Problem Statement:Breast Cancer Prediction

1)Data Collection

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

In [2]:
```
n=pd.read_csv(r"C:\Users\chinta pavani\Documents\BreastCancerPrediction.csv
n
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smooth
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]:
```
n.head()
```

Out[3]:

	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	

5 rows × 33 columns

In [4]:

```
n.tail()
```

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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 × 33 columns

In [5]:

```
n.shape
```

Out[5]:

```
(569, 33)
```

In [6]:

```
n.describe
```

Out[6]:

```
<bound method NDFrame.describe of            id diagnosis  radius_mean  t
exture_mean  perimeter_mean  area_mean
0      842302         M        17.99         10.38          122.80     10
01.0  \
1      842517         M        20.57         17.77          132.90     13
26.0
2    84300903         M        19.69         21.25          130.00     12
03.0
3    84348301         M        11.42         20.38           77.58      3
86.1
4    84358402         M        20.29         14.34          135.10     12
97.0
..        ...       ...          ...           ...             ...
...
564    926424         M        21.56         22.39          142.00     14
79.0
565    926682         M        20.13         28.25          131.20     12
61.0
566    926954         M        16.60         28.08          108.30      8
58.1
567    927241         M        20.60         29.33          140.10     12
65.0
568     92751         B         7.76         24.54           47.92      1
81.0

     smoothness_mean  compactness_mean  concavity_mean  concave points_me
an
0            0.11840           0.27760         0.30010              0.147
10  \
1            0.08474           0.07864         0.08690              0.070
17
2            0.10960           0.15990         0.19740              0.127
90
3            0.14250           0.28390         0.24140              0.105
20
4            0.10030           0.13280         0.19800              0.104
30
..               ...               ...             ...
...
564          0.11100           0.11590         0.24390              0.138
90
565          0.09780           0.10340         0.14400              0.097
91
566          0.08455           0.10230         0.09251              0.053
02
567          0.11780           0.27700         0.35140              0.152
00
568          0.05263           0.04362         0.00000              0.000
00

     ...  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_w
orst
0              0.66560           0.7119                0.2654          0.
4601  \
1              0.18660           0.2416                0.1860          0.
2750
2              0.42450           0.4504                0.2430          0.
3613
3              0.86630           0.6869                0.2575          0.
6638
4              0.20500           0.4000                0.1625          0.
2364
..                 ...              ...                   ...
...
564            0.21130           0.4107                0.2216          0.
2060
565            0.19220           0.3215                0.1628          0.
2572
566            0.30940           0.3403                0.1418          0.
2218
567            0.86810           0.9387                0.2650          0.
4087
568            0.06444           0.0000                0.0000          0.
2871

     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]
```

In [7]:

```
n.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

In [8]:

```
n.drop(['Unnamed: 32'],axis=1)
```

Out[8]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smooth
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	

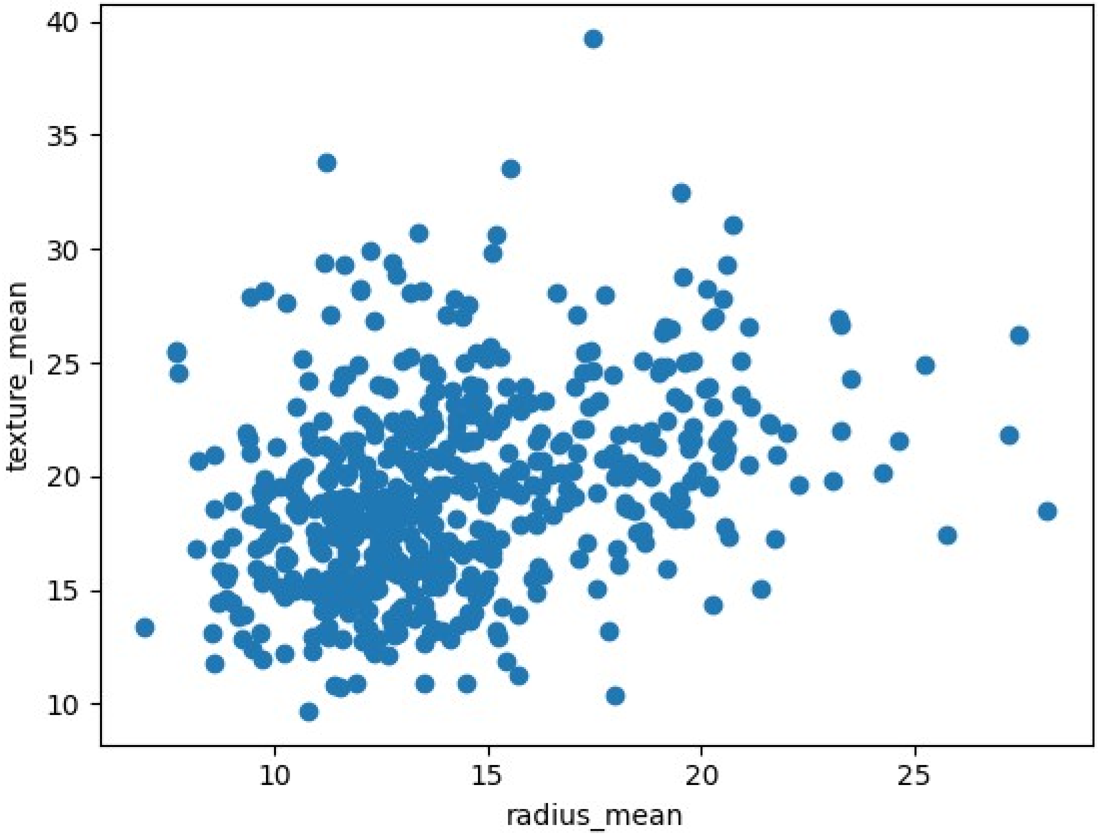
569 rows × 32 columns

In [11]:

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

Out[11]:

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



In [12]:

```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[12]:

```
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 [13]:

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

```
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
```

Out[13]:

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

In [14]:

```
n["cluster"]=y_predicted
n.head()
```

Out[14]:

	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	

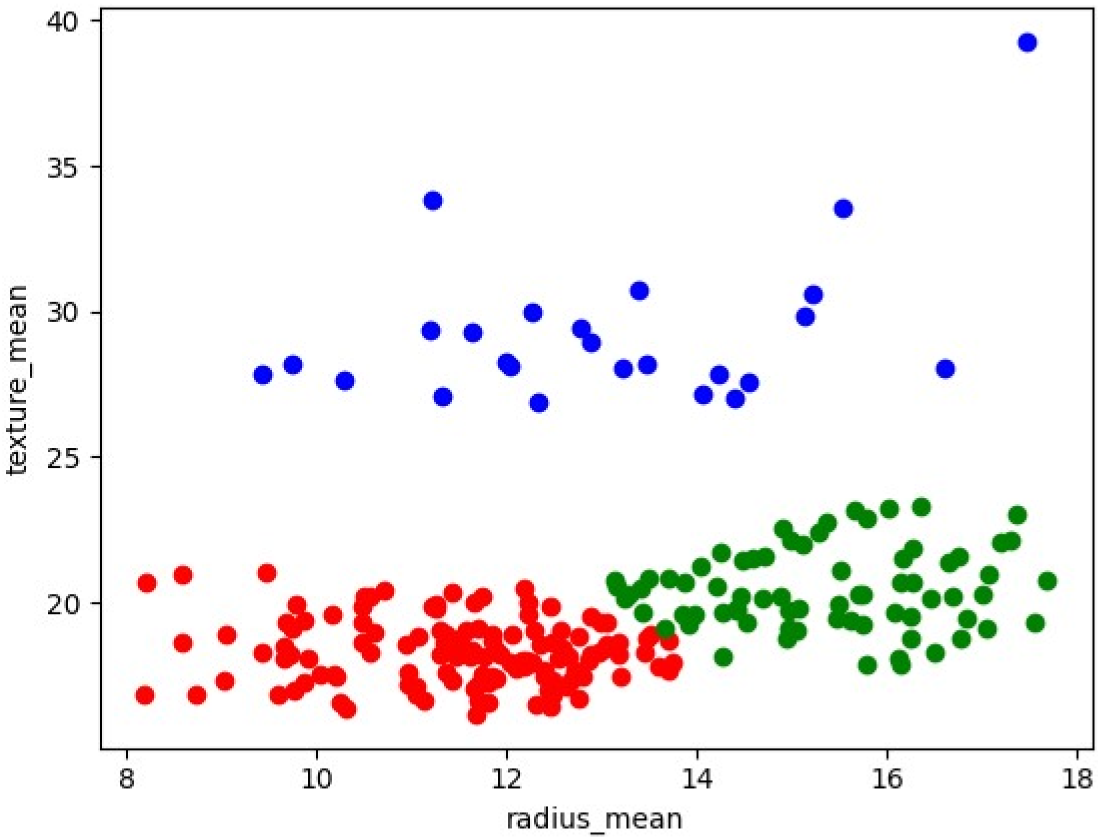
5 rows × 34 columns

In [17]:

```
df1=n[n.cluster==0]
df2=n[n.cluster==1]
df3=n[n.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[17]:

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



In [21]:

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

Out[21]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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 × 34 columns

In [23]:

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

Out[23]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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 [25]:

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

```
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
```

Out[25]:

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

In [26]:

```
n["New Cluster"]=y_predicted
n.head()
```

Out[26]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
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	

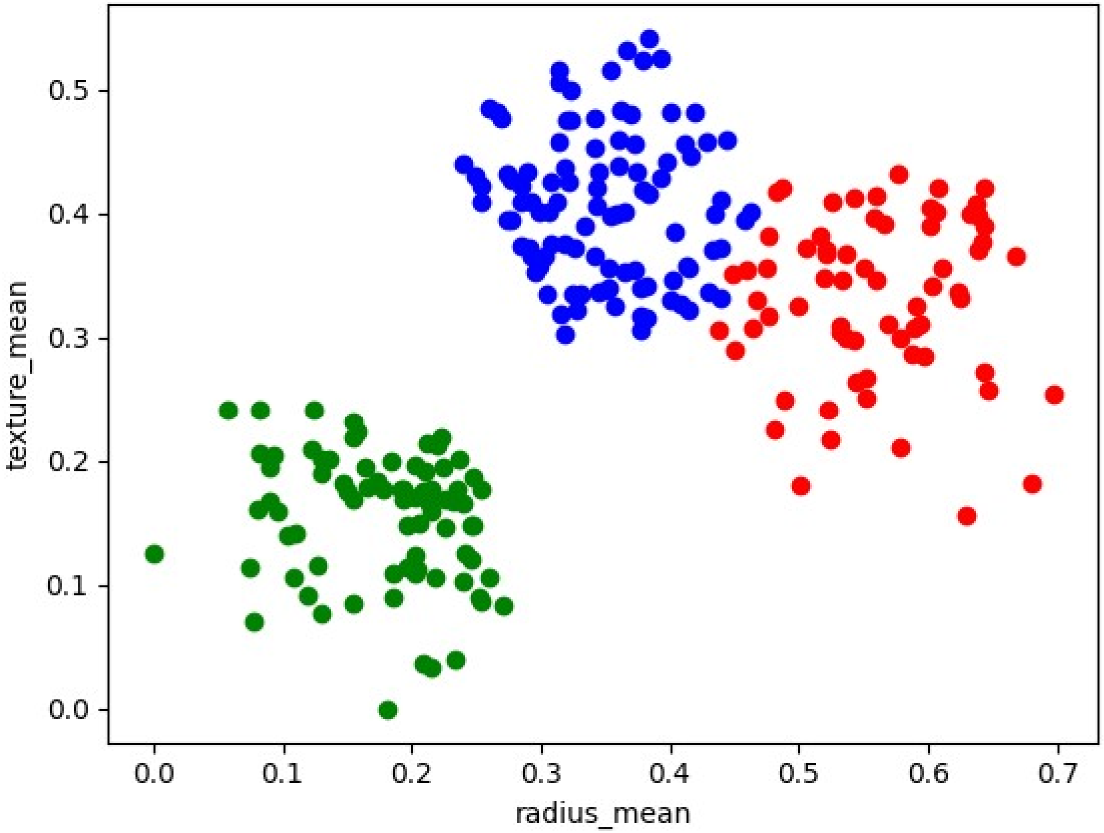
5 rows × 35 columns

In [27]:

```
df1=n[n["New Cluster"]==0]
df2=n[n["New Cluster"]==1]
df3=n[n["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[27]:

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



In [28]:

```
km.cluster_centers_
```

Out[28]:

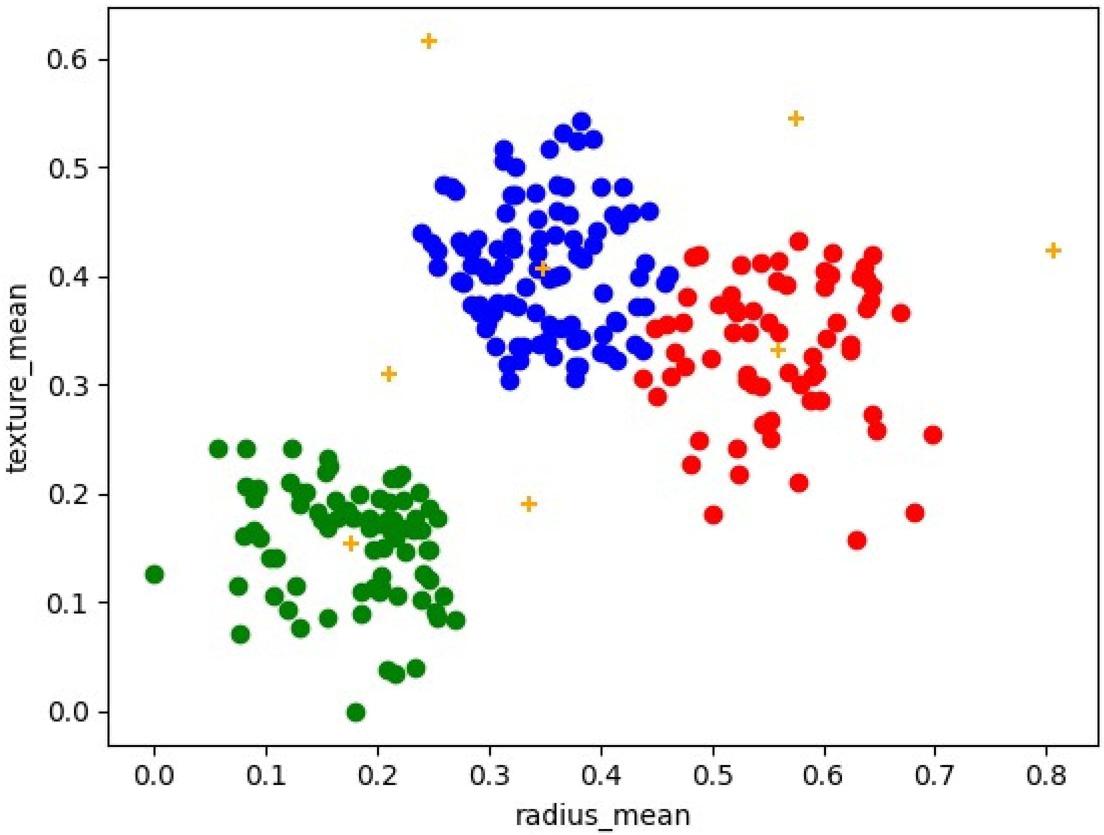
```
array([[0.55936641, 0.33176013],
       [0.17750575, 0.15412045],
       [0.34875763, 0.40662496],
       [0.24753115, 0.61622301],
       [0.80589822, 0.42316338],
       [0.57605341, 0.54408687],
       [0.33570532, 0.19063107],
       [0.21063269, 0.30993347]])
```

In [29]:

```
df1=n[n["New Cluster"]==0]
df2=n[n["New Cluster"]==1]
df3=n[n["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="orang
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[29]:

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



In [30]:

```
k_rng=range(1,10)
sse=[]
```

In [34]:

```
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(n[["radius_mean","texture_mean"]])
    sse.append(km.inertia_)
```

```
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
C:\Users\chinta pavani\AppData\Local\Programs\Python\Python311\Lib\site-p
ackages\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_ini
t` explicitly to suppress the warning
  warnings.warn(
```

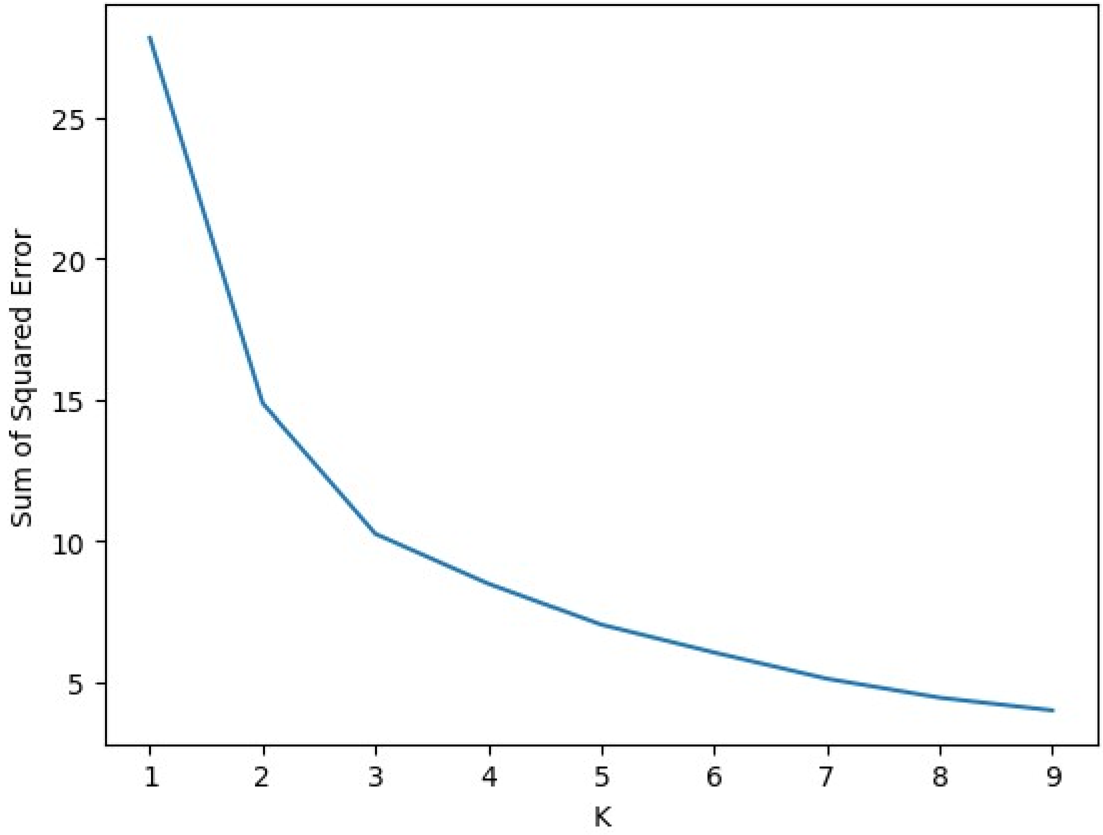
In [35]:

```
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
[27.81750759504308, 14.872032958271172, 10.2527514961052, 8.4873814407379
8, 7.034260811831778, 6.046311983660964, 5.117379110317933, 4.44439527370
828, 3.993997310928424]
```

Out[35]:

```
Text(0, 0.5, 'Sum of Squared Error')
```



Conclusion:- In Above DataSet we can use any models to get different accuracies.But by using clustering technique we can get best accuracy for the Dataset. Therefore we can conclude that breast Cancer prediction DataSet is best fit for "k-Means clustering Model.