Project on Breast Cancer prediction

In [33]:

```
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
df=pd.read_csv(r"C:\Users\RAMADEVI SURIPAKA\Downloads\BreastCancerPrediction.csv")
df.head()
```

Out[2]:

	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 [3]:

```
df.tail()
```

Out[3]:

	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 [4]:

```
df.describe()
```

Out[4]:

worst	perimeter_worst	area_worst	smoothness_worst	compactness_worst	concavity_worst
)0000	569.000000	569.000000	569.000000	569.000000	569.000000
77223	107.261213	880.583128	0.132369	0.254265	0.272188
46258	33.602542	569.356993	0.022832	0.157336	0.208624
20000	50.410000	185.200000	0.071170	0.027290	0.000000
30000	84.110000	515.300000	0.116600	0.147200	0.114500
10000	97.660000	686.500000	0.131300	0.211900	0.226700
20000	125.400000	1084.000000	0.146000	0.339100	0.382900
40000	251.200000	4254.000000	0.222600	1.058000	1.252000

In [5]:

```
df.tail()
```

Out[5]:

	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 [6]:

```
df.shape
```

Out[6]:

```
(569, 33)
```

In [7]:

```
df.columns
```

Out[7]:

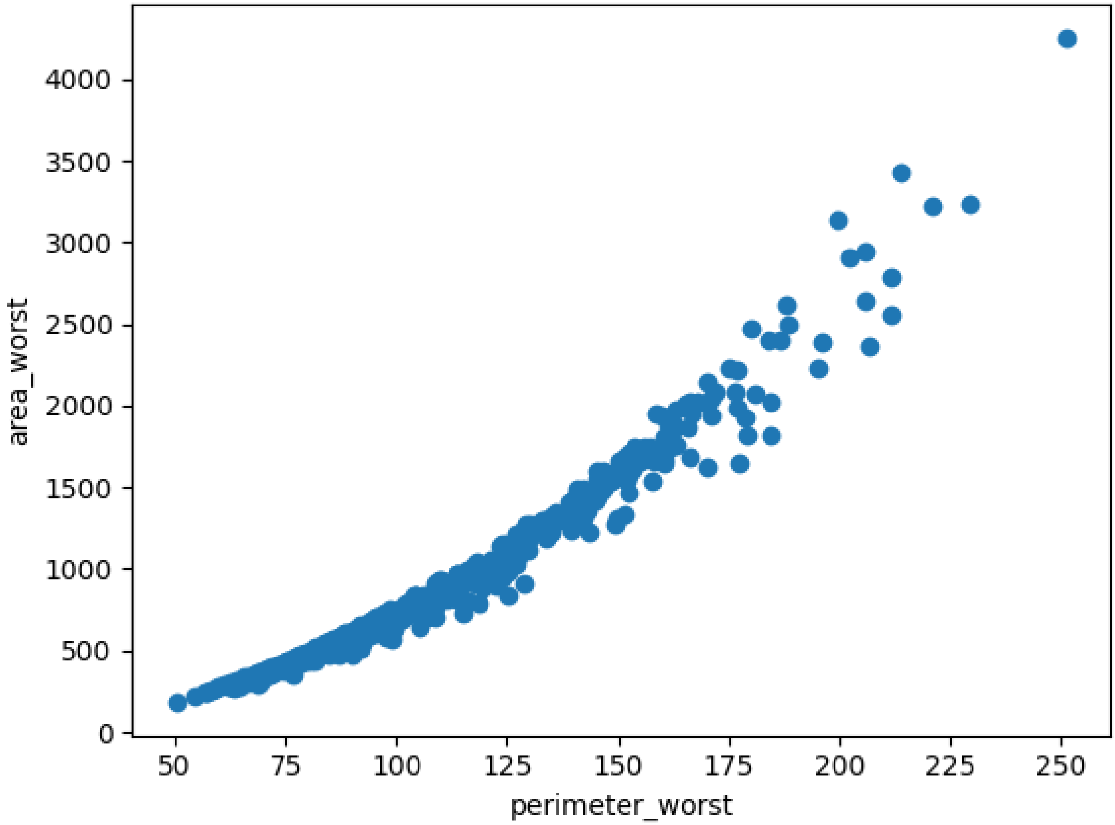
```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mea
n',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_s
e',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_s
e',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

In [8]:

```
plt.scatter(df["perimeter_worst"],df["area_worst"])
plt.xlabel("perimeter_worst")
plt.ylabel("area_worst")
```

Out[8]:

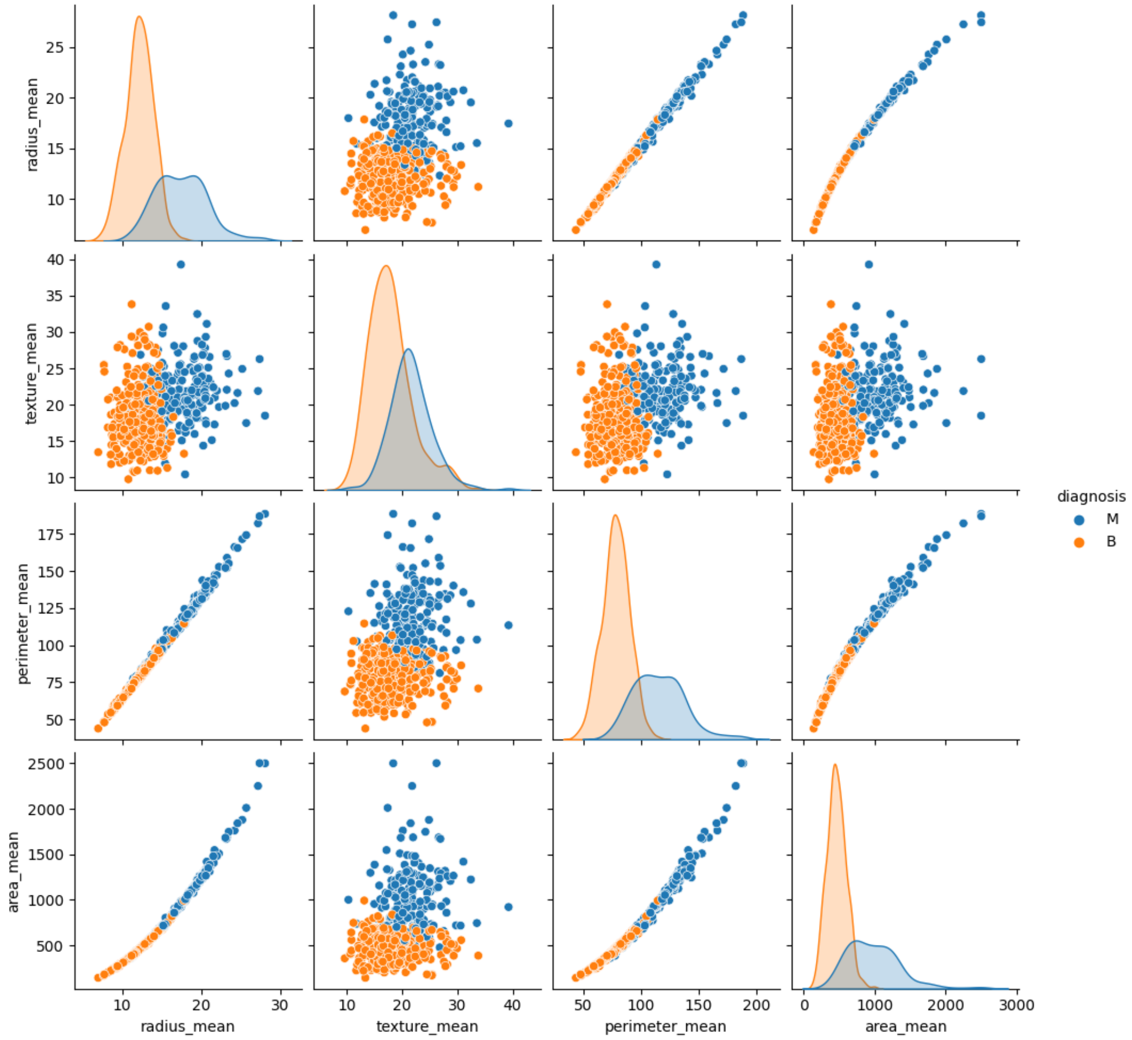
```
Text(0, 0.5, 'area_worst')
```



In [35]:

```
cols = ["diagnosis", "radius_mean", "texture_mean", "perimeter_mean", "area_mean"]

sns.pairplot(df[cols], hue="diagnosis")
plt.show()
```



In [ ]:

```

```

In [9]:

```
from sklearn.cluster import KMeans
```

In [10]:

```
km=KMeans()
km
```

Out[10]:

```
▾ KMeans
KMeans()
```

In [11]:

```
y_predicted=km.fit_predict(df[["perimeter_worst","area_worst"]])
y_predicted
```

```
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
```

Out[11]:

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

In [12]:

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

Out[12]:

	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	

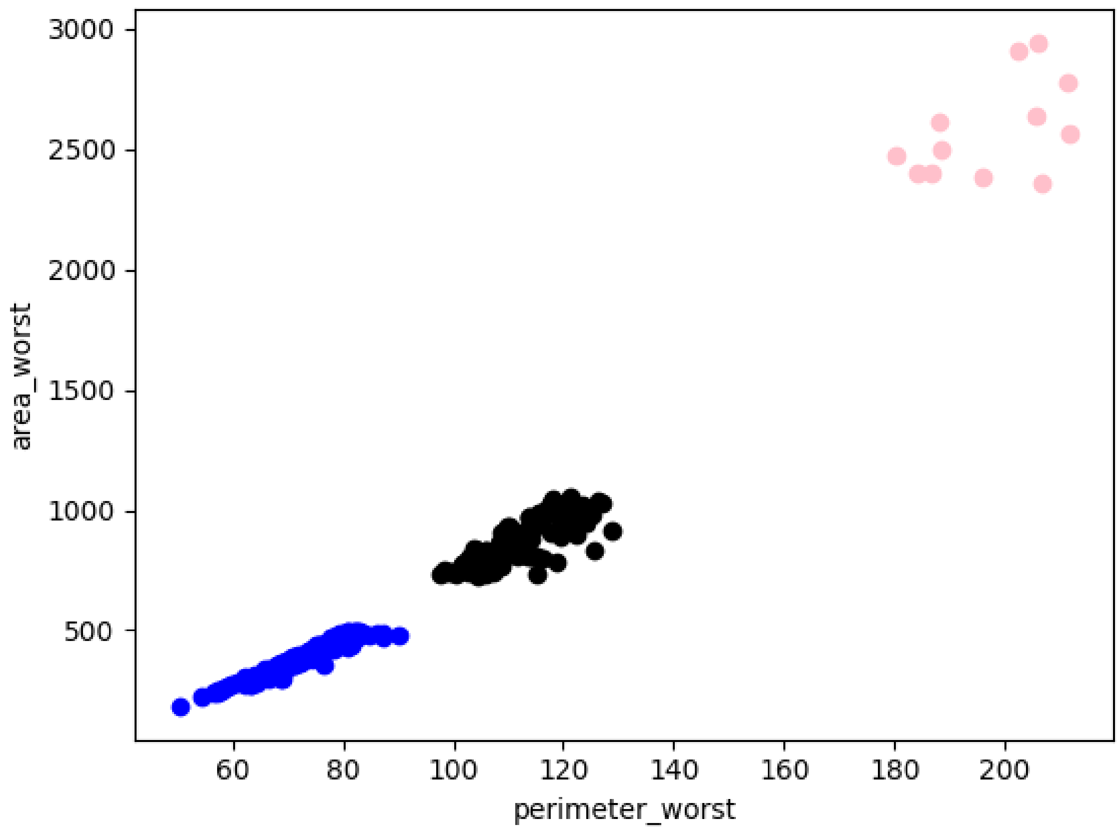
5 rows × 34 columns

In [16]:

```
df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["perimeter_worst"],df1["area_worst"],color="black")
plt.scatter(df2["perimeter_worst"],df2["area_worst"],color="pink")
plt.scatter(df3["perimeter_worst"],df3["area_worst"],color="blue")
plt.xlabel("perimeter_worst")
plt.ylabel("area_worst")
```

Out[16]:

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



In [17]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [18]:

```
Scaler=MinMaxScaler()
```

In [20]:

```
Scaler.fit(df[["perimeter_worst"]])
df["perimeter_worst"]=Scaler.transform(df[["perimeter_worst"]])
df.head()
```

Out[20]:

	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 [21]:

```
Scaler.fit(df[["area_worst"]])
df["area_worst"]=Scaler.transform(df[["area_worst"]])
df.head()
```

Out[21]:

	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 [22]:

```
km=KMeans()
km
```

Out[22]:

```
▾ KMeans
KMeans()
```

In [23]:

```
y_predicted=km.fit_predict(df[["perimeter_worst","area_worst"]])
y_predicted
```

```
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
```

Out[23]:

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

In [24]:

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

Out[24]:

	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	

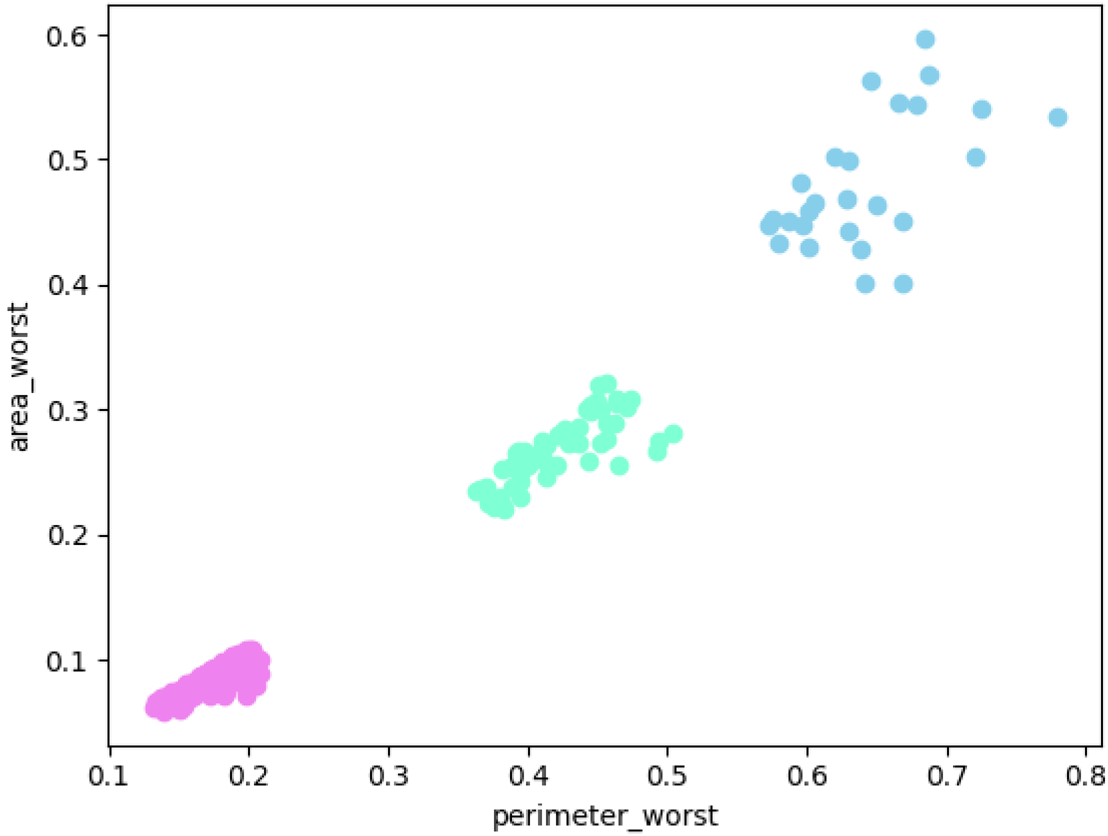
5 rows × 35 columns

In [25]:

```
df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["perimeter_worst"],df1["area_worst"],color="aquamarine")
plt.scatter(df2["perimeter_worst"],df2["area_worst"],color="violet")
plt.scatter(df3["perimeter_worst"],df3["area_worst"],color="skyblue")
plt.xlabel("perimeter_worst")
plt.ylabel("area_worst")
```

Out[25]:

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



In [26]:

```
km.cluster_centers_
```

Out[26]:

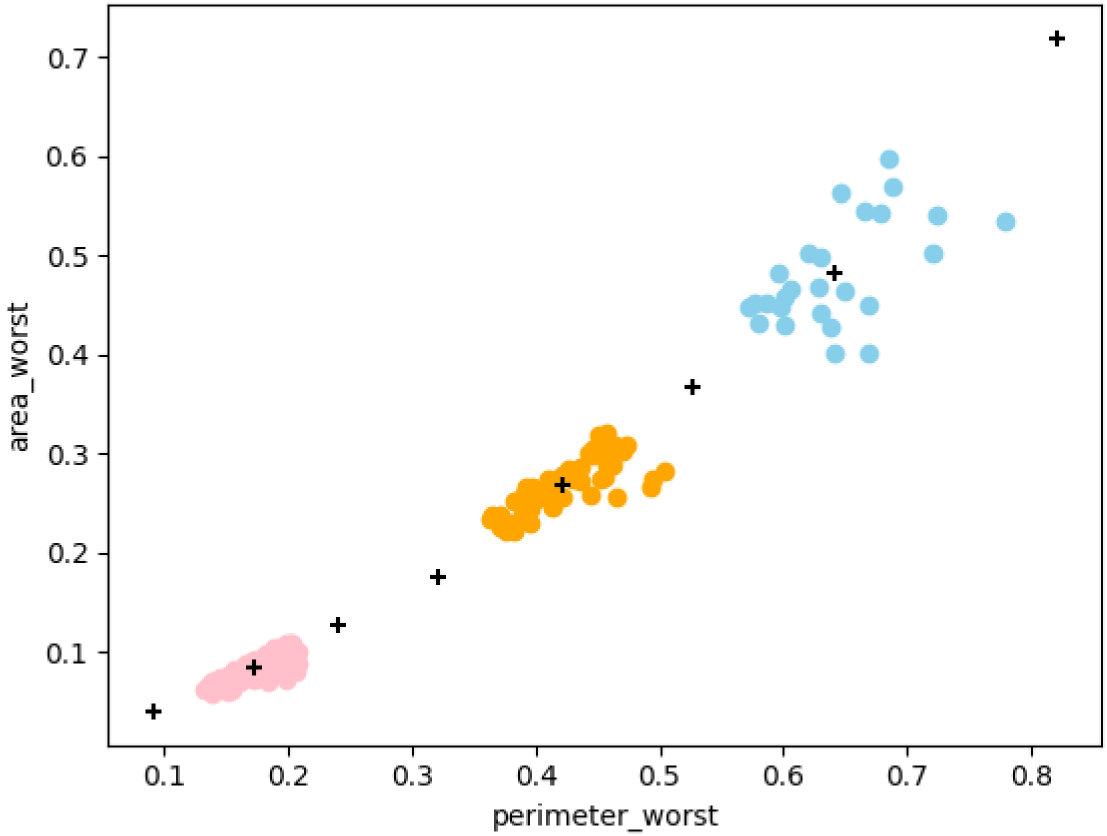
```
array([[0.4209081 , 0.26761004],
       [0.17196876, 0.08365685],
       [0.64149303, 0.48170317],
       [0.09096722, 0.04022389],
       [0.82075801, 0.71920468],
       [0.32102406, 0.17555771],
       [0.52618788, 0.36784751],
       [0.2409109 , 0.12571867]])
```

In [27]:

```
df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["perimeter_worst"],df1["area_worst"],color="orange")
plt.scatter(df2["perimeter_worst"],df2["area_worst"],color="pink")
plt.scatter(df3["perimeter_worst"],df3["area_worst"],color="skyblue")
plt.scatter(km.cluster_centers_[:,0],km.cluster_centers_[:,1],color="black",marker="+")
plt.xlabel("perimeter_worst")
plt.ylabel("area_worst")
```

Out[27]:

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



In [28]:

```
k_rng=range(1,10)
sse=[]
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["perimeter_worst","area_worst"]])
    sse.append(km.inertia_)
    sse
```

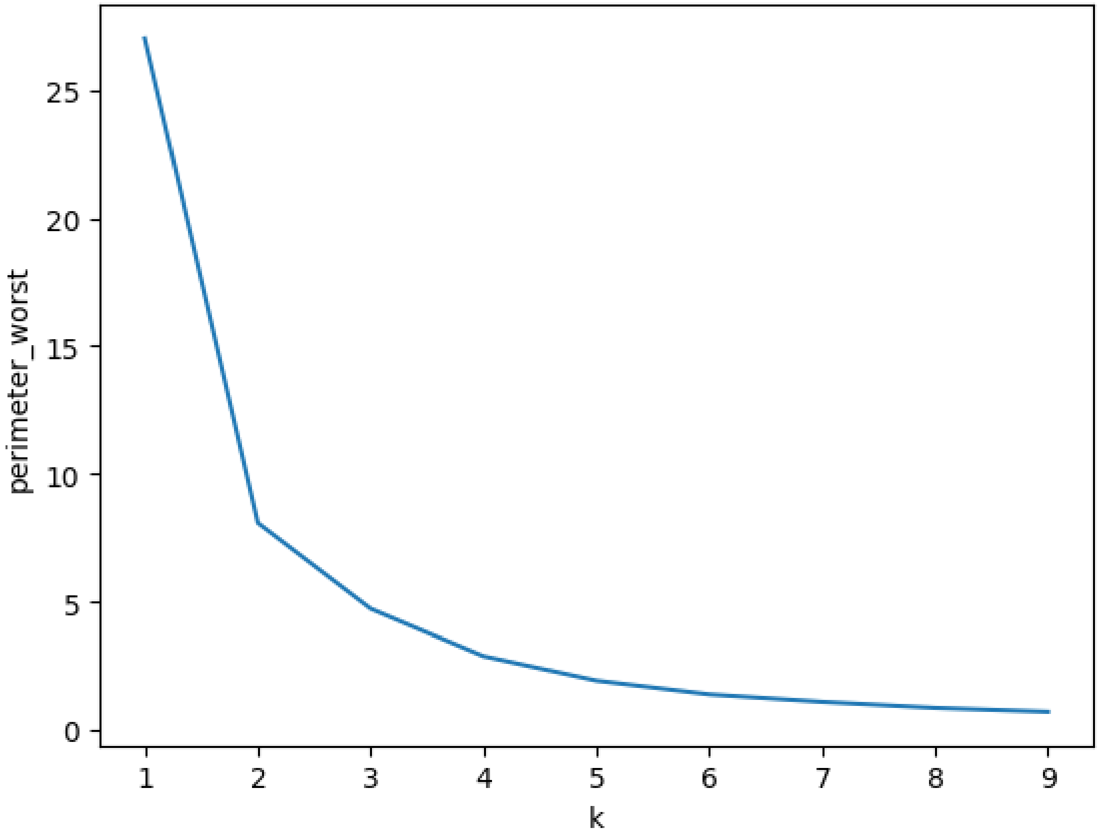
```
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
C:\Users\RAMADEVI SURIPAKA\AppData\Local\Programs\Python\Python310\lib\sit
e-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default valu
e of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_in
it` explicitly to suppress the warning
  warnings.warn(
```

In [30]:

```
plt.plot(k_rng,sse)
plt.xlabel("k")
plt.ylabel("perimeter_worst")
```

Out[30]:

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



Conclusion:

In this project, We had seen the data clearning and EDA using pandas methods and show some visual graphs to know the behaviour of this dataset and finnaly we train some model for it . I have wroted some basic codes in this notebook. So, After successfully completed we can deploye our models to the live production mode using exporting models,based on the above data the groups devided into cluster.

In [ ]: