5c-K-Means.Clustering

November 8, 2024

[1]:
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Example with Making Blobs

[ ]:
```
X, y_true = make_blobs(
        n_samples=300,
        centers=4,
        cluster_std=0.60,
        random_state=0,
)
# plt.scatter(X[:, 0], X[:, 1], s=50, );
sns.scatterplot(x = X[:, 0], y = X[:, 1], s=50) ;
;
```

1.1 KMeans

[ ]:
```
kmeans = KMeans(n_clusters=4, n_init=10).fit(X)
kmeans.labels_
```

[ ]:
```
sns.scatterplot(x = X[:,0], y = X[:,1], hue = kmeans.labels_, s=50) ;
```

1.2 Example with Pima Indians Dataset

[ ]:
```
col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree',␣
  ↪'age', 'label']
url = "https://ddc-datascience.s3.amazonaws.com/pima-indians-diabetes.csv"
pima = pd.read_csv( url, header=None, names=col_names)
pima.head()
```

[ ]:
```
pima["label"].value_counts()
```

```
[ ]: # Drop the label & some of the other variables for simplicity
     pima_X = pima.drop(['label', 'skin', 'pedigree', 'bp'], axis = 1).copy()
     pima_X.head()
```

```
[ ]: # Scale data
     scaler = MinMaxScaler()
     scaler.fit(pima_X)
     pima_X_scaled = scaler.transform(pima_X)
     # Convert back to data frame
     pima_X_scaled = pd.DataFrame(pima_X_scaled, columns = pima_X.columns)
     pima_X_scaled.head()
```

1.3 KMeans

```
[ ]: # Fit k-means w/ 4 clusters
     kmeans = KMeans(n_clusters=4, n_init=10).fit(pima_X_scaled)
     kmeans.labels_
```

```
[ ]: # Add a new column to pima_X_scaled with the cluster assignment
     pima_X_scaled['cluster'] = kmeans.labels_
     pima_X_scaled['cluster'].value_counts()
```

```
[ ]: sns.pairplot(pima_X_scaled, hue='cluster') ;
```

1.4 Choosing K - the elbow method

```
[ ]: # Drop cluster column
     pima_X_scaled.drop('cluster', axis = 1, inplace = True)
```

```
[ ]: distortions = []
     K = range(1,10)
     for k in K:
         kmeans = KMeans(n_clusters=k, n_init = 10)
         kmeans.fit(pima_X_scaled)
         distortions.append(kmeans.inertia_)
```

```
[ ]: plt.figure(figsize=(16,8))
     plt.plot(K, distortions, 'bx-')
     plt.xlabel('k')
     plt.ylabel('Distortion')
     plt.show()
```

```
[ ]: %%capture
     !pip install -U yellowbrick
```

```
[ ]: from yellowbrick.cluster.elbow import kelbow_visualizer
[ ]: # Use the quick method and immediately show the figure
     kelbow_visualizer(KMeans(random_state=4, n_init=10), pima_X_scaled, k=(1,10)) ;
     ;
```

2 Fit again with k = 3

```
[ ]: kmeans = KMeans(n_clusters=3, n_init=10).fit(pima_X_scaled)
```

```
[ ]: sns.pairplot(pima_X_scaled.assign(cluster=kmeans.labels_), hue='cluster') ;
```

```
[ ]:
```