

## **SECONDHAND DATASET – WEEK 2**

### **A. Tujuan**

Percobaan ini dilakukan dengan membangun model Multi-Layer Perceptron (MLP) menggunakan dua framework deep learning populer: PyTorch dan TensorFlow. Model ini akan digunakan untuk memprediksi harga mobil berdasarkan berbagai fitur seperti usia kendaraan, jarak tempuh, kondisi, ekonomi bahan bakar, kecepatan maksimal, tenaga (HP), dan torsi.

### **B. Dataset**

Dataset yang digunakan berisi informasi tentang harga mobil saat ini dan sebelumnya, serta berbagai fitur yang dapat memengaruhi harga mobil. Berikut adalah beberapa kolom dalam dataset:

- on road old: Harga lama saat pertama kali di jalan.
- on road now: Harga mobil saat ini.
- years: Usia kendaraan dalam tahun.
- km: Jarak tempuh dalam kilometer.
- rating: Peringkat kendaraan.
- condition: Kondisi kendaraan.
- economy: Efisiensi bahan bakar.
- top speed: Kecepatan maksimum kendaraan.
- hp: Tenaga (horsepower) kendaraan.
- torque: Torsi kendaraan.
- current price: Harga saat ini (target prediksi).

### **C. Arsitektur Model**

Model yang digunakan adalah Multi-Layer Perceptron (MLP) yang terdiri dari:

- Input Layer: Jumlah neuron sesuai dengan jumlah fitur dalam dataset.
- Hidden Layers: Dua hingga tiga lapisan tersembunyi dengan fungsi aktivasi ReLU.
- Output Layer: Satu neuron dengan aktivasi linear untuk regresi.

### **D. Implementasi PyTorch**

PyTorch digunakan untuk membangun model dengan pendekatan imperative programming yang fleksibel.

- Preprocessing Data: Dataset dinormalisasi menggunakan StandarScaler dari scikit-learn.
- Membangun Model: Model dibuat menggunakan torch.nn.Sequential.
- Loss Function: Mean Squared Error Loss (MSE) digunakan.
- Optimizer: Adam Optimizer digunakan untuk pembelajaran model.
- Training: Model dilatih dengan loop iteratif menggunakan backpropagation dan optimizer.

### **E. Implementasi Tensor Flow**

TensorFlow digunakan untuk membangun model dengan pendekatan deklaratif menggunakan Keras API.

- Preprocessing Data: Dataset dinormalisasi menggunakan TensorFlow Data API.
- Membangun Model: Model dibuat menggunakan tf.keras.Sequential.
- Loss Function: Mean Squared Error Loss (MSE) digunakan.
- Optimizer: Adam Optimizer digunakan untuk pembelajaran model.
- Training: Model dilatih dengan metode .fit() pada data latih.

### **F. Hasil Training**

Setelah melatih model dengan berbagai konfigurasi, diperoleh hasil evaluasi sebagai berikut:

Model	MSE	RMSE	R <sup>2</sup>
PyTorch	113842823168,0	337406,021	-5,771
TensorFlow	312656288,0	17682,089	0,9814

Hasil Eksperimen dengan PyTorch:

- Loss sangat besar dan tidak menunjukkan penurunan yang signifikan.
- MSE dan RMSE sangat tinggi.
- R<sup>2</sup> negatif, menunjukkan model gagal menjelaskan varians dalam data.

Hasil Eksperimen dengan TensorFlow:

- Loss menunjukkan penurunan signifikan selama epoch awal.
- Loss akhirnya berkisar di ratusan juta, jauh lebih baik dibandingkan PyTorch.
- Model tampaknya mulai konvergen setelah beberapa epoch.

#### G. Kesimpulan

Dari eksperimen ini, dapat disimpulkan bahwa model yang dilatih dengan TensorFlow menunjukkan kinerja yang jauh lebih baik dibandingkan dengan PyTorch. Model PyTorch mengalami kesulitan dalam menurunkan loss, sementara model TensorFlow berhasil mencapai konvergensi dengan hasil yang lebih baik. Kemungkinan besar ada kesalahan implementasi atau pengaturan pada model PyTorch yang menyebabkan hasilnya buruk.

#### H. Link Google Colab

[https://colab.research.google.com/drive/1BqI5tKuhZG\\_LEr1eDm5N\\_8Xf\\_1srJc0?usp=sharing](https://colab.research.google.com/drive/1BqI5tKuhZG_LEr1eDm5N_8Xf_1srJc0?usp=sharing)