project-4

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

In [2]:
```
df=pd.read_csv(r"C:\Users\Mastan Reddy\Downloads\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	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.
...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	0.
565	926682	M	20.13	28.25	131.20	1261.0	0.
566	926954	M	16.60	28.08	108.30	858.1	0.
567	927241	M	20.60	29.33	140.10	1265.0	0.
568	92751	B	7.76	24.54	47.92	181.0	0.

569 rows × 32 columns

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

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	17.99	10.38	122.80	1001.0	0.11
1	842517	M	20.57	17.77	132.90	1326.0	0.08
2	84300903	M	19.69	21.25	130.00	1203.0	0.10
3	84348301	M	11.42	20.38	77.58	386.1	0.14
4	84358402	M	20.29	14.34	135.10	1297.0	0.10

5 rows × 32 columns

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

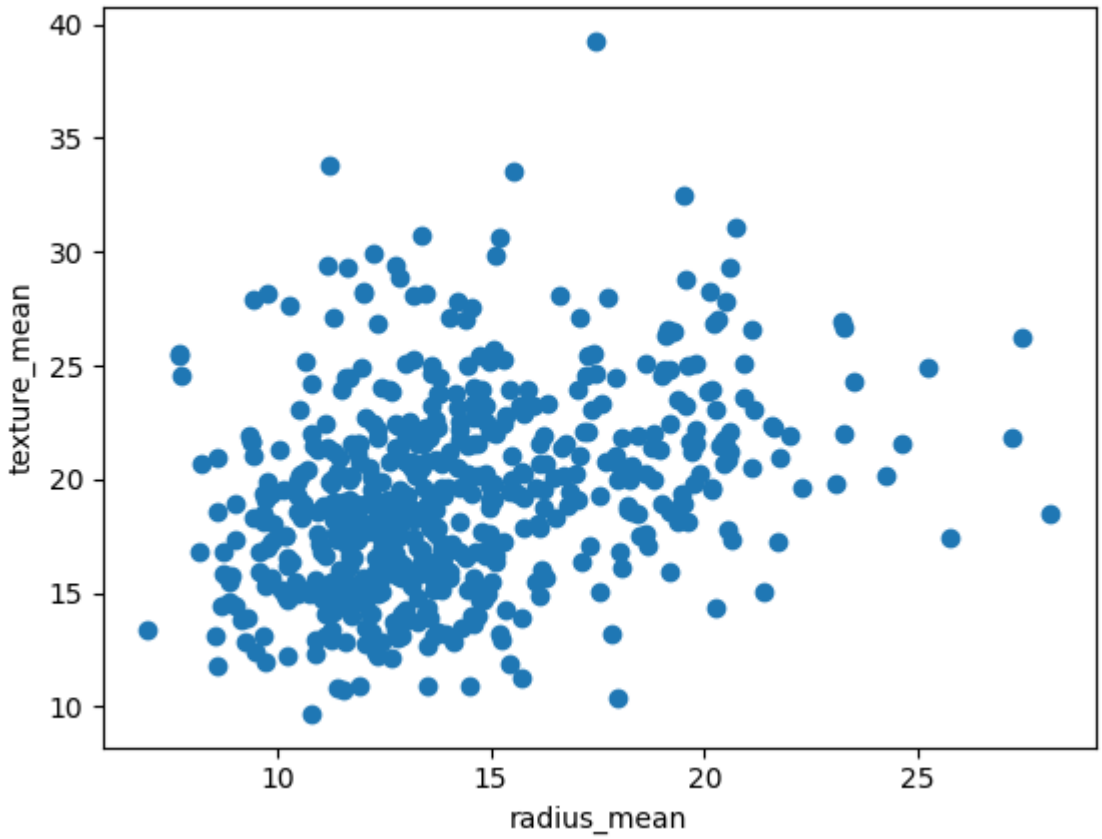
Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
564	926424	M	21.56	22.39	142.00	1479.0	0.11
565	926682	M	20.13	28.25	131.20	1261.0	0.09
566	926954	M	16.60	28.08	108.30	858.1	0.08
567	927241	M	20.60	29.33	140.10	1265.0	0.11
568	92751	B	7.76	24.54	47.92	181.0	0.05

5 rows × 32 columns

In [5]:
```
plt.scatter(df["radius_mean"],df["texture_mean"])
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[5]:
```
Text(0, 0.5, 'texture_mean')
```



In [6]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[6]:
```
KMeans()
```

In [7]:
```
y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

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

In [8]:
```
df["cluster"]=y_predicted
df.head()
```

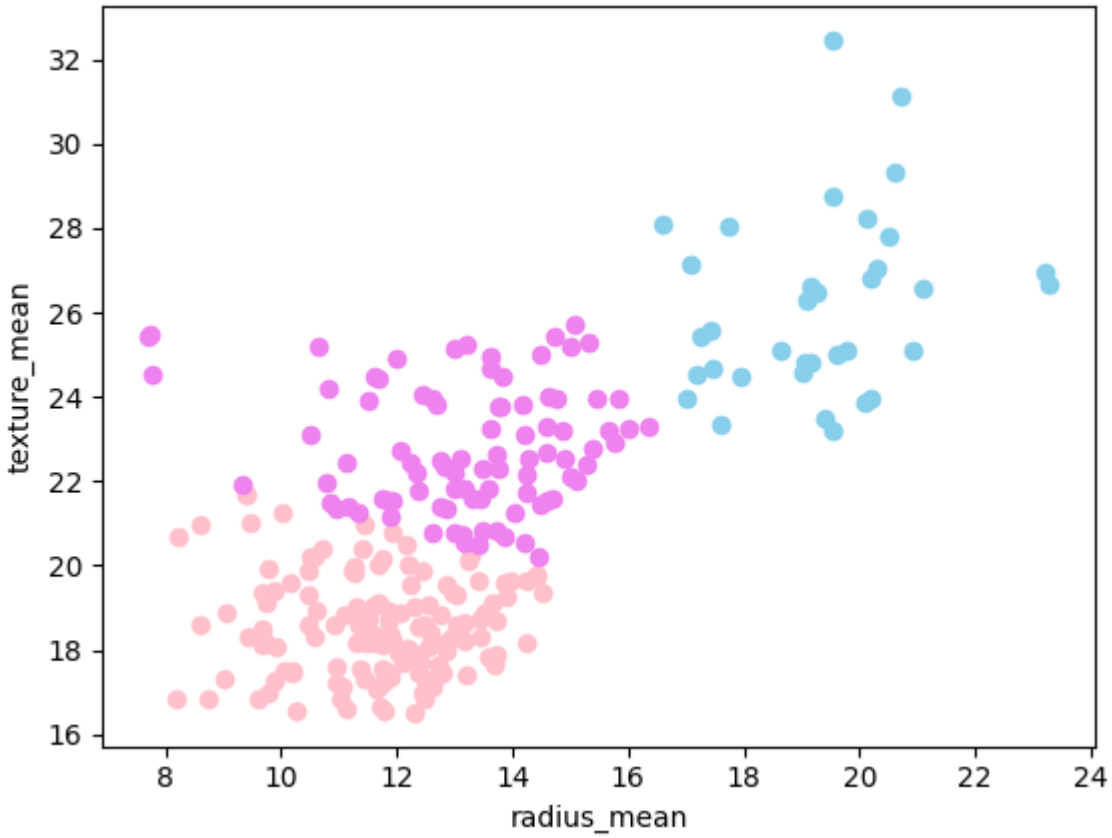
Out[8]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	17.99	10.38	122.80	1001.0	0.11
1	842517	M	20.57	17.77	132.90	1326.0	0.08
2	84300903	M	19.69	21.25	130.00	1203.0	0.10
3	84348301	M	11.42	20.38	77.58	386.1	0.14
4	84358402	M	20.29	14.34	135.10	1297.0	0.10

5 rows × 33 columns

In [9]:
```
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[9]:
```
Text(0, 0.5, 'texture_mean')
```



In [10]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["texture_mean"]])
df["texture_mean"]=scaler.transform(df[["texture_mean"]])
df.head()
```

Out[10]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	17.99	0.022658	122.80	1001.0	0.11
1	842517	M	20.57	0.272574	132.90	1326.0	0.08
2	84300903	M	19.69	0.390260	130.00	1203.0	0.10
3	84348301	M	11.42	0.360839	77.58	386.1	0.14
4	84358402	M	20.29	0.156578	135.10	1297.0	0.10

5 rows × 33 columns

In [11]:
```
scaler.fit(df[["radius_mean"]])
df["radius_mean"]=scaler.transform(df[["radius_mean"]])
df.head()
```

Out[11]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	0.521037	0.022658	122.80	1001.0	0.11
1	842517	M	0.643144	0.272574	132.90	1326.0	0.08
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.10
3	84348301	M	0.210090	0.360839	77.58	386.1	0.14
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.10

5 rows × 33 columns

In [12]:
```
y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

Out[12]:
```
array([4, 7, 7, 6, 7, 4, 7, 2, 2, 0, 2, 4, 5, 2, 2, 0, 2, 2, 7, 4, 4, 1,
       4, 3, 2, 7, 2, 7, 2, 4, 5, 6, 5, 5, 4, 2, 2, 6, 2, 2, 2, 6, 5, 2,
       2, 7, 1, 6, 1, 2, 6, 4, 6, 7, 2, 6, 7, 2, 6, 1, 1, 6, 2, 1, 0, 2,
       6, 6, 6, 4, 7, 1, 5, 4, 6, 2, 4, 7, 5, 6, 6, 4, 3, 5, 1, 7, 2, 5,
       2, 4, 2, 2, 4, 6, 2, 5, 6, 6, 1, 2, 0, 1, 6, 6, 6, 4, 6, 6, 3, 6,
       6, 6, 2, 6, 1, 6, 1, 4, 2, 7, 1, 7, 3, 4, 4, 4, 0, 7, 4, 5, 1, 2,
       2, 4, 7, 2, 6, 1, 4, 1, 1, 4, 6, 4, 1, 1, 6, 2, 4, 4, 2, 6, 1, 1,
       4, 6, 7, 7, 1, 1, 6, 7, 7, 2, 3, 2, 1, 7, 5, 4, 1, 2, 4, 1, 1, 1,
       6, 2, 2, 4, 3, 5, 2, 1, 2, 1, 7, 6, 6, 4, 2, 2, 6, 0, 2, 4, 2, 7,
       7, 2, 6, 7, 3, 2, 6, 4, 6, 7, 2, 4, 7, 6, 3, 5, 2, 4, 6, 6, 7, 5,
       4, 4, 6, 2, 4, 4, 1, 4, 0, 2, 7, 0, 0, 5, 1, 2, 3, 7, 0, 5, 4, 4,
       6, 2, 5, 6, 4, 4, 0, 1, 5, 6, 7, 7, 7, 4, 5, 4, 2, 0, 5, 7, 7, 2,
       7, 5, 6, 2, 4, 6, 4, 1, 3, 1, 5, 6, 1, 7, 4, 4, 5, 1, 7, 2, 4, 6,
       6, 4, 6, 6, 2, 2, 4, 6, 4, 4, 1, 6, 4, 6, 7, 6, 5, 6, 6, 0, 4, 1,
       4, 4, 6, 4, 4, 1, 6, 6, 1, 7, 6, 6, 1, 7, 4, 7, 1, 6, 4, 6, 2, 2,
       4, 6, 6, 1, 6, 7, 4, 7, 6, 3, 4, 1, 1, 7, 1, 1, 6, 4, 1, 1, 6, 2,
       3, 0, 1, 6, 6, 4, 1, 6, 6, 2, 6, 7, 4, 7, 5, 6, 7, 3, 2, 4, 7, 7,
       4, 4, 6, 0, 4, 6, 1, 1, 2, 6, 4, 2, 1, 4, 1, 5, 1, 1, 2, 3, 6, 4,
       2, 6, 1, 6, 7, 1, 6, 4, 1, 6, 4, 2, 7, 6, 6, 6, 6, 2, 0, 6, 6, 2,
       1, 6, 6, 4, 1, 2, 6, 6, 1, 6, 6, 6, 2, 6, 7, 7, 4, 2, 6, 4, 2, 4,
       6, 5, 4, 6, 7, 0, 5, 4, 2, 7, 6, 5, 0, 4, 6, 0, 0, 0, 0, 0, 5, 3,
       0, 6, 6, 2, 2, 6, 5, 6, 6, 0, 4, 0, 1, 4, 2, 4, 1, 2, 6, 2, 4, 4,
       4, 4, 4, 7, 1, 7, 2, 4, 7, 1, 2, 2, 6, 6, 7, 7, 4, 0, 4, 3, 1, 1,
       6, 6, 4, 2, 1, 4, 2, 4, 2, 6, 7, 7, 6, 4, 1, 3, 6, 2, 1, 1, 6, 1,
       4, 1, 6, 6, 4, 7, 6, 7, 2, 0, 0, 0, 1, 0, 0, 0, 2, 2, 1, 1, 6, 0,
       6, 6, 0, 6, 0, 0, 6, 0, 2, 0, 0, 0, 0, 5, 3, 5, 5, 5, 0])
```

In [13]:
```
df["New Cluster"]=y_predicted
df.head()
```

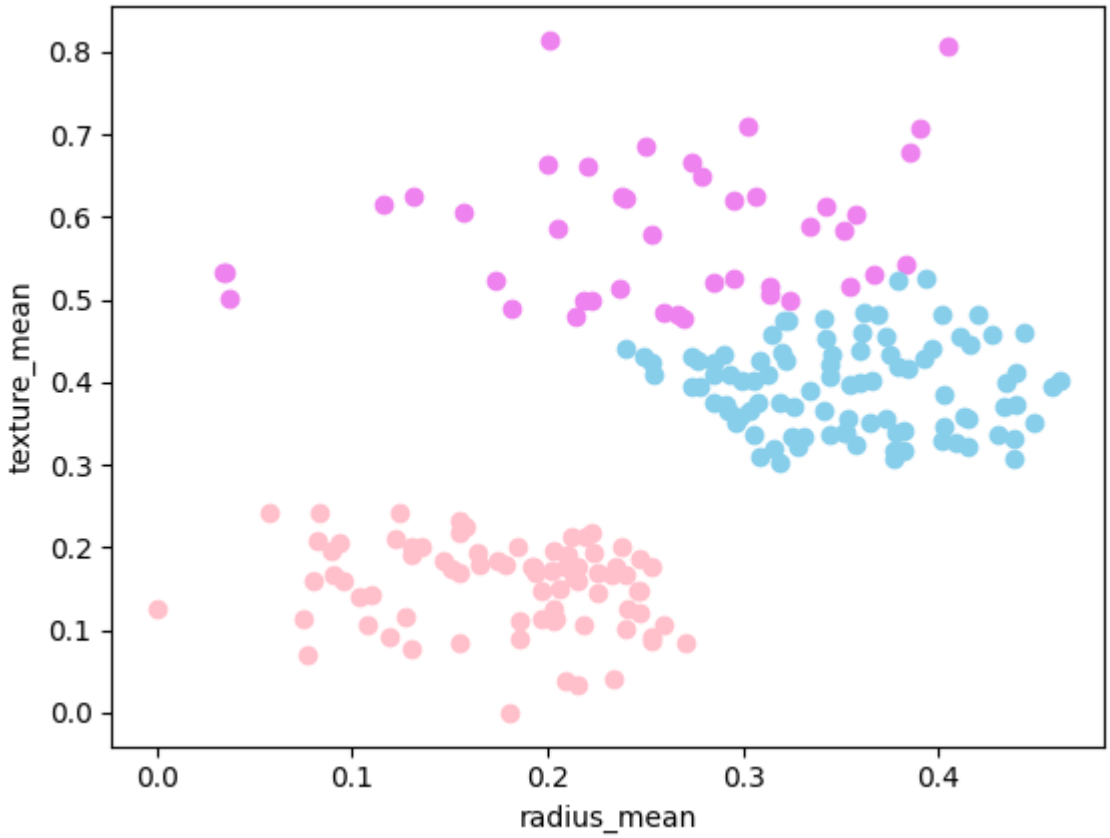
Out[13]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	0.521037	0.022658	122.80	1001.0	0.11
1	842517	M	0.643144	0.272574	132.90	1326.0	0.08
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.10
3	84348301	M	0.210090	0.360839	77.58	386.1	0.14
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.10

5 rows × 34 columns

In [14]:
```
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[14]:
```
Text(0, 0.5, 'texture_mean')
```

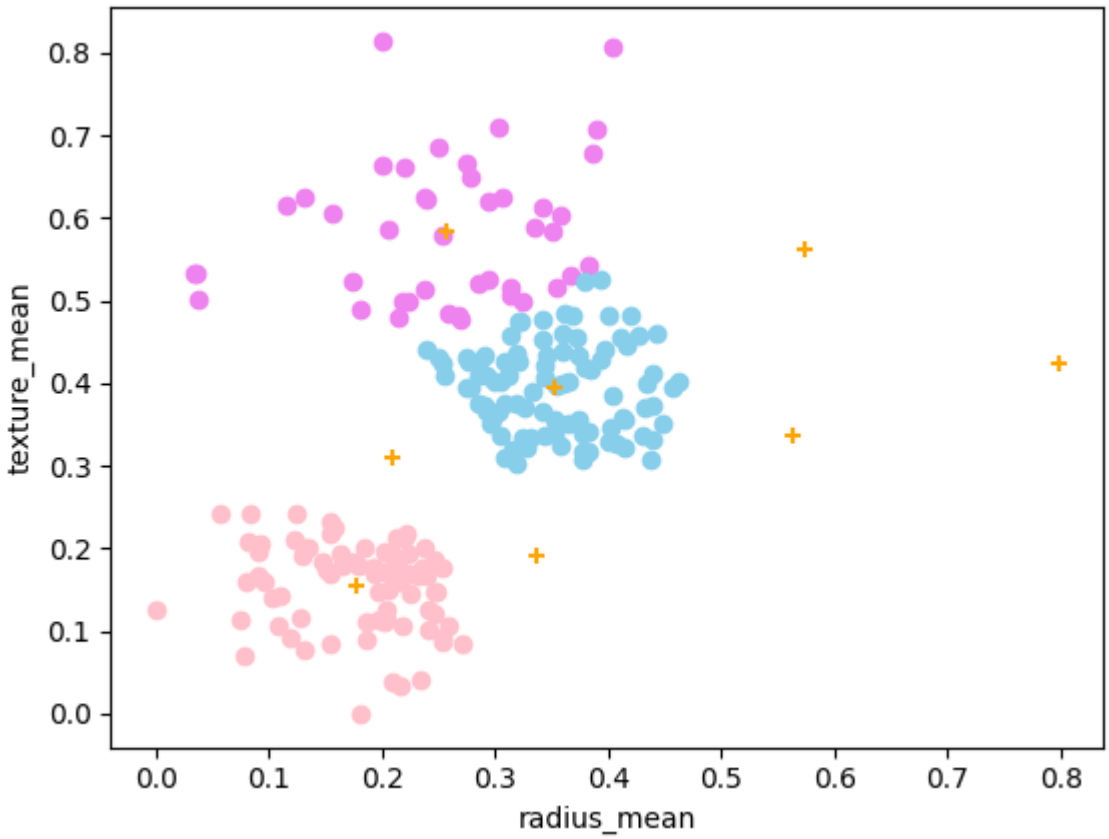


In [15]:
```
km.cluster_centers_
```

Out[15]:
```
array([[0.25627183, 0.58431314],
       [0.17750575, 0.15412045],
       [0.35339953, 0.39439771],
       [0.79840767, 0.42469846],
       [0.33731167, 0.19053357],
       [0.57355872, 0.56191523],
       [0.20987596, 0.3099295 ],
       [0.56272221, 0.33594655]])
```

In [16]:
```
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",m
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

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

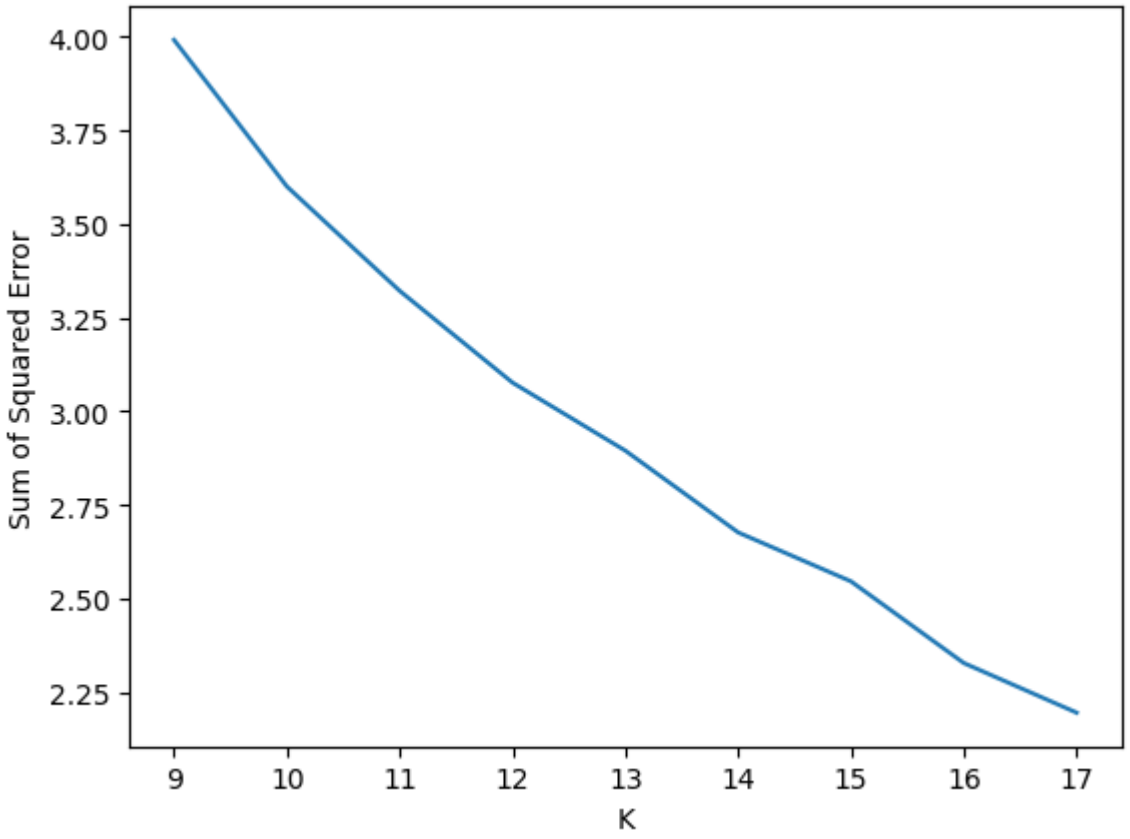


In [17]:
```
k_rng=range(9,18)
sse=[]
```

In [18]:
```
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")
```

```
[3.991752934887513, 3.600358909956173, 3.3216499091044813, 3.076609147188867,
2.8949841696830485, 2.676593050946156, 2.5459080581466798, 2.327761938956805,
2.195306888617589]
```

Out[18]: 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

In [ ]:
```
```