

Analisis Hasil Dataset RegresiUTSTelkom dengan Model PyTorch

A. MLP Regression PyTorch

1. Analisis Training

- Terjadi penurunan loss yang signifikan dari Epoch 1 hingga 30
- Loss awal: ~3 juta → Akhir: ~25 ribu → Training convergence terlihat baik

2. Evaluasi Matriks

- MSE: 289.54
- RMSE: 19.74
- R^2 Score: -2.26 (negatif menandakan model *sangat buruk* dibanding model rata-rata)

3. Catatan

Meski training loss menurun, model overfitting atau gagal generalisasi, terbukti dari R^2 negatif.

B. MLP Classification PyTorch

1. Analisis Training

- Loss konsisten menurun namun lambat dan stagnan di akhir.
- Dari 3.24 → 3.08 (cukup kecil perubahannya).

2. Evaluasi Matriks

- Accuracy: 10.26%
- Precision: 5.86%
- Recall: 3.64%
- F1-Score: 3.17%

3. Catatan

- Akurasi sangat rendah, mendekati tebak acak (jika banyak kelas).
- Terdapat peringatan dari sklearn: banyak label tidak terprediksi sama sekali.
- Menunjukkan underfitting parah, kemungkinan disebabkan model tidak cukup kompleks, data imbalance, atau preprocessing tidak optimal.

C. CNN Regression PyTorch

1. Analisis Training

- Loss stagnan dari Epoch 2 hingga 10 (tidak banyak perubahan setelah epoch awal).

- Penurunan hanya sedikit dari 62.3k ke 61.8k → Model tidak belajar optimal.
2. Evaluasi Matriks
 - MSE: 1558.97
 - RMSE: 39.48
 - R^2 Score: -12.05 → Sangat buruk, bahkan lebih buruk dari MLP Regressor.
 3. Catatan
CNN kurang cocok untuk data non-visual seperti dataset tabular, kemungkinan besar penyebab performa buruk.

D. Kesimpulan

Model	RMSE	R^2 Score	Catatan
MLP Regressor	19.74	-2.26	Cukup baik dalam training, namun overfitting
MLP Classifier	-	-	Akurasi dan F1-Score rendah
CNN Regressor	39.48	-12.05	Jauh lebih buruk dari MLP

Model Terbaik adalah MLP Regression, karena:

- Meskipun R^2 masih negatif (artinya performa di test set buruk), MLP Regression **lebih stabil dan menunjukkan proses training yang baik**.
- RMSE-nya juga paling kecil, menunjukkan prediksi numerik lebih dekat ke target.
- CNN tidak cocok untuk data tabular, dan model klasifikasi saat ini belum mampu mengenali pola signifikan.

Rekomendasi Perbaikan:

1. Untuk Regression:
 - Tambahkan regularisasi (dropout/L2) untuk mengurangi overfitting.
 - Uji arsitektur model yang lebih dalam atau lebih kompleks.
 - Normalisasi target/label (min-max atau standar).
 - Hyperparameter tuning (learning rate, batch size).
2. Untuk Classification:
 - Gunakan metode balancing (misalnya: SMOTE, class weight).
 - Coba model lebih kompleks (MLP lebih dalam).
 - Gunakan feature selection/reduksi dimensi.
 - Evaluasi distribusi label target (cek apakah imbalance).
3. Untuk CNN:
 - Hanya gunakan jika data memiliki struktur spasial (misal: citra).
 - Jika tetap ingin pakai CNN, ubah format data (misalnya reshape menjadi gambar 2D logis jika memungkinkan).

