DATASET: Breast Cancer Prediction

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

DATA COLLECTION

In [2]:
```
df=pd.read_csv(r"C:\Users\91720\Downloads\BreastCancerPrediction.csv")
df
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	poin
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	
...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	

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_mean	compactness_mean	concavity_mean	co points_
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0

5 rows × 33 columns

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

Out[4]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	con points_m
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.00
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.04
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.03
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.00
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.02
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.03
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.07
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.20

8 rows × 32 columns

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
Unnamed: 32                569
dtype: int64
```

In [6]:
```
df.duplicated().sum()
```

Out[6]:
```
0
```

In [7]:
```
df.shape
```

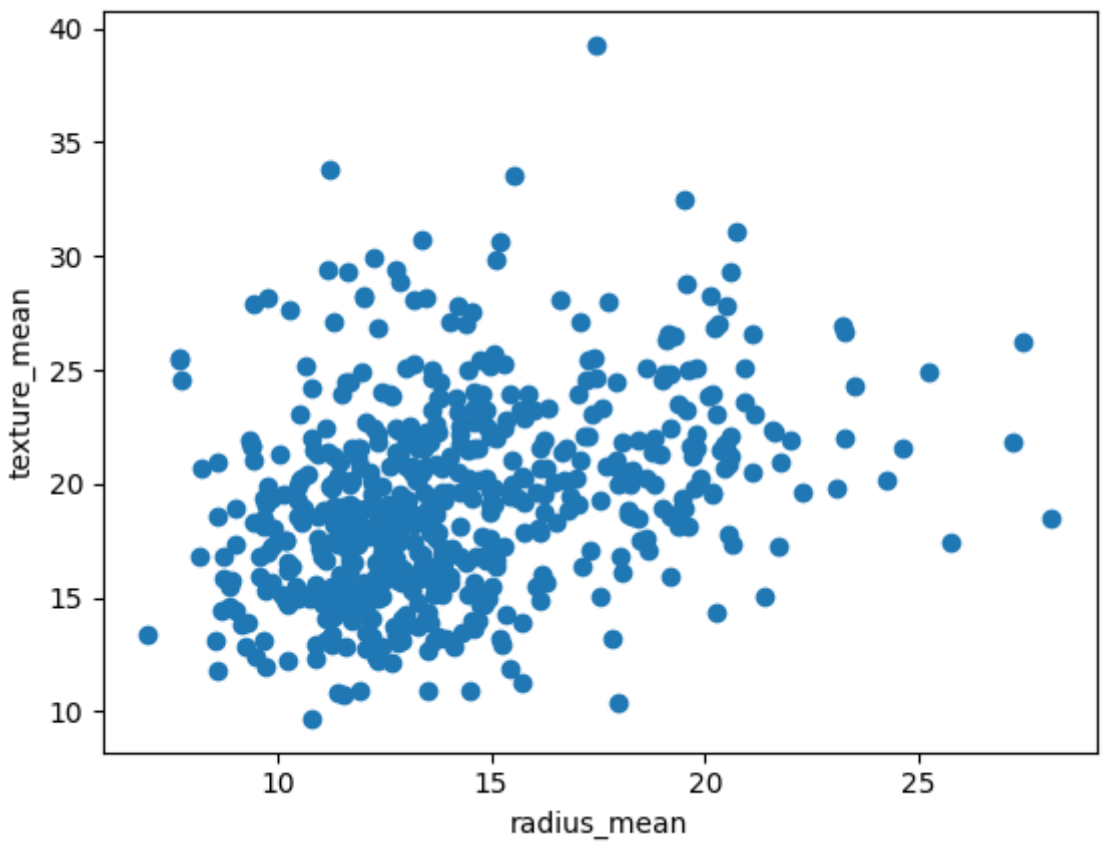
Out[7]:
```
(569, 33)
```

In [8]:
```
df.sum()
```

Out[8]:
```
id                                                      17281572085
diagnosis                  MMMMMMMMMMMMMMMMMMMBBBMMMMMMMMMMMMMMMMBMMMMMMMMM...
radius_mean                                                 8038.429
texture_mean                                                10975.81
perimeter_mean                                              52330.38
area_mean                                                   372631.9
smoothness_mean                                               54.829
compactness_mean                                            59.37002
concavity_mean                                              50.526811
concave points_mean                                         27.834994
symmetry_mean                                                103.0811
fractal_dimension_mean                                       35.73184
radius_se                                                    230.5429
texture_se                                                   692.3896
perimeter_se                                               1630.7877
area_se                                                     22951.798
smoothness_se                                                4.006317
compactness_se                                              14.497061
concavity_se                                                18.147525
concave points_se                                            6.712002
symmetry_se                                                 11.688568
fractal_dimension_se                                          2.1593
radius_worst                                                 9257.169
texture_worst                                                14610.34
perimeter_worst                                              61031.63
area_worst                                                   501051.8
smoothness_worst                                             75.31773
compactness_worst                                           144.67681
concavity_worst                                            154.875247
concave points_worst                                        65.210941
symmetry_worst                                                165.053
fractal_dimension_worst                                       47.76517
Unnamed: 32                                                        0.0
dtype: object
```

In [9]:
```
plt.scatter(df["radius_mean"],df["texture_mean"])
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

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



using KMeans Cluster

In [10]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[10]:
```
KMeans()
```

In [11]:
```
y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

```
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarni
ng: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to supp
ress the warning
  warnings.warn(
```

Out[11]:
```
array([3, 1, 1, 7, 1, 3, 1, 4, 2, 2, 4, 4, 5, 2, 2, 6, 4, 4, 1, 3, 3, 0,
       3, 5, 4, 3, 4, 1, 2, 3, 5, 7, 5, 5, 4, 4, 4, 7, 2, 4, 2, 2, 5, 4,
       2, 1, 7, 7, 0, 2, 2, 3, 7, 1, 4, 7, 1, 4, 7, 0, 0, 7, 2, 0, 2, 2,
       7, 7, 7, 3, 1, 0, 5, 3, 7, 4, 0, 3, 5, 7, 2, 3, 5, 5, 0, 1, 4, 5,
       2, 3, 2, 4, 3, 7, 4, 5, 7, 7, 0, 4, 2, 0, 7, 7, 7, 3, 7, 7, 1, 2,
       7, 2, 4, 7, 0, 2, 0, 3, 4, 1, 0, 1, 1, 3, 3, 3, 2, 1, 3, 5, 0, 4,
       4, 3, 1, 2, 7, 0, 3, 0, 0, 4, 7, 3, 0, 0, 7, 4, 3, 7, 2, 7, 0, 0,
       3, 7, 4, 4, 0, 0, 7, 1, 1, 2, 1, 4, 0, 4, 5, 3, 0, 4, 3, 0, 0, 0,
       7, 4, 2, 0, 1, 5, 4, 0, 4, 0, 1, 7, 7, 3, 2, 2, 7, 6, 2, 3, 2, 1,
       1, 4, 7, 4, 5, 2, 7, 3, 7, 4, 2, 3, 1, 7, 1, 5, 2, 3, 7, 7, 1, 5,
       3, 3, 7, 4, 3, 3, 0, 3, 2, 2, 4, 6, 6, 5, 0, 4, 5, 1, 6, 6, 3, 0,
       7, 2, 5, 7, 7, 3, 2, 0, 5, 7, 1, 3, 1, 3, 5, 3, 4, 6, 5, 4, 4, 4,
       4, 5, 7, 2, 3, 7, 3, 0, 1, 0, 5, 7, 0, 1, 7, 3, 5, 0, 1, 4, 3, 7,
       2, 0, 7, 7, 4, 4, 3, 7, 0, 3, 0, 7, 4, 2, 1, 7, 5, 7, 7, 2, 3, 0,
       3, 3, 7, 3, 0, 0, 7, 7, 0, 1, 7, 7, 0, 1, 0, 1, 0, 7, 3, 7, 4, 4,
       3, 7, 7, 0, 7, 4, 3, 1, 7, 5, 3, 7, 0, 1, 0, 0, 7, 3, 0, 0, 7, 4,
       1, 2, 0, 7, 7, 3, 0, 7, 7, 2, 7, 4, 3, 1, 5, 7, 1, 1, 4, 3, 1, 1,
       3, 3, 7, 6, 3, 7, 0, 0, 2, 7, 3, 2, 0, 3, 0, 5, 0, 7, 4, 1, 7, 3,
       7, 7, 0, 7, 1, 0, 7, 3, 0, 7, 3, 2, 1, 7, 7, 7, 2, 4, 6, 2, 2, 4,
       0, 2, 7, 3, 0, 4, 7, 2, 0, 2, 7, 7, 4, 7, 1, 1, 3, 4, 7, 3, 4, 3,
       7, 5, 3, 7, 1, 2, 5, 3, 4, 1, 2, 5, 6, 3, 7, 6, 6, 2, 2, 6, 5, 5,
       6, 7, 7, 4, 4, 7, 5, 7, 7, 6, 3, 6, 0, 3, 4, 3, 0, 4, 7, 4, 3, 7,
       3, 7, 3, 1, 7, 4, 2, 3, 1, 0, 4, 4, 7, 7, 1, 1, 3, 2, 3, 1, 0, 0,
       7, 7, 3, 2, 0, 3, 4, 3, 4, 7, 1, 1, 7, 7, 0, 1, 7, 7, 0, 0, 7, 0,
       3, 0, 7, 7, 3, 1, 7, 1, 2, 2, 2, 2, 0, 2, 2, 6, 4, 2, 7, 7, 7, 2,
       2, 2, 6, 2, 6, 6, 7, 6, 2, 2, 6, 6, 6, 5, 1, 5, 6, 5, 2])
```

In [12]:
```
df["cluster"]=y_predicted
df.head()
```

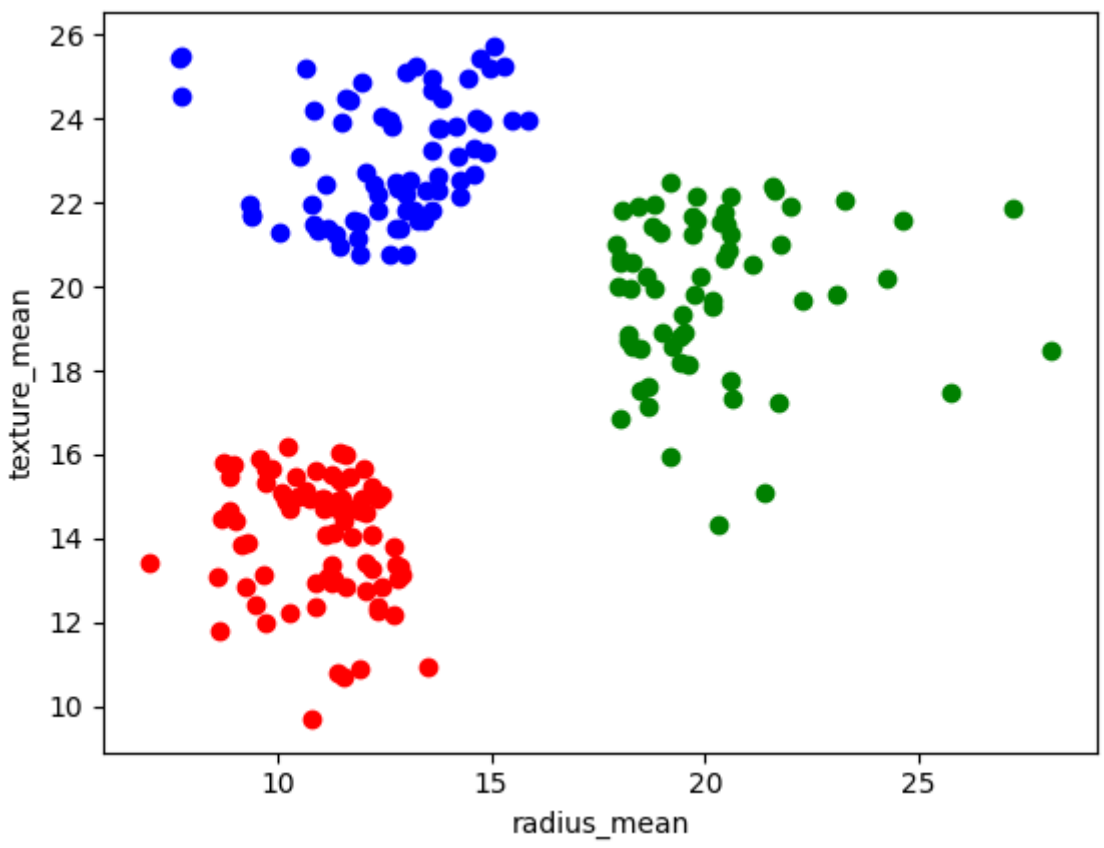
Out[12]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points_
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0

5 rows × 34 columns

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



In [15]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["texture_mean"]])
df["texture_mean"]=scaler.transform(df[["texture_mean"]])
df.head()
```

Out[15]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points_
0	842302	M	17.99	0.022658	122.80	1001.0	0.11840	0.27760	0.3001	0
1	842517	M	20.57	0.272574	132.90	1326.0	0.08474	0.07864	0.0869	0
2	84300903	M	19.69	0.390260	130.00	1203.0	0.10960	0.15990	0.1974	0
3	84348301	M	11.42	0.360839	77.58	386.1	0.14250	0.28390	0.2414	0
4	84358402	M	20.29	0.156578	135.10	1297.0	0.10030	0.13280	0.1980	0

5 rows × 34 columns

In [16]:
```
scaler.fit(df[["radius_mean"]])
df["radius_mean"]=scaler.transform(df[["radius_mean"]])
df.head()
```

Out[16]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points_
0	842302	M	0.521037	0.022658	122.80	1001.0	0.11840	0.27760	0.3001	0
1	842517	M	0.643144	0.272574	132.90	1326.0	0.08474	0.07864	0.0869	0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.10960	0.15990	0.1974	0
3	84348301	M	0.210090	0.360839	77.58	386.1	0.14250	0.28390	0.2414	0
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.10030	0.13280	0.1980	0

5 rows × 34 columns

In [17]:
```
km.cluster_centers_
```

Out[17]:
```
array([[10.9873    , 14.001625  ],
       [20.21428571, 19.85968254],
       [12.59061644, 23.00191781],
       [14.31370787, 15.29797753],
       [15.31726027, 20.43589041],
       [19.9335    , 25.8125    ],
       [13.05352   , 29.3064    ],
       [11.51644444, 18.30857143]])
```

In [18]:
```
df["New Cluster"]=y_predicted
df.head()
```

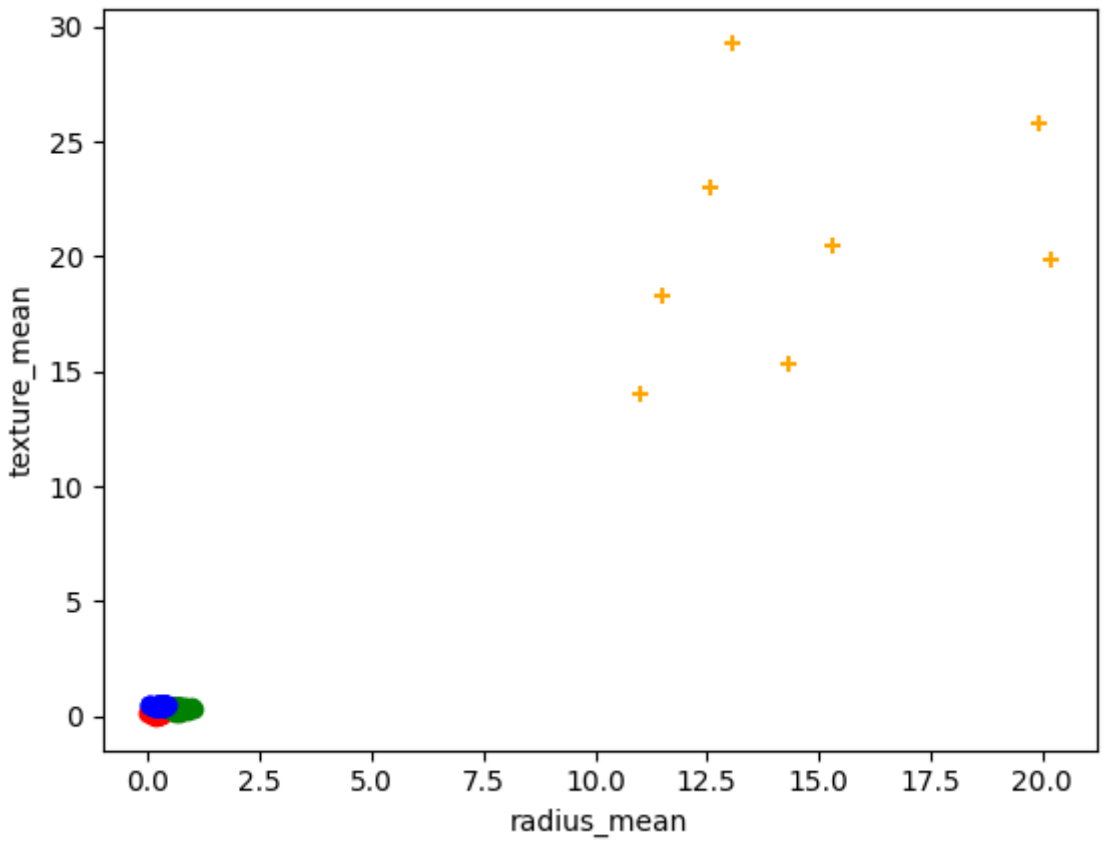
Out[18]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	co points_
0	842302	M	0.521037	0.022658	122.80	1001.0	0.11840	0.27760	0.3001	0
1	842517	M	0.643144	0.272574	132.90	1326.0	0.08474	0.07864	0.0869	0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.10960	0.15990	0.1974	0
3	84348301	M	0.210090	0.360839	77.58	386.1	0.14250	0.28390	0.2414	0
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.10030	0.13280	0.1980	0

5 rows × 35 columns

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"],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="orange",marker="+")
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[19]:
```
Text(0, 0.5, 'texture_mean')
```



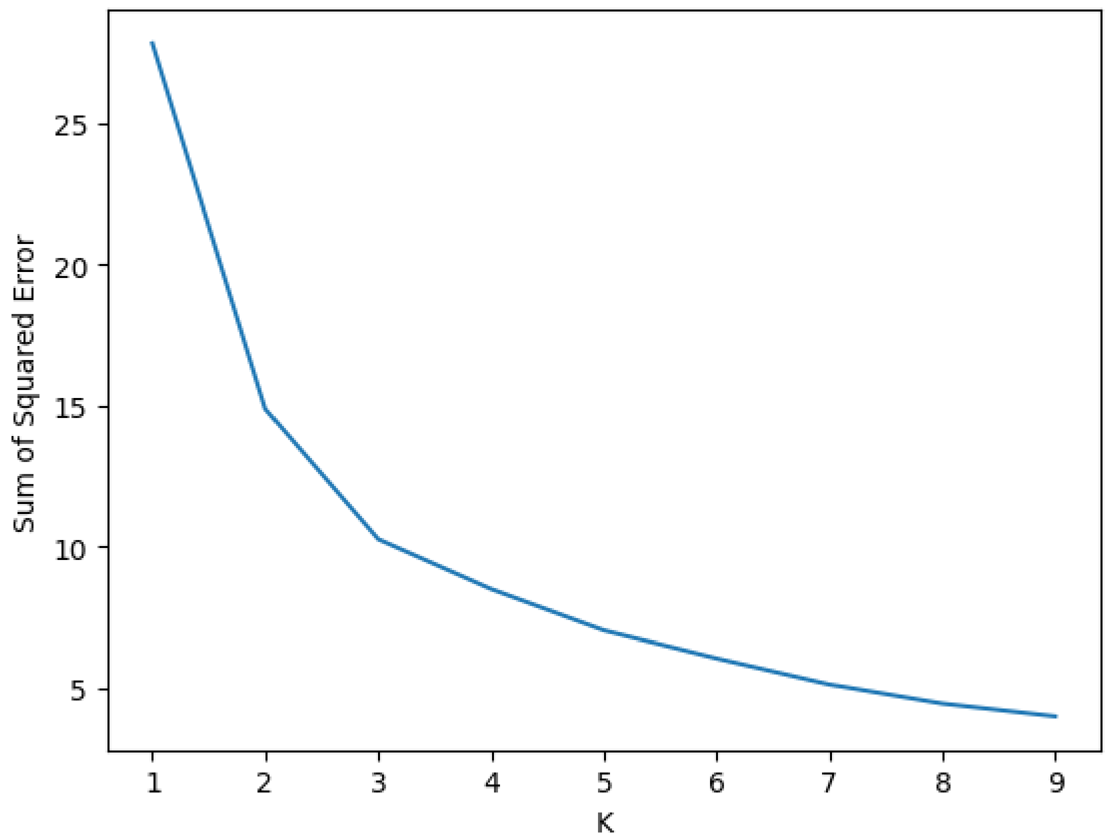
In [20]:
```
k_rng=range(1,10)
sse=[]
```

In [22]:
```
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["radius_mean","texture_mean"]])
 sse.append(km.inertia_)
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(
C:\Users\91720\AppData\Local\Programs\Python\Python311\Lib\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 warning
  warnings.warn(

[27.81750759504308, 14.872032958271172, 10.2527514961052, 8.490050221511442, 7.030668267339053, 6.038177754620531, 5.117927753802226, 4.44413586208949, 3.991627647771391]
```

Out[22]:
```
Text(0, 0.5, 'Sum of Squared Error')
```



CONCLUSION

For the given dataset we can use multiple models,forthat models we get different types of accuracies but that accuracies is not good so,that's why we will take it as a clustering and done with K-Means Clustering