Problem Statement:-

Predictive study using the breast cancer diagnostic dataset.

In [1]:

```python
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

Data Collection

In [2]:

```python
df=pd.read_csv(r"C:\Users\91955\Downloads\BreastCancerPrediction.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 × 33 columns

Data Cleaning and Preprocessing

In [3]:

```
1  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 × 33 columns

In [4]:

```
1  df.tail()
```

Out[4]:

|	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
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]:

```python
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]:
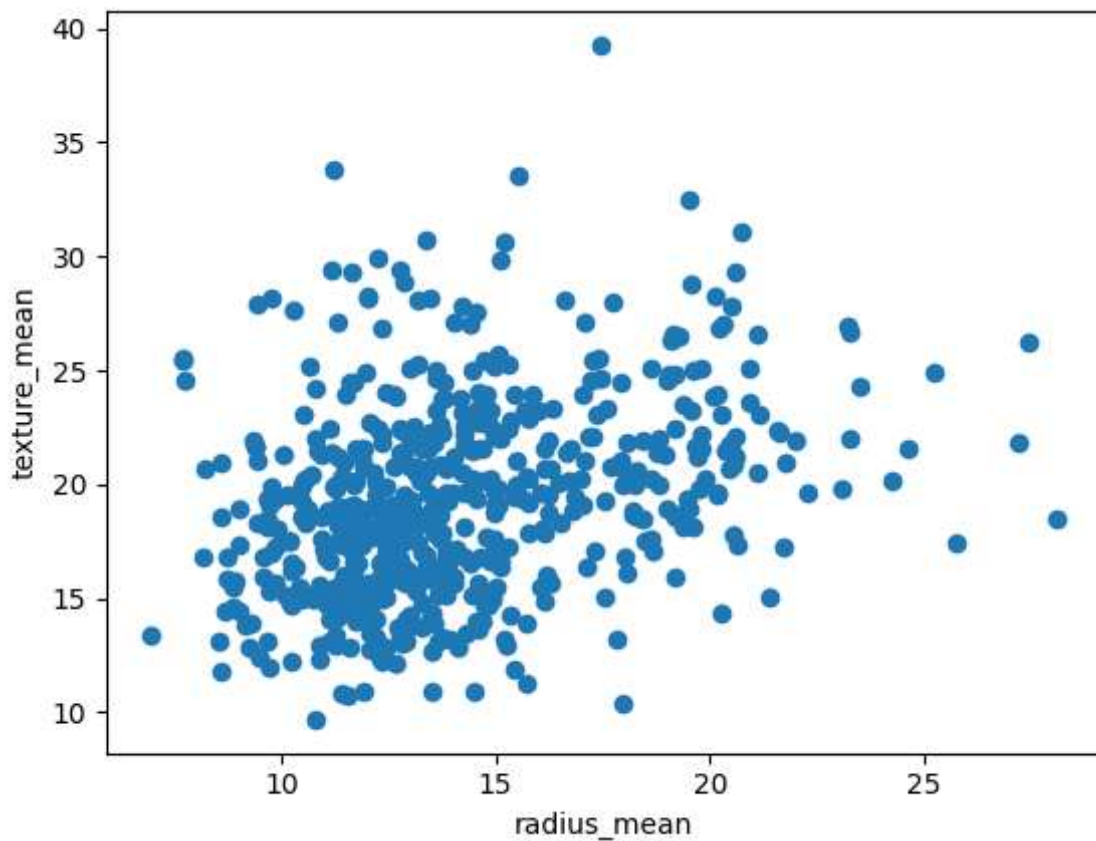
```python
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[6]:

▼ KMeans

KMeans()

In [7]:

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

C:\Users\91955\AppData\Local\Programs\Python\Python310\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[7]:

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

In [8]:

```
1  df["cluster"]=y_predicted
2  df.head()
```

Out[8]:

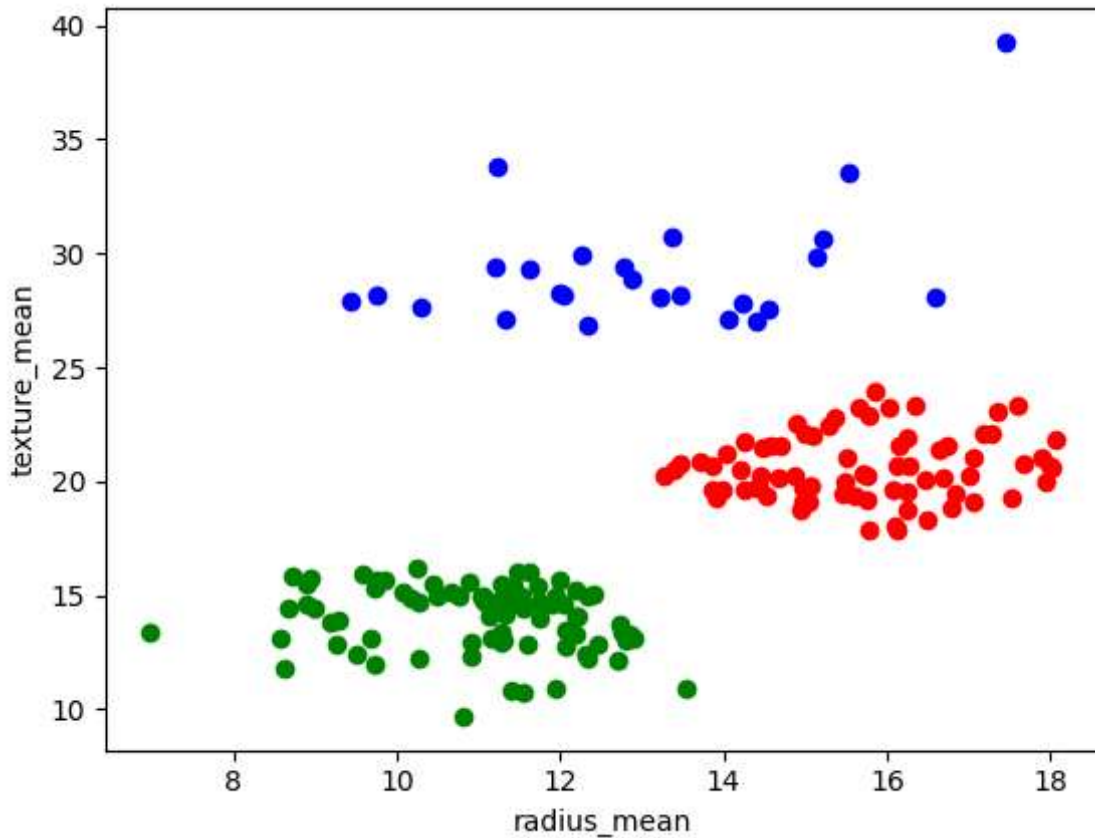|	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothnes
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

5 rows × 34 columns

In [9]:

```python
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.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[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	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 × 34 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	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 [12]:

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

C:\Users\91955\AppData\Local\Programs\Python\Python310\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[12]:

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

In [13]:

```
1  df["New Cluster"]=y_predicted
2  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 × 35 columns

In [14]:

```
1  df1=df[df["New Cluster"]==0]
2  df2=df[df["New Cluster"]==1]
3  df3=df[df["New Cluster"]==2]
4  plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
5  plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
6  plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
7  plt.xlabel("radius_mean")
8  plt.ylabel("texture_mean")
```

Out[14]:

Text(0, 0.5, 'texture_mean')
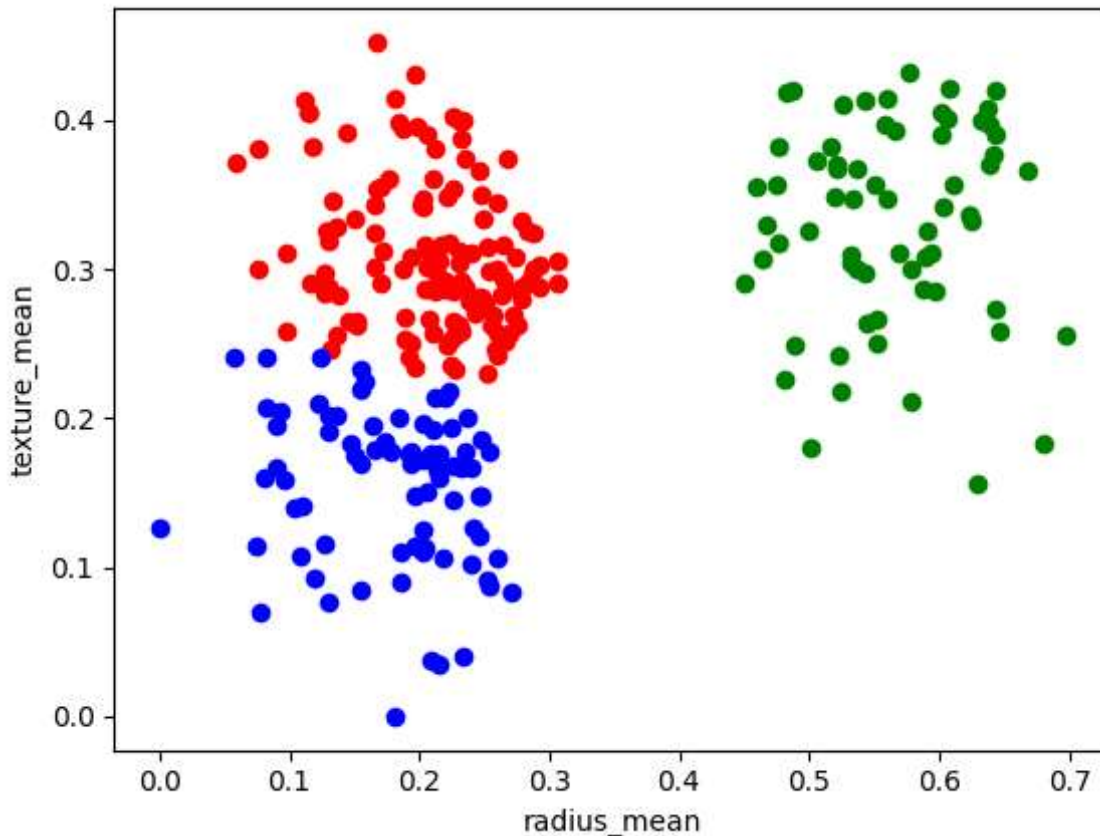
In [15]:

```
1  km.cluster_centers_
```

Out[15]:

```
array([[0.2104771 , 0.31042356],
       [0.56287997, 0.33184226],
       [0.17694105, 0.15527139],
       [0.35339953, 0.39439771],
       [0.57132058, 0.55893025],
       [0.79840767, 0.42469846],
       [0.33570532, 0.19063107],
       [0.25627183, 0.58431314]])
```

In [16]:

```
1  df1=df[df["New Cluster"]==0]
2  df2=df[df["New Cluster"]==1]
3  df3=df[df["New Cluster"]==2]
4  plt.scatter(df1["radius_mean"],df1["texture_mean"],color="red")
5  plt.scatter(df2["radius_mean"],df2["texture_mean"],color="green")
6  plt.scatter(df3["radius_mean"],df3["texture_mean"],color="blue")
7  plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker=
8  plt.xlabel("radius_mean")
9  plt.ylabel("texture_mean")
```

Out[16]:

Text(0, 0.5, 'texture_mean')
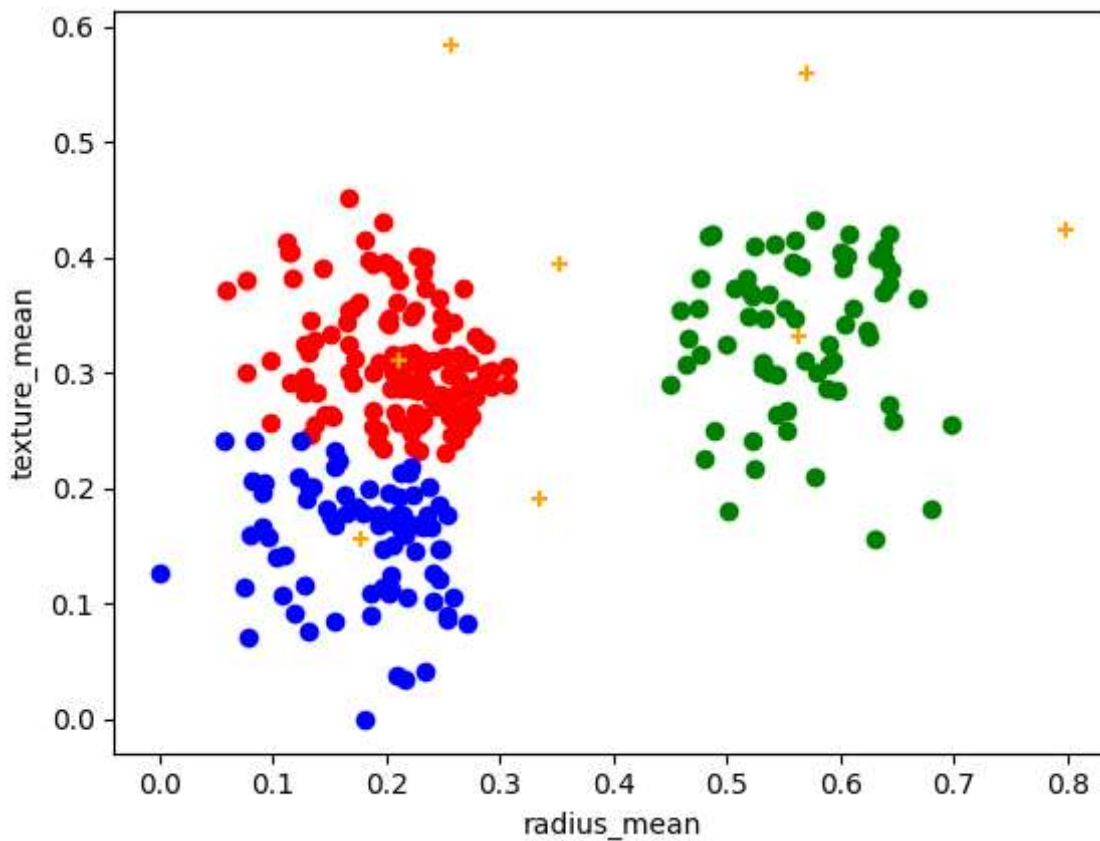
In [17]:

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

In [18]:

```python
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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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\91955\AppData\Local\Programs\Python\Python310\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(

[27.817507595043075, 14.87203295827117, 10.252751496105198, 8.487131283091
337, 7.030021844241492, 6.0300900841176945, 5.118334570400575, 4.444287355
88123, 4.027491396780893]
```
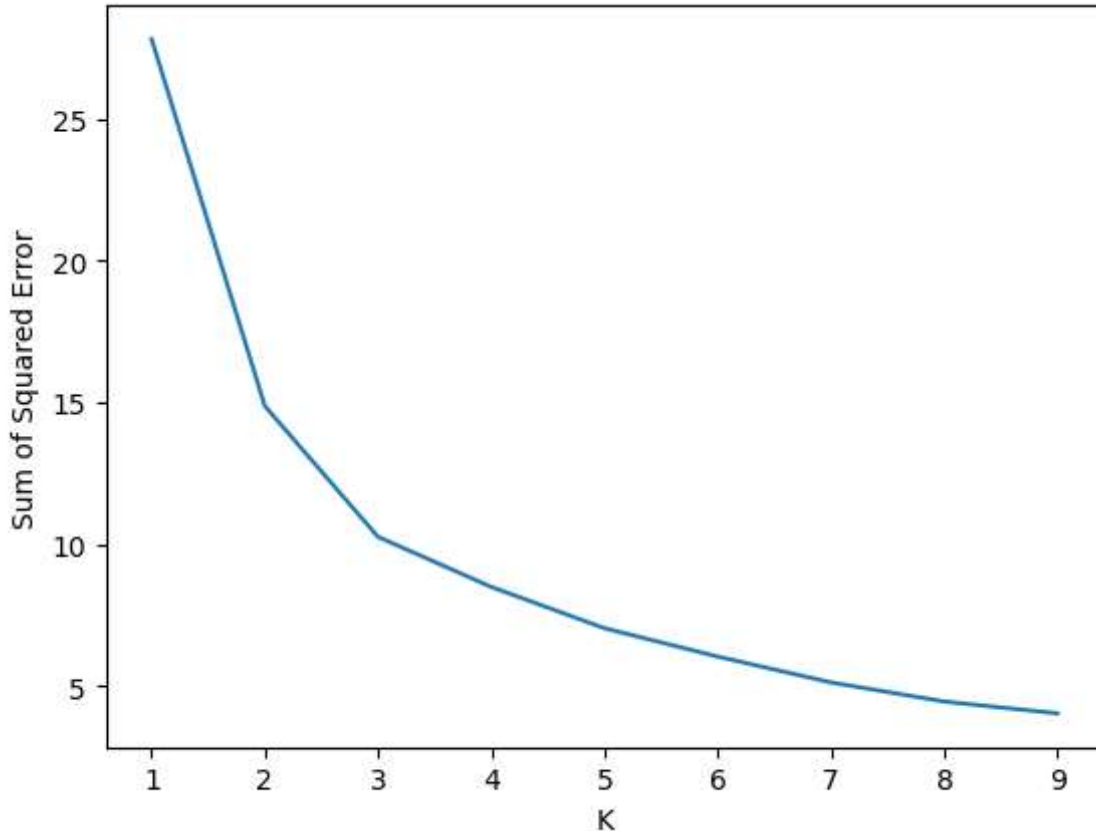
```
C:\Users\91955\AppData\Local\Programs\Python\Python310\lib\site-packages\s
klearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init
` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicit
ly to suppress the warning
  warnings.warn(
```

Out[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 that accuracies is not good so,that is why we
will take it as a clustering and done with K-Means Clustering.**

In [ ]:

```
1
```