



<<<<

# ARTIFICIAL INTELLIGENCE (AI)

## PRAKTIKUM 13

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

---

oleh : Dukhaan Kamimpangan  
2 D4 IT A  
3122600003



<<<<

01.

# SINGLE PERCEPTRON



# DATA AND

## Analisis :

- Model Perceptron berhasil mencapai akurasi 100% pada dataset AND, menghasilkan prediksi [0, 0, 0, 1]. Ini menunjukkan bahwa model berhasil mempelajari fungsi logika AND dengan benar. Dataset AND, yang bersifat linier, cocok untuk model Perceptron yang juga merupakan model linier.

```
import pandas as pd
from sklearn.linear_model import Perceptron
dataset = pd.read_csv('AND.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:,-1]
clf = Perceptron(tol=1e-3, shuffle=False, max_iter=1000, eta0=0.1)
clf.fit(data, label)
print(clf.score(data,label))
print(clf.predict(data))
```

```
1.0
[0 0 0 1]
```

# DATA OR

## Analisis :

- Model Perceptron berhasil mencapai akurasi 100% pada dataset OR, menghasilkan prediksi [0, 1, 1, 1]. Ini menunjukkan bahwa model berhasil mempelajari fungsi logika OR dengan benar. Dataset OR, yang bersifat linier, cocok untuk model Perceptron yang juga merupakan model linier.

```
0s ✓ #Single Preceptor OR
import pandas as pd
from sklearn.linear_model import Perceptron
dataset = pd.read_csv('OR.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:,-1]
clf = Perceptron(tol=1e-3, shuffle=False, max_iter=1000, eta0=0.1)
clf.fit(data, label)
print(clf.score(data,label))
print(clf.predict(data))

1.0
[0 1 1 1]
```

# DATA AND\_NOT

## Analisis :

- Model Perceptron berhasil mencapai akurasi 100% pada dataset AND-NOT, menghasilkan prediksi [0, 0, 1, 0]. Ini menunjukkan bahwa model berhasil mempelajari fungsi logika AND-NOT dengan benar. Dataset AND-NOT, yang bersifat linier, cocok untuk model Perceptron yang juga merupakan model linier.

```
✓ 0s #AND NOT
import pandas as pd
from sklearn.linear_model import Perceptron
dataset = pd.read_csv('AND_NOT.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:, -1]
clf = Perceptron(tol=1e-3, shuffle=False, max_iter=1000, eta0=0.1)
clf.fit(data, label)
print(clf.score(data, label))
print(clf.predict(data))
```

```
1.0
[0 0 1 0]
```

# DATA ANGKA

## Analisis :

- Model Perceptron gagal mempelajari fungsi, hanya mencapai akurasi 66% pada dataset AND. Ini karena Angka tidak dapat dipisahkan secara linier, sedangkan Perceptron hanya dapat memecahkan masalah yang dapat diselesaikan secara linier.

```
#Angka
import pandas as pd
from sklearn.linear_model import Perceptron
dataset = pd.read_csv('Angka.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:,-1]
clf = Perceptron(tol=1e-3, shuffle=False, max_iter=1000, eta0=0.1)
clf.fit(data, label)
print(clf.score(data,label))
print(clf.predict(data))
```

0.6625

[0 0 0 4 7 9 6 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 9 3 3 9 9 9 4 9 4 4 9  
4 4 4 9 9 9 5 9 5 9 5 6 6 8 8 6 6 6 9 7 7 7 9 7 7 7 9 8 9 9 9 9 9 9 9  
9 9 9 9 9]

# DATA XOR

## Analisis :

- Model Perceptron gagal mempelajari fungsi, hanya mencapai akurasi 50% dan prediksi [0, 0, 0, 0] pada dataset XOR. Ini karena XOR tidak dapat dipisahkan secara linier, sedangkan Perceptron hanya dapat memecahkan masalah yang dapat diselesaikan secara linier.

```
#XOR
import pandas as pd
from sklearn.linear_model import Perceptron
dataset = pd.read_csv('XOR.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:, -1]
clf = Perceptron(tol=1e-3, shuffle=False, max_iter=1000, eta0=0.1)
clf.fit(data, label)
print(clf.score(data, label))
print(clf.predict(data))
```

0.5  
[0 0 0 0]

02.

ANALISA GRADE

XOR < 1.0

---



# GET DATA SEMUA ATTRIBUTE

Analisis :

- Perceptron adalah model linier yang hanya dapat memisahkan data yang bisa dipisahkan dengan garis lurus. Namun, operasi XOR (exclusive OR) tidak bisa dipisahkan dengan garis lurus, yang berarti tidak ada garis lurus yang dapat memisahkan kelas-kelasnya dalam ruang dua dimensi. Oleh karena itu, Perceptron tidak mampu mempelajari pola dalam dataset XOR dengan benar.

```
import pandas as pd
from sklearn.neural_network import MLPClassifier as MLP

dataset = pd.read_csv('XOR.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:,1]

model = MLP(hidden_layer_sizes=(2), max_iter=100, activation = 'relu', learning_rate_init=0.1,shuffle=False)
model.fit(data,label)

print('score:', model.score(data,label))
print('predictions:', model.predict(data))
print('expected:',label)
```

score: 0.75  
predictions: [0 0 1 0]  
expected: 0 0  
1 1  
2 1  
3 0  
Name: Y, dtype: int64

/ [AI]

**03.**

**MULTIPLAYER  
PERCEPTRON XOR**

---

# MULTIPLAYER PERCEPTRON XOR DENGAN PARAMETER YANG TEPAT

## Analisis :

- berhasil melatih model MLP untuk memprediksi dataset XOR dengan akurasi 100%. Penyesuaian jumlah neuron di lapisan tersembunyi menjadi 10 dan peningkatan jumlah iterasi hingga 1000 memungkinkan model untuk menangkap pola non-linear dalam dataset XOR. Hasil prediksi model sesuai dengan label yang diharapkan.

```
import pandas as pd
from sklearn.neural_network import MLPClassifier as MLP

dataset = pd.read_csv('XOR.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:, -1]

model = MLP(hidden_layer_sizes=(10), max_iter=1000, activation = 'relu', learning_rate_init=0.1, shuffle=False)
model.fit(data, label)

print('score:', model.score(data, label))
print('predictions:', model.predict(data))
print('expected:', label)

score: 1.0
predictions: [0 1 1 0]
expected: 0    0
1    1
2    1
3    0
Name: Y, dtype: int64
```

04.

# ANALISA JARINGAN DAN SKORE TERBAIK



# CLUSTER DATA MENGGUNAKAN K-MEANS

Analisis :

- Model MLP di atas berhasil memprediksi dataset "Angka" dengan akurasi 100%. Dengan menggunakan konfigurasi lapisan tersembunyi yang memiliki 6 neuron dan menjalankan 1000 iterasi, model mampu mempelajari dan mengenali pola dalam data dengan sangat baik. Hasil prediksi model sepenuhnya sesuai dengan label yang diharapkan, menunjukkan performa yang optimal.

```
import pandas as pd
from sklearn.neural_network import MLPClassifier as MLP

dataset = pd.read_csv('Angka.csv')
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:, -1]

model = MLP(hidden_layer_sizes=(6), max_iter=1000, activation = 'relu', learning_rate_init=0.1, shuffle=False)
model.fit(data, label)

print('score:', model.score(data, label))
print('predictions:', model.predict(data))
print('expected:', label)
```

```
score: 1.0
predictions: [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 4 4 4
4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 9 9
9 9 9 9 9 9]
expected: 0      0
1      0
2      0
3      0
4      0
..
75     9
76     9
77     9
78     9
79     9
Name: Class, Length: 80, dtype: int64
```

/[AI]/[AI]/



**TERIMA  
KASIH**

ARTI  
CIAL  
INTE  
IGEN  
[AI]