mall_data

June 24, 2024

[ ]:
```
import dataidea as di
import numpy as np
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

[ ]:
```
mall_data= di.loadDataset('mall')
mall_data.head()
```

[ ]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

[ ]:
```
numeric_cols = ["Age", "Annual Income (k$)", "Spending Score (1-100)"]
X = mall_data[numeric_cols]
X.head()
```

[ ]:

	Age	Annual Income (k$)	Spending Score (1-100)
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40

[ ]:
```
#fitting the model
model= KMeans(n_clusters=3, random_state=42)
model.fit(X)
```

[ ]:
```
KMeans(n_clusters=3, random_state=42)
```

[ ]:
```
#checking the cluster centers of each cluster
model.cluster_centers_
```

[ ]:
```
array([[44.48387097, 59.87903226, 35.42741935],
       [32.97560976, 88.73170732, 79.24390244],
       [25.77142857, 29.97142857, 68.51428571]])
```

[ ]:
```
#making predicions on X ( clustering)
preds = model.predict(X)
```

[ ]:
```
#assigning each row to a cluster
X['Clusters']= preds
X.head(n=5)
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_10980\1126609120.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['Clusters']= preds
```

[ ]:

	Age	Annual Income (k$)	Spending Score (1-100)	Clusters
0	19	15	39	2
1	21	15	81	2
2	20	16	6	0
3	23	16	77	2
4	31	17	40	0

[ ]:
```
sns.scatterplot(data=X, x='Spending Score (1-100)', y='Annual Income (k$)',␣
↪hue=preds)
centers_x, centers_y = model.cluster_centers_[:,0], model.cluster_centers_[:,2]
plt.plot(centers_x, centers_y, 'xb')
plt.title('Spending Score (1-100) against Annual Income (k$)')
plt.ylabel('Annual Income (k$)')
plt.xlabel('Spending Score (1-100)')
plt.show()
```

[ ]:
```
# calculating the inertia
model.inertia_
```

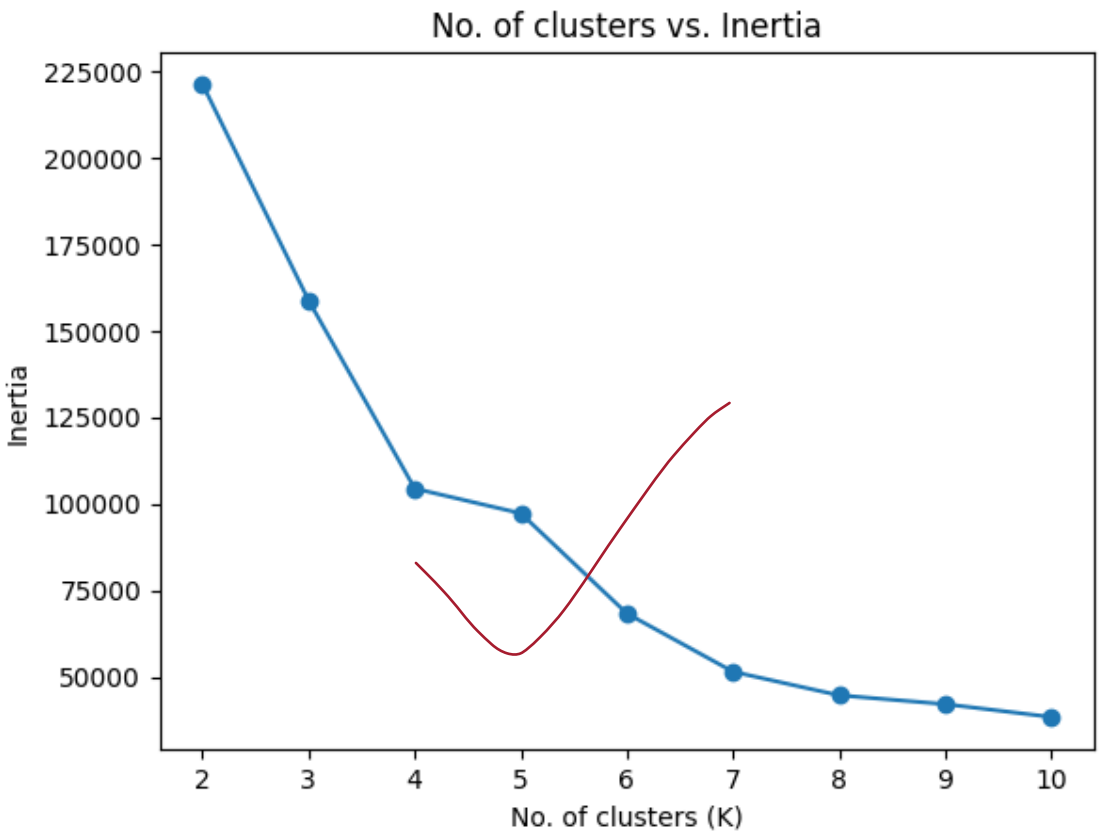
[ ]: 158744.9710801394

[ ]:
```
options = range(2, 11)
inertias = []

for n_clusters in options:
    model = KMeans(n_clusters, random_state=42).fit(X)
    inertias.append(model.inertia_)

plt.plot(options, inertias, linestyle='-', marker='o')
plt.title("No. of clusters vs. Inertia")
plt.xlabel('No. of clusters (K)')
plt.ylabel('Inertia')
```

[ ]: Text(0, 0.5, 'Inertia')



The best number of clusters to use is 4

[ ]:
```
model= KMeans(n_clusters=4, random_state=42)
model.fit(X)
```

[ ]:
```
KMeans(n_clusters=4, random_state=42)
```

[ ]:
```
model.cluster_centers_
```

[ ]:
```
array([[44.89473684, 48.70526316, 42.63157895,  0.23157895],
       [32.69230769, 86.53846154, 82.12820513,  3.15384615],
       [24.82142857, 28.71428571, 74.25      ,  2.        ],
       [40.39473684, 87.        , 18.63157895,  3.        ]])
```

[ ]:
```
preds=model.predict(X)
```

[ ]:
```
X['Clusters']=preds
X.tail()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_10980\263820305.py:1:
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['Clusters']=preds
```

[ ]:

	Age	Annual Income (k$)	Spending Score (1-100)	Clusters
195	35	120	79	1
196	45	126	28	3
197	32	126	74	1
198	32	137	18	3
199	30	137	83	1

[ ]:
```
sns.scatterplot(data=X, x='Spending Score (1-100)', y='Annual Income (k$)',␣
 ↪hue=preds)
centers_x, centers_y = model.cluster_centers_[:,0], model.cluster_centers_[:,2]
plt.plot(centers_x, centers_y, 'xb')
plt.title('Spending Score (1-100) against Annual Income (k$)')
plt.ylabel('Annual Income (k$)')
plt.xlabel('Spending Score (1-100)')
plt.show()
```