In [2]:

```
#problem stmt:To find the best fit model for the given dataset by using the k-means c
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Reading the dataFrame

In [4]:

```
df=pd.read_csv(r"C:\Users\MY HOME\Desktop\BreastCancerPrediction.csv")
df
```

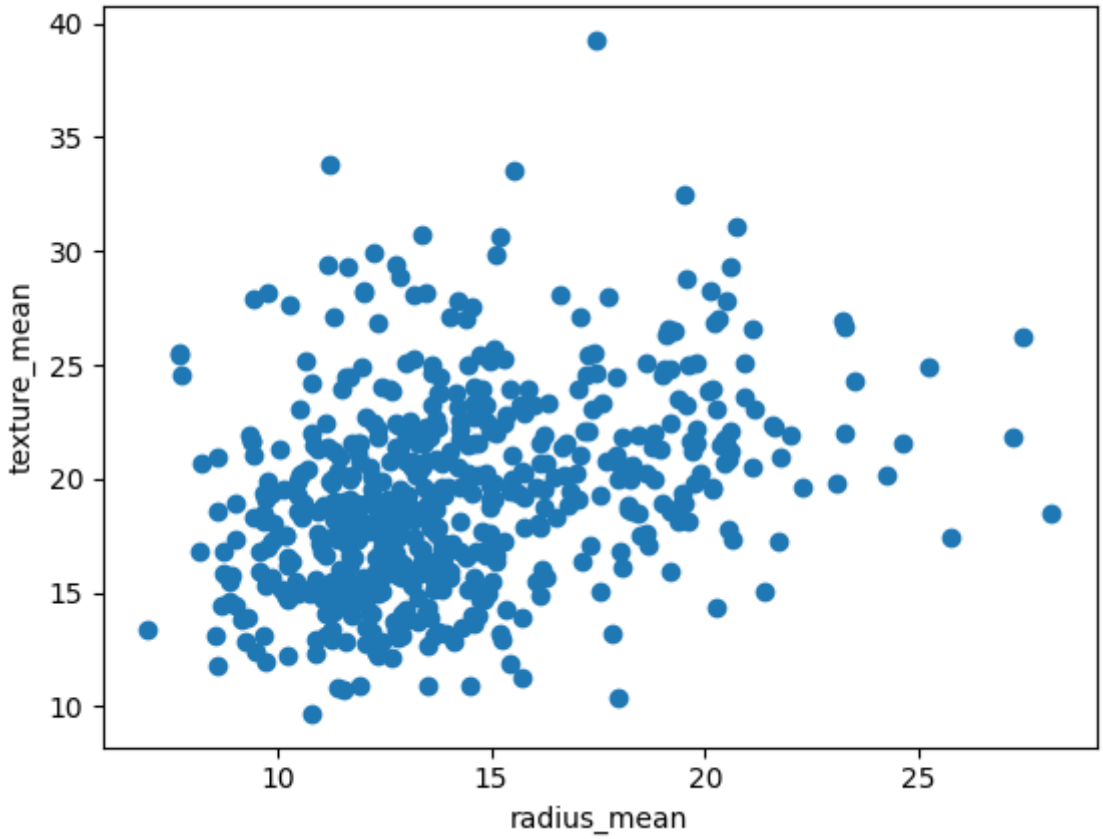
Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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 [5]:

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



In [8]:

```
from sklearn.cluster import KMeans
```

In [9]:

```
km=KMeans()
km
```

Out[9]:

KMeans()

In [10]:

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

```
C:\Users\MY HOME\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[10]:

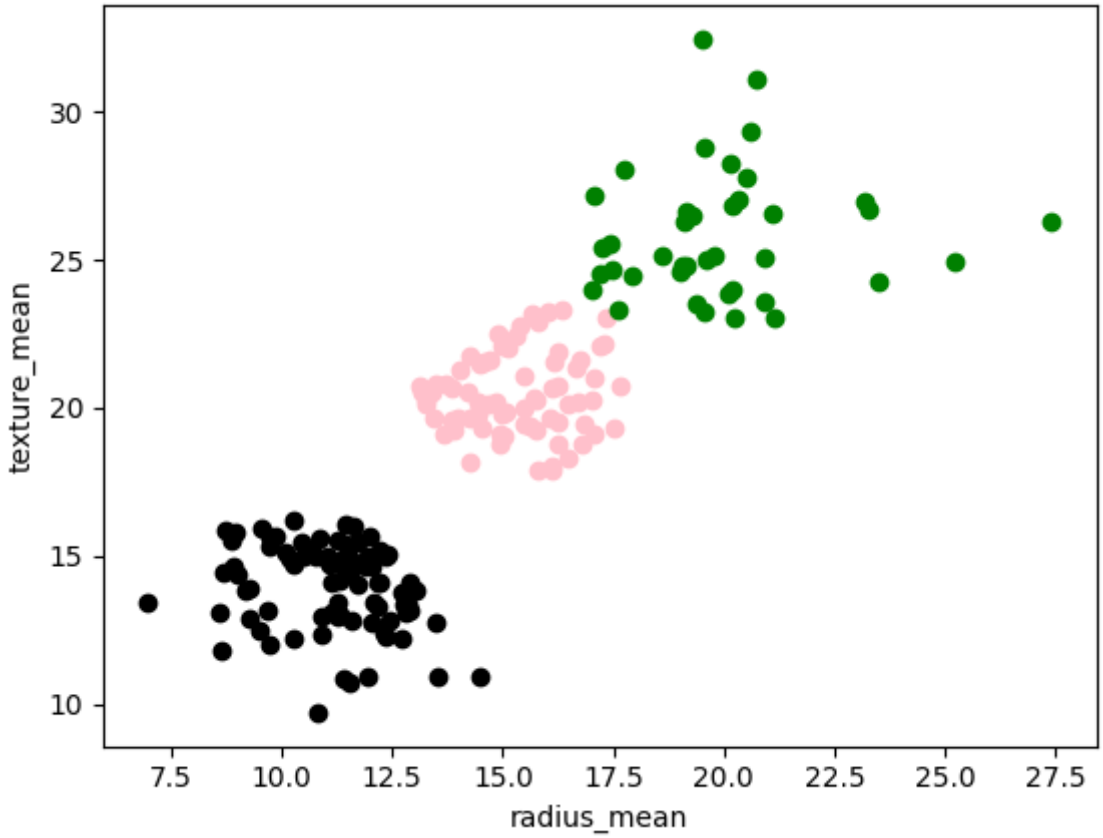
```
array([4, 7, 7, 6, 7, 4, 7, 0, 5, 5, 0, 0, 1, 5, 5, 3, 0, 0, 7, 4, 4, 2,
       4, 1, 0, 4, 0, 7, 5, 4, 1, 6, 1, 1, 0, 0, 0, 6, 5, 0, 5, 5, 1, 0,
       5, 7, 6, 6, 2, 5, 5, 4, 6, 7, 0, 6, 7, 0, 6, 2, 2, 6, 5, 2, 5, 5,
       6, 6, 6, 4, 7, 2, 1, 4, 6, 0, 2, 4, 1, 6, 5, 4, 1, 1, 2, 7, 0, 1,
    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

In [11]:

```
df["cluster"]=y_predicted
```

In [12]:

```
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="green")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="black")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
plt.show()
```



In [13]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [14]:

```
scaler=MinMaxScaler()
scaler.fit(df[["radius_mean"]])
s=scaler.transform(df[["radius_mean"]])
df.tail()
```

Out[14]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness
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 × 34 columns

In [15]:

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

Out[15]:

	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	

5 rows × 34 columns

In [16]:

```
km=KMeans()
km

```

Out[16]:

```
KMeans()
```

In [17]:

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

```
C:\Users\MY HOME\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[17]:

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

In [18]:

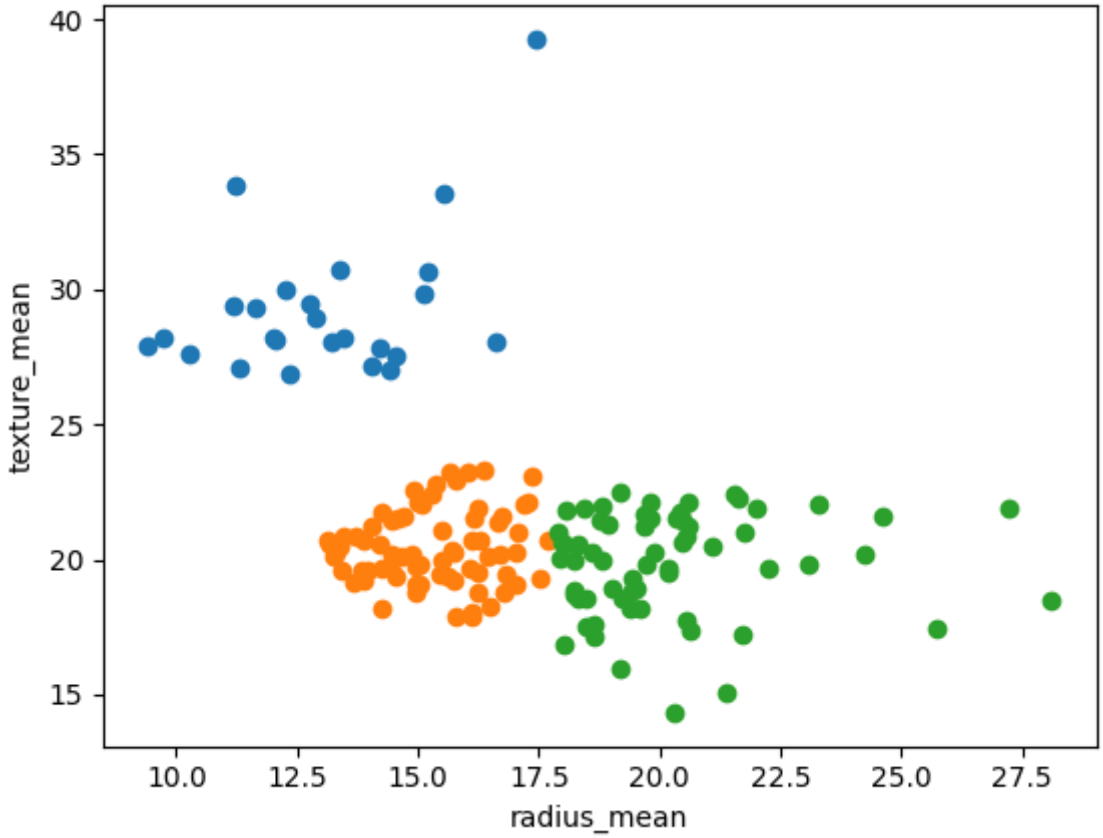
```
df["New cluster"]=y_predicted
y_predicted
```

Out[18]:

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

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"])
plt.scatter(df2["radius_mean"],df2["texture_mean"])
plt.scatter(df3["radius_mean"],df3["texture_mean"])
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
plt.show()
```



In [20]:

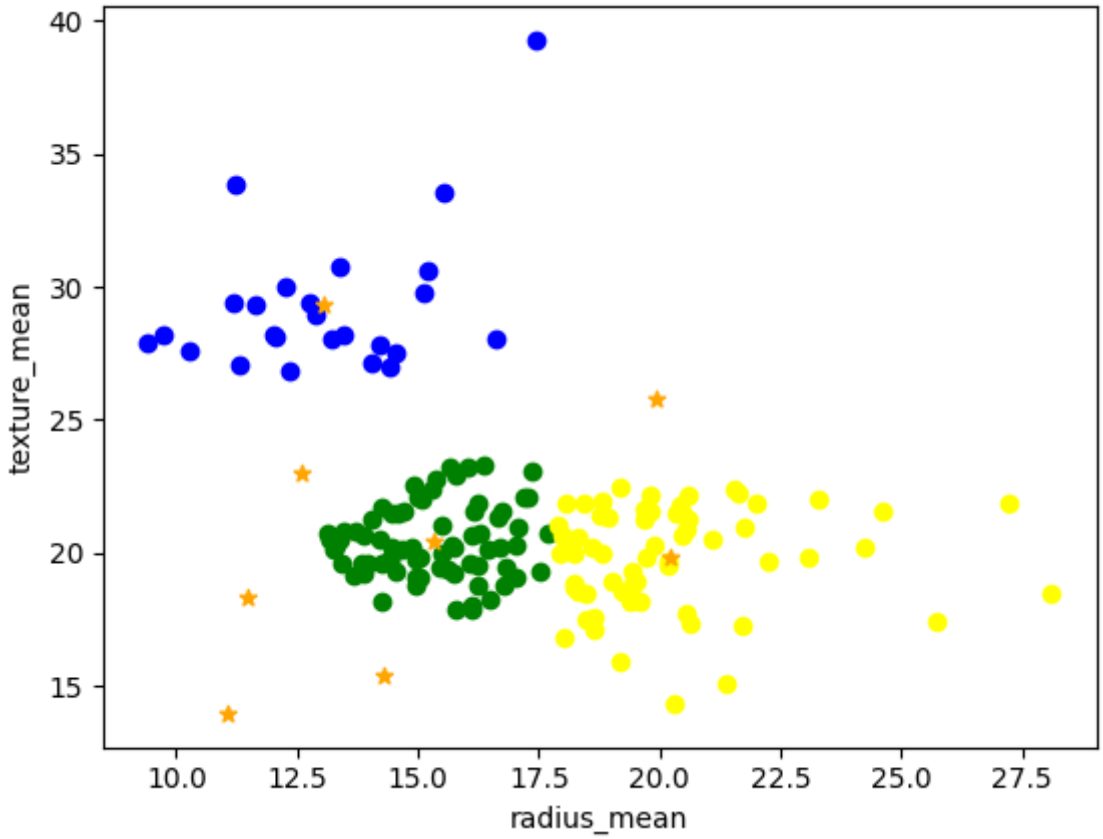
```
km.cluster_centers_
```

Out[20]:

```
array([[13.05352   , 29.3064    ],
       [15.31726027, 20.43589041],
       [20.21428571, 19.85968254],
       [11.0653494 , 13.98542169],
       [12.59061644, 23.00191781],
       [11.48107317, 18.34252033],
       [19.9335    , 25.8125    ],
       [14.30764045, 15.41134831]])
```

In [21]:

```
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="blue")
plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
plt.scatter(df3["radius_mean"],df3["texture_mean"],color="yellow")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker=
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
plt.show()
```



In [22]:

```
t=range(1,10)
sse=[]
for k in t:
    km=KMeans(n_clusters=k)
    km.fit(df[["radius_mean","texture_mean"]])
    sse.append(km.inertia_)
```

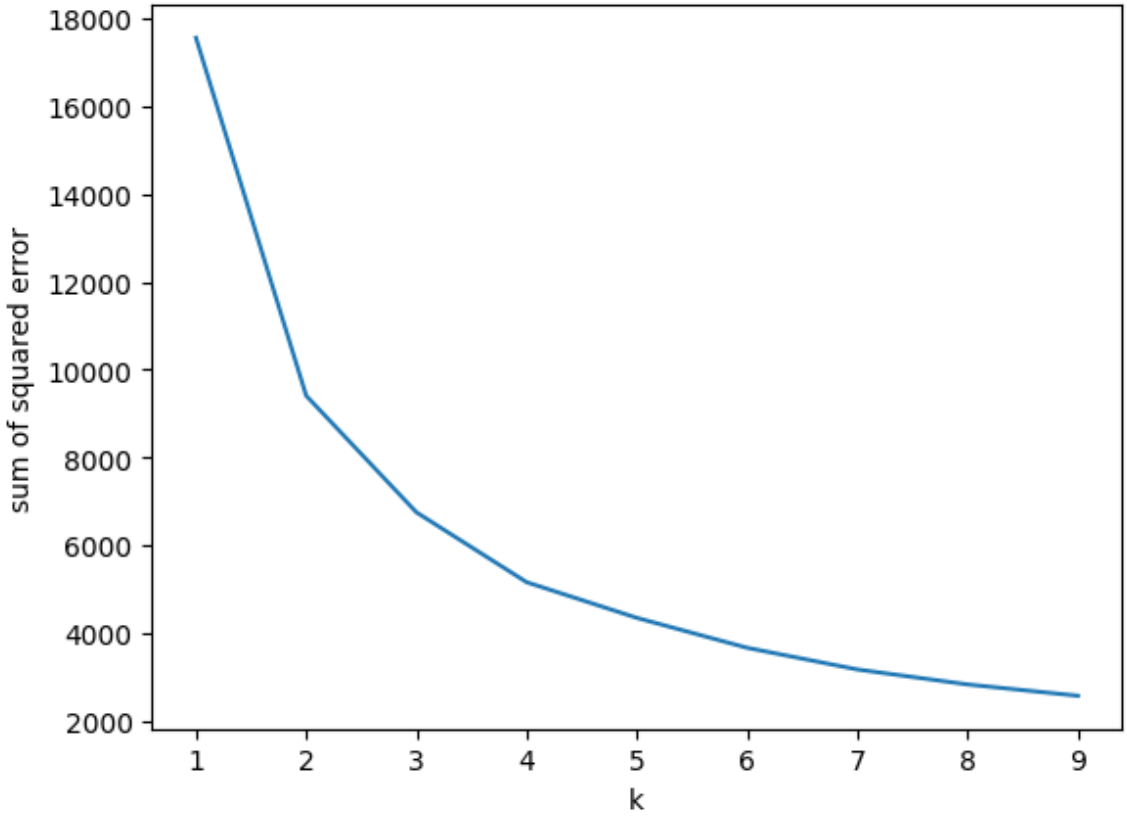
```
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages
\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_in
it` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages
\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\MY HOME\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\MY HOME\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\MY HOME\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\MY HOME\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\MY HOME\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\MY HOME\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\MY HOME\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 [24]:

```
plt.plot(t,sse)
plt.xlabel("k")
plt.ylabel("sum of squared error")
```

Out[24]:

```
Text(0, 0.5, 'sum of squared error')
```



conclusion:

In [ ]:

```
The conclusion that i have concluded that its a best fit model for the given dataset

```