

Nama : Raihana Fawaz

NIM : 1103210102

Kelas : TK-45-05

WEEK 10 MACHINE LEARNING

MLP CLASSIFICATION

Loading dataset

```
import pandas as pd
import torch
import torch.nn as nn
import torch.optim as optim
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
```

Menampilkan 5 baris dataset

```
# Memuat dataset
df = pd.read_csv('ObesityDataSet_raw_and_data_synthetic.csv')
df.head()
```

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS
0	Female	21.0	1.62	64.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	0.0	1.0	no	Public_Transportation
1	Female	21.0	1.52	56.0	yes	no	3.0	3.0	Sometimes	yes	3.0	yes	3.0	0.0	Sometimes	Public_Transportation
2	Male	23.0	1.80	77.0	yes	no	2.0	3.0	Sometimes	no	2.0	no	2.0	1.0	Frequently	Public_Transportation
3	Male	27.0	1.80	87.0	no	no	3.0	3.0	Sometimes	no	2.0	no	2.0	0.0	Frequently	Walking
4	Male	22.0	1.78	89.8	no	no	2.0	1.0	Sometimes	no	2.0	no	0.0	0.0	Sometimes	Public_Transportation

Menampilkan informasi pada dataset

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2111 entries, 0 to 2110
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype
---  -
 0   Gender                                  2111 non-null   object
 1   Age                                     2111 non-null   float64
 2   Height                                 2111 non-null   float64
 3   Weight                                 2111 non-null   float64
 4   family_history_with_overweight         2111 non-null   object
 5   FAVC                                    2111 non-null   object
 6   FCVC                                    2111 non-null   float64
 7   NCP                                     2111 non-null   float64
 8   CAEC                                    2111 non-null   object
 9   SMOKE                                  2111 non-null   object
10  CH2O                                    2111 non-null   float64
11  SCC                                     2111 non-null   object
12  FAF                                     2111 non-null   float64
13  TUE                                     2111 non-null   float64
14  CALC                                    2111 non-null   object
15  MTRANS                                  2111 non-null   object
16  NObeyesdad                             2111 non-null   object
dtypes: float64(8), object(9)
```

Menampilkan kolom pada dataset

```
df.columns

Index(['Gender', 'Age', 'Height', 'Weight', 'family_history_with_overweight',
       'FAVC', 'FCVC', 'NCP', 'CAEC', 'SMOKE', 'CH2O', 'SCC', 'FAF', 'TUE',
       'CALC', 'MTRANS', 'NObeyesdad'],
      dtype='object')
```

Melakukan encoding pada kolom kategorikal menggunakan LabelEncoder

```
# Melakukan encoding pada kolom kategorikal menggunakan LabelEncoder
label_cols = ['Gender', 'family_history_with_overweight', 'FAVC', 'CAEC', 'SMOKE', 'SCC', 'CALC', 'MTRANS', 'NObeyesdad']
label_encoder = LabelEncoder()

for col in label_cols:
    df[col] = label_encoder.fit_transform(df[col])
```

Menentukan fitur dan target untuk split data training dan data testing

```
# Menentukan fitur (X) dan target (y)
X = df.drop(['NObeyesdad'], axis=1) # Menghapus kolom target
y = df['NObeyesdad'] # Kolom target

# Split data menjadi training dan testing (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[+ Code](#)[+ Text](#)

```
[ ] # Standardisasi fitur numerik
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
```

Membuat standard scaler

```
[ ] # Konversi data ke tensor
    X_train_tensor = torch.tensor(X_train_scaled, dtype=torch.float32)
    y_train_tensor = torch.tensor(y_train.values, dtype=torch.float32)
    X_test_tensor = torch.tensor(X_test_scaled, dtype=torch.float32)
    y_test_tensor = torch.tensor(y_test.values, dtype=torch.float32)
```

Grafik di atas menunjukkan hubungan antara hyperparameter yang diuji (jumlah lapisan tersembunyi, jumlah neuron, fungsi aktivasi, jumlah epoch, learning rate, dan ukuran batch) dengan rata-rata Mean Absolute Error (MAE). MAE digunakan untuk mengevaluasi performa model, dimana semakin kecil nilai MAE, semakin baik model dalam memprediksi hasil. Grafik ini memberikan gambaran bahwa kombinasi tertentu dari hyperparameter menghasilkan MAE yang lebih rendah, mengindikasikan model dengan konfigurasi tersebut lebih efektif dalam melakukan prediksi dibandingkan dengan konfigurasi lainnya.

```
# Menyusun model MLP untuk regresi
class MLPRegression(nn.Module):
    def __init__(self, input_size, hidden_layers, neurons, activation):
        super(MLPRegression, self).__init__()
        self.input_size = input_size
        self.hidden_layers = hidden_layers
        self.neurons = neurons
        self.activation = activation

        # Membuat layer input ke layer tersembunyi
        layers = []
        layers.append(nn.Linear(self.input_size, self.neurons))

        # Menambahkan hidden layers
        for _ in range(self.hidden_layers - 1):
            layers.append(self.activation())
            layers.append(nn.Linear(self.neurons, self.neurons))

        layers.append(nn.Linear(self.neurons, 1)) # Layer output

        self.model = nn.Sequential(*layers)

    def forward(self, x):
        return self.model(x).squeeze()
```

Grafik ini menggambarkan pengaruh berbagai kombinasi hyperparameter terhadap Mean Squared Error (MSE) dalam model MLP (Multi-Layer Perceptron). MSE menunjukkan seberapa jauh prediksi model dari nilai sebenarnya, dengan nilai MSE yang lebih rendah menunjukkan kinerja model yang lebih baik. Dari grafik ini, terlihat bahwa beberapa kombinasi hyperparameter,

seperti jumlah lapisan tersembunyi, jumlah neuron, fungsi aktivasi, jumlah epoch, learning rate, dan ukuran batch, menghasilkan nilai MSE yang lebih kecil. Hal ini menunjukkan bahwa pengaturan yang tepat dari hyperparameter sangat mempengaruhi akurasi dan kinerja model dalam memprediksi.

```
# Setup perangkat (GPU jika tersedia)
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

# Hyperparameter yang akan diuji
hidden_layers = [1, 2, 3]
neurons = [4, 8, 16, 32, 64]
activations = [nn.Sigmoid, nn.ReLU, nn.Softmax, nn.Tanh] # Changed nn.ReLU to nn.ReLU
epochs_list = [1, 10, 25, 50, 100, 250]
learning_rates = [10, 1, 0.1, 0.01, 0.001, 0.0001]
batch_sizes = [16, 32, 64, 128, 256, 512]

# Menyimpan hasil
results = []

# Melakukan eksperimen dengan kombinasi hyperparameter
for layers in hidden_layers:
    for neuron in neurons:
        for activation in activations:
            for epochs in epochs_list:
                for lr in learning_rates:
                    for batch_size in batch_sizes:
                        # Membuat dan memindahkan model ke perangkat (GPU atau CPU)
                        model = MLPRegression(input_size=X_train_tensor.shape[1],
```

Kode di atas bertujuan untuk melakukan eksperimen dengan berbagai kombinasi hyperparameter pada model Multi-Layer Perceptron (MLP) untuk tugas regresi. Hyperparameter yang diuji meliputi jumlah lapisan tersembunyi, jumlah neuron, fungsi aktivasi, jumlah epoch, learning rate, dan ukuran batch. Setiap kombinasi hyperparameter digunakan untuk melatih model, yang kemudian dievaluasi menggunakan metrik seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared, dan akurasi. Hasil eksperimen, yang mencakup metrik kinerja dan kombinasi hyperparameter, disimpan untuk analisis lebih lanjut, dengan tujuan mencari pengaturan terbaik yang memberikan kinerja model optimal.







