

# KMeans

February 27, 2023

## 1 Explorasi K-Means

Load libraries yang diperlukan untuk melakukan pembelajaran.

```
[1]: # Load libraries
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.cluster import KMeans
import numpy as np
import random
import pickle
```

Baca datanya dan bagi menjadi data training dan test.

```
[2]: # Load the data
breast_cancer = load_breast_cancer()

# split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(breast_cancer.data,
↪breast_cancer.target, test_size=0.2, random_state=42)
```

Latih algoritma dengan dataset training. Inisialisasi cluster supaya label sama.

```
[3]: # Using random predetermined data as seed
kmeans_init = np.array([random.choice(X_train[y_train == 0]), random.
↪choice(X_train[y_train == 1])])
# Make an object and fit the data
KM = KMeans(n_clusters=2, init=kmeans_init, n_init=1)
KM.fit(X_train)
```

```
[3]: KMeans(init=array([[1.603e+01, 1.551e+01, 1.058e+02, 7.932e+02, 9.491e-02,
1.371e-01,
1.204e-01, 7.041e-02, 1.782e-01, 5.976e-02, 3.371e-01, 7.476e-01,
2.629e+00, 3.327e+01, 5.839e-03, 3.245e-02, 3.715e-02, 1.459e-02,
1.467e-02, 3.121e-03, 1.876e+01, 2.198e+01, 1.243e+02, 1.070e+03,
1.435e-01, 4.478e-01, 4.956e-01, 1.981e-01, 3.019e-01, 9.124e-02],
```

```
[1.220e+01, 1.521e+01, 7.801e+01, 4.579e+02, 8.673e-02, 6.545e-02,
 1.994e-02, 1.692e-02, 1.638e-01, 6.129e-02, 2.575e-01, 8.073e-01,
 1.959e+00, 1.901e+01, 5.403e-03, 1.418e-02, 1.051e-02, 5.142e-03,
 1.333e-02, 2.065e-03, 1.375e+01, 2.138e+01, 9.111e+01, 5.831e+02,
 1.256e-01, 1.928e-01, 1.167e-01, 5.556e-02, 2.661e-01, 7.961e-02]]),
n_clusters=2, n_init=1)
```

Simpan model pada suatu file, kemudian load kembali file tersebut.

```
[4]: # Save the model
with open('KMeans_model.pkl', 'wb') as f:
    pickle.dump(KM, f)
```

```
[5]: # Load the model
with open('KMeans_model.pkl', 'rb') as f:
    KM = pickle.load(f)
```

Lakukan prediksi menggunakan algoritma yang telah di-train menggunakan dataset test.

```
[6]: # Predict result
y_pred = KM.predict(X_test)
```

Tampilkan confusion matrix dari prediksi

```
[7]: cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
[[29 14]
 [ 0 71]]
```

Tampilkan laporan perbandingan hasil prediksi dengan hasil asli prediksi.

```
[8]: report = classification_report(y_test, y_pred)
print(report)
```

	precision	recall	f1-score	support
0	1.00	0.67	0.81	43
1	0.84	1.00	0.91	71
accuracy			0.88	114
macro avg	0.92	0.84	0.86	114
weighted avg	0.90	0.88	0.87	114

Terdapat tiga metrik penilaian yaitu Precision, Recall, dan F1-score. Berikut penjelasan singkat masing-masing: - Precision: Dari semua tebakan positif, berapa banyak yang benar-benar positif? - Recall: Dari semua data yang positif, berapa banyak yang benar ditebak positif? - F1-score: Harmonic Mean dari Precision dan Recall.

Nilai tersebut merupakan nilai per kelas, sehingga untuk didapatkan nilai keseluruhan dilakukan rata-rata dengan macro average dan weigted average.