

```
In [1]: 1 import pandas as pd
        2 from matplotlib import pyplot as plt
        3 %matplotlib inline
```

```
In [2]: 1 df=pd.read_csv(r"C:\Users\teppa\Downloads\Income.csv")
        2 df
```

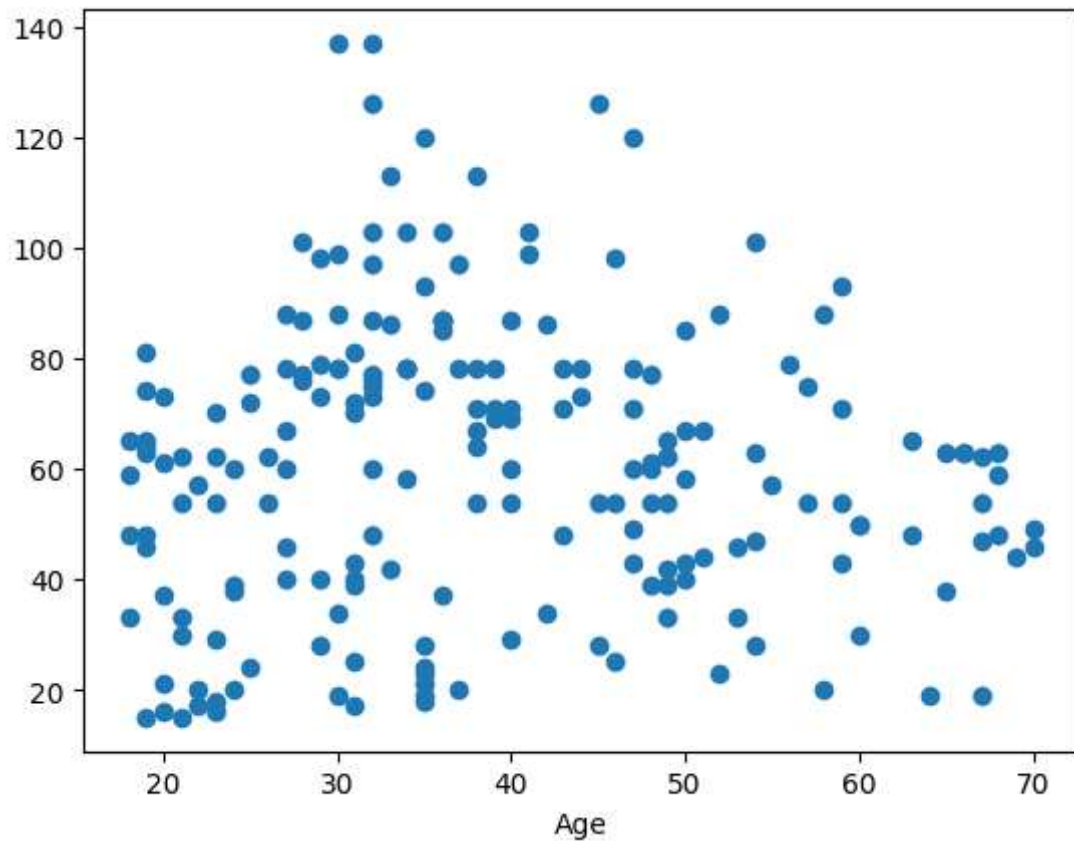
Out[2]:

	Gender	Age	Income(\$)
0	Male	19	15
1	Male	21	15
2	Female	20	16
3	Female	23	16
4	Female	31	17
...
195	Female	35	120
196	Female	45	126
197	Male	32	126
198	Male	32	137
199	Male	30	137

200 rows × 3 columns

```
In [3]: 1 plt.scatter(df['Age'],df['Income($)'])  
2 plt.xlabel("Age")
```

Out[3]: Text(0.5, 0, 'Age')



```
In [4]: 1 from sklearn.cluster import KMeans
```

```
In [13]: 1 Km=KMeans()  
2 Km
```

Out[13]:

▼ KMeans

KMeans()

```
In [15]: 1 y_predicted=Knn.fit_predict(df[["Age","Income($)"]])
        2 y_predicted
```

C:\Users\teppa\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[15]: array([0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 4, 0, 4, 0, 0, 0, 0, 0, 4, 0, 0, 0,
        4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 6, 4, 6, 4, 6, 6, 6, 4, 6, 4, 6,
        4, 6, 4, 6, 6, 6, 4, 6, 6, 4, 4, 4, 4, 2, 6, 4, 2, 6, 2, 4, 2, 6,
        4, 2, 6, 6, 2, 4, 2, 2, 2, 6, 3, 3, 6, 3, 2, 3, 2, 3, 6, 3, 2, 7,
        3, 3, 2, 7, 3, 3, 7, 7, 3, 7, 3, 7, 7, 3, 2, 7, 3, 7, 2, 3, 2, 2,
        2, 7, 3, 7, 7, 7, 2, 3, 3, 3, 7, 3, 3, 3, 7, 7, 3, 3, 3, 3, 3, 3,
        7, 7, 7, 7, 3, 7, 7, 7, 3, 7, 7, 7, 7, 3, 7, 7, 7, 3, 3, 3, 7,
        3, 7, 7, 7, 7, 7, 3, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 5, 5, 5, 5, 5, 5,
        5, 5])
```

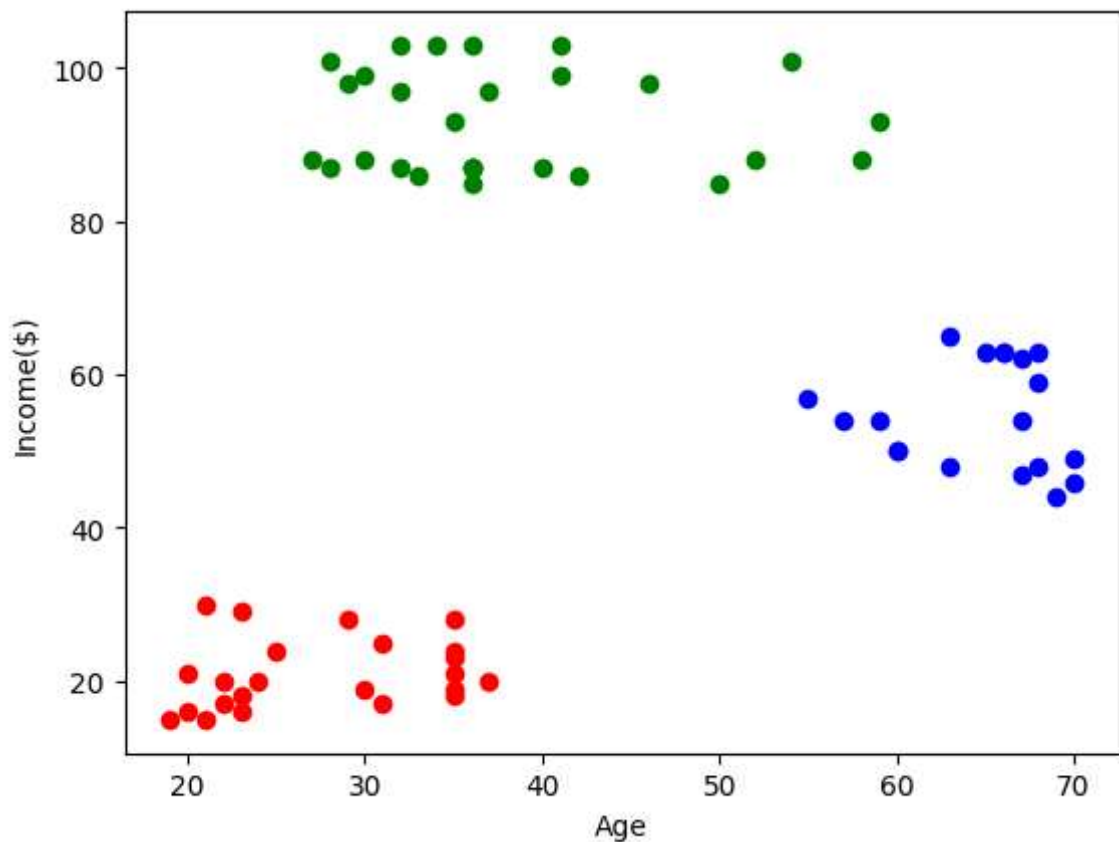
```
In [16]: 1 df['cluster']=y_predicted
        2 df.head()
```

Out[16]:

	Gender	Age	Income(\$)	cluster
0	Male	19	15	0
1	Male	21	15	0
2	Female	20	16	0
3	Female	23	16	0
4	Female	31	17	0

```
In [19]: 1 df1=df[df.cluster==0]
2 df2=df[df.cluster==1]
3 df3=df[df.cluster==2]
4 plt.scatter(df1["Age"],df1["Income($)"],color="red")
5 plt.scatter(df2["Age"],df2["Income($)"],color="green")
6 plt.scatter(df3["Age"],df3["Income($)"],color="blue")
7 plt.xlabel("Age")
8 plt.ylabel("Income($)")
9
```

Out[19]: Text(0, 0.5, 'Income(\$))')



```
In [24]: 1 from sklearn.preprocessing import MinMaxScaler
2 scaler=MinMaxScaler()
3 scaler.fit(df[["Income($)"]])
4 df["Income($)"]=scaler.transform(df[["Income($)"]])
5 df.head()
```

Out[24]:

	Gender	Age	Income(\$)	cluster
0	Male	19	0.000000	0
1	Male	21	0.000000	0
2	Female	20	0.008197	0
3	Female	23	0.008197	0
4	Female	31	0.016393	0

```
In [26]: 1 Scaler.fit(df[["Age"]])
2 df["Age"]=Scaler.transform(df[["Age"]])
3 df.head()
```

Out[26]:

	Gender	Age	Income(\$)	cluster
0	Male	0.019231	0.000000	0
1	Male	0.057692	0.000000	0
2	Female	0.038462	0.008197	0
3	Female	0.096154	0.008197	0
4	Female	0.250000	0.016393	0

```
In [28]: 1 Km=KMeans()
2
```

```
In [33]: 1 y_predicted=Km.fit_predict(df[['Age', 'Income($)']])
2 y_predicted
```

C:\Users\teppa\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[33]: array([4, 4, 4, 4, 2, 4, 2, 4, 5, 2, 5, 2, 1, 4, 2, 4, 2, 4, 1, 2, 2, 4, 1, 2, 1, 2, 1, 2, 2, 4, 5, 4, 1, 4, 1, 4, 1, 2, 2, 4, 5, 4, 1, 2, 1, 4, 1, 2, 2, 2, 1, 2, 2, 5, 1, 1, 1, 5, 0, 1, 5, 0, 5, 1, 5, 0, 1, 5, 0, 2, 5, 1, 5, 5, 5, 0, 1, 1, 0, 1, 5, 3, 5, 1, 0, 1, 1, 0, 3, 1, 5, 0, 6, 3, 3, 0, 6, 0, 6, 0, 0, 6, 5, 0, 6, 0, 5, 6, 5, 5, 5, 0, 3, 0, 0, 0, 5, 6, 6, 6, 0, 3, 3, 3, 0, 3, 6, 3, 6, 3, 6, 3, 0, 3, 0, 3, 6, 3, 0, 3, 6, 3, 3, 3, 0, 3, 6, 3, 3, 3, 6, 3, 6, 3, 6, 3, 3, 3, 3, 3, 6, 3, 0, 3, 6, 3, 6, 3, 3, 3, 3, 3, 3, 3, 6, 3, 6, 3, 6, 3, 7, 7, 6, 7, 7, 7, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7])

```
In [34]: 1 df['New cluster']=y_predicted
2 df.head()
```

Out[34]:

	Gender	Age	Income(\$)	cluster	New cluster
0	Male	0.019231	0.000000	0	4
1	Male	0.057692	0.000000	0	4
2	Female	0.038462	0.008197	0	4
3	Female	0.096154	0.008197	0	4
4	Female	0.250000	0.016393	0	2

```

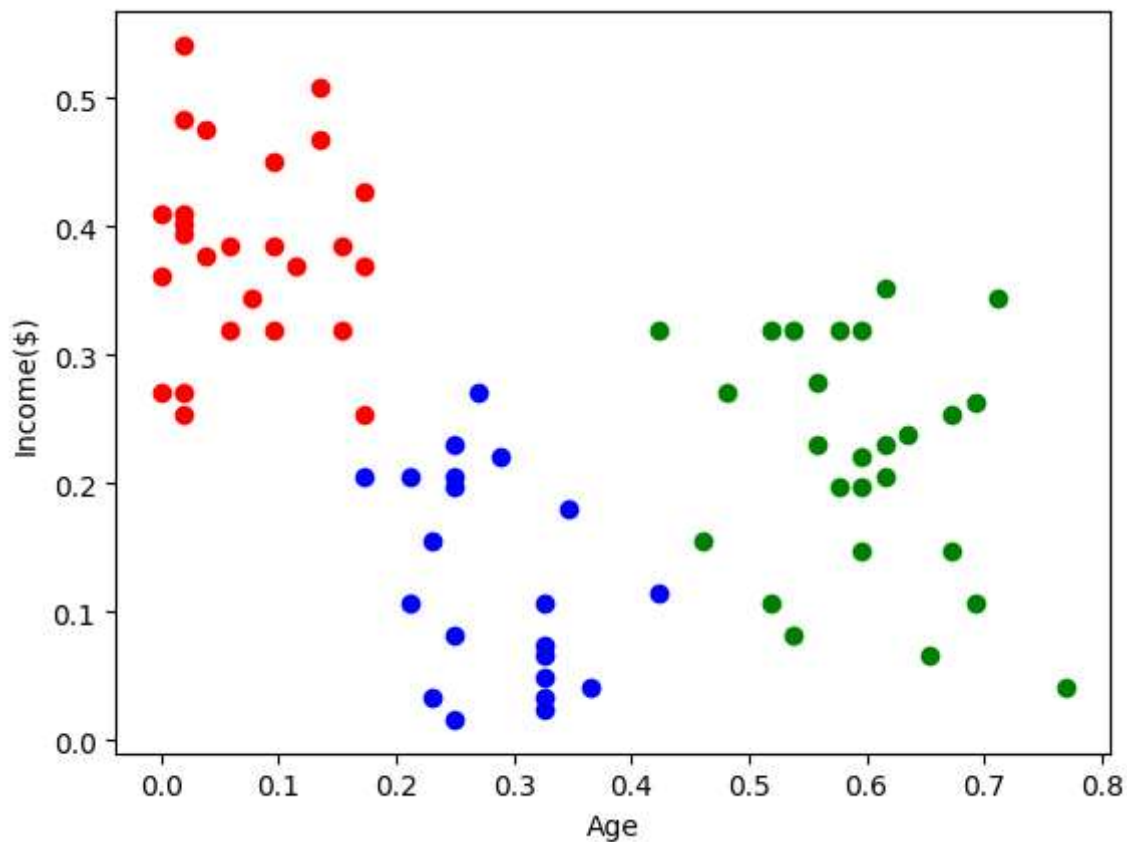
In [40]: 1 df1=df[df["New cluster"]==0]
          2 df2=df[df["New cluster"]==1]
          3 df3=df[df["New cluster"]==2]
          4 plt.scatter(df1["Age"],df1["Income($)"],color="red")
          5 plt.scatter(df2["Age"],df2["Income($)"],color="green")
          6 plt.scatter(df3["Age"],df3["Income($)"],color="blue")
          7 plt.xlabel("Age")
          8 plt.ylabel("Income($)")

```

```

Out[40]: Text(0, 0.5, 'Income($)')

```



```

In [54]: 1 Km.cluster_centers_

```

```

Out[54]: array([[0.07322485, 0.38272383],
                [0.5954142 , 0.2203657 ],
                [0.28388278, 0.1245121 ],
                [0.3059034 , 0.50247808],
                [0.07239819, 0.08003857],
                [0.89799331, 0.28011404],
                [0.61094675, 0.49401009],
                [0.32905983, 0.78551913]])

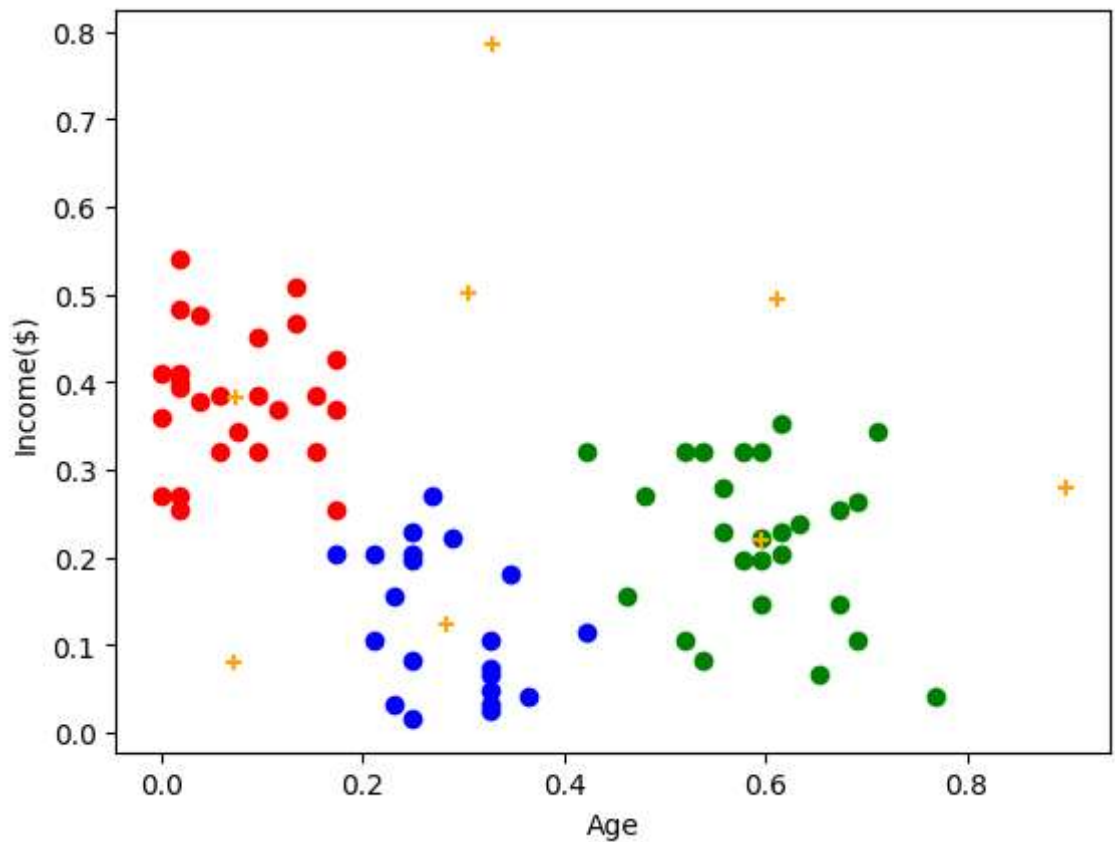
```

```

In [56]: 1 df1=df[df["New cluster"]==0]
          2 df2=df[df["New cluster"]==1]
          3 df3=df[df["New cluster"]==2]
          4 plt.scatter(df1["Age"],df1["Income($)"],color="red")
          5 plt.scatter(df2["Age"],df2["Income($)"],color="green")
          6 plt.scatter(df3["Age"],df3["Income($)"],color="blue")
          7 plt.scatter(Km.cluster_centers_[0],Km.cluster_centers_[1],color="orange")
          8 plt.xlabel("Age")
          9 plt.ylabel("Income($)")

```

Out[56]: Text(0, 0.5, 'Income(\$'))



In [60]:

```
1 k_mg=range(1,10)
2 sse=[]
3 for k in k_mg:
4     Km=KMeans(n_clusters=k)
5     Km.fit(df[["Age", "Income($)"]])
6     sse.append(Km.inertia_)
7     sse
```

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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(

C:\Users\teppa\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

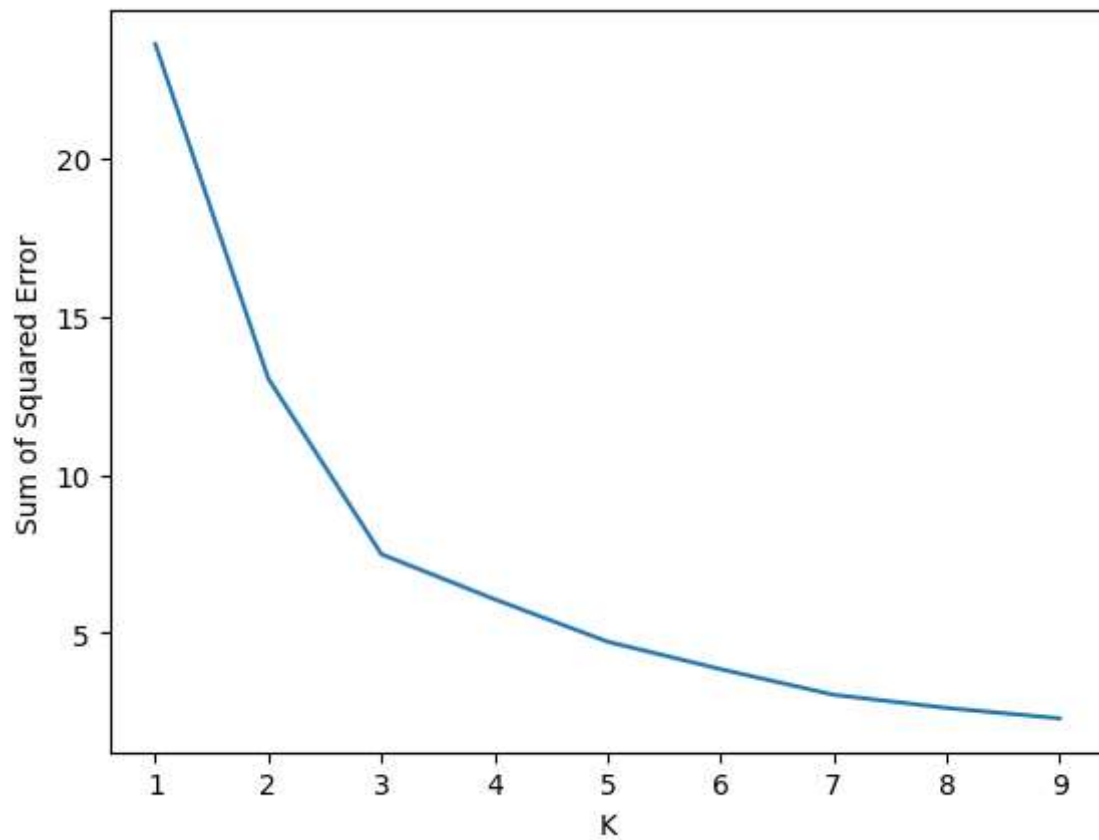
warnings.warn(

C:\Users\teppa\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(


```
In [65]: 1 plt.plot(k_mg,sse)
          2 plt.xlabel("K")
          3 plt.ylabel("Sum of Squared Error")
```

Out[65]: Text(0, 0.5, 'Sum of Squared Error')



```
In [ ]: 1
```

```
In [ ]: 1
```