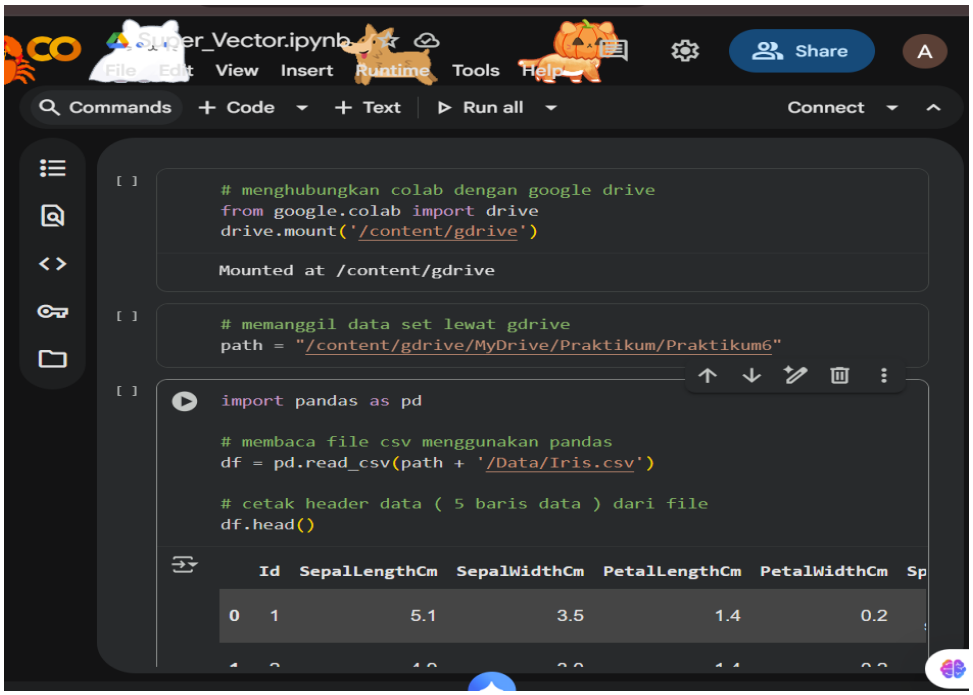
Tugas 6: Tugas Praktikum Mandiri 6 – Machine Learning

Alia Maisyarah 1 - 0110224095 [1*]

[1] Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail:40110224095@student.nurulfikri.ac.id – **email mahasiswa 1**

1. Latihan Mandiri 6



1. # menghubungkan colab dengan google drive
   from google.colab import drive
   drive.mount('/content/gdrive')

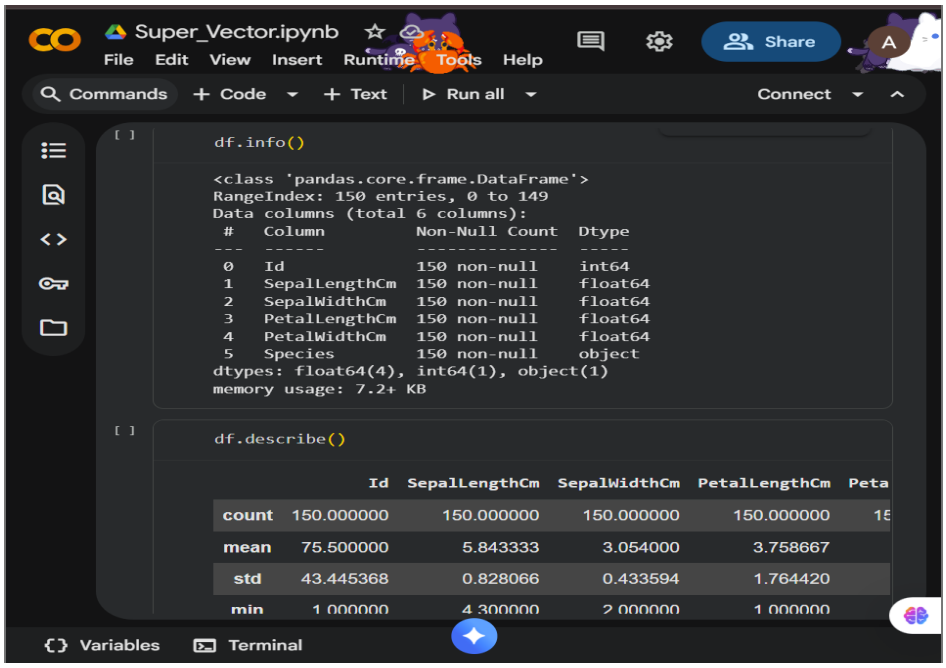
   (Perintah ini digunakan supaya file di Google Drive bisa dipakai di Colab — misalnya buat membaca dataset (day.csv, hour.csv), menyimpan hasil, atau memuat model.)

2. # memanggil data set lewat gdrive
   path = "/content/gdrive/MyDrive/Praktikum/Praktikum4"

   (Kode ini digunakan untuk menunjukkan lokasi dataset di Google Drive, supaya nanti ketika kamu memanggil data (misalnya dengan pd.read_csv(path + '/day.csv')), Python tahu harus mencari file-nya di mana.)

3. Import library
   import pandas as pd
   from sklearn.model_selection import train_test_split
   from sklearn.tree import DecisionTreeClassifier, plot_tree
   from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
   import matplotlib.pyplot as plt

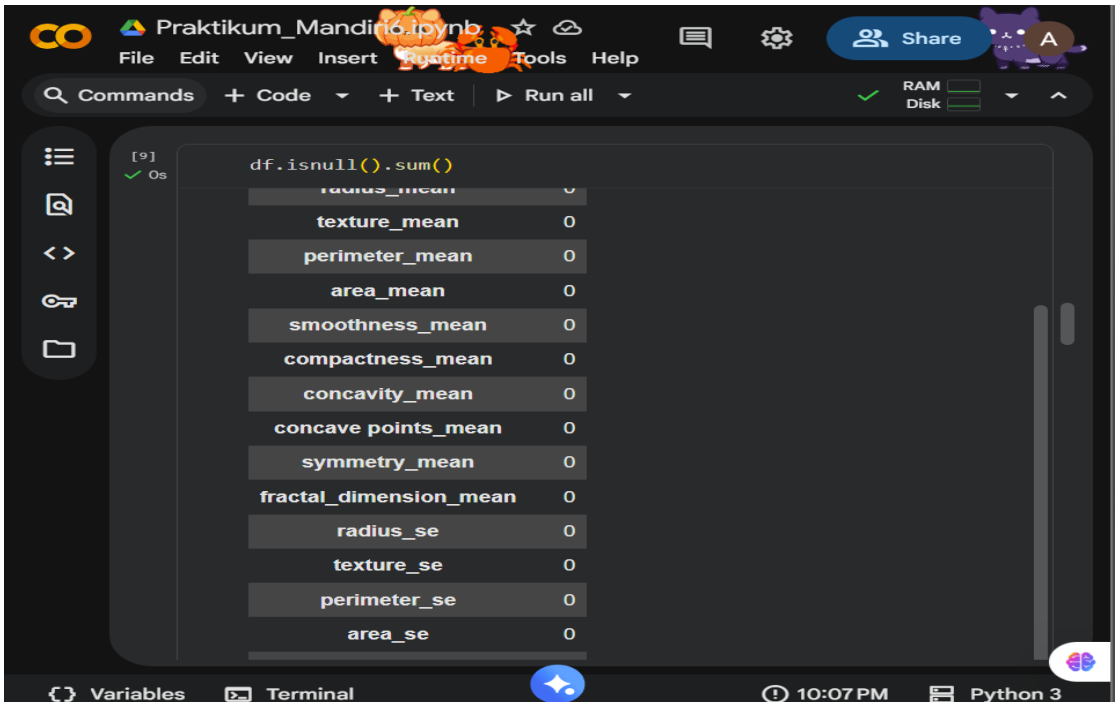
(Kode ini berfungsi untuk menyiapkan semua pustaka yang diperlukan dalam pembuatan, pelatihan, evaluasi, dan visualisasi model *Decision Tree* menggunakan dataset Iris.)



4. df.info()

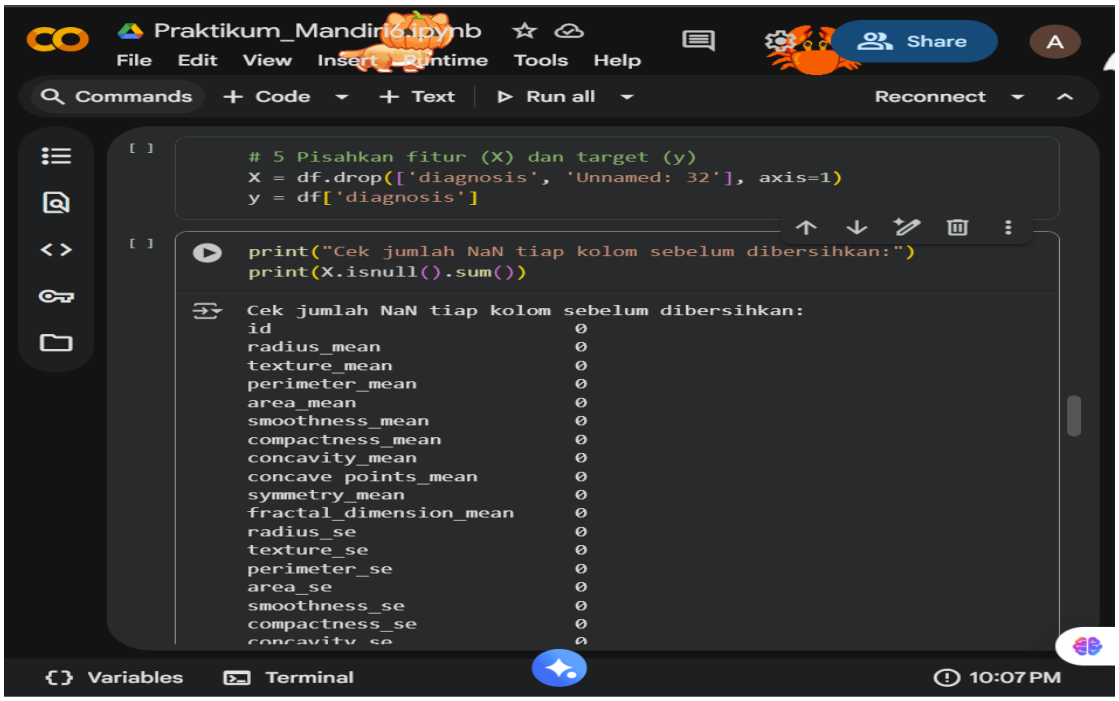
(Dipakai buat melihat struktur data, tipe data, dan ada tidaknya data kosong di dataset kamu.Biasanya ini dijalankan di awal analisis supaya kamu paham dulu bentuk data yang kamu hadapi.)

5. df.describe()
(dipakai untuk melihat ringkasan statistik dari data numerik membantu kamu memahami sebaran data, rata-rata, dan rentang nilainya sebelum analisis lebih lanjut.)



6. df.isnull().sum()

(Artinya semua kolom tidak ada data yang kosong (0 = tidak ada nilai hilang)

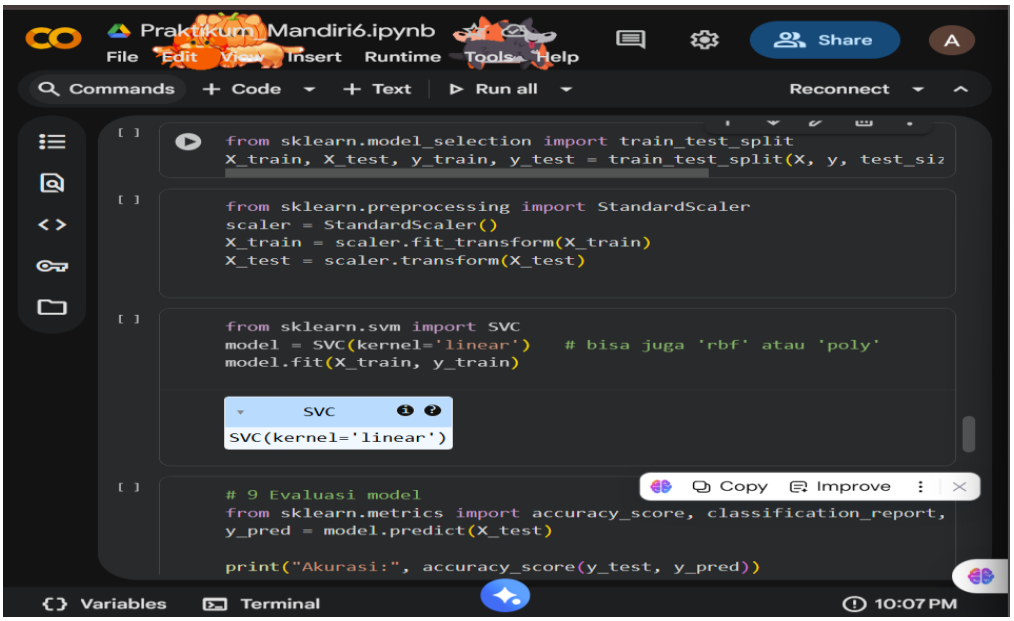


7. Pisahkan fitur (X) dan target (y)
X = df.drop(['diagnosis', 'Unnamed: 32'], axis=1)
y = df['diagnosis']

(Kode ini digunakan untuk memisahkan data fitur (X) dan data target (y) pada dataset. Biasanya dilakukan sebelum pelatihan model Machine Learning.)

8. print("Cek jumlah NaN tiap kolom sebelum dibersihkan:")
print(X.isnull().sum())

(digunakan untuk mengecek apakah ada data kosong (NaN) di setiap kolom pada dataset fitur X.)



9. from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

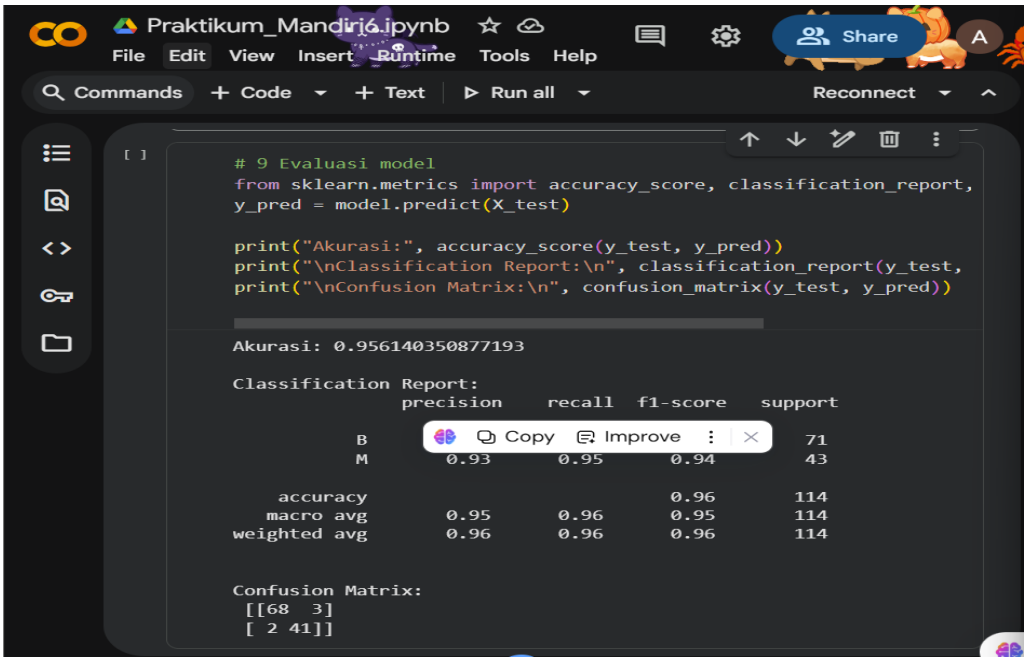
(Kode ini fungsinya untuk memisahkan data menjadi data latih dan data uji sebelum model dilatih.)

10. from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

(Kode ini digunakan untuk menormalkan atau menstandarkan data fitur (X) sebelum dimasukkan ke model.)

11. from sklearn.svm import SVC
model = SVC(kernel='linear') # bisa juga 'rbf' atau 'poly'
model.fit(X_train, y_train)

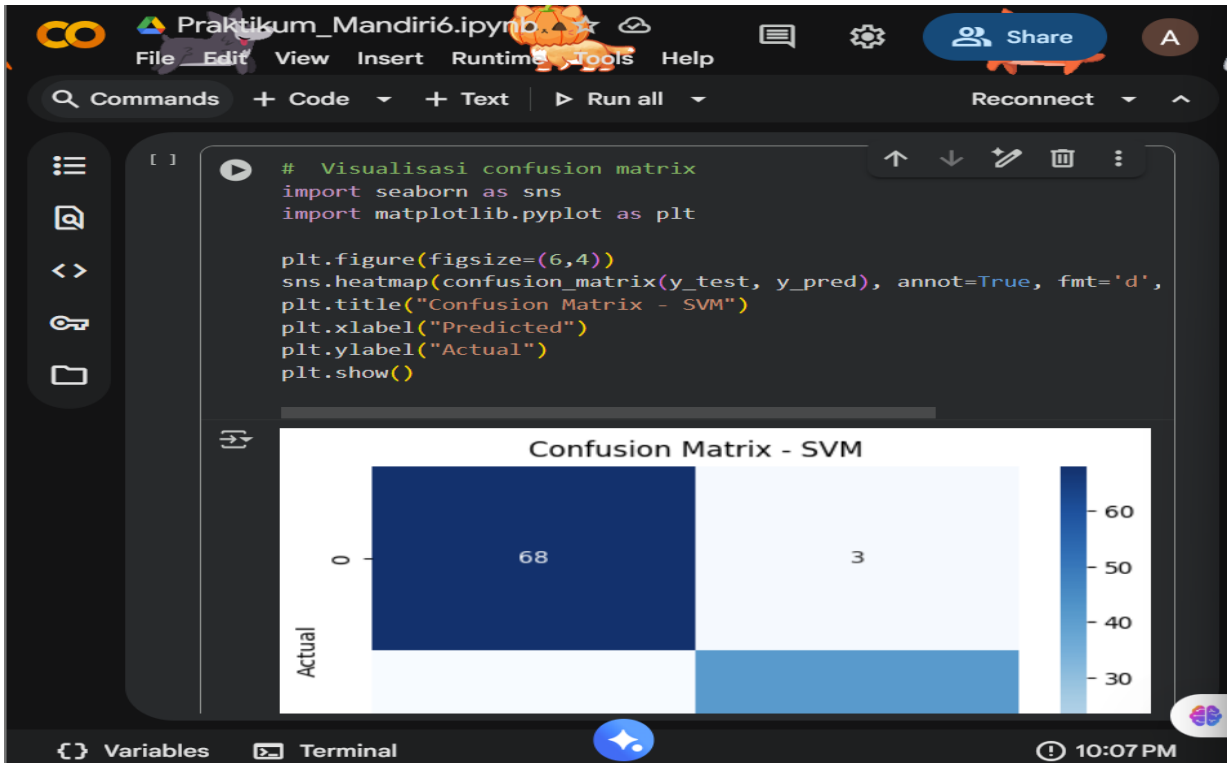
(Kode ini digunakan untuk membuat dan melatih model klasifikasi menggunakan algoritma Support Vector Machine (SVM).



12. 9 Evaluasi model
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
    y_pred = model.predict(X_test)

    print("Akurasi:", accuracy_score(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))
    print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

    (Kode ini digunakan untuk mengevaluasi seberapa baik model SVM kamu bekerja dalam memprediksi data uji (X_test).



13. Visualisasi confusion matrix
    import seaborn as sns
    import matplotlib.pyplot as plt

    plt.figure(figsize=(6,4))
    sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
    plt.title("Confusion Matrix - SVM")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.show()

    (Kode ini digunakan untuk menampilkan hasil evaluasi model dalam bentuk visual (grafik), yaitu confusion matrix.)

LINK GITHUB UPLOAD TUGAS :

https://github.com/AliaMaisyarah14/Praktikum02ML.git