

k-means-mall-customer

February 4, 2024

```
[50]: import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
import seaborn as sns
```

```
[6]: df = pd.read_csv("Mall_Customers.csv")
df.head()
```

```
[6]:
```

	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

```
[68]: df= df.drop(columns="CustomerID")
```

```
[69]: df.describe().T
```

```
[69]:
```

	count	mean	std	min	25%	50% \
Gender	200.0	0.440000	0.497633	0.0	0.000000	0.000000
Age	200.0	0.400962	0.268635	0.0	0.206731	0.346154
Annual Income (k\$)	200.0	0.373443	0.215285	0.0	0.217213	0.381148
Spending Score (1-100)	200.0	0.502041	0.263505	0.0	0.344388	0.500000

	75%	max
Gender	1.000000	1.0
Age	0.596154	1.0
Annual Income (k\$)	0.516393	1.0
Spending Score (1-100)	0.734694	1.0

```
[70]: #Null values
df.isnull().sum()
```

```
[70]: Gender          0
      Age            0
      Annual Income (k$)  0
      Spending Score (1-100)  0
      dtype: int64
```

```
[71]: #Categorical Values to numerical

encoder = LabelEncoder()
df['Gender'] = encoder.fit_transform(df['Gender'])

#Normalize data and categorical values
scaler = MinMaxScaler()
df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']] = scaler.
    fit_transform(df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
```

```
[85]: df
```

```
[85]:      Gender      Age  Annual Income (k$)  Spending Score (1-100)
0         1  0.019231          0.000000          0.387755
1         1  0.057692          0.000000          0.816327
2         0  0.038462          0.008197          0.051020
3         0  0.096154          0.008197          0.775510
4         0  0.250000          0.016393          0.397959
..      ...      ...      ...      ...
195        0  0.326923          0.860656          0.795918
196        0  0.519231          0.909836          0.275510
197        1  0.269231          0.909836          0.744898
198        1  0.269231          1.000000          0.173469
199        1  0.230769          1.000000          0.836735
```

```
[200 rows x 4 columns]
```

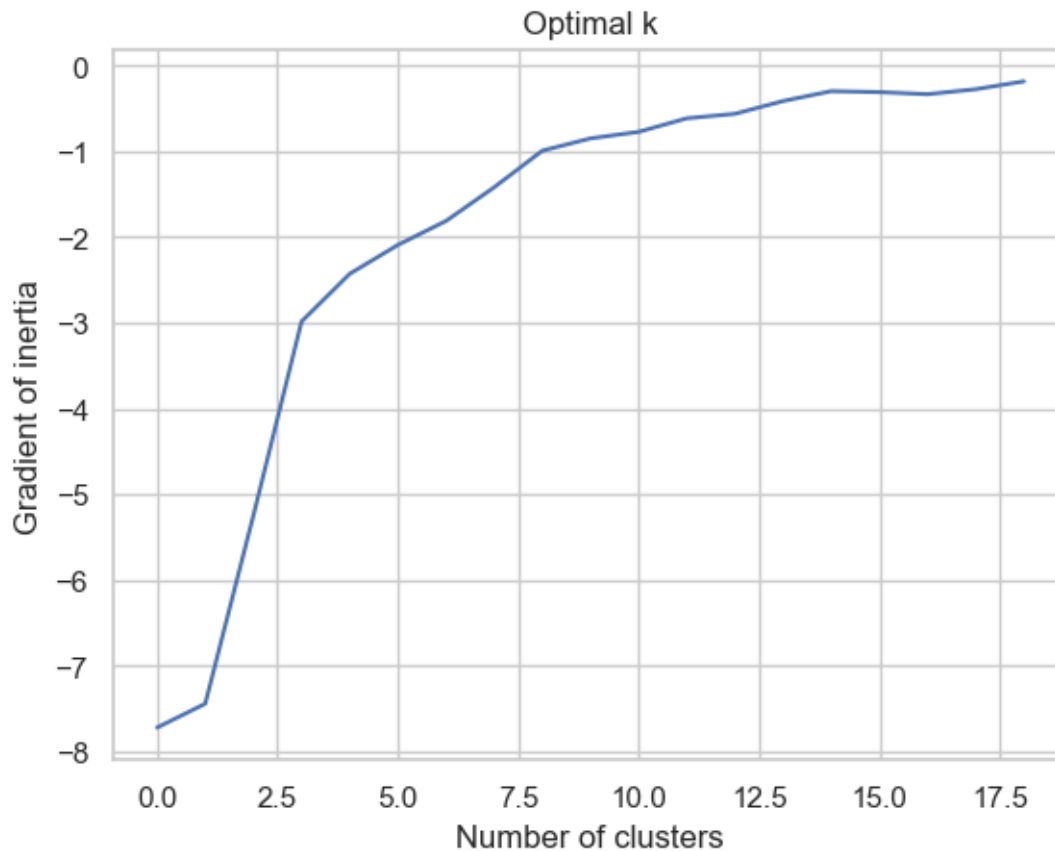
```
[89]: min_nb_clusters = 2
      max_nb_clusters = 20
      inertias = np.zeros(shape=(max_nb_clusters - min_nb_clusters + 1,))
      for i in range(min_nb_clusters, max_nb_clusters + 1):
          km = KMeans(n_clusters=i, random_state=1000)
          km.fit(df)
          inertias[i - min_nb_clusters] = km.inertia_
```

```
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-
packages\sklearn\cluster\_kmeans.py:1416: 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
    super()._check_params_vs_input(X, default_n_init=10)
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-
```



```
plt.ylabel('Gradient of inertia')
plt.title('Optimal k')

# Muestra el gráfico
plt.show()
```



```
[120]: km = KMeans(n_clusters=3, random_state=1000)
Y = km.fit_predict(df.iloc[:, :3])
```

C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-packages\sklearn\cluster_kmeans.py:1416: 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

```
super()._check_params_vs_input(X, default_n_init=10)
```

```
[121]: from sklearn.manifold import TSNE
tsne = TSNE(n_components=2, perplexity=10.0, random_state=1000)
X_tsne = tsne.fit_transform(df.iloc[:, :3])
```

```
[122]: sns.scatterplot(x = X_tsne[:,0], y = X_tsne[:,1], hue= Y, palette = sns.  
        ↪hls_palette(10), legend = 'full');
```

```
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-  
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is  
deprecated and will be removed in a future version. Use isinstance(dtype,  
CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-  
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is  
deprecated and will be removed in a future version. Use isinstance(dtype,  
CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-  
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is  
deprecated and will be removed in a future version. Use isinstance(dtype,  
CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\HECTOR\PycharmProjects\pythonProject4\venv\lib\site-  
packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype is  
deprecated and will be removed in a future version. Use isinstance(dtype,  
CategoricalDtype) instead  
    if pd.api.types.is_categorical_dtype(vector):  
C:\Users\HECTOR\AppData\Local\Temp\ipykernel_17244\3194591602.py:1: UserWarning:  
The palette list has more values (10) than needed (3), which may not be  
intended.  
    sns.scatterplot(x = X_tsne[:,0], y = X_tsne[:,1], hue= Y, palette =  
sns.hls_palette(10), legend = 'full');
```

