

# K Means clustering

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

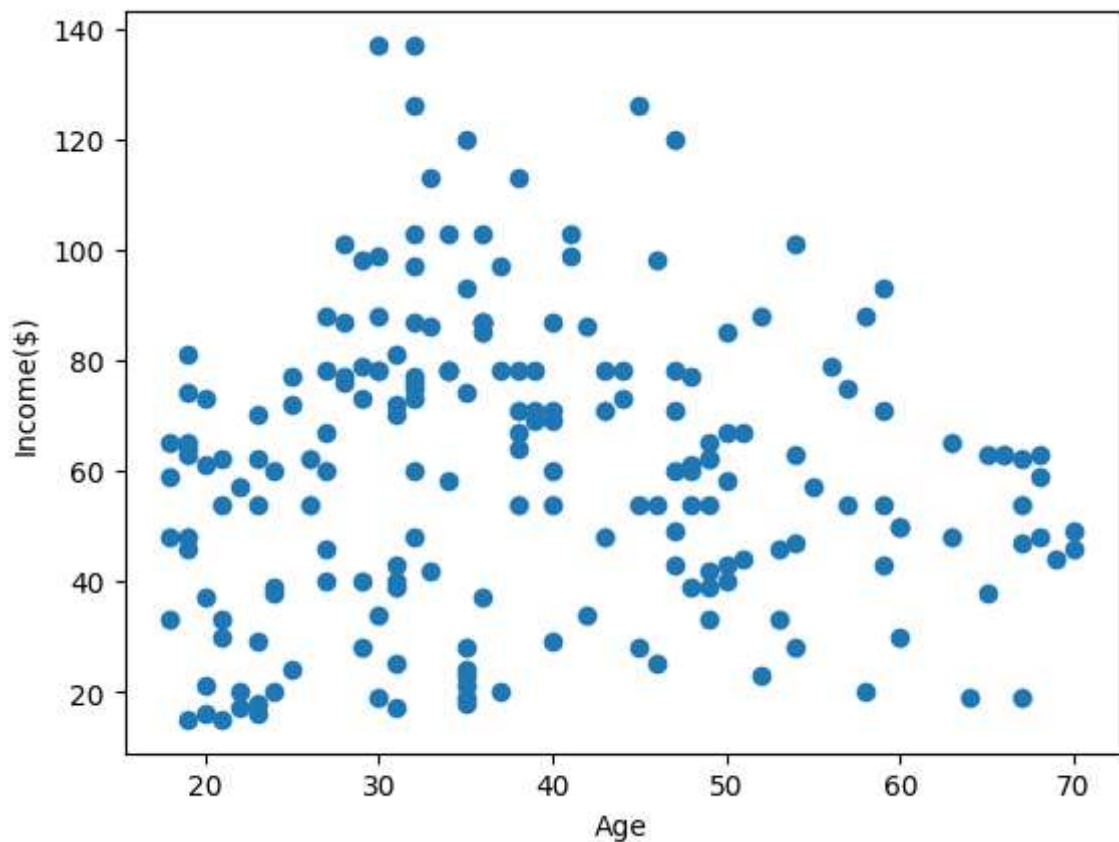
```
In [2]: df=pd.read_csv(r"C:\Users\sudheer\Downloads\Income.csv")
        df.head()
```

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

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

Out[3]: Text(0, 0.5, 'Income(\$)')



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

```
In [5]: km = KMeans()  
km
```

```
Out[5]: 
▼ KMeans
KMeans()
```

```
In [6]: y_predicted = km.fit_predict(df[["Age", "Income($)"]])  
y_predicted
```

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

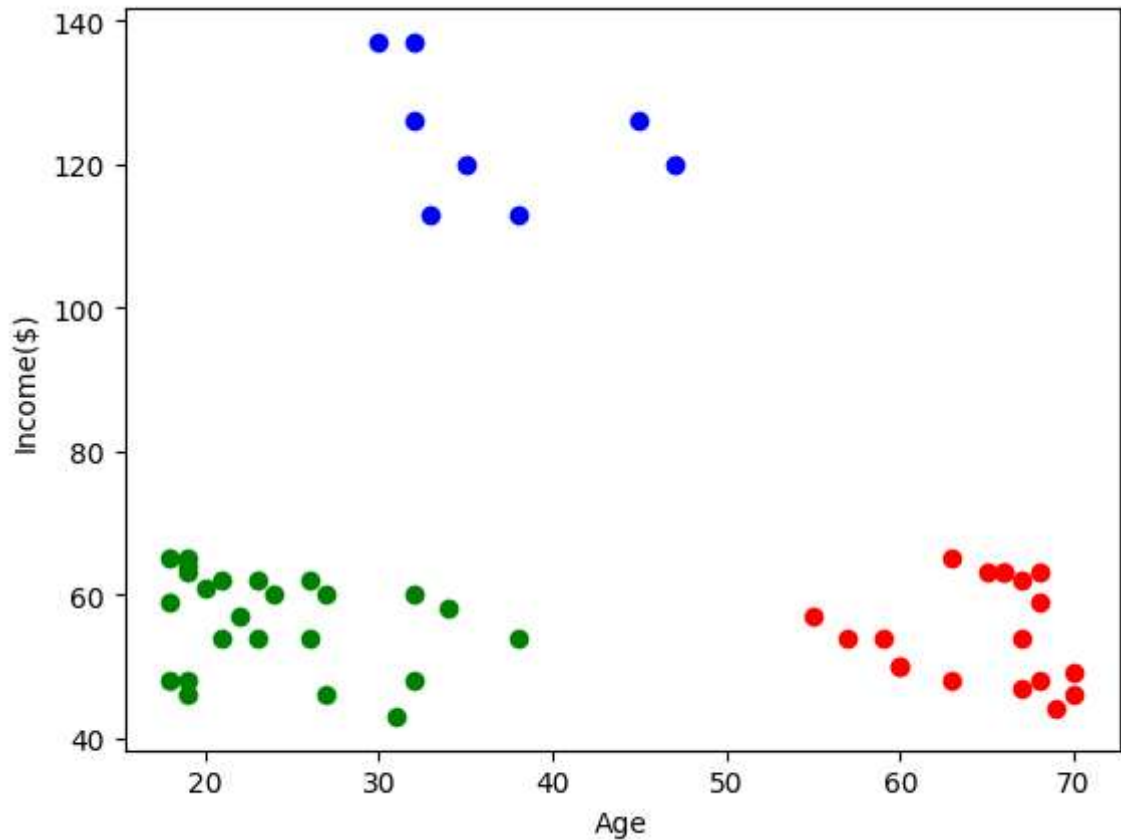
```
In [7]: df["Cluster"]=y_predicted  
df.head()
```

```
Out[7]:
```

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

```
In [10]: df1 = df[df.Cluster==0]
df2 = df[df.Cluster==1]
df3 = df[df.Cluster==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

```
Out[10]: Text(0, 0.5, 'Income($)')
```



```
In [11]: from sklearn.preprocessing import MinMaxScaler
```

```
In [12]: scaler = MinMaxScaler()
```

```
In [13]: scaler.fit(df[["Income($)"]])
df["Income($)"] = scaler.transform(df[["Income($)"]])
df.head()
```

```
Out[13]:
```

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

```
In [14]: scaler.fit(df[["Age"]])
df["Age"] = scaler.transform(df[["Age"]])
df.head()
```

```
Out[14]:
```

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

```
In [15]: km = KMeans()
```

```
In [16]: y_predicted=km.fit_predict(df[["Age", "Income($)"]])
y_predicted
```

C:\Users\sudheer\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[16]: array([5, 5, 5, 5, 0, 5, 0, 5, 6, 0, 6, 0, 3, 5, 0, 5, 0, 5, 3, 0, 0, 5,
 3, 0, 3, 0, 3, 0, 0, 5, 6, 5, 3, 5, 3, 5, 3, 0, 0, 5, 6, 5, 3, 0,
 3, 5, 3, 0, 0, 0, 3, 0, 0, 6, 3, 3, 3, 6, 2, 3, 6, 2, 6, 3, 6, 2,
 3, 6, 2, 0, 6, 3, 6, 6, 6, 2, 3, 3, 2, 3, 6, 7, 6, 3, 2, 3, 1, 2,
 7, 1, 6, 2, 1, 7, 7, 2, 1, 2, 1, 2, 2, 1, 6, 2, 1, 2, 6, 1, 6, 6,
 6, 2, 7, 2, 2, 2, 6, 1, 1, 1, 2, 7, 7, 7, 2, 7, 1, 7, 1, 7, 1, 7,
 2, 7, 2, 7, 1, 7, 2, 7, 1, 7, 7, 7, 2, 7, 1, 7, 7, 7, 1, 7, 1, 7,
 1, 7, 7, 7, 7, 7, 1, 7, 2, 7, 1, 7, 7, 7, 7, 7, 7, 7, 7, 1, 7,
 1, 7, 1, 7, 4, 4, 1, 4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
 4, 4])
```

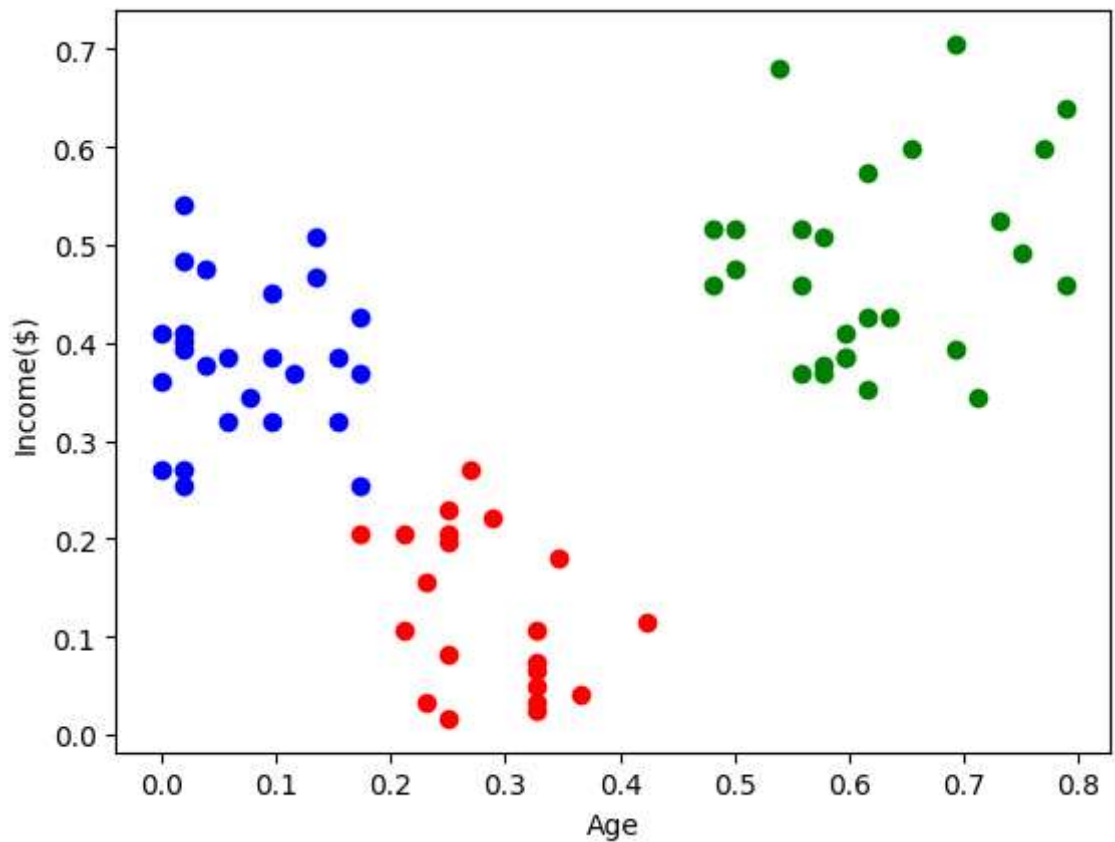
```
In [17]: df["New Cluster"] = y_predicted
df.head()
```

Out[17]:

	Gender	Age	Income(\$)	Cluster	New Cluster
0	Male	0.019231	0.000000	3	5
1	Male	0.057692	0.000000	3	5
2	Female	0.038462	0.008197	3	5
3	Female	0.096154	0.008197	3	5
4	Female	0.250000	0.016393	3	0

```
In [18]: df1 = df[df["New Cluster"]==0]
df2 = df[df["New Cluster"]==1]
df3 = df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.xlabel("Age")
plt.ylabel("Income($)")
```

Out[18]: Text(0, 0.5, 'Income(\$)')

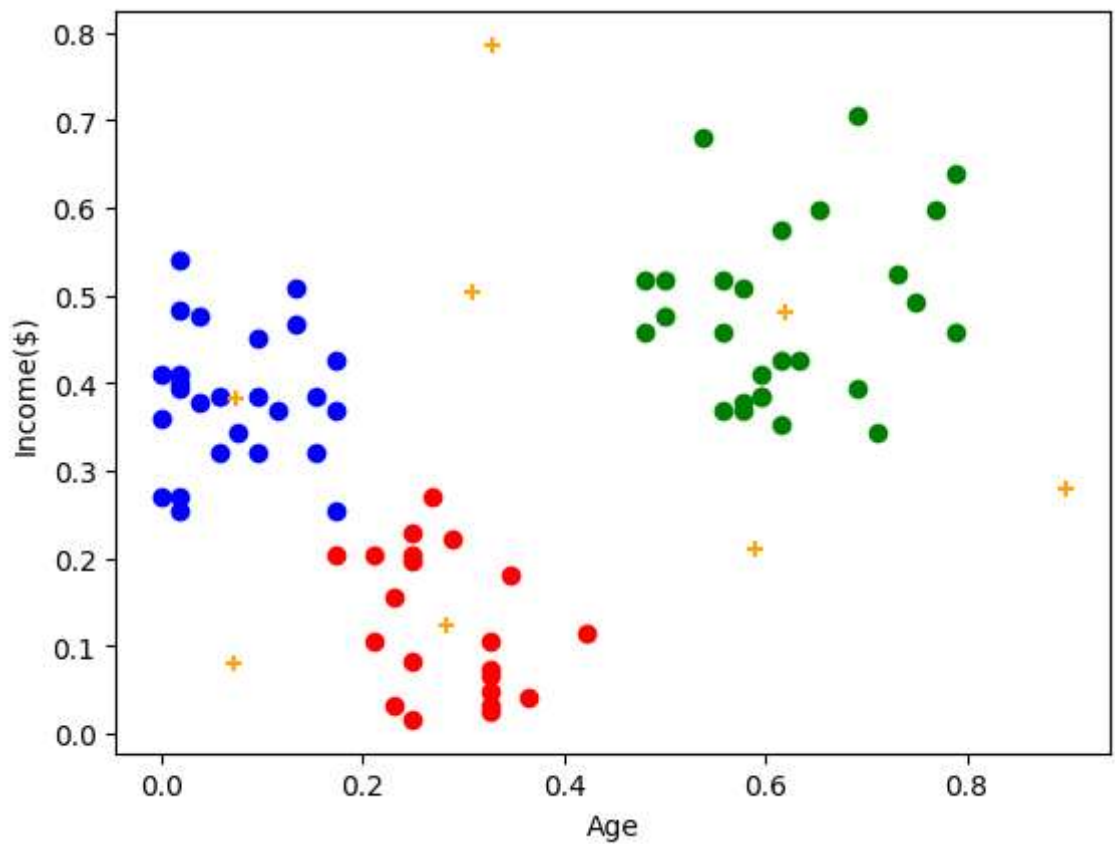


```
In [19]: km.cluster_centers_
```

```
Out[19]: array([[0.28388278, 0.1245121 ],
 [0.62037037, 0.47996357],
 [0.07322485, 0.38272383],
 [0.58974359, 0.20969945],
 [0.32905983, 0.78551913],
 [0.07239819, 0.08003857],
 [0.89799331, 0.28011404],
 [0.30944056, 0.50428465]])
```

```
In [20]: df1 = df[df["New Cluster"]==0]
df2 = df[df["New Cluster"]==1]
df3 = df[df["New Cluster"]==2]
plt.scatter(df1["Age"],df1["Income($)"],color="red")
plt.scatter(df2["Age"],df2["Income($)"],color="green")
plt.scatter(df3["Age"],df3["Income($)"],color="blue")
plt.scatter(km.cluster_centers_[0],km.cluster_centers_[1],color="orange",marker='x')
plt.xlabel("Age")
plt.ylabel("Income($)")
```

```
Out[20]: Text(0, 0.5, 'Income($)')
```



```
In [25]: k_rng = range(1,10)
sse = []
for k in k_rng:
    km = KMeans(n_clusters=k)
    km.fit(df[["Age", "Income($)"]])
    sse.append(km.inertia_)
sse
```

C:\Users\sudheer\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\sudheer\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\sudheer\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\sudheer\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\sudheer\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\sudheer\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\sudheer\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\sudheer\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(
```

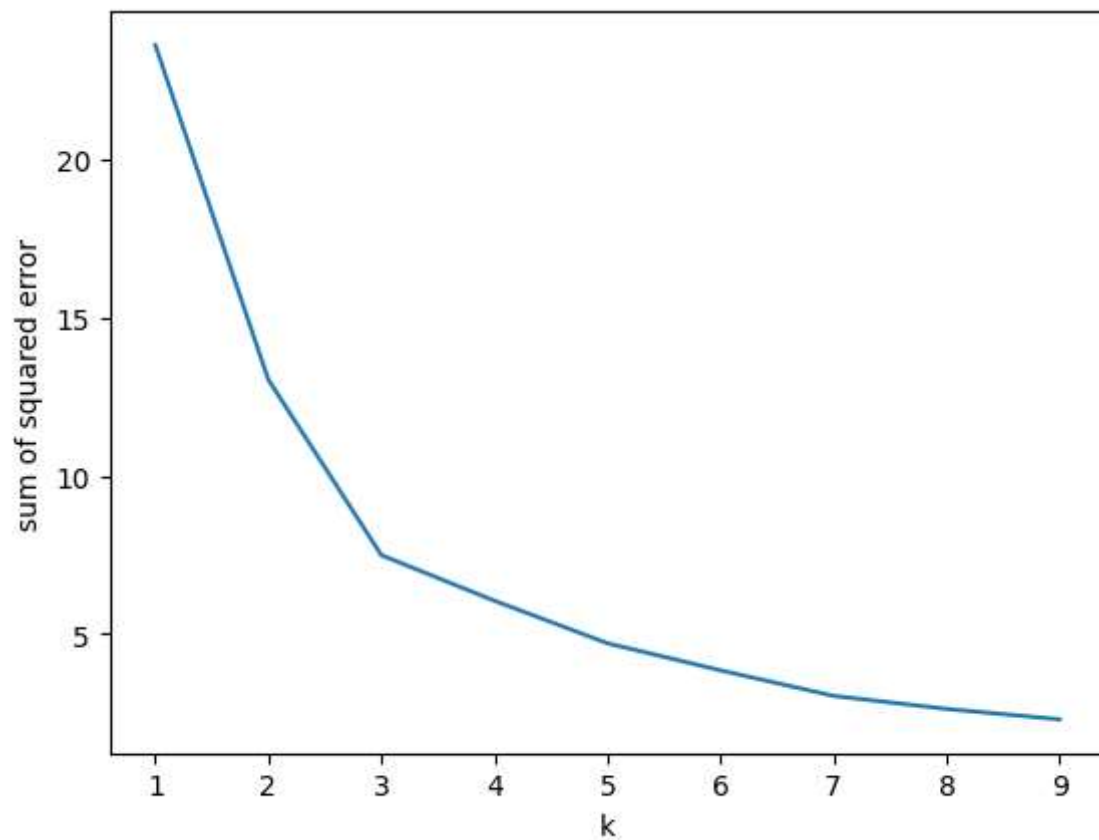
C:\Users\sudheer\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\cluster\\_kmeans.py:870: FutureWarning: The default value of `n\_init` will change from 10 to 'auto' in 1.4. Set the value of `n\_init` explicitly to suppress the warning

```
warnings.warn(
```

```
Out[25]: [23.583906150363603,  
          13.028938428018286,  
          7.493024843304991,  
          6.055824667599623,  
          4.713416604872824,  
          3.860037034685059,  
          3.054717436369358,  
          2.6426939469218094,  
          2.3135720353543285]
```

```
In [26]: plt.plot(k_rng,sse)  
plt.xlabel("k")  
plt.ylabel("sum of squared error")
```

```
Out[26]: Text(0, 0.5, 'sum of squared error')
```



```
In [ ]:
```