

```

from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot as plt
%matplotlib inline

df = pd.read_csv("/content/house_price_full.csv")
df.head()

```

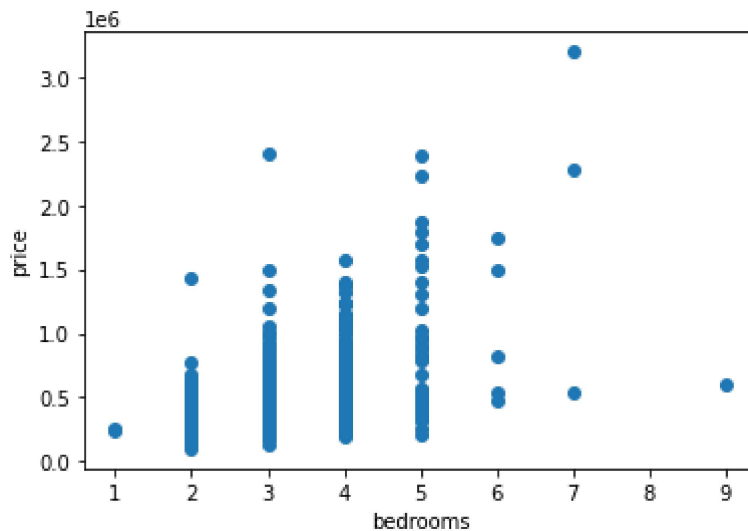
	bedrooms	sqft_living	price	
0	3	1340	313000	
1	5	3650	2384000	
2	3	1930	342000	
3	3	2000	420000	
4	4	1940	550000	

```

plt.scatter(df.bedrooms,df['price'])
plt.xlabel('bedrooms')
plt.ylabel('price')

```

☞ Text(0, 0.5, 'price')



```

km = KMeans(n_clusters=3)
y_predicted = km.fit_predict(df[['bedrooms','price']])
y_predicted

```

```

array([0, 2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1,

```

```
0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 2, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0,
1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 2, 0, 1, 0, 0, 0, 0, 0, 1, 0,
0, 2, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
1, 0, 1, 2, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 2, 0, 0,
1, 0, 0, 0, 1, 1, 0, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1,
0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 2, 2,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
0, 2, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 2, 1, 0, 2, 1, 0, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0, 0, 1,
0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
1, 1, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0,
1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 1, 0, 0, 1, 0, 0, 0, 1,
2, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 2, 1,
0, 0, 1, 1, 2, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0], dtype=int32)
```

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

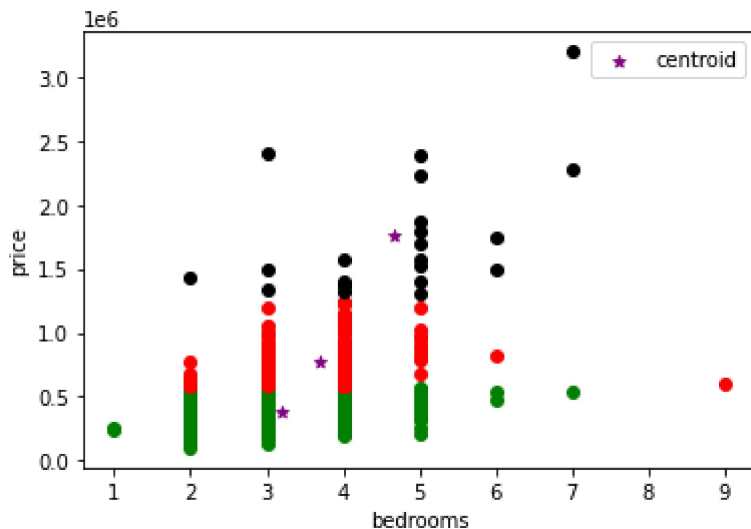
	bedrooms	sqft_living	price	cluster	
0	3	1340	313000	0	
1	5	3650	2384000	2	
2	3	1930	342000	0	
3	3	2000	420000	0	
4	4	1940	550000	0	

```
km.cluster_centers_
```

```
array([[3.20597015e+00, 3.82550343e+05],
       [3.70629371e+00, 7.68208503e+05],
       [4.66666667e+00, 1.75547086e+06]])
```

```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.bedrooms,df1['price'],color='green')
plt.scatter(df2.bedrooms,df2['price'],color='red')
plt.scatter(df3.bedrooms,df3['price'],color='black')
plt.scatter(km.cluster_centers_[0,0],km.cluster_centers_[0,1],color='purple',marker='*',label=0)
plt.xlabel('bedrooms')
plt.ylabel('price')
plt.legend()
```

<matplotlib.legend.Legend at 0x7fde1b06d550>



```
#Preprocessing using min max scaler
```

```
scaler = MinMaxScaler()
```

```
scaler.fit(df[['price']])
```

```
df['price'] = scaler.transform(df[['price']])
```

```
scaler.fit(df[['bedrooms']])
```

```
df['bedrooms'] = scaler.transform(df[['bedrooms']])
```

```
df.head()
```

	bedrooms	sqft_living	price	cluster	
0	0.250	1340	0.068710	0	
1	0.500	3650	0.736774	2	
2	0.250	1930	0.078065	0	
3	0.250	2000	0.103226	0	
4	0.375	1940	0.145161	0	

```
plt.scatter(df.bedrooms,df['price'])
```

&lt;matplotlib.collections.PathCollection at 0x7fde1b11e490&gt;



```
km = KMeans(n_clusters=3)
```

```
y_predicted = km.fit_predict(df[['bedrooms', 'price']])
```

```
y_predicted
```

```
array([1, 2, 1, 1, 0, 1, 1, 0, 1, 0, 1, 2, 1, 1, 2, 1, 1, 0, 1, 1, 1, 0,
       1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1,
       1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 2, 0, 1, 1, 0, 0, 0, 0, 2, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 2, 1, 1, 1, 0, 1, 1, 0, 1, 1,
       0, 2, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 2, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1,
       0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 2, 1, 1, 0, 0,
       0, 1, 0, 2, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 2, 1, 1,
       1, 1, 1, 0, 1, 0, 1, 2, 0, 1, 0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 2, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 2, 2,
       1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,
       1, 2, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0,
       0, 1, 0, 1, 2, 1, 0, 2, 1, 0, 1, 1, 1, 1, 2, 1, 1, 0, 1, 1, 0, 2,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 1, 0, 0, 1, 0, 2, 1, 1, 1, 1, 1, 1, 1, 0, 1, 2, 0, 1, 1,
       2, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 2, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       2, 1, 0, 1, 0, 0, 1, 0, 1, 2, 0, 1, 0, 1, 1, 1, 2, 1, 0, 1, 2, 1,
       0, 1, 0, 0, 2, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1], dtype=int32)
```

```
df['cluster']=y_predicted
```

```
df.head()
```

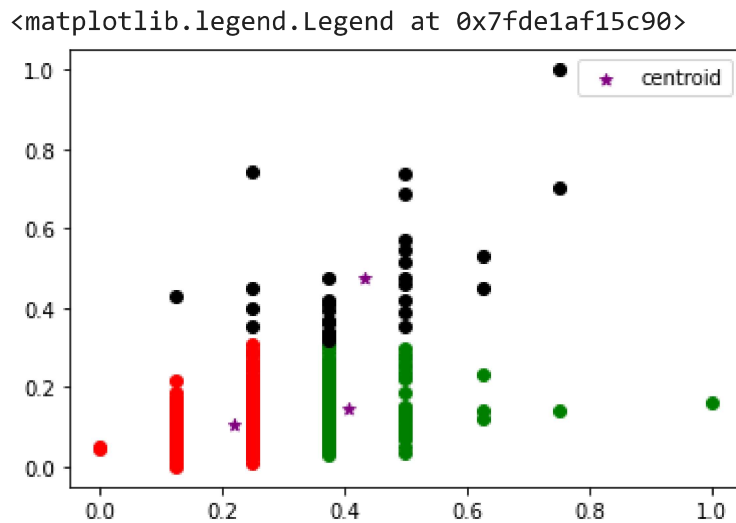
	bedrooms	sqft_living	price	cluster	
0	0.250	1340	0.068710	1	
1	0.500	3650	0.736774	2	
2	0.250	1930	0.078065	1	
3	0.250	2000	0.103226	1	
4	0.375	1940	0.145161	0	

```
km.cluster_centers_
```

```
array([[0.40508021, 0.14871579],
       [0.21797153, 0.1070548 ]],
```

```
[0.43145161, 0.4735389 ]])
```

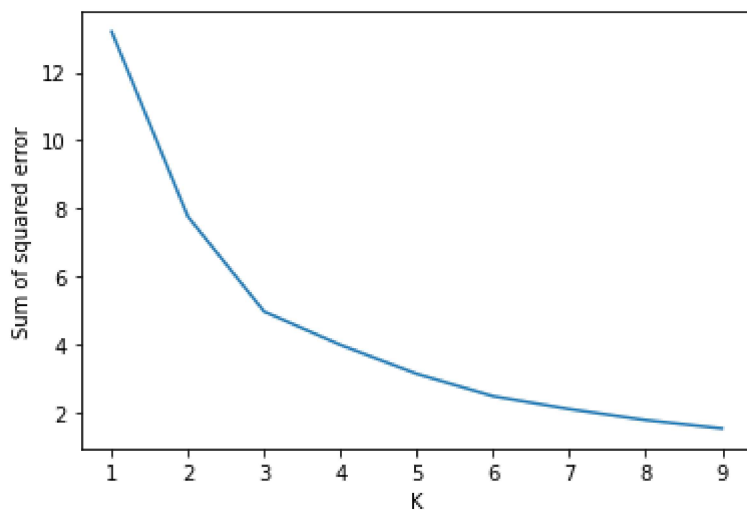
```
df1 = df[df.cluster==0]
df2 = df[df.cluster==1]
df3 = df[df.cluster==2]
plt.scatter(df1.bedrooms,df1['price'],color='green')
plt.scatter(df2.bedrooms,df2['price'],color='red')
plt.scatter(df3.bedrooms,df3['price'],color='black')
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color='purple',marker='*',label=0)
plt.legend()
```



```
#Elbow plot
sse = []
k_rng = range(1,10)
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[['bedrooms','price']])
    sse.append(km.inertia_)
```

```
plt.xlabel('K')
plt.ylabel('Sum of squared error')
plt.plot(k_rng,sse)
```

```
[<matplotlib.lines.Line2D at 0x7fde1669ab10>]
```



[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 10:57 PM

