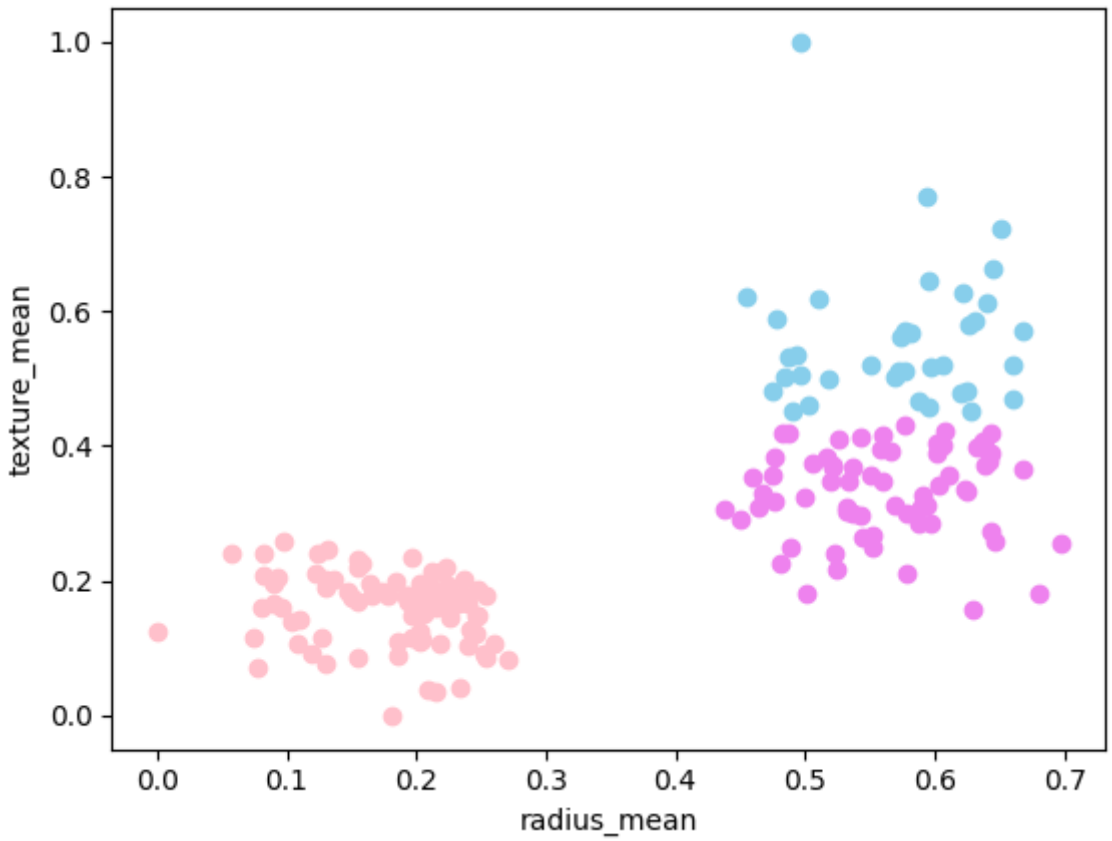
5 rows × 34 columns

In [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="violet")
plt.scatter(df2["radius_mean"],df2["texture_mean"],color="pink")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="skyblue")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[17]:

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



In [18]:

```
km.cluster_centers_
```

Out[18]:

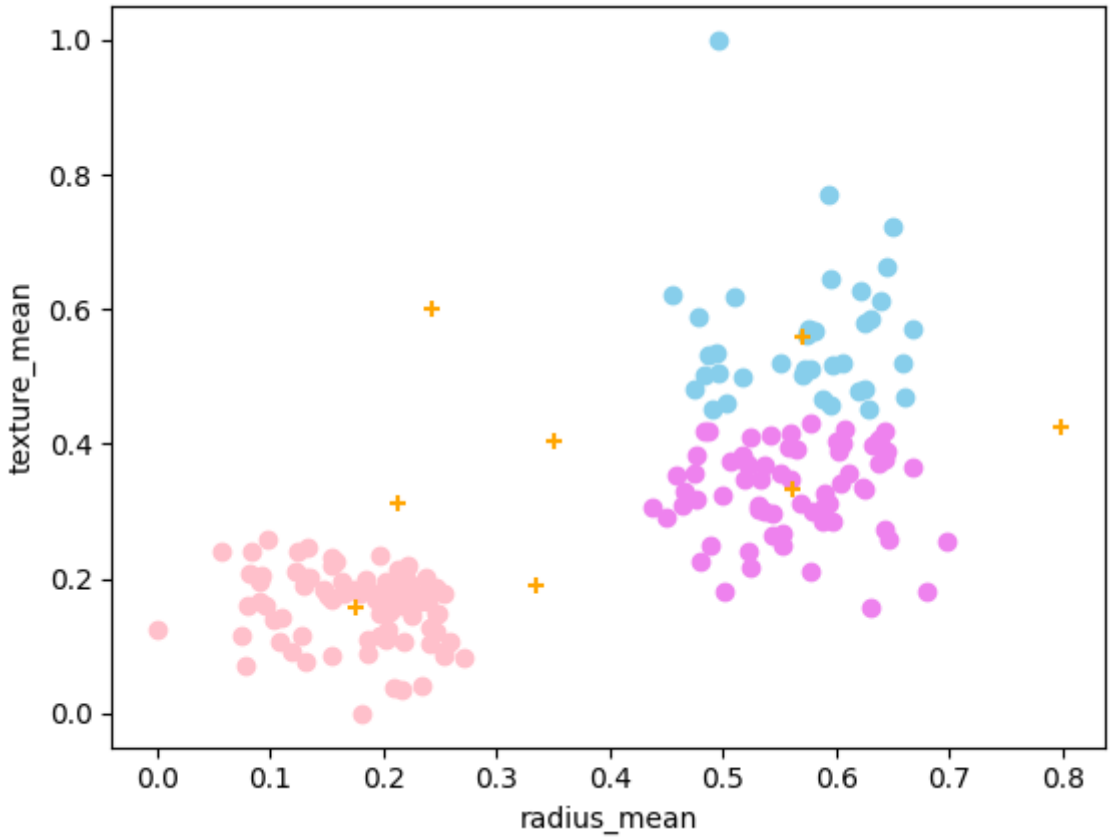
```
array([[0.56101927, 0.3314624 ],
       [0.17620217, 0.15747668],
       [0.57132058, 0.55893025],
       [0.35135576, 0.40520246],
       [0.21306768, 0.31137257],
       [0.79840767, 0.42469846],
       [0.33570532, 0.19063107],
       [0.24279689, 0.59913388]])
```

In [19]:

```
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="violet")
plt.scatter(df2["radius_mean"],df2["texture_mean"],color="pink")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="skyblue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker="+")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[19]:

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



In [23]:

```
k_rng=range(9,18)
sse=[]
```

In [24]:

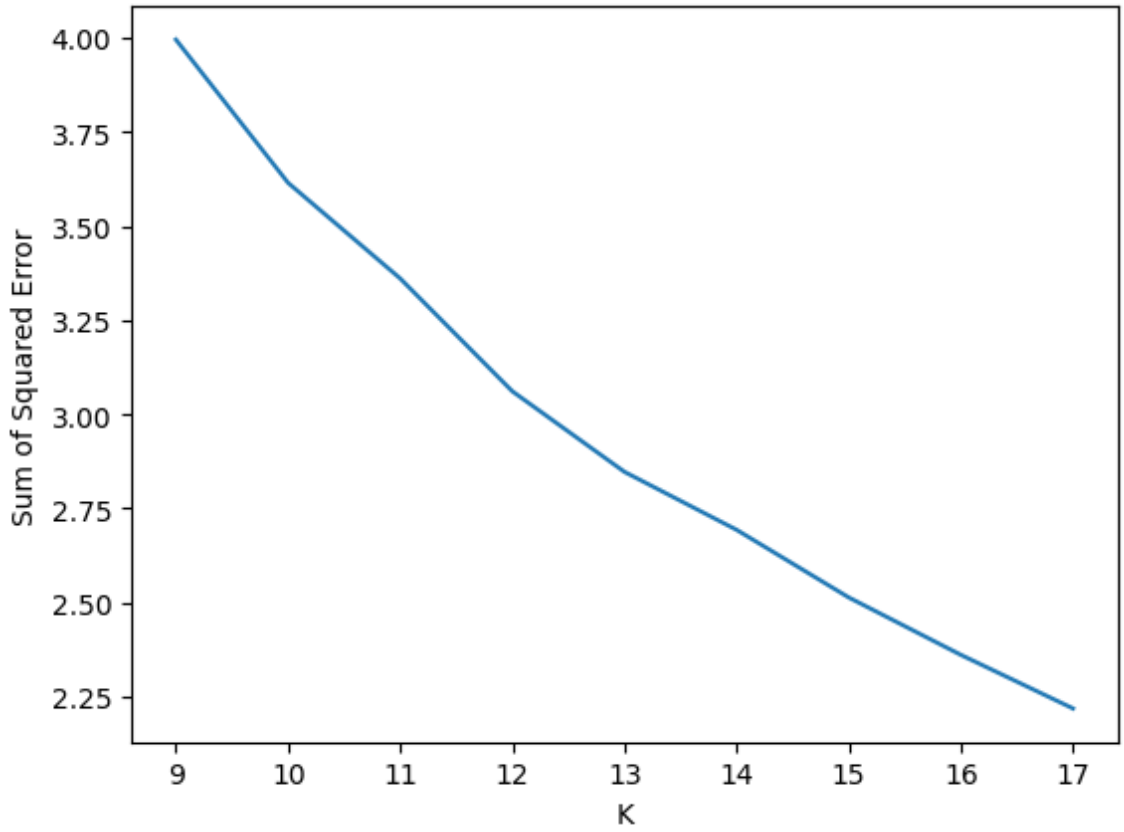
```
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["radius_mean","texture_mean"]])
    sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square error
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\smb06\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\smb06\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\smb06\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\smb06\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\smb06\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\smb06\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\smb06\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\smb06\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\smb06\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(
```

```
[3.994645114512035, 3.614558770100311, 3.3606574010709807, 3.0611560269213
68, 2.8472784333896923, 2.693425089905679, 2.5139350612513347, 2.361533262
6492003, 2.2192489516023146]
```

Out[24]:

```
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 [ ]: