

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

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

Out[2]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_
0	842302	M	17.99	10.38	122.80	1001.0	0.
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
...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.
565	926682	M	20.13	28.25	131.20	1261.0	0.0
566	926954	M	16.60	28.08	108.30	858.1	0.0
567	927241	M	20.60	29.33	140.10	1265.0	0.
568	92751	B	7.76	24.54	47.92	181.0	0.0

569 rows × 33 columns



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

Out[3]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 33 columns



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

Out[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
564	926424	M	21.56	22.39	142.00	1479.0	0.11
565	926682	M	20.13	28.25	131.20	1261.0	0.097
566	926954	M	16.60	28.08	108.30	858.1	0.084
567	927241	M	20.60	29.33	140.10	1265.0	0.117
568	92751	B	7.76	24.54	47.92	181.0	0.052

5 rows × 33 columns



In [5]: `df.shape`

Out[5]: (569, 33)

In [6]: `df.describe()`

Out[6]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mea
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.09636
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.01406
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.05263
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.08637
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.09587
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.10530
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.16340

8 rows × 32 columns



In [7]: `df.columns`

Out[7]: Index(['id', 'diagnosis', 'radius\_mean', 'texture\_mean', 'perimeter\_mean', 'area\_mean', 'smoothness\_mean', 'compactness\_mean', 'concavity\_mean', 'concave points\_mean', 'symmetry\_mean', 'fractal\_dimension\_mean', 'radius\_se', 'texture\_se', 'perimeter\_se', 'area\_se', 'smoothness\_se', 'compactness\_se', 'concavity\_se', 'concave points\_se', 'symmetry\_se', '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')

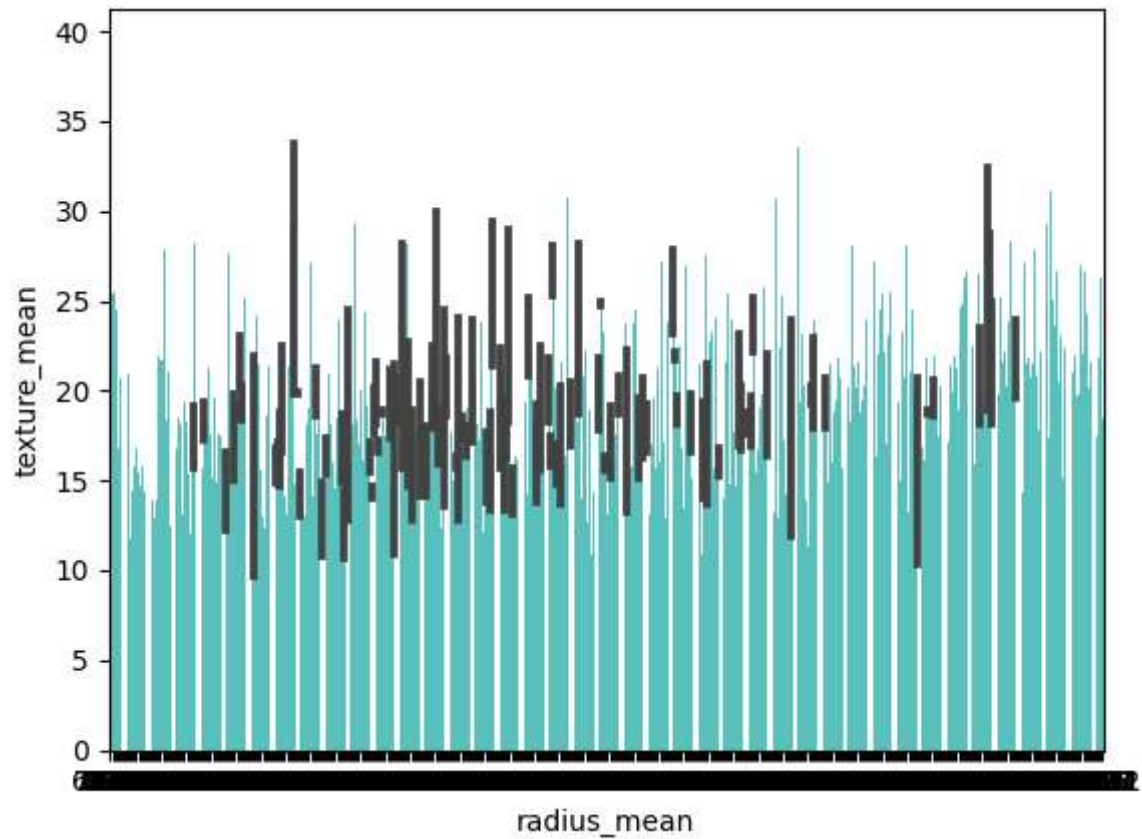
In [8]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   id                                    569 non-null    int64
1   diagnosis                            569 non-null    object
2   radius_mean                          569 non-null    float64
3   texture_mean                         569 non-null    float64
4   perimeter_mean                      569 non-null    float64
5   area_mean                           569 non-null    float64
6   smoothness_mean                     569 non-null    float64
7   compactness_mean                    569 non-null    float64
8   concavity_mean                      569 non-null    float64
9   concave points_mean                 569 non-null    float64
10  symmetry_mean                       569 non-null    float64
11  fractal_dimension_mean              569 non-null    float64
12  radius_se                           569 non-null    float64
13  texture_se                          569 non-null    float64
14  perimeter_se                        569 non-null    float64
15  area_se                             569 non-null    float64
16  smoothness_se                       569 non-null    float64
17  compactness_se                      569 non-null    float64
18  concavity_se                        569 non-null    float64
19  concave points_se                   569 non-null    float64
20  symmetry_se                         569 non-null    float64
21  fractal_dimension_se                569 non-null    float64
22  radius_worst                       569 non-null    float64
23  texture_worst                       569 non-null    float64
24  perimeter_worst                     569 non-null    float64
25  area_worst                          569 non-null    float64
26  smoothness_worst                   569 non-null    float64
27  compactness_worst                   569 non-null    float64
28  concavity_worst                     569 non-null    float64
29  concave points_worst                569 non-null    float64
30  symmetry_worst                      569 non-null    float64
31  fractal_dimension_worst             569 non-null    float64
32  Unnamed: 32                         0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

```
In [9]: df.isnull().sum()
```

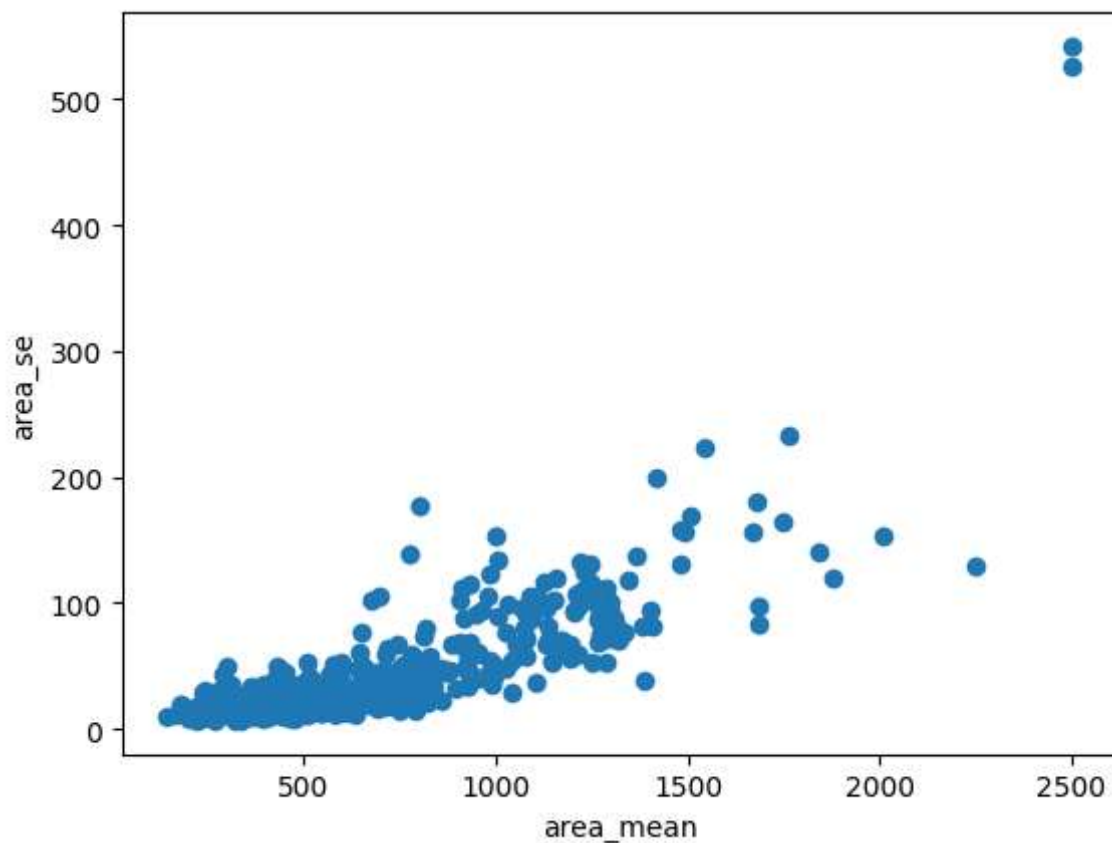
```
Out[9]: id                                0
        diagnosis                         0
        radius_mean                       0
        texture_mean                      0
        perimeter_mean                   0
        area_mean                        0
        smoothness_mean                  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 [10]: sns.barplot(x='radius_mean', y='texture_mean', data=df, color="mediumturquoise")  
plt.show()
```



```
In [11]: plt.scatter(df["area_mean"],df["area_se"])  
plt.xlabel("area_mean")  
plt.ylabel("area_se")
```

```
Out[11]: Text(0, 0.5, 'area_se')
```



```
In [12]: from sklearn.cluster import KMeans
```

```
In [13]: km=KMeans()  
km
```

```
Out[13]: 

▼ KMeans  
KMeans()


```

```
In [14]: y_predicted=km.fit_predict(df[["symmetry_se", "symmetry_worst"]])
y_predicted
```

C:\Users\DELL\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(

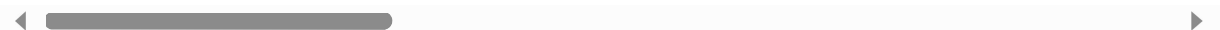
```
Out[14]: array([7, 0, 5, 1, 3, 4, 2, 2, 4, 4, 2, 5, 2, 0, 5, 4, 2, 5, 0, 2, 2, 3,
 7, 0, 5, 4, 4, 3, 4, 0, 5, 7, 5, 5, 4, 7, 5, 6, 6, 0, 2, 2, 7, 5,
 5, 5, 2, 4, 0, 0, 3, 3, 0, 2, 0, 2, 5, 5, 3, 2, 5, 2, 0, 2, 5, 2,
 0, 3, 4, 3, 3, 3, 2, 0, 0, 0, 0, 5, 1, 0, 0, 5, 3, 3, 5, 5, 2, 4,
 2, 2, 3, 3, 6, 0, 0, 5, 3, 6, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 4, 0,
 3, 3, 3, 3, 0, 3, 6, 5, 2, 7, 2, 0, 2, 0, 6, 3, 5, 0, 0, 2, 5, 0,
 5, 0, 2, 0, 6, 0, 5, 6, 2, 0, 0, 5, 3, 0, 1, 0, 0, 3, 2, 2, 2, 0,
 4, 2, 3, 3, 3, 0, 2, 3, 5, 3, 5, 0, 6, 0, 3, 3, 0, 0, 2, 6, 0, 0,
 0, 2, 3, 6, 0, 4, 5, 6, 2, 0, 2, 0, 2, 3, 7, 3, 6, 2, 2, 2, 2, 3,
 2, 7, 0, 0, 2, 4, 2, 5, 2, 2, 4, 3, 0, 3, 6, 6, 7, 5, 5, 0, 2, 0,
 3, 2, 2, 4, 3, 2, 0, 2, 0, 5, 2, 0, 0, 3, 0, 3, 2, 3, 6, 0, 3, 0,
 2, 0, 0, 0, 0, 0, 5, 0, 2, 0, 0, 2, 5, 2, 0, 2, 2, 5, 2, 3, 2, 0,
 2, 0, 2, 3, 5, 0, 3, 0, 0, 0, 3, 3, 0, 3, 3, 2, 2, 2, 5, 0, 6, 3,
 3, 3, 2, 2, 3, 2, 5, 2, 3, 0, 6, 6, 0, 3, 2, 0, 2, 3, 3, 2, 0, 2,
 3, 6, 5, 3, 3, 5, 2, 6, 3, 0, 2, 6, 0, 2, 3, 1, 0, 0, 6, 3, 2, 0,
 2, 5, 2, 0, 3, 0, 3, 5, 0, 0, 2, 2, 2, 4, 0, 3, 0, 2, 0, 2, 0, 4,
 5, 0, 0, 6, 2, 3, 3, 3, 3, 0, 0, 3, 0, 0, 2, 2, 3, 0, 7, 3, 0, 0,
 2, 2, 0, 0, 5, 4, 2, 2, 3, 0, 0, 3, 0, 3, 3, 0, 0, 3, 2, 5, 2, 3,
 0, 6, 0, 0, 2, 3, 2, 2, 3, 3, 0, 3, 2, 5, 2, 2, 3, 2, 2, 2, 0, 2,
 0, 2, 2, 0, 0, 0, 2, 3, 2, 2, 3, 3, 0, 3, 2, 2, 3, 2, 2, 0, 3, 6,
 3, 3, 6, 0, 0, 0, 2, 5, 3, 3, 3, 3, 3, 0, 2, 3, 0, 3, 3, 3, 2, 0,
 3, 2, 3, 0, 3, 2, 3, 0, 2, 3, 0, 3, 0, 2, 3, 0, 2, 2, 3, 3, 2, 3,
 3, 2, 3, 2, 0, 7, 2, 6, 2, 3, 0, 3, 2, 2, 3, 3, 3, 5, 2, 0, 2, 0,
 0, 0, 3, 2, 0, 3, 5, 0, 3, 2, 2, 3, 0, 2, 5, 2, 3, 0, 3, 2, 5, 2,
 3, 0, 3, 2, 0, 2, 3, 2, 0, 0, 2, 0, 3, 2, 0, 3, 3, 0, 0, 0, 3, 2,
 3, 2, 3, 3, 3, 3, 3, 3, 3, 6, 3, 6, 4, 0, 6, 0, 3, 4, 0])
```

```
In [15]: df["cluster"]=y_predicted
df.head()
```

```
Out[15]:
```

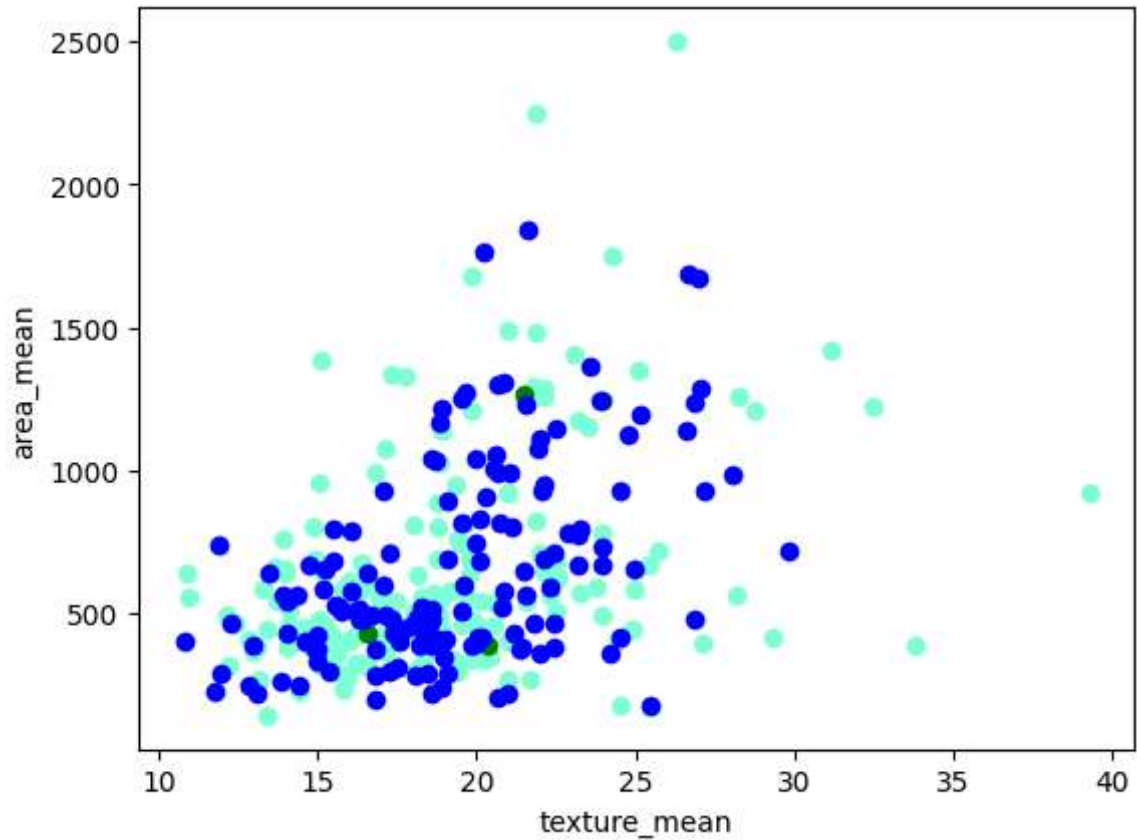
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 34 columns



```
In [25]: df1=df[df.cluster==0]
df2=df[df.cluster==1]
df3=df[df.cluster==2]
plt.scatter(df1["texture_mean"],df1["area_mean"],color="aquamarine")
plt.scatter(df2["texture_mean"],df2["area_mean"],color="green")
plt.scatter(df3["texture_mean"],df3["area_mean"],color="blue")
plt.xlabel("texture_mean")
plt.ylabel("area_mean")
```

Out[25]: Text(0, 0.5, 'area\_mean')



```
In [17]: from sklearn.preprocessing import MinMaxScaler
```

```
In [18]: scaler=MinMaxScaler()
```



```
In [19]: scaler.fit(df[["concavity_se"]])
df["concavity_se"]=scaler.transform(df[["concavity_se"]])
df.head()
```

Out[19]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 34 columns

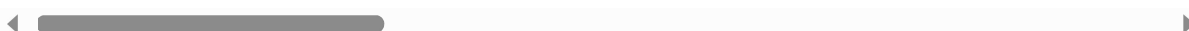


```
In [20]: scaler.fit(df[["compactness_mean"]])
df["compactness_mean"]=scaler.transform(df[["compactness_mean"]])
df.head()
```

Out[20]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 34 columns



```
In [21]: km=KMeans()
```

```
In [22]: y_predicted=km.fit_predict(df[["radius_mean","area_mean"]])
y_predicted
```

C:\Users\DELL\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(

```
Out[22]: array([7, 3, 3, 1, 3, 1, 7, 4, 4, 1, 0, 0, 3, 0, 4, 0, 0, 0, 3, 4, 4, 5,
 0, 3, 7, 7, 0, 7, 0, 7, 7, 1, 7, 3, 0, 7, 4, 4, 0, 4, 4, 1, 7, 4,
 4, 7, 5, 4, 1, 4, 1, 4, 1, 7, 0, 1, 3, 0, 4, 5, 5, 5, 0, 5, 4, 0,
 5, 1, 5, 4, 3, 5, 7, 4, 1, 0, 4, 7, 3, 4, 1, 4, 2, 3, 1, 7, 0, 7,
 1, 0, 0, 0, 4, 4, 0, 3, 1, 5, 1, 0, 4, 5, 1, 5, 5, 4, 1, 1, 2, 1,
 5, 1, 4, 5, 5, 1, 5, 0, 0, 7, 1, 7, 2, 0, 4, 4, 4, 3, 0, 3, 1, 0,
 0, 0, 7, 4, 1, 1, 0, 1, 5, 0, 1, 4, 1, 1, 1, 0, 0, 4, 4, 5, 5, 1,
 4, 1, 7, 7, 1, 1, 1, 3, 3, 1, 2, 0, 1, 7, 7, 0, 1, 4, 0, 1, 5, 5,
 5, 0, 4, 4, 6, 3, 0, 1, 0, 5, 7, 1, 1, 1, 4, 4, 5, 1, 0, 4, 4, 7,
 3, 0, 1, 7, 2, 4, 1, 0, 5, 7, 4, 0, 3, 1, 6, 7, 4, 4, 1, 5, 3, 3,
 4, 4, 5, 0, 4, 0, 5, 0, 4, 4, 7, 1, 1, 3, 5, 4, 2, 3, 4, 7, 4, 1,
 1, 4, 3, 5, 4, 4, 5, 1, 3, 1, 3, 7, 3, 4, 3, 0, 0, 0, 3, 7, 7, 0,
 7, 3, 5, 4, 4, 5, 4, 1, 2, 5, 7, 1, 1, 7, 4, 4, 3, 1, 3, 0, 4, 4,
 1, 4, 1, 1, 0, 0, 4, 1, 4, 4, 1, 1, 4, 5, 3, 1, 3, 5, 1, 1, 4, 5,
 4, 4, 1, 0, 4, 1, 5, 1, 1, 7, 5, 1, 5, 3, 4, 3, 1, 4, 4, 1, 0, 0,
 0, 4, 1, 1, 1, 7, 4, 7, 5, 2, 0, 5, 1, 3, 1, 5, 1, 0, 1, 1, 1, 0,
 2, 0, 1, 1, 4, 4, 5, 5, 4, 4, 4, 0, 4, 3, 3, 1, 2, 2, 0, 0, 3, 3,
 4, 0, 5, 4, 4, 1, 1, 1, 1, 1, 4, 0, 1, 4, 1, 3, 5, 5, 0, 3, 1, 4,
 4, 4, 1, 1, 7, 1, 4, 4, 1, 1, 0, 4, 7, 1, 1, 1, 5, 0, 0, 1, 5, 0,
 4, 1, 1, 0, 1, 4, 5, 5, 5, 1, 1, 4, 0, 1, 3, 7, 0, 4, 4, 4, 4, 4,
 1, 7, 4, 5, 7, 1, 7, 0, 0, 3, 1, 3, 1, 0, 4, 4, 1, 4, 4, 5, 7, 6,
 0, 1, 4, 4, 4, 5, 7, 1, 5, 1, 0, 1, 1, 4, 4, 4, 1, 0, 1, 4, 4, 4,
 0, 1, 0, 3, 1, 7, 1, 7, 7, 1, 4, 0, 1, 1, 7, 3, 0, 4, 1, 2, 5, 5,
 1, 1, 0, 0, 1, 0, 4, 0, 0, 1, 7, 3, 4, 4, 5, 2, 1, 4, 5, 5, 4, 1,
 4, 1, 1, 1, 4, 3, 1, 3, 4, 1, 5, 5, 1, 0, 0, 4, 4, 4, 5, 5, 5, 1,
 1, 1, 4, 5, 4, 5, 5, 5, 0, 1, 4, 1, 0, 3, 2, 3, 7, 3, 5])
```

```
In [23]: df["New cluster"]=y_predicted
df.head()
```

```
Out[23]:
```

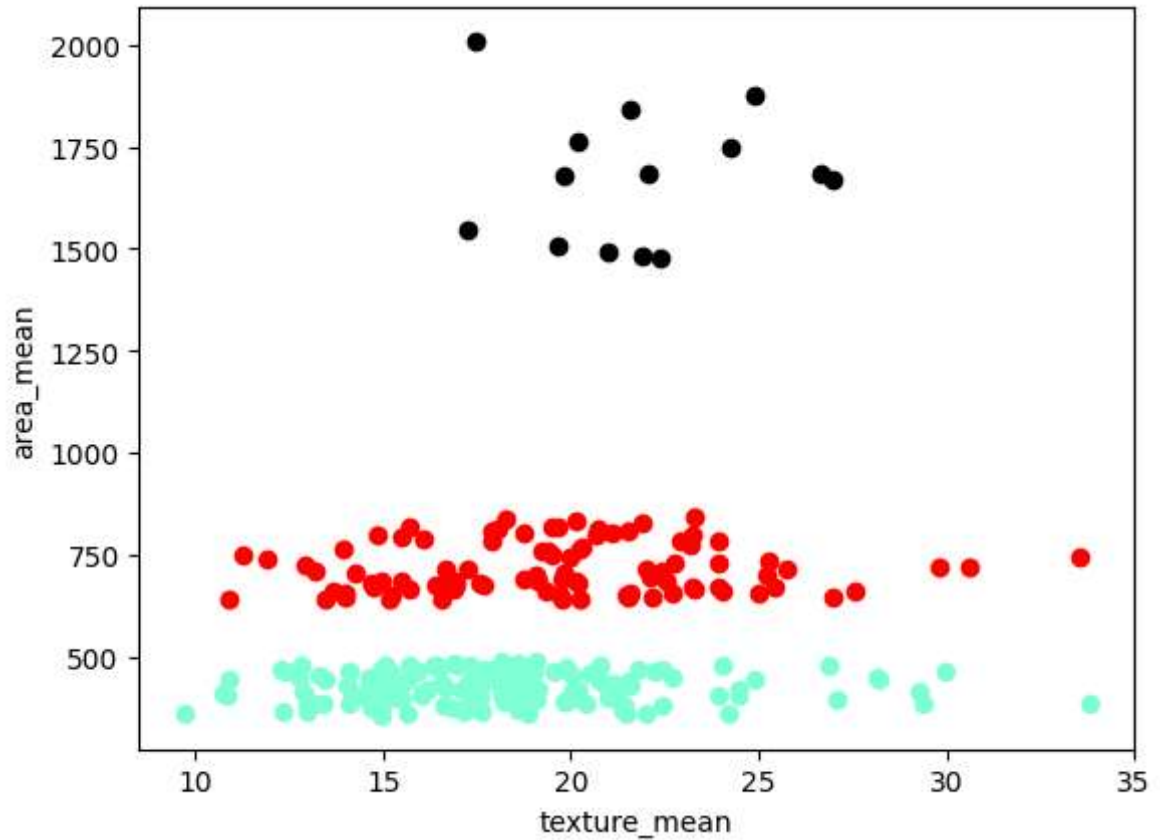
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_me
0	842302	M	17.99	10.38	122.80	1001.0	0.118
1	842517	M	20.57	17.77	132.90	1326.0	0.084
2	84300903	M	19.69	21.25	130.00	1203.0	0.109
3	84348301	M	11.42	20.38	77.58	386.1	0.142
4	84358402	M	20.29	14.34	135.10	1297.0	0.100

5 rows × 35 columns



```
In [27]: df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["texture_mean"],df1["area_mean"],color="red")
plt.scatter(df2["texture_mean"],df2["area_mean"],color="aquamarine")
plt.scatter(df3["texture_mean"],df3["area_mean"],color="black")
plt.xlabel("texture_mean")
plt.ylabel("area_mean")
```

Out[27]: Text(0, 0.5, 'area\_mean')

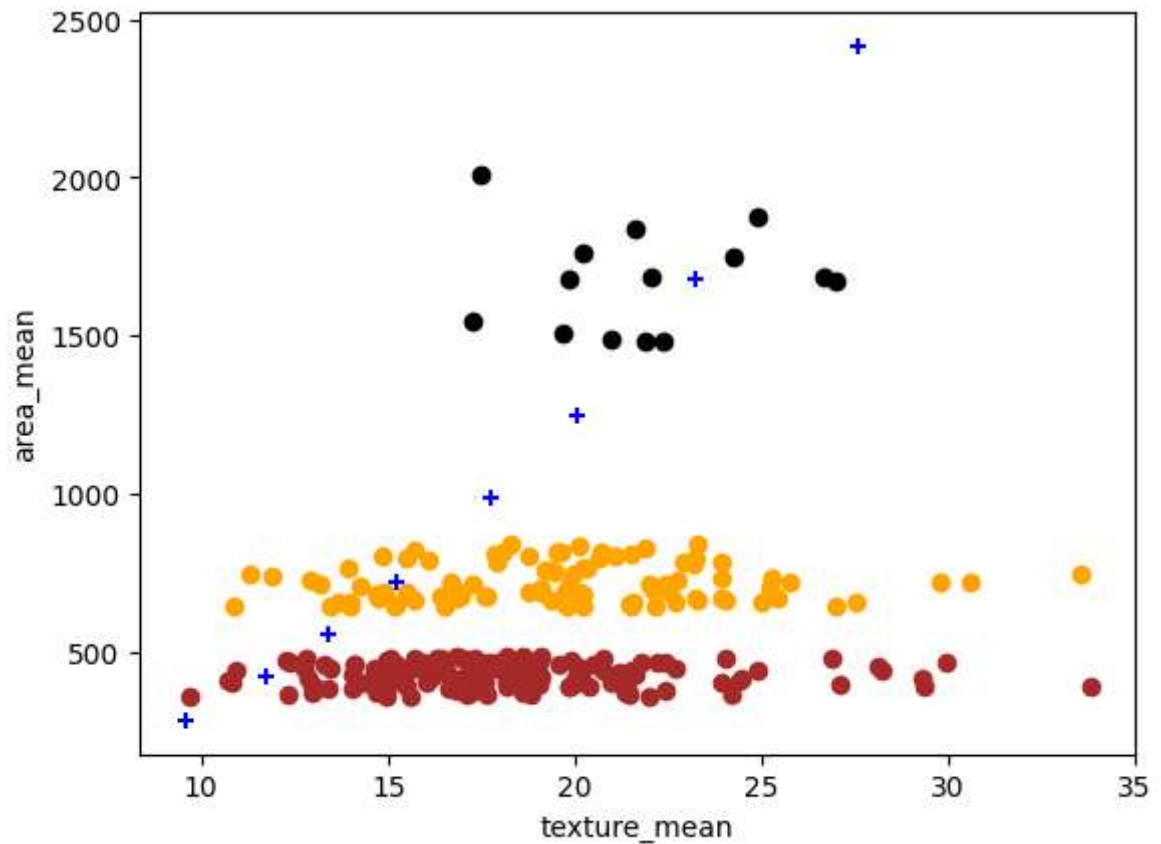


```
In [28]: km.cluster_centers_
```

```
Out[28]: array([[ 15.21344086,  716.75913978],
 [ 11.72890411,  422.90479452],
 [ 23.25        , 1676.21428571],
 [ 20.03944444, 1247.18518519],
 [ 13.37503759,  551.89849624],
 [  9.58368493,  281.46027397],
 [ 27.58333333, 2416.66666667],
 [ 17.7790566 ,  985.19056604]])
```

```
In [30]: df1=df[df["New cluster"]==0]
df2=df[df["New cluster"]==1]
df3=df[df["New cluster"]==2]
plt.scatter(df1["texture_mean"],df1["area_mean"],color="orange")
plt.scatter(df2["texture_mean"],df2["area_mean"],color="brown")
plt.scatter(df3["texture_mean"],df3["area_mean"],color="black")
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color="blue",marker="+")
plt.xlabel("texture_mean")
plt.ylabel("area_mean")
```

Out[30]: Text(0, 0.5, 'area\_mean')



```
In [32]: k_rng=range(1,10)
sse=[]
for k in k_rng:
    km=KMeans(n_clusters=k)
    km.fit(df[["texture_mean","area_mean"]])
    sse.append(km.inertia_)
sse
```

C:\Users\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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\DELL\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

```
In [33]: plt.plot(k_rng,sse)
plt.xlabel("k")
plt.ylabel("sum of squared Error")
```

```
Out[33]: Text(0, 0.5, 'sum of squared Error')
```

