Problem Statement:To Predict How Best the Datafits and To predict the BreastCancer based on the given feature

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

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\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 × 33 columns

Data cleaning and preprocessing

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

Out[3]:

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

5 rows × 33 columns

In [5]:
```
df.info
```

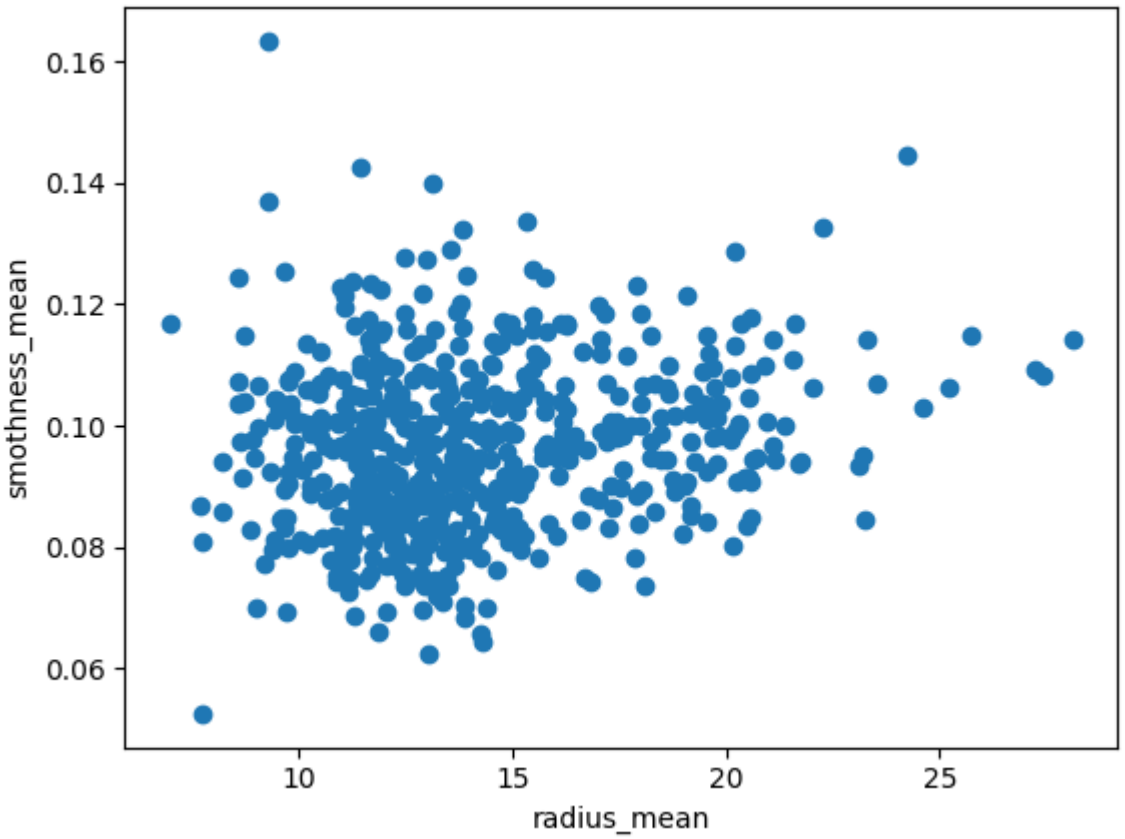
Out[5]:
```
<bound method DataFrame.info of            id diagnosis  radius_mean  textur
e_mean  perimeter_mean  area_mean
0      842302         M        17.99         10.38          122.80     1001.
0  \
1      842517         M        20.57         17.77          132.90     1326.
0
2    84300903         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

     smoothness_mean  compactness_mean  concavity_mean  concave points_mean
0            0.11840           0.27760         0.30010              0.14710
\
1            0.08474           0.07864         0.08690              0.07017
2            0.10960           0.15990         0.19740              0.12790
3            0.14250           0.28390         0.24140              0.10520
4            0.10030           0.13280         0.19800              0.10430
..               ...               ...             ...                  ...
564          0.11100           0.11590         0.24390              0.13890
565          0.09780           0.10340         0.14400              0.09791
566          0.08455           0.10230         0.09251              0.05302
567          0.11780           0.27700         0.35140              0.15200
568          0.05263           0.04362         0.00000              0.00000

     ...  texture_worst  perimeter_worst  area_worst  smoothness_worst
0    ...          17.33           184.60      2019.0           0.16220  \
1    ...          23.41           158.80      1956.0           0.12380
2    ...          25.53           152.50      1709.0           0.14440
3    ...          26.50            98.87       567.7           0.20980
4    ...          16.67           152.20      1575.0           0.13740
..   ...            ...              ...         ...               ...
564  ...          26.40           166.10      2027.0           0.14100
565  ...          38.25           155.00      1731.0           0.11660
566  ...          34.12           126.70      1124.0           0.11390
567  ...          39.42           184.60      1821.0           0.16500
568  ...          30.37            59.16       268.6           0.08996

     compactness_worst  concavity_worst  concave points_worst  symmetry_wors
t
0              0.66560           0.7119                0.2654         0.460
1  \
1              0.18660           0.2416                0.1860         0.275
```

```
0
2              0.42450          0.4504               0.2430          0.361
3
3              0.86630          0.6869               0.2575          0.663
8
4              0.20500          0.4000               0.1625          0.236
4
..                 ...             ...                  ...
...
564            0.21130          0.4107               0.2216          0.206
0
565            0.19220          0.3215               0.1628          0.257
2
566            0.30940          0.3403               0.1418          0.221
8
567            0.86810          0.9387               0.2650          0.408
7
568            0.06444          0.0000               0.0000          0.287
1

     fractal_dimension_worst  Unnamed: 32
0                    0.11890          NaN
1                    0.08902          NaN
2                    0.08758          NaN
3                    0.17300          NaN
4                    0.07678          NaN
..                       ...          ...
564                  0.07115          NaN
565                  0.06637          NaN
566                  0.07820          NaN
567                  0.12400          NaN
568                  0.07039          NaN

[569 rows x 33 columns]>
```

Exploratory data Analysis

In [7]:
```
plt.scatter(df["radius_mean"],df["smoothness_mean"])
plt.xlabel("radius_mean")
plt.ylabel("smothness_mean")
```

Out[7]:
```
Text(0, 0.5, 'smothness_mean')
```



K-Means Clustering

In [8]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[8]:
```
KMeans()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [9]:
```
y_predicted=km.fit_predict(df[["radius_mean","smoothness_mean"]])
y_predicted
```

```
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
```

Out[9]:
```
array([1, 6, 6, 5, 6, 0, 1, 7, 0, 0, 4, 4, 1, 4, 7, 7, 7, 4, 6, 7, 0, 2,
       4, 6, 4, 1, 7, 1, 4, 1, 1, 5, 1, 6, 4, 4, 7, 0, 4, 7, 7, 5, 1, 0,
       0, 1, 2, 0, 0, 7, 5, 7, 5, 1, 4, 5, 1, 7, 0, 2, 2, 2, 7, 2, 0, 7,
       2, 5, 2, 0, 1, 2, 1, 7, 0, 4, 7, 1, 6, 0, 5, 0, 3, 1, 5, 1, 7, 1,
       0, 7, 7, 4, 0, 7, 4, 6, 0, 2, 5, 7, 7, 2, 0, 2, 5, 0, 5, 0, 6, 5,
       2, 0, 7, 5, 2, 5, 2, 4, 4, 1, 5, 1, 3, 7, 7, 7, 7, 1, 4, 6, 0, 4,
       4, 4, 1, 0, 5, 5, 4, 5, 2, 4, 5, 0, 5, 5, 5, 4, 7, 7, 0, 2, 2, 5,
       0, 0, 1, 4, 0, 5, 5, 1, 6, 0, 3, 4, 5, 4, 1, 4, 0, 7, 4, 5, 5, 2,
       2, 4, 0, 0, 3, 6, 4, 5, 4, 2, 1, 5, 5, 0, 7, 0, 2, 0, 4, 0, 7, 1,
       1, 7, 0, 1, 3, 7, 0, 4, 2, 1, 0, 4, 6, 5, 3, 1, 7, 7, 5, 2, 6, 6,
       7, 7, 2, 4, 0, 7, 5, 4, 0, 0, 1, 5, 5, 6, 2, 7, 3, 6, 7, 1, 7, 0,
       5, 7, 6, 5, 0, 0, 5, 5, 6, 5, 6, 1, 6, 7, 6, 4, 4, 4, 6, 1, 1, 4,
       1, 6, 5, 7, 0, 5, 7, 5, 6, 2, 1, 5, 5, 1, 7, 7, 1, 5, 6, 4, 0, 0,
       5, 0, 5, 5, 7, 4, 0, 5, 0, 7, 5, 5, 7, 5, 6, 0, 6, 5, 5, 5, 0, 2,
       7, 0, 5, 7, 0, 5, 2, 0, 0, 1, 2, 0, 2, 6, 0, 6, 0, 0, 7, 0, 4, 4,
       4, 0, 5, 5, 0, 1, 0, 1, 2, 3, 7, 2, 5, 6, 5, 2, 0, 7, 5, 5, 5, 4,
       3, 4, 5, 0, 0, 7, 2, 2, 0, 0, 0, 4, 7, 6, 6, 0, 6, 6, 4, 4, 6, 6,
       7, 4, 5, 7, 7, 5, 5, 5, 0, 0, 0, 7, 0, 7, 5, 6, 2, 2, 4, 6, 0, 7,
       7, 0, 5, 5, 1, 5, 0, 0, 0, 5, 4, 0, 1, 0, 5, 5, 2, 4, 4, 5, 2, 4,
       0, 5, 5, 7, 5, 7, 2, 2, 5, 5, 5, 0, 4, 0, 6, 1, 4, 7, 0, 7, 7, 7,
       5, 1, 7, 5, 1, 5, 1, 7, 7, 6, 5, 6, 5, 7, 0, 7, 5, 0, 0, 2, 1, 3,
       7, 5, 0, 0, 0, 2, 1, 5, 2, 0, 4, 0, 5, 0, 7, 7, 5, 4, 0, 7, 7, 7,
       4, 0, 7, 6, 5, 4, 0, 1, 1, 0, 0, 4, 0, 0, 1, 6, 4, 7, 0, 3, 2, 2,
       0, 5, 4, 4, 5, 7, 7, 7, 4, 5, 1, 6, 0, 0, 2, 3, 5, 7, 2, 2, 7, 0,
       7, 0, 5, 5, 7, 6, 5, 6, 7, 5, 2, 2, 5, 7, 7, 0, 7, 7, 2, 2, 2, 5,
       5, 5, 0, 2, 0, 2, 2, 2, 7, 5, 7, 5, 4, 6, 6, 6, 4, 6, 2])
```

In [10]:
```
df["cluster"]=y_predicted
df.head()
```

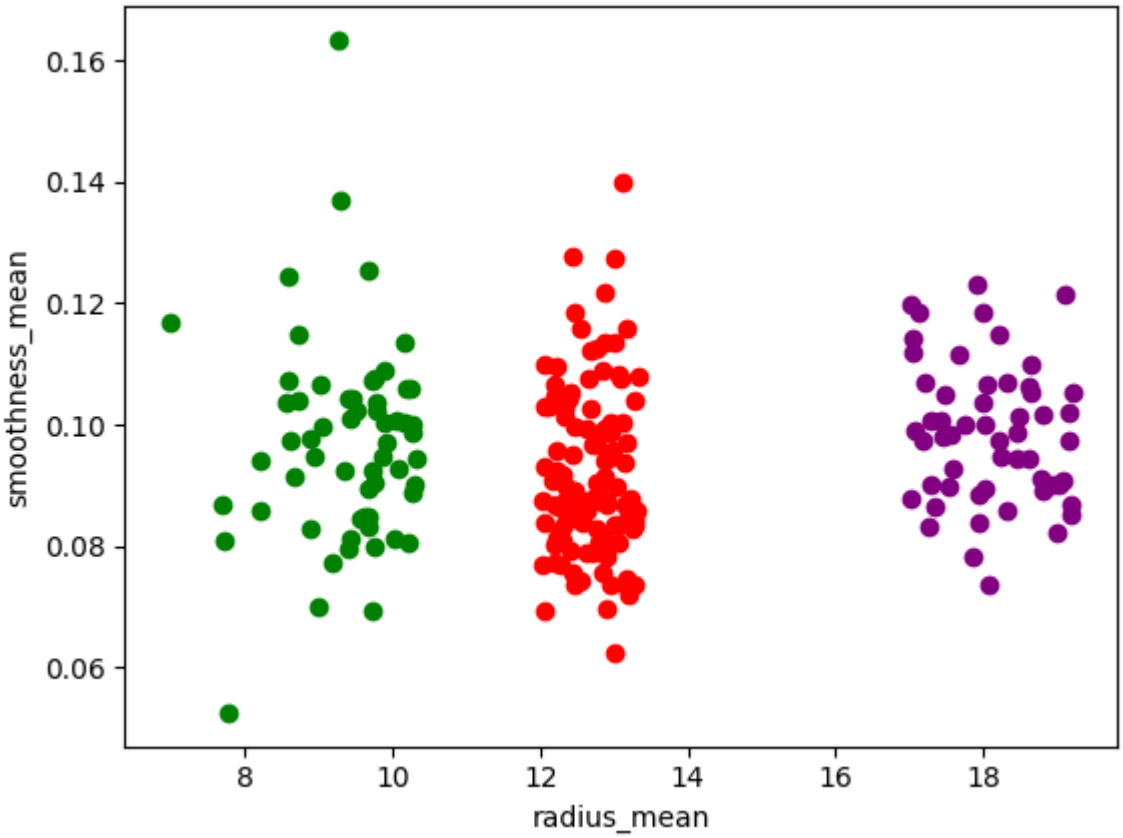
Out[10]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_n
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 [21]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["radius_mean"],df1["smoothness_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["smoothness_mean"],color="purple")
plt.scatter(df3["radius_mean"],df3["smoothness_mean"],color="green")
plt.xlabel("radius_mean")
plt.ylabel("smoothness_mean")
```

Out[21]:
```
Text(0, 0.5, 'smoothness_mean')
```



In [22]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [23]:
```
scaler=MinMaxScaler()
```

In [24]:
```
scaler.fit(df[["smoothness_mean"]])
df["smoothness_mean"]=scaler.transform(df[["smoothness_mean"]])
df.head()
```

Out[24]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_n
0	842302	M	17.99	10.38	122.80	1001.0	0.59
1	842517	M	20.57	17.77	132.90	1326.0	0.28
2	84300903	M	19.69	21.25	130.00	1203.0	0.51
3	84348301	M	11.42	20.38	77.58	386.1	0.81
4	84358402	M	20.29	14.34	135.10	1297.0	0.43

5 rows × 34 columns

In [25]:
```
km=KMeans()
```

In [27]:
```
y_predicted=km.fit_predict(df[["radius_mean","smoothness_mean"]])
y_predicted
```

```
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
```

Out[27]:
```
array([1, 4, 4, 7, 4, 0, 1, 2, 0, 0, 6, 6, 4, 6, 2, 2, 6, 6, 4, 2, 0, 5,
       6, 4, 1, 1, 2, 1, 6, 1, 1, 7, 1, 4, 6, 1, 2, 0, 6, 2, 2, 7, 4, 2,
       0, 1, 5, 0, 0, 2, 7, 2, 0, 1, 6, 7, 4, 6, 0, 5, 5, 5, 2, 5, 0, 6,
       5, 7, 5, 0, 1, 5, 1, 2, 0, 6, 2, 1, 4, 0, 7, 2, 3, 4, 0, 1, 2, 1,
       0, 6, 2, 6, 2, 2, 6, 4, 0, 5, 7, 2, 2, 5, 0, 5, 7, 0, 7, 0, 4, 7,
       5, 0, 2, 7, 5, 0, 5, 6, 6, 1, 7, 1, 3, 2, 2, 2, 2, 1, 6, 4, 0, 6,
       6, 6, 1, 0, 7, 7, 6, 7, 5, 6, 7, 0, 7, 0, 7, 6, 2, 2, 0, 5, 5, 7,
       0, 0, 1, 1, 0, 7, 7, 4, 4, 0, 3, 6, 7, 1, 1, 6, 0, 2, 6, 7, 7, 5,
       5, 6, 0, 0, 3, 4, 6, 7, 6, 5, 1, 7, 7, 0, 2, 0, 5, 0, 6, 0, 2, 1,
       4, 2, 0, 1, 3, 2, 0, 6, 5, 1, 0, 6, 4, 7, 3, 1, 2, 2, 0, 5, 4, 4,
       2, 2, 5, 6, 2, 2, 7, 6, 0, 0, 1, 7, 7, 4, 5, 2, 3, 4, 2, 1, 2, 0,
       7, 2, 4, 7, 2, 0, 7, 7, 4, 7, 4, 1, 4, 2, 4, 6, 6, 6, 4, 1, 1, 6,
       1, 4, 7, 2, 0, 7, 2, 7, 4, 5, 1, 0, 7, 1, 2, 2, 4, 7, 4, 6, 0, 0,
       0, 0, 7, 7, 2, 6, 0, 7, 0, 2, 7, 7, 2, 7, 4, 0, 4, 7, 7, 7, 2, 5,
       2, 0, 7, 2, 0, 7, 5, 0, 0, 1, 5, 0, 7, 4, 0, 4, 0, 0, 2, 0, 6, 6,
       6, 0, 7, 7, 0, 1, 0, 1, 5, 3, 2, 5, 7, 4, 7, 7, 0, 6, 7, 0, 7, 6,
       3, 6, 7, 0, 0, 2, 5, 5, 0, 2, 0, 6, 2, 4, 4, 0, 4, 4, 6, 6, 4, 4,
       2, 6, 7, 2, 2, 7, 7, 7, 0, 0, 2, 2, 0, 2, 7, 4, 7, 5, 6, 4, 0, 2,
       2, 0, 7, 7, 1, 0, 0, 0, 0, 7, 6, 0, 1, 0, 7, 7, 5, 6, 6, 0, 5, 6,
       0, 7, 7, 6, 7, 2, 5, 5, 7, 7, 7, 0, 6, 0, 4, 1, 6, 2, 0, 2, 2, 2,
       7, 1, 2, 7, 1, 0, 1, 6, 2, 4, 0, 4, 0, 2, 0, 2, 7, 2, 0, 5, 1, 3,
       2, 7, 0, 2, 0, 5, 1, 7, 5, 0, 6, 0, 7, 0, 2, 2, 7, 6, 0, 2, 2, 2,
       6, 0, 6, 4, 7, 1, 0, 1, 1, 0, 0, 6, 0, 0, 1, 4, 6, 2, 0, 3, 5, 5,
       0, 7, 6, 6, 7, 6, 2, 2, 6, 7, 1, 4, 0, 0, 5, 3, 7, 2, 5, 5, 2, 0,
       2, 0, 7, 7, 2, 4, 7, 4, 2, 7, 5, 5, 7, 2, 6, 2, 2, 2, 7, 7, 5, 7,
       7, 7, 0, 5, 0, 7, 5, 5, 2, 7, 2, 7, 6, 4, 4, 4, 6, 4, 5])
```

In [28]:
```
df["New cluster"]=y_predicted
df.head()
```

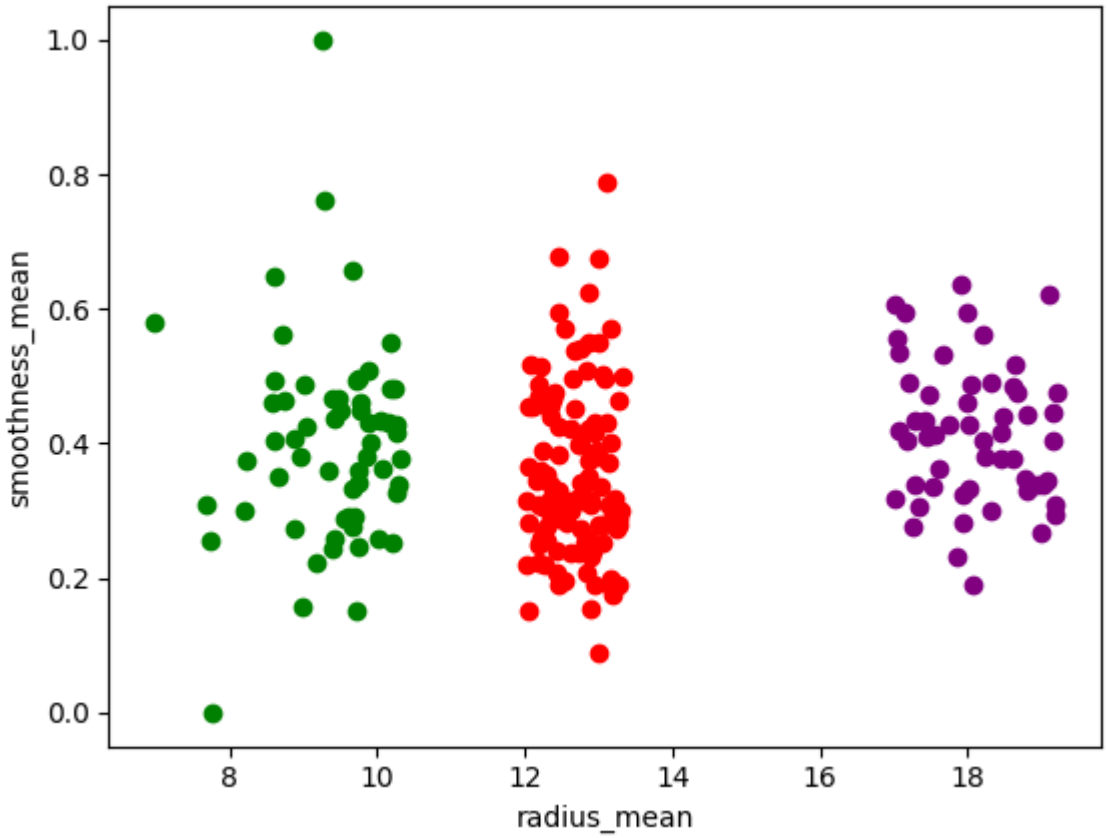
Out[28]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_n
0	842302	M	17.99	10.38	122.80	1001.0	0.59
1	842517	M	20.57	17.77	132.90	1326.0	0.28
2	84300903	M	19.69	21.25	130.00	1203.0	0.51
3	84348301	M	11.42	20.38	77.58	386.1	0.81
4	84358402	M	20.29	14.34	135.10	1297.0	0.43

5 rows × 35 columns

In [29]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["radius_mean"],df1["smoothness_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["smoothness_mean"],color="purple")
plt.scatter(df3["radius_mean"],df3["smoothness_mean"],color="green")
plt.xlabel("radius_mean")
plt.ylabel("smoothness_mean")
```

Out[29]:
```
Text(0, 0.5, 'smoothness_mean')
```

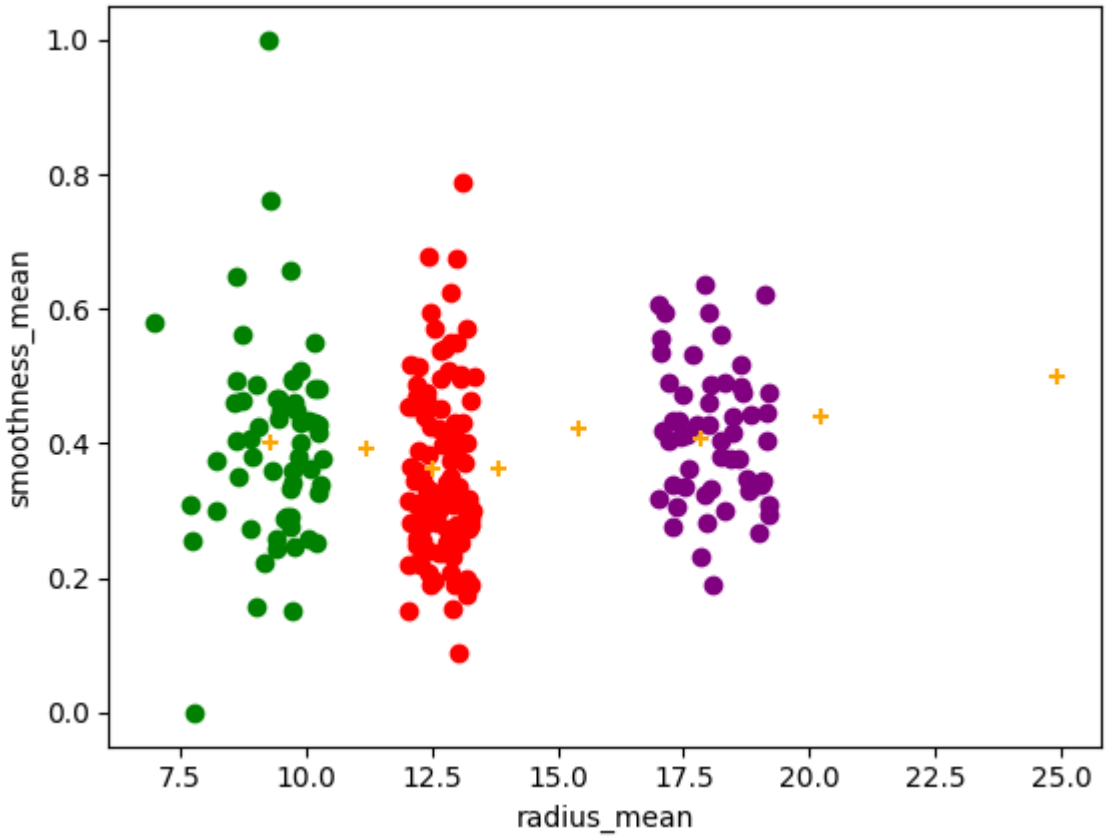


In [30]:
```
km.cluster_centers_
```

Out[30]:
```
array([[12.51557522,  0.36344942],
       [17.82188679,  0.40700857],
       [13.84166667,  0.36379479],
       [24.9125    ,  0.49896181],
       [20.21655172,  0.44015559],
       [ 9.27424074,  0.40037749],
       [15.42824324,  0.42313877],
       [11.20145631,  0.39257151]])
```

In [32]:
```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["radius_mean"],df1["smoothness_mean"],color="red")
plt.scatter(df2["radius_mean"],df2["smoothness_mean"],color="purple")
plt.scatter(df3["radius_mean"],df3["smoothness_mean"],color="green")
plt.xlabel("radius_mean")
plt.ylabel("smoothness_mean")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",m
plt.xlabel("radius_mean")
plt.ylabel("smoothness_mean")
```

Out[32]:
```
Text(0, 0.5, 'smoothness_mean')
```



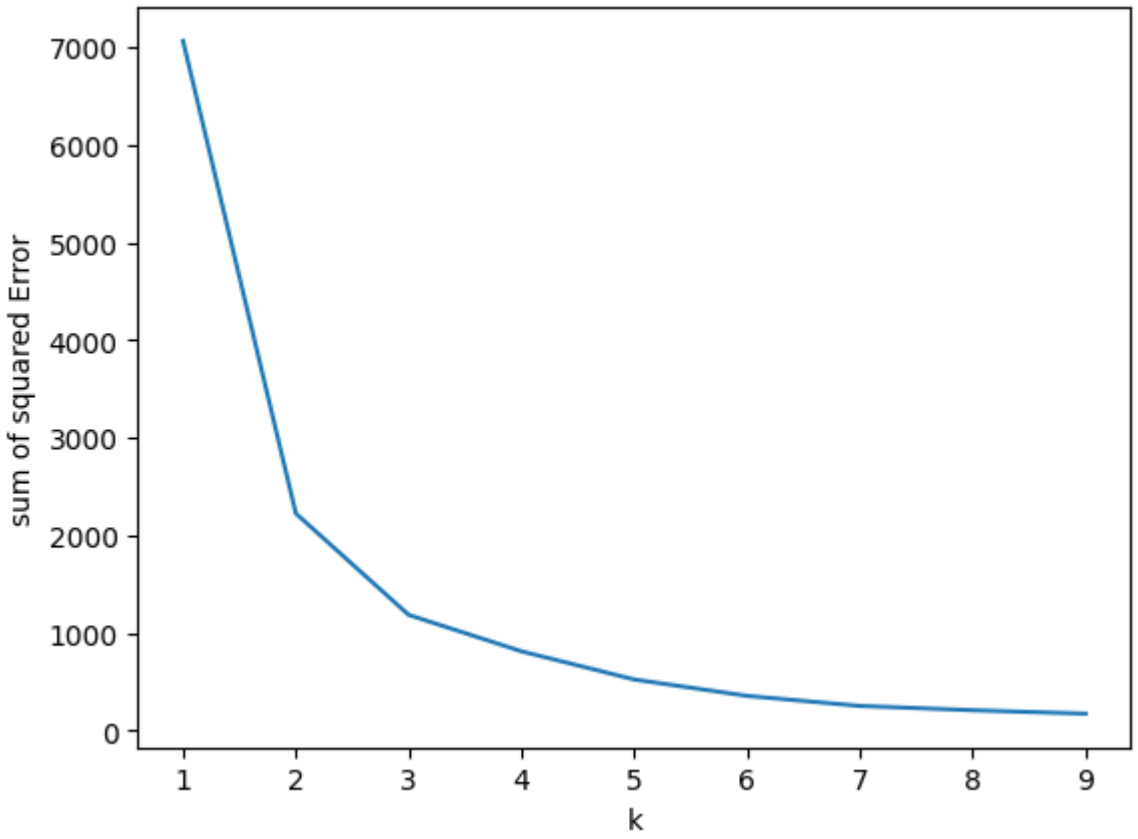
In [33]:
```
k_rng=range(1,10)
sse=[]
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["radius_mean","smoothness_mean"]])
    sse.append(km.inertia_)
sse
```

```
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
C:\Users\HP\AppData\Roaming\Python\Python310\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 wa
rning
  warnings.warn(
```

Out[33]:
```
[7063.103136812105,
 2224.638745124554,
 1186.1413422504445,
 813.0376722583254,
 524.5108878231828,
 357.1400543520583,
 254.11853395680947,
 210.23755992555377,
 174.92985533539218]
```

In [34]:
```
plt.plot(k_rng,sse)
plt.xlabel("k")
plt.ylabel("sum of squared Error")
```

Out[34]:
```
Text(0, 0.5, 'sum of squared Error')
```



CONCLUSION:The given data set is "BreastCancer Prediction".For this Dataset we have used Kmeans model.Based on the given data set we have divided Dataset in to Different Clusters.