# PROBLEM STATEMENT:

```
                To perform an analytics report on 100 years of Rainfall data"
```

importing required libraries

In [1]:

```
import numpy
import matplotlib.pyplot as plt
import pygad
import pandas as pd
```

Data collection

In [2]:

```
df=pd.read_csv(r"C:\Users\MY HOME\Downloads\BreastCancerPrediction.csv")
df
```

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mea
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980
...	...	...	...	...	...	...	...	...	.
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.2439
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.1440
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.0925
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.3514
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.0000

569 rows × 33 columns

data cleaning

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

Out[3]:

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

5 rows × 33 columns

In [4]:
```
df.shape
```

Out[4]:
```
(569, 33)
```

In [5]:

```
df.describe()
```

Out[5]:

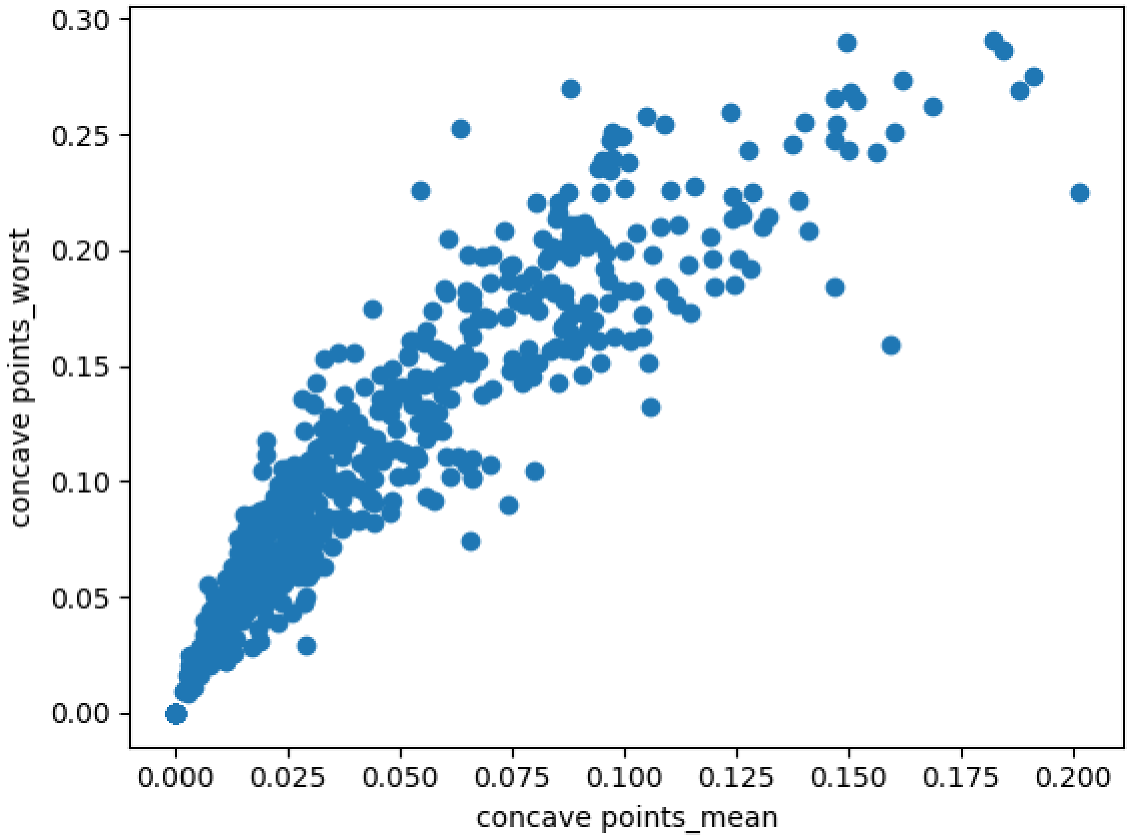
```
                 id  radius_mean  texture_mean  perimeter_mean   area_mean  smoothness_mean  compactness_mean  concavity_mean  p

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

8 rows × 32 columns
```

```
In [8]: plt.scatter(df["concave points_mean"],df["concave points_worst"])
        plt.xlabel("concave points_mean")
        plt.ylabel("concave points_worst")
```

```
Out[8]: Text(0, 0.5, 'concave points_worst')
```



In [9]:

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

Out[9]:

```
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 [10]:

```
y_predicted=km.fit_predict(df[["concave points_mean","concave points_worst"]])
y_predicted
```

```
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
```

Out[10]:

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

In [11]:

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

Out[11]:

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

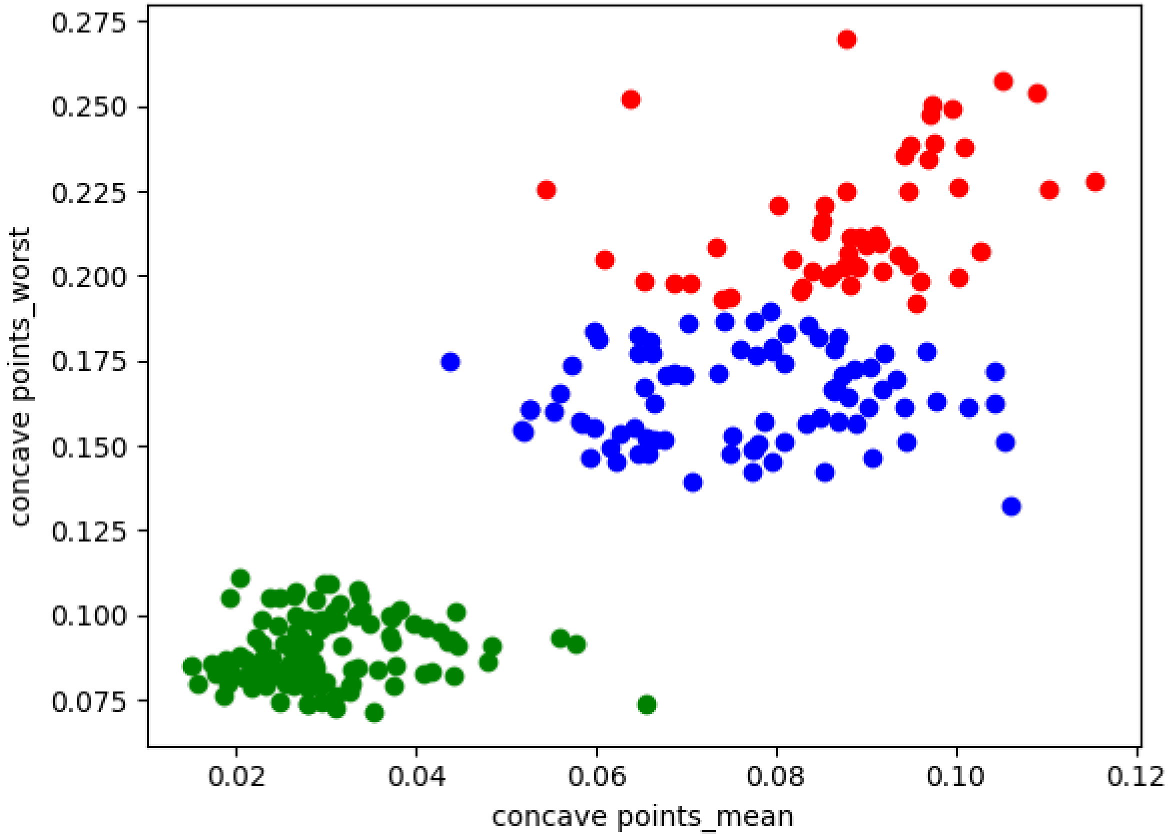
5 rows × 34 columns

In [12]:

```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["concave points_mean"],df1["concave points_worst"],color="red")
plt.scatter(df2["concave points_mean"],df2["concave points_worst"],color="green")
plt.scatter(df3["concave points_mean"],df3["concave points_worst"],color="blue")
plt.xlabel("concave points_mean")
plt.ylabel("concave points_worst")
```

Out[12]:

```
Text(0, 0.5, 'concave points_worst')
```



In [13]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
scaler.fit(df[["concave points_worst"]])
df["concave points_worst"]=scaler.transform(df[["concave points_worst"]])
df.head()
```

Out[13]:

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

5 rows × 34 columns

In [14]:

```
scaler.fit(df[["concave points_mean"]])
df["Age"]=scaler.transform(df[["concave points_mean"]])
df.head()
```

Out[14]:

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

5 rows × 35 columns

In [15]:

```
km=KMeans()
```

In [16]:

```
y_predicted=km.fit_predict(df[["concave points_mean","concave points_worst"]])
y_predicted
```

```
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
```

Out[16]:

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

In [17]:

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

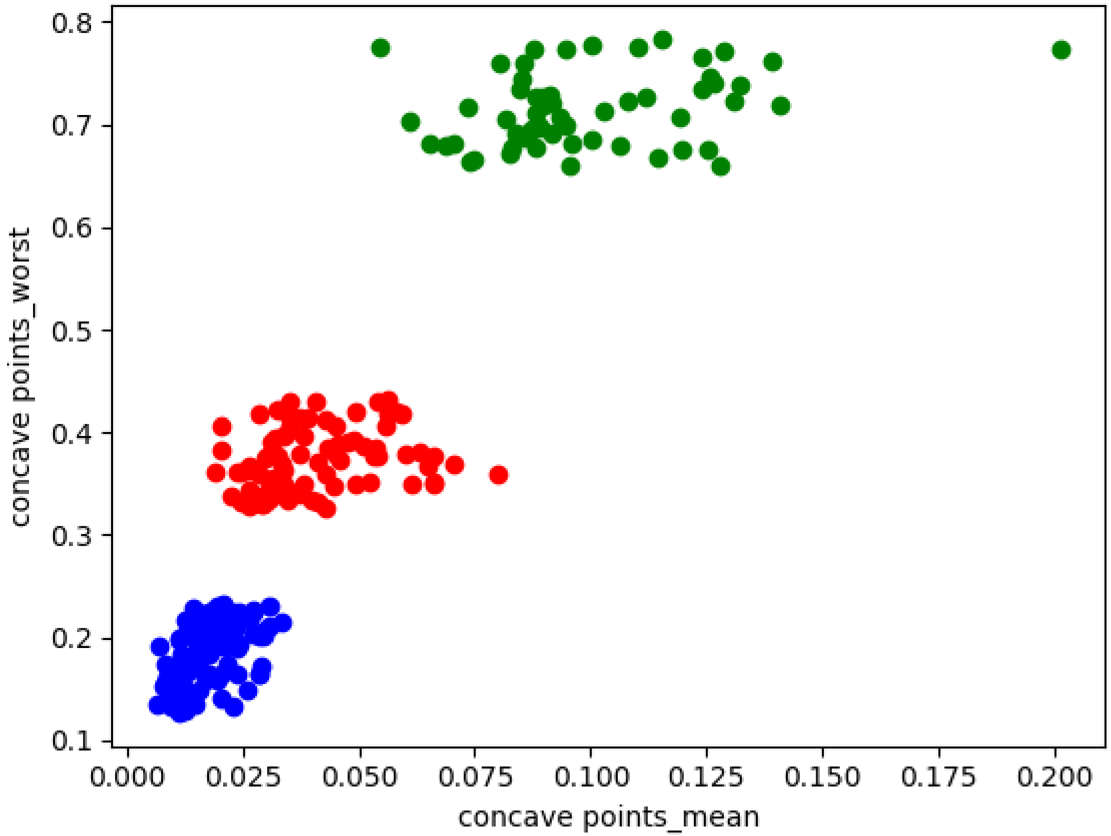
Out[17]:

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

5 rows × 36 columns

In [20]:
```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["concave points_mean"],df1["concave points_worst"],color="red")
plt.scatter(df2["concave points_mean"],df2["concave points_worst"],color="green")
plt.scatter(df3["concave points_mean"],df3["concave points_worst"],color="blue")
plt.xlabel("concave points_mean")
plt.ylabel("concave points_worst")
```

Out[20]:
```
Text(0, 0.5, 'concave points_worst')
```



In [21]:

```
km.cluster_centers_
```

Out[21]:

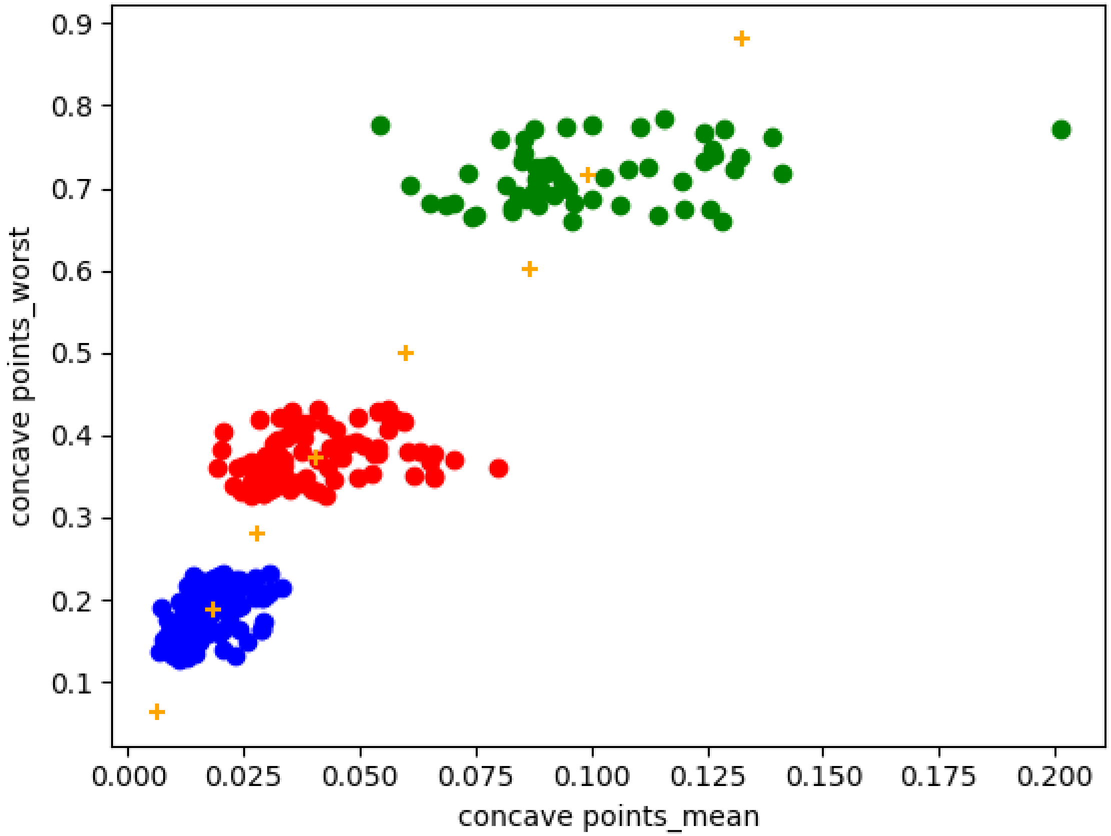
```
array([[0.04060904, 0.37325964],
       [0.09937034, 0.71544614],
       [0.01822572, 0.18850648],
       [0.02782514, 0.27945112],
       [0.05999863, 0.49852187],
       [0.13261516, 0.88198648],
       [0.00627369, 0.06437132],
       [0.08671579, 0.60248387]])
```

In [22]:

```
df1=df[df["New Cluster"]==0]
df2=df[df["New Cluster"]==1]
df3=df[df["New Cluster"]==2]
plt.scatter(df1["concave points_mean"],df1["concave points_worst"],color="red")
plt.scatter(df2["concave points_mean"],df2["concave points_worst"],color="green")
plt.scatter(df3["concave points_mean"],df3["concave points_worst"],color="blue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="orange",marker="+")
plt.xlabel("concave points_mean")
plt.ylabel("concave points_worst")
```

Out[22]:

```
Text(0, 0.5, 'concave points_worst')
```



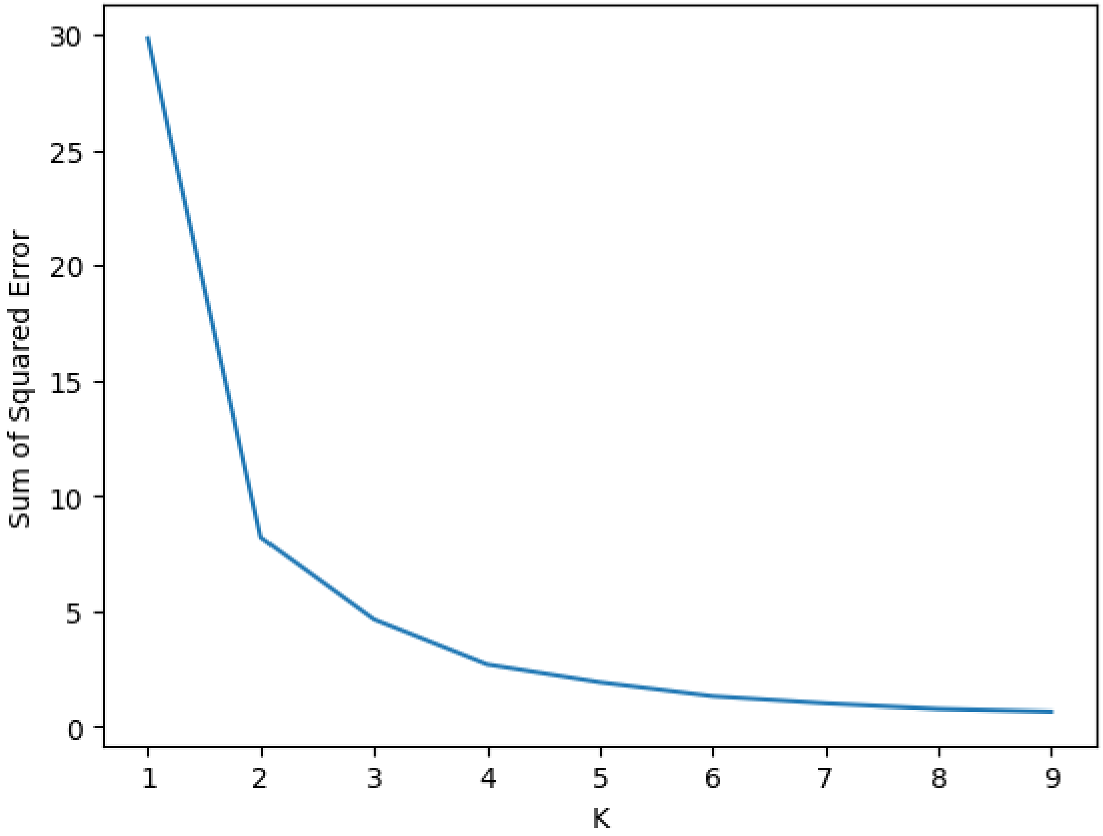
In [23]:

```
k_rng=range(1,10)
sse=[]
for k in k_rng:
 km=KMeans(n_clusters=k)
 km.fit(df[["concave points_mean","concave points_worst"]])
 sse.append(km.inertia_) #km.inertia_ will give you the value of sum of squa
print(sse)
plt.plot(k_rng,sse)
plt.xlabel("K")
plt.ylabel("Sum of Squared Error")
```

```
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(
C:\Users\MY HOME\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:870: Futu
reWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explic
itly to suppress the warning
  warnings.warn(

[29.83669529147959, 8.192771108043772, 4.632302769625648, 2.682394795326975, 1.902932932070331, 1.2992164832655
178, 0.9926335747828674, 0.7560326485435538, 0.6391630303515827]
```

```
Out[23]: Text(0, 0.5, 'Sum of Squared Error')
```



# CONCLUSION:

---------->for the given dataset we perform kmeans algorithm and we have divided the dataset into several clusters.