Nama: FARHAN ALJIHADILLAH

KELAS: 05TPLE017

MATKUL: MACHINE LEARNING

PERTEMUAN PEKAN 7: Artificial Neural Network



1. Kode ini digunakan untuk menyiapkan data sebelum pelatihan model machine learning. Data dibaca dari file CSV, kolom target "Lulus" dipisahkan, lalu fitur dinormalisasi dengan StandardScaler. Setelah itu, data dibagi menjadi tiga bagian train (60%), validation (20%), dan test (20%)

 Kode ini membuat model neural network dengan TensorFlow Keras untuk klasifikasi biner. Model memiliki dua lapisan tersembunyi dengan aktivasi ReLU, satu lapisan output dengan sigmoid, serta Dropout untuk mencegah overfitting. Model dikompilasi dengan optimizer Adam, loss binary\_crossentropy, dan metrik accuracy serta AUC. Perintah model.summary() menampilkan struktur dan jumlah parameternya

```
import tensorflow as tf
from tensorflow.keras import layers

model = keras.Sequential([
    layers.Input(shape=(X_train.shape[1],)),
    layers.Dense(32, activation="relu"),
    layers.Dropout(0.3),
    layers.Dense(16, activation="relu"),
    layers.Dense(1, activation="sigmoid") # klasifikasi biner
])

model.compile(optimizer=keras.optimizers.Adam(1e-3),
    loss="binary_crossentropy",
    metrics=["accuracy","AUC"])
model.summary()
```

Dan hasilnya seperti dibawah ini:

3. Kode ini melatih model dengan **early stopping**, yang menghentikan training jika **val\_loss** tidak membaik selama 10 epoch dan mengembalikan bobot terbaik. Model dilatih hingga 100 epoch dengan batch size 32 menggunakan data latih dan validasi

```
es = keras.callbacks.EarlyStopping(
    monitor="val_loss", patience=10, restore_best_weights=True
)

history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100, batch_size=32,
    callbacks=[es], verbose=1
)
```

Dan hasilnya seperti dibawah ini:

```
− 3s 3s/step - AUC: 0.2222 - accuracy: 0.1667 - loss: 0.7283 - val AUC: 0.0000e+00 - val accuracy: 0.5000 -
 1/1 -
Epoch 2/100
1/1
Epoch 3/100
1/1
                                                         — 0s 190ms/step - AUC: 0.6667 - accuracy: 0.5000 - loss: 0.6785 - val_AUC: 0.0000e+00 - val_accuracy: 0.5000
 Epoch 4/100
 0s 361ms/step - AUC: 0.5000 - accuracy: 0.5000 - loss: 0.7060 - val_AUC: 1.0000 - val_accuracy: 0.5000 - val_acc
 1/1 -
                                                         — 0s 179ms/step - AUC: 0.7222 - accuracy: 0.6667 - loss: 0.6706 - val_AUC: 1.0000 - val_accuracy: 0.5000 - va
 Epoch 6/100
 1/1 -
                                                          – 0s 190ms/step - AUC: 1.0000 - accuracy: 0.5000 - loss: 0.6578 - val AUC: 1.0000 - val accuracy: 0.5000 - va
Epoch 7/100
                                                          - 0s 182ms/step - AUC: 0.7778 - accuracy: 0.8333 - loss: 0.6428 - val AUC: 1.0000 - val accuracy: 0.5000 - v
Epoch 8/100
Epoch 9/100
                                                           - 0s 174ms/step - AUC: 0.7778 - accuracy: 0.5000 - loss: 0.6129 - val_AUC: 1.0000 - val_accuracy: 1.0000 - val_
                                                            0s 167ms/step - AUC: 1.0000 - accuracy: 0.8333 - loss: 0.5826 - val_AUC: 1.0000 - val_accuracy: 1.0000 - val_accuracy
 Epoch 10/100
 Epoch 11/100
                                                         – 0s 320ms/step - AUC: 0.7778 - accuracy: 0.8333 - loss: 0.6623 - val AUC: 1.0000 - val accuracy: 1.0000 - va
 1/1 -
                                                          – 0s 200ms/step - AUC: 0.7778 - accuracy: 0.8333 - loss: 0.6300 - val AUC: 1.0000 - val accuracy: 1.0000 - va
 .-
Epoch 13/100
 Epoch 99/100
 Epoch 100/100
                                                            0s 130ms/step - AUC: 1.0000 - accuracy: 1.0000 - loss: 0.1861 - val_AUC: 1.0000 - val_accuracy: 1.0000 - val
```

4. Kode ini mengevaluasi model pada data uji dengan menghitung loss, akurasi, dan AUC, lalu menghasilkan prediksi biner dari probabilitas. Hasilnya dievaluasi menggunakan confusion matrix dan classification report untuk melihat kinerja model secara detail

```
from sklearn.metrics import classification_report, confusion_matrix

loss, acc, auc = model.evaluate(X_test, y_test, verbose=0)
print("Test Acc:", acc, "AUC:", auc)

y_proba = model.predict(X_test).ravel()
y_pred = (y_proba >= 0.5).astype(int)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred, digits=3))

[5]
```

Dan hasilnya seperti dibawah ini:

```
Test Acc: 1.0 AUC: 1.0
1/1
                          0s 141ms/step
[[1 0]
[0 1]]
              precision
                            recall f1-score
                                                support
           0
                   1.000
                             1.000
                                        1.000
                                                       1
                             1.000
           1
                   1.000
                                        1.000
    accuracy
                                        1.000
   macro avg
                   1.000
                             1.000
                                        1.000
                                                       2
weighted avg
                   1.000
                             1.000
                                        1.000
```

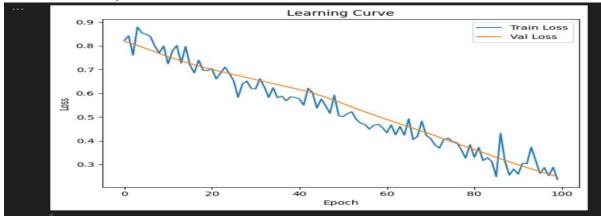
5. Kode ini digunakan untuk menampilkan kurva pembelajaran (learning curve) yang menunjukkan perubahan nilai loss pada data pelatihan dan validasi selama proses training model. Dengan menggunakan matplotlib, grafik dibuat untuk membandingkan Train Loss dan Val Loss terhadap jumlah epoch, membantu melihat apakah model mengalami overfitting atau underfitting. Grafik kemudian diberi label, judul, dan disimpan sebagai file gambar "learning\_curve.png"

6.

```
import matplotlib.pyplot as plt

plt.plot(history.history["loss"], label="Train Loss")
    plt.plot(history.history["val_loss"], label="Val Loss")
    plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.legend()
    plt.title("Learning Curve")
    plt.tight_layout(); plt.savefig("learning_curve.png", dpi=120)
[6]
```

Dan berikut hasil nya



\_\_\_\_\_

## Jika diubah jumlah neuron menjadi 64

