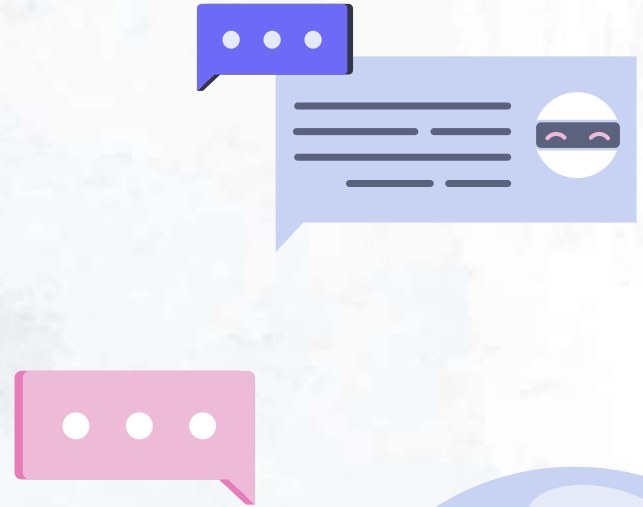


Jaringan Syaraf Tiruan

(AI)



Assignment #14

- 01 → Lakukan Single Perceptron untuk data AND, OR, XOR
- 02 → Lakukan analisa mengapa pada XOR tidak bisa memperoleh grade 1.0 (akurasi 100%)
- 03 → Lakukan Multilayer Perceptron untuk XOR dengan parameter yang tepat dan tampilkan akurasi
- 04 → Lakukan split untuk data heart.csv dengan metode hold-out
- 05 → Lakukan training data dan pilih parameter yang tepat dengan multilayer perceptron
- 06 → Lakukan testing dan tampilkan akurasi

01 →

**Lakukan Single Perceptron untuk data
AND, OR, XOR**

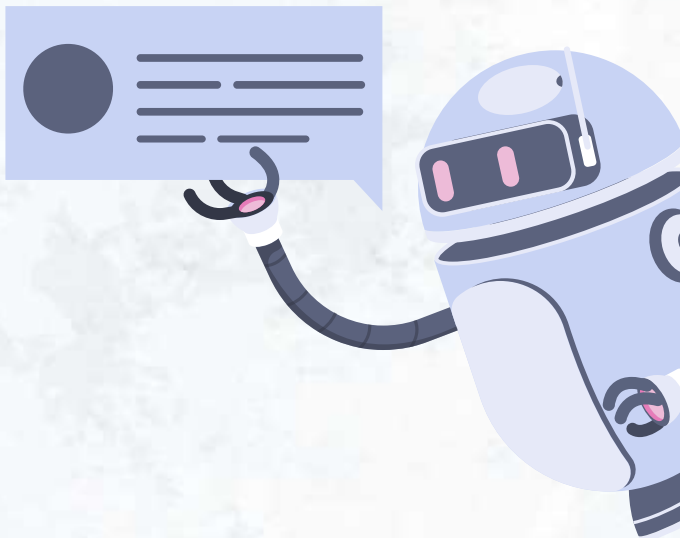
✓
0d



```
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, random_state=0)
clf.fit(data, label)
print(clf.score(data, label))
print(clf.predict(data))
```

1.0

[0 0 0 1]



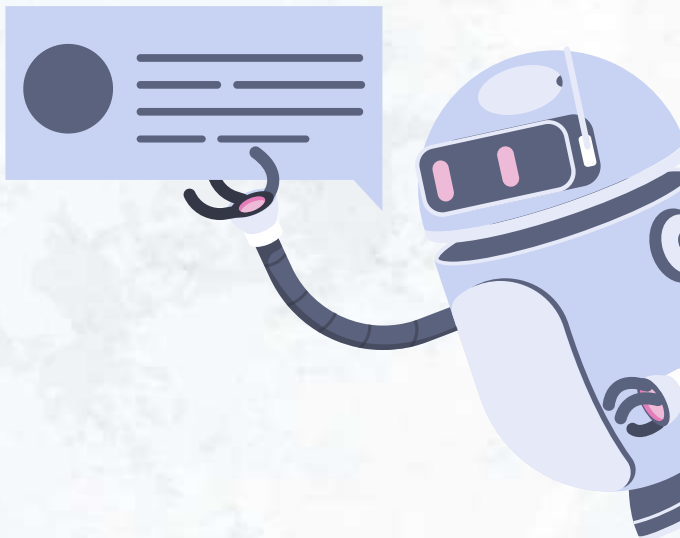
✓
0d



```
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, random_state=0)
clf.fit(data, label)
print(clf.score(data, label))
print(clf.predict(data))
```

1.0

[0 1 1 1]



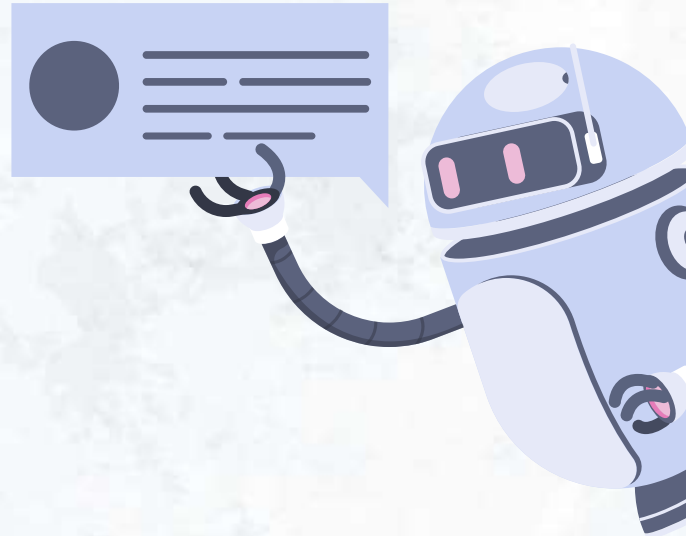
✓
0d



```
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, random_state=0)
clf.fit(data, label)
print(clf.score(data, label))
print(clf.predict(data))
```

0.5

[0 0 0 0]



02 →

**Lakukan analisa mengapa pada XOR
tidak bisa memperoleh grade 1.0
(akurasi 100%)**

Pada XOR, tidak mungkin memperoleh akurasi 100% atau grade 1.0 dengan menggunakan single perceptron. Hal ini disebabkan oleh sifat XOR yang tidak dapat dipisahkan secara linier oleh single perceptron. XOR dapat dipecahkan dengan menggunakan jaringan saraf tiruan yang lebih kompleks, seperti multi-layer perceptron (MLP) dengan setidaknya satu lapisan tersembunyi (hidden layer). Lapisan tersembunyi dalam MLP memungkinkan model untuk mempelajari hubungan non-linier yang diperlukan untuk memodelkan XOR dengan akurasi yang tinggi



03 →

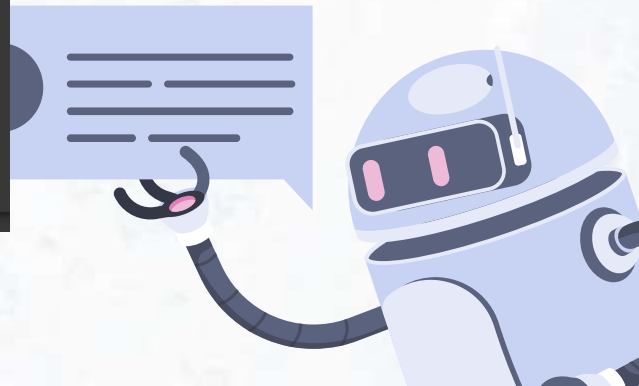
Lakukan Multilayer Perceptron untuk XOR dengan parameter yang tepat dan tampilkan akurasi

✓
0d



```
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=(4), max_iter=500,
             activation = 'relu', learning_rate_init=0.1, solver='sgd')
model.fit(data,label)
print('score:', model.score(data,label))
print('predictions:', model.predict(data))
print('expected:',label)
```

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



04 →

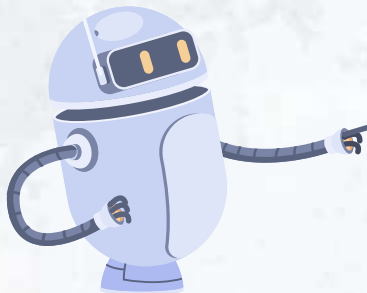
Lakukan split untuk data heart.csv
dengan metode Hold-out

✓
0d



```
import pandas as pd
from sklearn.model_selection import train_test_split

dataset = pd.read_csv('heart.csv')
X = dataset.drop('trestbps', axis=1)
y = dataset['trestbps']
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.30, random_state=100)
```



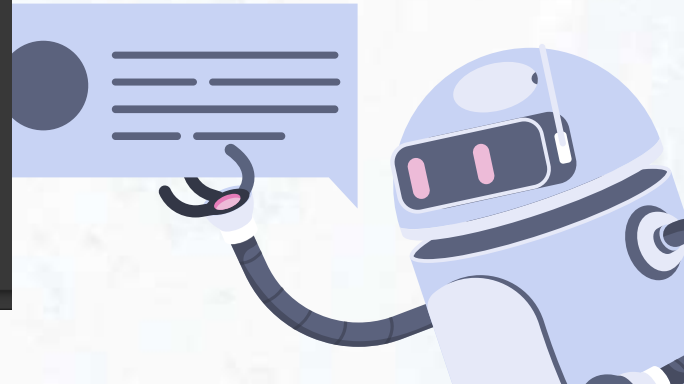
05 →

Lakukan Multilayer Perceptron untuk XOR dengan parameter yang tepat dan tampilkan akurasi



```
dataset = X_train
data = dataset.iloc[:,0:-1]
label = dataset.iloc[:,1]
model = MLP(hidden_layer_sizes=(4), max_iter=1000, activation='relu',
             learning_rate_init=0.1, solver='sgd')
model.fit(data,label)

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

[illegible]

06 →

**Lakukan testing untuk data testing
dan tampilkan akurasi**


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

[illegible]

300 0

220 0

134 1

7 1

203 0

237 0

29 1

161 1

266 0

```
Name: target, Length: 91, dtype: int64
```

