Ade Tirta Rahmat Hidayat – 1103203212 – Deep Learning TK 45

INCOME CLASSIFICATION

A. Tujuan

Pendapatan seseorang sering kali dipengaruhi oleh berbagai faktor seperti tingkat pendidikan, jenis pekerjaan, dan status perkawinan. Dalam proyek ini, dikembangkan model deep learning untuk mengklasifikasikan apakah seseorang memiliki pendapatan di atas atau di bawah ambang batas tertentu berdasarkan fitur-fitur yang tersedia dalam dataset.

B. Preprocessing Data

Langkah ini dilakukan untuk mempersiapkan dataset:

1. Pembersihan Data:
   - Menghapus data dengan nilai kosong (missing values) atau menggantinya dengan metode imputasi seperti median atau modus.
   - Menghilangkan duplikasi jika ada.
2. Encoding Kategorikal:
   Fitur kategorikal diubah menjadi numerik. Pada kasus ini, fitur income diubah menjadi <=50K: 0, >50K: 1.
3. Normalisasi dan Standarisasi:
   - Normalisasi dilakukan menggunakan MinMaxScaler agar nilai fitur berada dalam rentang [0,1].
   - Standarisasi dilakukan dengan StandardScaler agar distribusi fitur memiliki nilai rata-rata 0 dan standar deviasi 1.
4. Split Data
   Dataset dibagi menjadi data latih (80%) dan data uji (20%) untuk melatih dan menguji model.

C. Arsitektur Model

Model deep learning yang dikembangkan memiliki arsitektur sebagai berikut:

- Input Layer:
  Jumlah neuron sesuai dengan jumlah fitur setelah preprocessing.
- Hidden Layers:
  - Beberapa lapisan fully connected (dense layers) dengan jumlah neuron yang bervariasi.
  - Menggunakan fungsi aktivasi ReLU untuk menangani non-linearitas dalam data.
  - Menambahkan dropout untuk mengurangi overfitting.
- Output Layer:
  Menggunakan satu neuron dengan fungsi aktivasi sigmoid, karena klasifikasi bersifat biner.
- Optimasi dan Loss Function:
  - Optimizer: Adam dengan learning rate yang disesuaikan.
  - Loss Function: Binary Crossentropy, karena tugas klasifikasi biner.
  - Batch Size: 32 atau 64.
  - Epoch: 50 hingga 100, dengan early stopping untuk menghindari overfitting.

D. Evaluasi Model

Untuk menilai performa model, digunakan metrik evaluasi sebagai berikut:

- Akurasi: Persentase prediksi yang benar dibandingkan total data uji.

- Presisi & Recall: Presisi menunjukkan seberapa tepat model dalam memprediksi kelas positif, sementara recall menunjukkan seberapa banyak kelas positif yang berhasil diprediksi dengan benar.
- F1-score: Rata-rata harmonik dari presisi dan recall, memberikan gambaran keseimbangan antara keduanya.
- AUC & ROC Curve: Digunakan untuk melihat sejauh mana model dapat membedakan antara dua kelas

E. Hasil Training

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

Model Klasifikasi	**Akurasi**	**Presisi**	**Recall**	**F1-Score**	**AUC-ROC**
PyTorch	83,88%	71,60%	57,71%	63,91%	75,09%
TensorFlow	82,79%	66,89%	60,21%	63,37%	75,21%

F. Kesimpulan

Dari hasil evaluasi di atas, dapat disimpulkan bahwa model PyTorch memiliki performa terbaik berdasarkan nilai F1-score (63.91%) dibandingkan model TensorFlow (63.37%). F1-score lebih dipertimbangkan dalam kasus ini karena mencerminkan keseimbangan antara presisi dan recall.

Namun, model TensorFlow memiliki nilai AUC-ROC sedikit lebih tinggi (75.21%) dibandingkan model PyTorch (75.09%), yang menunjukkan bahwa model TensorFlow sedikit lebih baik dalam membedakan antara dua kelas.

G. Link Google Colab

https://colab.research.google.com/drive/1GWY1AvMX_0Kj89sTpWWwXaxU1Wjt_fAs?usp=sharing