

<<<<

ARTIFICIAL INTELLIGENCE (AI)

PRAKTIKUM 10

Dosen Pengampu : Renovita Edelani S.ST., M.Tr.Kom

oleh : Dukhaan Kamimpangan
2 D4 IT A
3122600003



01.

DATASET MILK.CSV



MENAMPILKAN DATASET MILK.CSV

Analisis : Kode ini mengimpor library pandas dan numpy untuk analisis data. Data set dari file CSV 'milk.csv' dibaca menggunakan pandas, kemudian dibagi menjadi train_data (fitur) dan train_label (label) dari 0 hingga -1 menggunakan numpy. Hasilnya ditampilkan dengan mencetak train_data dan train_label.

```
import pandas as pd
import numpy as np

dataset = pd.read_csv('milk.csv')
train_data = np.array(dataset[:,0:-1])
train_label = np.array(dataset[:, -1])

print(train_data)
print(train_label)
```

```
[[6.6 35 1 ... 1 0 254]
 [6.6 36 0 ... 0 1 253]
 [8.5 70 1 ... 1 1 246]
 ...
 [3.0 40 1 ... 1 1 255]
 [6.8 43 1 ... 1 0 250]
 [8.6 55 0 ... 1 1 255]]
['high' 'high' 'low' ... 'low' 'high' 'low']
```



02. KLASIFIKASI DENGAN K-NN

MENAMPILKAN DATATRAINING

Analisis : Dilakukan percobaan serupa dengan sebelumnya, tetapi terdapat baris tambahan yang mengimpor kelas KNeighborsClassifier dari pustaka scikit-learn. Setelah membaca dataset dari file 'milk.csv' menggunakan pandas, kode ini mengambil fitur dari dataset dan memisahkan labelnya. Kemudian, kedua array ini dicetak untuk ditampilkan. Kode ini kemungkinan akan diikuti dengan penggunaan model KNeighborsClassifier untuk melakukan klasifikasi berdasarkan fitur dan label yang disediakan.

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier

dataset = pd.read_csv('milk.csv')
train_data = np.array(dataset[:,0:-1])
train_label = np.array(dataset[:, -1])

print(train_data)
print(train_label)
```

```
[[6.6 35 1 ... 1 0 254]
 [6.6 36 0 ... 0 1 253]
 [8.5 70 1 ... 1 1 246]
 ...
 [3.0 40 1 ... 1 1 255]
 [6.8 43 1 ... 1 0 250]
 [8.6 55 0 ... 1 1 255]]
['high' 'high' 'low' ... 'low' 'high' 'low']
```

NORMALISASI TERHADAP TRAIN_DATA DENGAN METODE MIN-MAX(0-1)

Analisis : penambahan library MinMaxScaler dari scikit-learn. Setelah dataset dibaca menggunakan pandas, MinMaxScaler digunakan untuk menormalkan fitur-fitur dalam train_data ke dalam rentang antara 0 dan 1. Kemudian, hasil normalisasi disimpan dalam variabel data. Hasil normalisasi ini adalah fitur-fitur yang telah dinormalisasi. Dalam kasus ini, normalisasi ini berguna untuk memastikan bahwa semua fitur memiliki skala yang serupa, yang biasanya diinginkan dalam banyak algoritma pembelajaran mesin.

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk.csv')
sc = MinMaxScaler(feature_range=(0,1))
train_data = np.array(dataset[:,0:-1])
data = sc.fit_transform(train_data)

print(data)
```

[[0.55384615	0.01785714	1.	...	1.	0.	0.93333333]
[0.55384615	0.03571429	0.	...	0.	1.	0.86666667]
[0.84615385	0.64285714	1.	...	1.	1.	0.4
...
[0.	0.10714286	1.	...	1.	1.	1.
[0.58461538	0.16071429	1.	...	1.	0.	0.66666667]
[0.86153846	0.375	0.	...	1.	1.	1.]

KLASIFIKASI MENGGUNAKAN K-NN UNTUK 1 INPUT DATA TEST

Kode ini melakukan klasifikasi menggunakan algoritma k-NN (k-Nearest Neighbors) untuk satu data uji. Langkah-langkahnya adalah sebagai berikut:

1. Dataset dibaca dari file 'milk.csv' menggunakan pandas.
2. Fitur dan label dipisahkan dari dataset.
3. Fitur-fitur dinormalisasi menggunakan MinMaxScaler dari scikit-learn.
4. Model k-NN diinisialisasi dan dilatih dengan data latih yang telah dinormalisasi.
5. Satu contoh data uji disediakan.
6. Data uji dinormalisasi menggunakan MinMaxScaler yang sama.
7. Klasifikasi dilakukan pada data uji menggunakan model k-NN yang telah dilatih.
8. Hasil klasifikasi (label yang diprediksi) ditampilkan.

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk.csv')
train_data = np.array(dataset)[:, :-1]
train_label = np.array(dataset)[:, -1]

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_train_data = scaler.fit_transform(train_data)

knn = KNeighborsClassifier(n_neighbors=3, weights='distance', metric='euclidean')
knn.fit(scaled_train_data, train_label)

test_data = np.array([[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]])

scaled_test_data = scaler.transform(test_data)
predicted_label = knn.predict(scaled_test_data)

print("Predicted label:", predicted_label)

Predicted label: ['medium']
```

`train_label`: Array 1D yang berisi label dari dataset latih. Setiap elemen mewakili label dari satu sampel yang sesuai dalam `train_data`.

[illegible]

NORMALISASI TERHADAP TRAIN_DATA DENGAN METODE MIN-MAX(0-1)

Analisis :

Nilai-nilai dalam output telah dinormalisasi ke dalam rentang antara 0 dan 1, sesuai dengan parameter `feature_range=(0, 1)` yang diberikan saat inisialisasi `MinMaxScaler`. Hal ini dilakukan untuk memastikan bahwa semua fitur memiliki skala yang serupa, yang bisa meningkatkan performa algoritma pembelajaran mesin seperti k-NN.

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler

dataset = pd.read_csv('milk_training.csv')
sc = MinMaxScaler(feature_range=(0,1))
train_data = np.array(dataset)[:,:0:-1]
data = sc.fit_transform(train_data)

print(data)
```

[0.55384615	0.01785714	1.	...	1.	0.	0.93333333]
[0.55384615	0.03571429	0.	...	0.	1.	0.86666667]
[0.55384615	0.05357143	1.	...	1.	1.	1.
...
[0.56923077	0.125	1.	...	0.	0.	0.46666667]
[0.58461538	0.125	0.	...	0.	0.	1.
[0.58461538	0.07142857	0.	...	0.	0.	1.

train_label: Array 1D yang berisi label dari dataset latih. Setiap elemen mewakili label dari satu sampel yang sesuai dalam `train_data`.

[illegible]

KLASIFIKASI MENGGUNAKAN K-NN

Analisis :

Hasil akurasi yang diperoleh adalah sekitar 99.37%, yang menunjukkan tingkat keberhasilan yang tinggi dalam klasifikasi menggunakan model k-NN dengan parameter yang telah ditentukan. Akurasi ini mengindikasikan bahwa model dengan baik dapat memprediksi label kategori untuk data uji yang belum pernah dilihat sebelumnya, berdasarkan pada data latih yang diberikan

```
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import accuracy_score

train_dataset = pd.read_csv('milk_training.csv')
train_data = np.array(train_dataset)[:, 0:-1]
train_label = np.array(train_dataset)[:, -1]

test_dataset = pd.read_csv('milk_testing.csv')
test_data = np.array(test_dataset)[:, 0:-1]
test_label = np.array(test_dataset)[:, -1]

scaler = MinMaxScaler(feature_range=(0,1))
scaled_train_data = scaler.fit_transform(train_data)
scaled_test_data = scaler.transform(test_data)

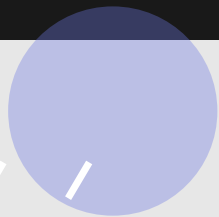
kNN = KNeighborsClassifier(n_neighbors=3, weights='distance', metric='euclidean')
kNN.fit(scaled_train_data, train_label)

predicted_label = kNN.predict(scaled_test_data)
accuracy = accuracy_score(test_label, predicted_label)

print("Accuracy Classifications:", accuracy)

Accuracy Classifications: 0.9936908517350158
```

/[AI]/[AI]/



**TERIMA
KASIH**

ARTI
CIAL
INTE
IGEN
[AI]