# kmeans

March 3, 2022

## 1 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import time
from sklearn.preprocessing import StandardScaler
```

---

## 2 Code Algorithm

[2]:
```
def kmeans_alg(train, clusters, actual_labels):
    ## track runtime
    start = time.time()

    ## run algorithm
    output = KMeans(n_clusters = clusters, init = 'k-means++', max_iter = 10000,
                    n_init = 10, random_state = 0).fit(train)

    ## predicted labels
    predicted = output.labels_

    ## track runtime
    end = time.time()
    elapsed = end - start

    ## evaluate accuracy
    correct = sum(predicted == actual_labels)
    accuracy = correct / len(actual_labels)

    return(round(accuracy, 4), round(elapsed, 4))
```

---

## 3 Iris Dataset

[3]:
```
## import data
iris = pd.read_csv('iris.csv')
iris.head()
```

[3]:

| | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|---|---|---|---|---|
| 0 | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 1 | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 2 | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 3 | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 4 | 5.0 | 3.6 | 1.4 | 0.2 | setosa |

[4]:
```
## extract explanatory variables
iris_train = iris.iloc[:, [0,1,2,3]].values

## extract actual species label
iris_species = iris.iloc[:, 4]

## re-label actual species to integers
iris_distinct_species = iris.Species.unique()

iris_dict = {iris_distinct_species[0]: 1,
                 iris_distinct_species[1]: 2,
                 iris_distinct_species[2]: 0}
iris_labels = iris_species.replace(iris_dict)

## run algorithm
iris_metrics = kmeans_alg(iris_train, 3, iris_labels)

## print results
print(iris_metrics)
```

```
(0.8933, 0.0125)
```

---

## 4 Penguins Dataset

[5]:
```
## import data
penguins = pd.read_csv('penguins.csv')
penguins = penguins.dropna()
penguins.head()
```

[5]:

| | species | island | bill_length_mm | bill_depth_mm | flipper_length_mm | \ |
|---|---|---|---|---|---|---|
| 0 | Adelie | Torgersen | 39.1 | 18.7 | 181.0 | |
| 1 | Adelie | Torgersen | 39.5 | 17.4 | 186.0 | |
| 2 | Adelie | Torgersen | 40.3 | 18.0 | 195.0 | |

```
4  Adelie  Torgersen            36.7           19.3              193.0
5  Adelie  Torgersen            39.3           20.6              190.0

   body_mass_g     sex  year
0       3750.0    male  2007
1       3800.0  female  2007
2       3250.0  female  2007
4       3450.0  female  2007
5       3650.0    male  2007
```

[6]:
```
## extract explanatory variables
penguins_train = penguins.iloc[:, [2,3,4,5]].values

## extract actual species label
penguins_species = penguins.iloc[:, 0]

## re-label actual species to integers
penguins_distinct_species = penguins.species.unique()
penguins_dict = {penguins_distinct_species[0]: 2,
                 penguins_distinct_species[1]: 1,
                 penguins_distinct_species[2]: 0}
penguins_labels = penguins_species.replace(penguins_dict)

## run algorithm
penguins_metrics = kmeans_alg(penguins_train, 3, penguins_labels)

## print results
print(penguins_metrics)
```

```
(0.5826, 0.0154)
```

---

## 5 Seeds Dataset

[7]:
```
## import data
seeds = pd.read_csv('seeds_dataset.csv')
seeds.head()
```

[7]:

| | Area | Perim | Compact | K.Length | K.Width | Assym | G.Length | Class |
|---|---|---|---|---|---|---|---|---|
| 0 | 15.26 | 14.84 | 0.8710 | 5.763 | 3.312 | 2.221 | 5.220 | 1 |
| 1 | 14.88 | 14.57 | 0.8811 | 5.554 | 3.333 | 1.018 | 4.956 | 1 |
| 2 | 14.29 | 14.09 | 0.9050 | 5.291 | 3.337 | 2.699 | 4.825 | 1 |
| 3 | 13.84 | 13.94 | 0.8955 | 5.324 | 3.379 | 2.259 | 4.805 | 1 |
| 4 | 16.14 | 14.99 | 0.9034 | 5.658 | 3.562 | 1.355 | 5.175 | 1 |

[8]:
```
## extract explanatory variables
seeds_train = seeds.iloc[:, [0,1,2,3,4,5,6]].values

## extract actual species label
seeds_species = seeds.iloc[:, 7]

## re-label actual species to integers
seeds_distinct_species = seeds.Class.unique()
seeds_dict = {seeds_distinct_species[0]: 2,
              seeds_distinct_species[1]: 1,
              seeds_distinct_species[2]: 0}
seeds_labels = seeds_species.replace(seeds_dict)

## run algorithm
seeds_metrics = kmeans_alg(seeds_train, 3, seeds_labels)

## print results
print(seeds_metrics)
```

```
(0.8952, 0.0135)
```

---

# 6 Principal Component Analysis

From the Principal Component Analysis *explained variance ratio* (shown below), we observe the following:

- Around 73% of the variance in the *iris* dataset can be explained by the first principal component and around 23% of the variance in the dataset can be explained by the second, for a total of 96%
- Around 69% of the variance in the *penguins* dataset can be explained by the first principal component and around 19% of the variance in the dataset can be explained by the second, for a total of 88%
- Around 72% of the variance in the *seeds* dataset can be explained by the first principal component and around 17% of the variance in the dataset can be explained by the second, for a total of 89%

Therefore, we will try using both only one and two principal components below.

[9]:
```
pca = PCA()
```

## 6.1 Iris PCA

[10]:
```
scaler = StandardScaler()
iris_std = scaler.fit_transform(iris_train)

iris_pca = pca.fit(iris_std)
```

```
iris_pca.explained_variance_ratio_
```

[10]:
```
array([0.72962445, 0.22850762, 0.03668922, 0.00517871])
```

### 6.2 Penguins PCA

[11]:
```
scaler = StandardScaler()
penguins_std = scaler.fit_transform(penguins_train)

penguins_pca = pca.fit(penguins_std)
penguins_pca.explained_variance_ratio_
```

[11]:
```
array([0.68633893, 0.19452929, 0.09216063, 0.02697115])
```

### 6.3 Seeds PCA

[12]:
```
scaler = StandardScaler()
seeds_std = scaler.fit_transform(seeds_train)

seeds_pca = pca.fit(seeds_std)
seeds_pca.explained_variance_ratio_
```

[12]:
```
array([7.18743027e-01, 1.71081835e-01, 9.68576341e-02, 9.76635386e-03,
       2.67337271e-03, 7.61720812e-04, 1.16056686e-04])
```

---

## 7 K-Means Using PCA Components

### 7.1 Iris Dataset

#### 7.1.1 One Principal Component

[13]:
```
## fit PCA
iris_pca1 = PCA(n_components = 1)
iris_pca1.fit(iris_std)
iris_scores1 = iris_pca1.transform(iris_std)

## re-label actual species to integers
iris_dict_pca1 = {iris_distinct_species[0]: 1,
                  iris_distinct_species[1]: 2,
                  iris_distinct_species[2]: 0}
iris_labels_pca1 = iris_species.replace(iris_dict_pca1)

## run algorithm
iris_metrics_pca1 = kmeans_alg(iris_scores1, 3, iris_labels_pca1)

## print results
```

```
print(iris_metrics_pca1)
```

```
(0.9267, 0.0146)
```

#### 7.1.2 Two Principal Components

[14]:
```
## fit PCA
iris_pca2 = PCA(n_components = 2)
iris_pca2.fit(iris_std)
iris_scores2 = iris_pca2.transform(iris_std)

## re-label actual species to integers
iris_dict_pca2 = {iris_distinct_species[0]: 1,
                  iris_distinct_species[1]: 2,
                  iris_distinct_species[2]: 0}
iris_labels_pca2 = iris_species.replace(iris_dict_pca2)

## run algorithm
iris_metrics_pca2 = kmeans_alg(iris_scores2, 3, iris_labels_pca2)

## print results
print(iris_metrics_pca2)
```

```
(0.8333, 0.015)
```

### 7.2 Penguins Dataset

#### 7.2.1 One Principal Component

[15]:
```
## fit PCA
penguins_pca1 = PCA(n_components = 1)
penguins_pca1.fit(penguins_std)
penguins_scores1 = penguins_pca1.transform(penguins_std)

## re-label actual species to integers
penguins_dict_pca1 = {penguins_distinct_species[0]: 2,
                      penguins_distinct_species[1]: 1,
                      penguins_distinct_species[2]: 0}
penguins_labels_pca1 = penguins_species.replace(penguins_dict_pca1)

## run algorithm
penguins_metrics_pca1 = kmeans_alg(penguins_scores1, 3, penguins_labels_pca1)

## print results
print(penguins_metrics_pca1)
```

```
(0.8589, 0.0199)
```

### 7.2.2 Two Principal Components

[16]:
```
## fit PCA
penguins_pca2 = PCA(n_components = 2)
penguins_pca2.fit(penguins_std)
penguins_scores2 = penguins_pca2.transform(penguins_std)

## re-label actual species to integers
penguins_dict_pca2 = {penguins_distinct_species[0]: 2,
                      penguins_distinct_species[1]: 0,
                      penguins_distinct_species[2]: 1}
penguins_labels_pca2 = penguins_species.replace(penguins_dict_pca2)

## run algorithm
penguins_metrics_pca2 = kmeans_alg(penguins_scores2, 3, penguins_labels_pca2)

## print results
print(penguins_metrics_pca2)
```

```
(0.8799, 0.0189)
```

## 7.3 Seeds Dataset

### 7.3.1 One Principal Component

[17]:
```
## fit PCA
seeds_pca1 = PCA(n_components = 1)
seeds_pca1.fit(seeds_std)
seeds_scores1 = seeds_pca1.transform(seeds_std)

## re-label actual species to integers
seeds_dict_pca1 = {seeds_distinct_species[0]: 2,
                   seeds_distinct_species[1]: 1,
                   seeds_distinct_species[2]: 0}
seeds_labels_pca1 = seeds_species.replace(seeds_dict_pca1)

## run algorithm
seeds_metrics_pca1 = kmeans_alg(seeds_scores1, 3, seeds_labels_pca1)

## print results
print(seeds_metrics_pca1)
```

```
(0.8571, 0.0125)
```

### 7.3.2 Two Principal Components

[18]:
```
## fit PCA
seeds_pca2 = PCA(n_components = 2)
seeds_pca2.fit(seeds_std)
seeds_scores2 = seeds_pca2.transform(seeds_std)

## re-label actual species to integers
seeds_dict_pca2 = {seeds_distinct_species[0]: 0,
                   seeds_distinct_species[1]: 2,
                   seeds_distinct_species[2]: 1}
seeds_labels_pca2 = seeds_species.replace(seeds_dict_pca2)

## run algorithm
seeds_metrics_pca2 = kmeans_alg(seeds_scores2, 3, seeds_labels_pca2)

## print results
print(seeds_metrics_pca2)
```

```
(0.9095, 0.0135)
```

---

# 8 Compare Full Model vs. PCA Model

## 8.1 Iris Dataset

[19]:
```
## 0 = Full Model (4 Predictors)
## 1 = PCA Model w/ One Component
## 2 = PCA Model w/ Two Components
iris_compare = [iris_metrics,
                iris_metrics_pca1,
                iris_metrics_pca2]
iris_compare = pd.DataFrame(iris_compare, columns=['Accuracy', 'Runtime'])
print(iris_compare)
```

```
   Accuracy  Runtime
0    0.8933   0.0125
1    0.9267   0.0146
2    0.8333   0.0150
```

## 8.2 Penguins Dataset

[20]:
```
## 0 = Full Model (4 Predictors)
## 1 = PCA Model w/ One Component
## 2 = PCA Model w/ Two Components
penguins_compare = [penguins_metrics,
                    penguins_metrics_pca1,
                    penguins_metrics_pca2]
```

```
penguins_compare = pd.DataFrame(penguins_compare, columns=['Accuracy',␣
↪'Runtime'])
print(penguins_compare)
```

```
   Accuracy  Runtime
0    0.5826   0.0154
1    0.8589   0.0199
2    0.8799   0.0189
```

### 8.3 Seeds Dataset

[21]:
```
## 0 = Full Model (7 Predictors)
## 1 = PCA Model w/ One Component
## 2 = PCA Model w/ Two Components
seeds_compare = [seeds_metrics,
                 seeds_metrics_pca1,
                 seeds_metrics_pca2]
seeds_compare = pd.DataFrame(seeds_compare, columns=['Accuracy', 'Runtime'])
print(seeds_compare)
```

```
   Accuracy  Runtime
0    0.8952   0.0135
1    0.8571   0.0125
2    0.9095   0.0135
```