Problem Statement : Predictive study using the breast cancer diagnostic data set

Data Collection

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

In [2]:
```
df=pd.read_csv(r"C:\Users\91903\Downloads\BreastCancerPrediction.csv")
df
```

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 x 32 columns

Data Cleaning and preprocessing

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

Out[3]:

hness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	texture_worst	
0.11840	0.27760	0.3001	0.14710	...	25.38	17.33	
0.08474	0.07864	0.0869	0.07017	...	24.99	23.41	
0.10960	0.15990	0.1974	0.12790	...	23.57	25.53	
0.14250	0.28390	0.2414	0.10520	...	14.91	26.50	
0.10030	0.13280	0.1980	0.10430	...	22.54	16.67	

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

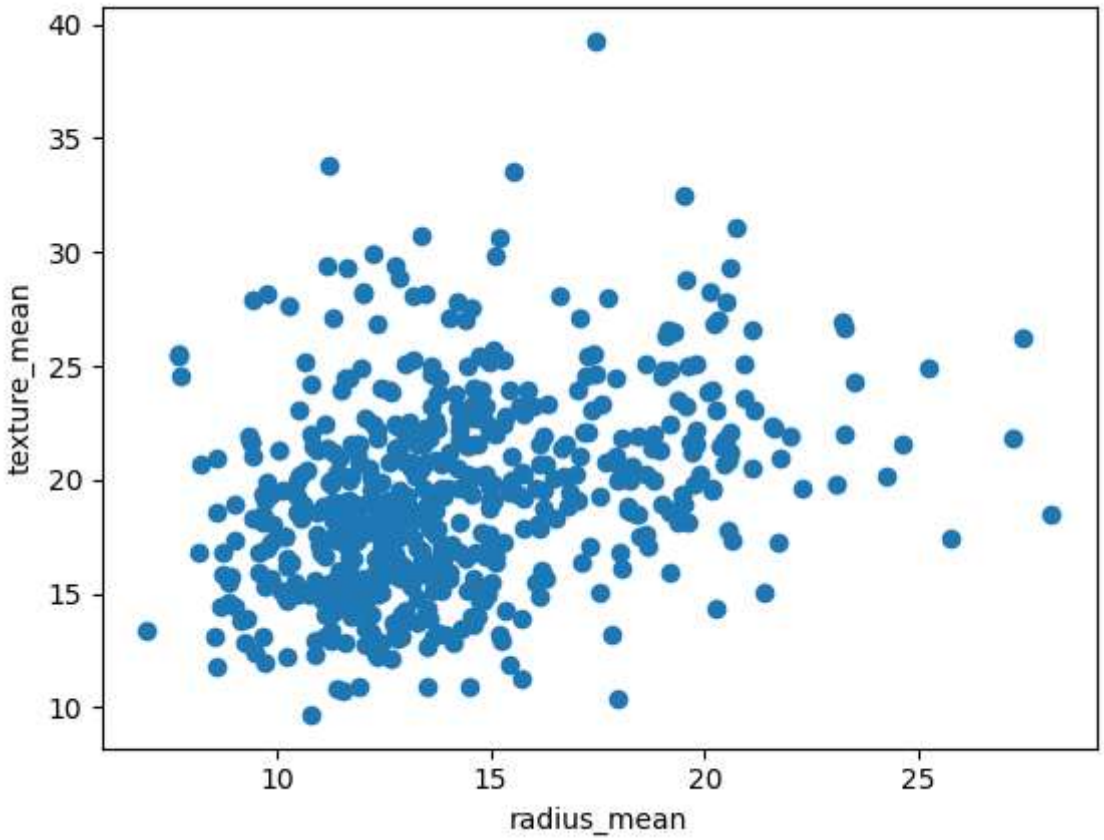
Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
564	926424	M	21.56	22.39	142.00	1479.0	0.1
565	926682	M	20.13	28.25	131.20	1261.0	0.09
566	926954	M	16.60	28.08	108.30	858.1	0.08
567	927241	M	20.60	29.33	140.10	1265.0	0.1
568	92751	B	7.76	24.54	47.92	181.0	0.05

5 rows × 32 columns

In [5]:
```
plt.scatter(df["radius_mean"],df["texture_mean"])
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[5]:
```
Text(0, 0.5, 'texture_mean')
```



K Means Clustering

In [6]:
```
from sklearn.cluster import KMeans
km=KMeans()
km
```

Out[6]:
```
KMeans()
```

In [7]:
```
y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

```
C:\Users\91903\AppData\Local\Programs\Python\Python310\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(
```

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

In [8]:
```
df["cluster"]=y_predicted
df.head()
```

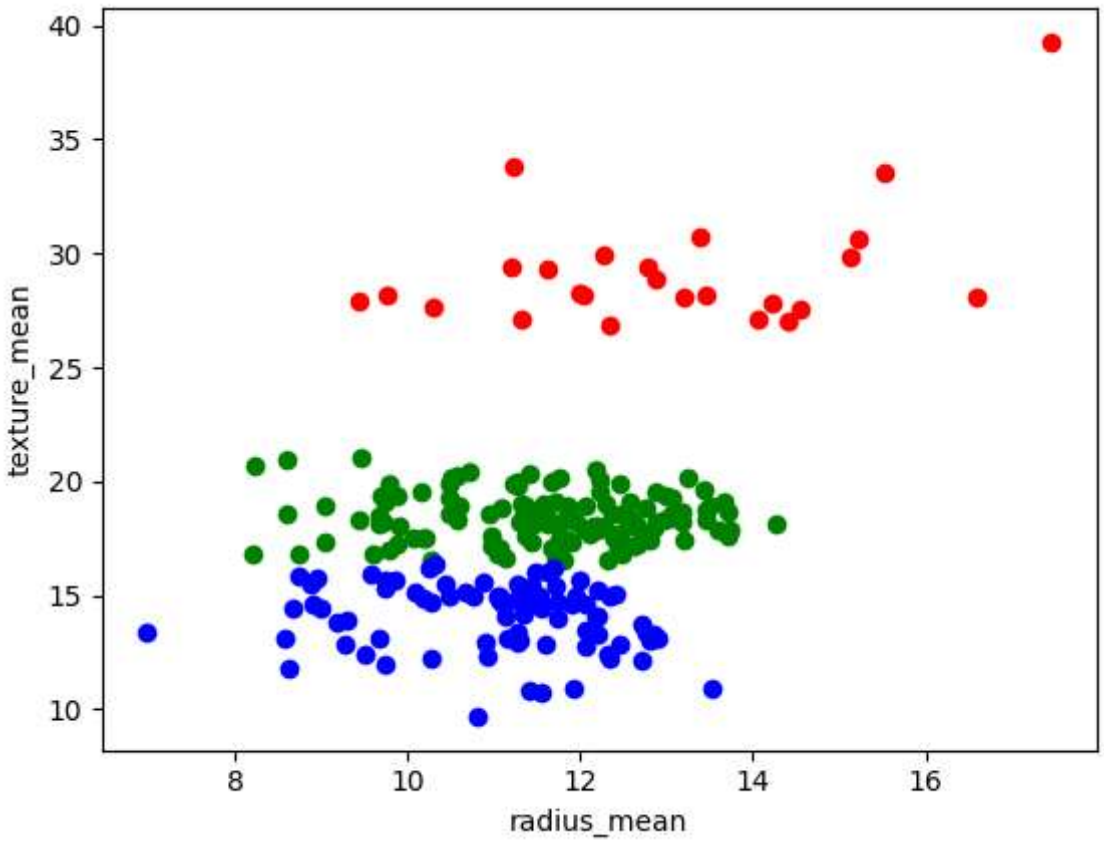
Out[8]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
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 [9]:
```
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	smoothness_m
0	842302	M	17.99	0.022658	122.80	1001.0	0.1
1	842517	M	20.57	0.272574	132.90	1326.0	0.0
2	84300903	M	19.69	0.390260	130.00	1203.0	0.1
3	84348301	M	11.42	0.360839	77.58	386.1	0.1
4	84358402	M	20.29	0.156578	135.10	1297.0	0.1

5 rows × 33 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	smoothness_m
0	842302	M	0.521037	0.022658	122.80	1001.0	0.1
1	842517	M	0.643144	0.272574	132.90	1326.0	0.0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.1
3	84348301	M	0.210090	0.360839	77.58	386.1	0.1
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.1

5 rows × 33 columns

In [12]:
```
y_predicted=km.fit_predict(df[["radius_mean","texture_mean"]])
y_predicted
```

```
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
```

Out[12]:
```
array([2, 6, 6, 5, 6, 2, 6, 1, 1, 4, 1, 2, 7, 1, 1, 4, 1, 1, 6, 2, 2, 3,
       2, 0, 1, 6, 1, 6, 1, 6, 7, 5, 7, 7, 2, 1, 1, 5, 1, 1, 1, 5, 7, 1,
       1, 6, 3, 5, 3, 1, 5, 2, 5, 6, 1, 5, 6, 1, 5, 3, 3, 5, 1, 3, 4, 1,
       5, 5, 5, 2, 6, 3, 7, 2, 5, 1, 2, 6, 7, 5, 5, 2, 0, 7, 3, 6, 1, 7,
       1, 2, 1, 1, 2, 5, 1, 7, 5, 5, 3, 1, 4, 3, 5, 5, 5, 2, 5, 5, 0, 5,
       3, 5, 1, 5, 3, 5, 3, 2, 1, 6, 3, 6, 0, 2, 2, 2, 4, 6, 2, 7, 3, 1,
       1, 2, 6, 1, 5, 3, 2, 3, 3, 2, 5, 2, 3, 3, 5, 1, 2, 2, 1, 5, 3, 3,
       2, 5, 6, 6, 3, 3, 5, 6, 6, 1, 0, 1, 3, 6, 7, 2, 3, 1, 2, 3, 3, 3,
       5, 1, 1, 2, 0, 7, 1, 3, 1, 3, 6, 5, 5, 2, 1, 1, 5, 4, 1, 2, 1, 6,
       6, 1, 5, 6, 0, 1, 5, 2, 5, 6, 1, 2, 6, 5, 0, 7, 1, 2, 5, 5, 6, 7,
       2, 2, 5, 1, 2, 2, 3, 2, 4, 1, 6, 4, 4, 7, 3, 1, 0, 6, 4, 7, 2, 2,
       5, 1, 7, 5, 2, 2, 4, 3, 7, 5, 6, 6, 6, 2, 7, 2, 1, 4, 7, 7, 6, 1,
       6, 7, 5, 1, 2, 5, 2, 3, 0, 3, 7, 5, 3, 6, 2, 2, 7, 3, 6, 6, 2, 5,
       5, 2, 5, 5, 1, 1, 2, 5, 2, 2, 3, 5, 2, 5, 6, 5, 7, 5, 5, 4, 2, 3,
       2, 2, 5, 2, 2, 3, 5, 5, 3, 6, 5, 5, 3, 6, 2, 6, 3, 5, 2, 5, 1, 1,
       2, 5, 5, 3, 5, 6, 2, 6, 5, 0, 2, 3, 3, 6, 3, 3, 5, 2, 3, 3, 5, 1,
       0, 4, 3, 5, 5, 2, 3, 5, 5, 1, 5, 6, 2, 6, 7, 5, 6, 0, 1, 2, 6, 6,
       2, 2, 5, 4, 2, 5, 3, 3, 1, 5, 2, 1, 3, 2, 3, 7, 3, 3, 1, 0, 5, 2,
       5, 5, 3, 5, 6, 3, 5, 2, 3, 5, 2, 1, 6, 5, 5, 5, 5, 1, 4, 5, 5, 1,
       3, 5, 5, 2, 3, 1, 5, 5, 3, 5, 5, 5, 1, 5, 6, 6, 2, 1, 5, 2, 1, 2,
       5, 7, 2, 5, 6, 4, 7, 2, 1, 6, 5, 7, 4, 2, 5, 4, 4, 4, 4, 4, 7, 0,
       4, 5, 5, 1, 1, 5, 7, 5, 5, 4, 2, 4, 3, 2, 1, 2, 3, 1, 5, 1, 2, 2,
       2, 2, 2, 6, 3, 6, 1, 2, 6, 3, 1, 1, 5, 5, 6, 6, 2, 1, 2, 0, 3, 3,
       5, 5, 2, 1, 3, 2, 1, 2, 1, 5, 6, 6, 5, 2, 3, 0, 5, 1, 3, 3, 5, 3,
       2, 3, 5, 5, 2, 6, 5, 6, 1, 4, 4, 4, 3, 1, 4, 4, 1, 1, 3, 3, 5, 4,
       5, 5, 4, 5, 4, 4, 5, 4, 1, 4, 4, 4, 4, 7, 0, 7, 7, 7, 4])
```

In [13]:
```
df["New Cluster"]=y_predicted
df.head()
```

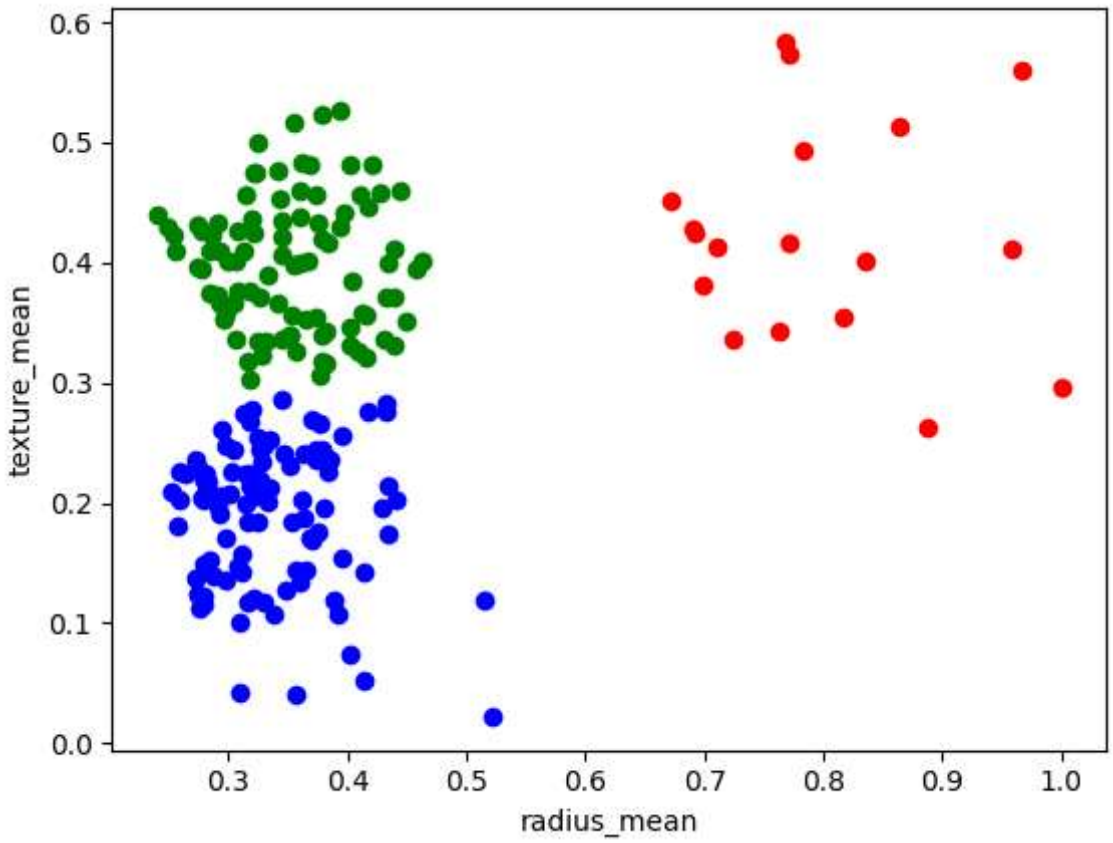
Out[13]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_m
0	842302	M	0.521037	0.022658	122.80	1001.0	0.1
1	842517	M	0.643144	0.272574	132.90	1326.0	0.0
2	84300903	M	0.601496	0.390260	130.00	1203.0	0.1
3	84348301	M	0.210090	0.360839	77.58	386.1	0.1
4	84358402	M	0.629893	0.156578	135.10	1297.0	0.1

5 rows × 34 columns

In [14]:
```
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.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

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

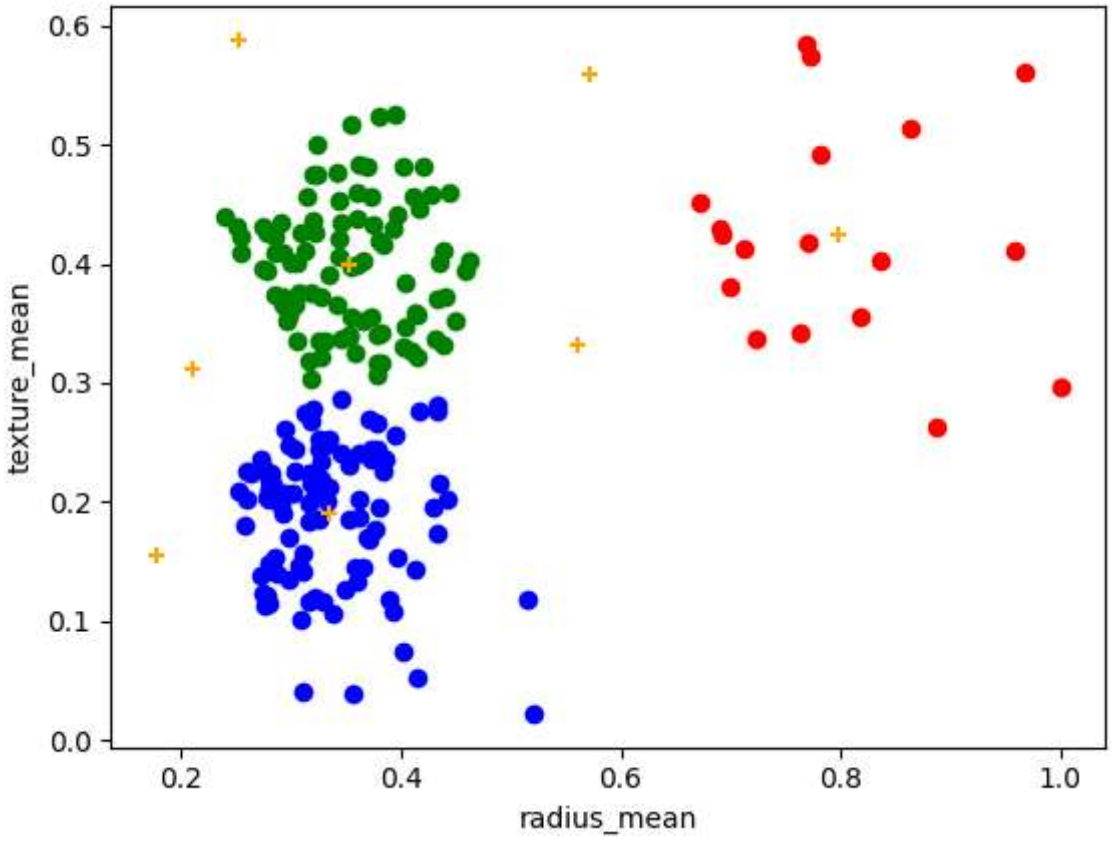


In [15]:
```
km.cluster_centers_
```

Out[15]:
```
array([[0.79840767, 0.42469846],
       [0.35266443, 0.39865016],
       [0.33489471, 0.19101622],
       [0.17694105, 0.15527139],
       [0.25223338, 0.58802181],
       [0.21091736, 0.31104487],
       [0.56101927, 0.3314624 ],
       [0.57132058, 0.55893025]])
```

In [16]:
```
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",m
plt.xlabel("radius_mean")
plt.ylabel("texture_mean")
```

Out[16]:
```
Text(0, 0.5, 'texture_mean')
```



In [17]:
```
k_rng=range(1,10)
sse=[]
```

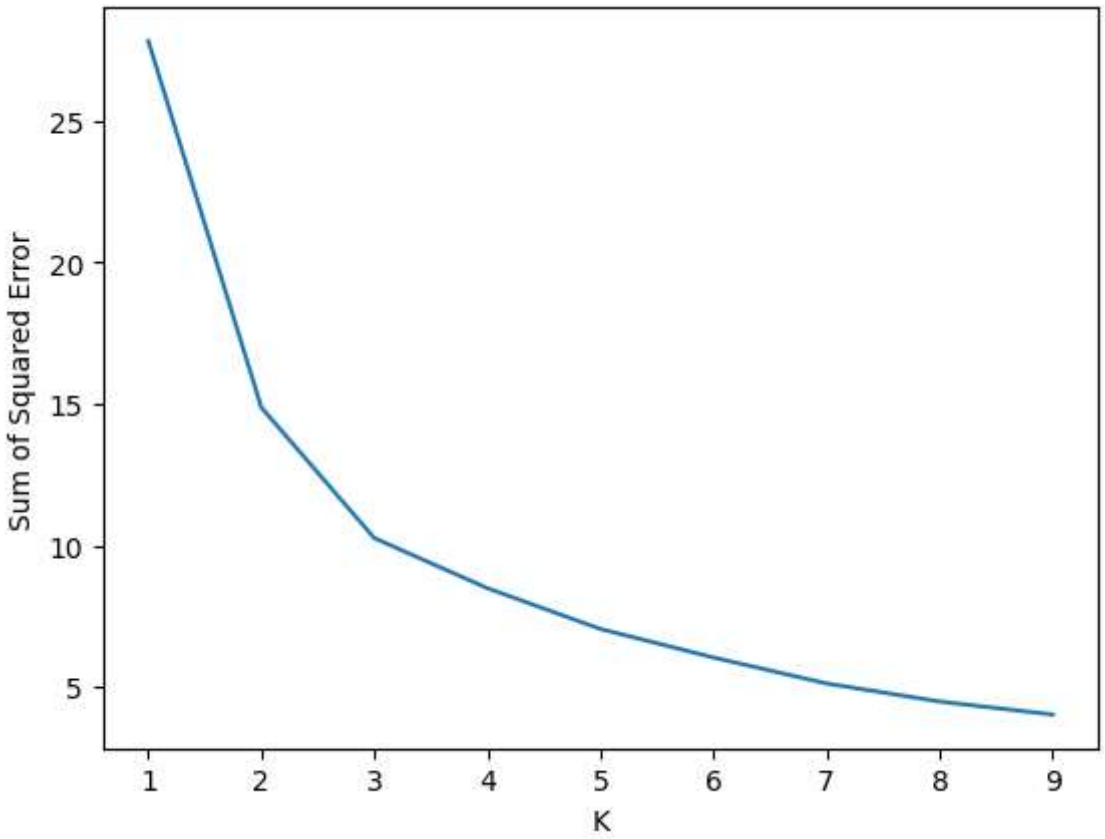
In [18]:
```
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["radius_mean","texture_mean"]])
    sse.append(km.inertia_)
#km.inertia_ will give you the value of sum of square error
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(
C:\Users\91903\AppData\Local\Programs\Python\Python310\lib\site-packages\skle
arn\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 supp
ress the warning
  warnings.warn(

[27.817507595043075, 14.872296449956036, 10.252751496105198, 8.48459193556418
8, 7.0434235336341064, 6.0360159475616415, 5.117629404113856, 4.4800639959361
47, 4.021641907033482]
```

```
C:\Users\91903\AppData\Local\Programs\Python\Python310\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(
```

Out[18]: Text(0, 0.5, 'Sum of Squared Error')



Conclusion :

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's why we will take it as a clustering and done with K-Means Clustering

In [ ]: