DATA SCIENCE

Klasifikasi Iris Dataset menggunakan
Algoritma Support Vector Machine (SVM)

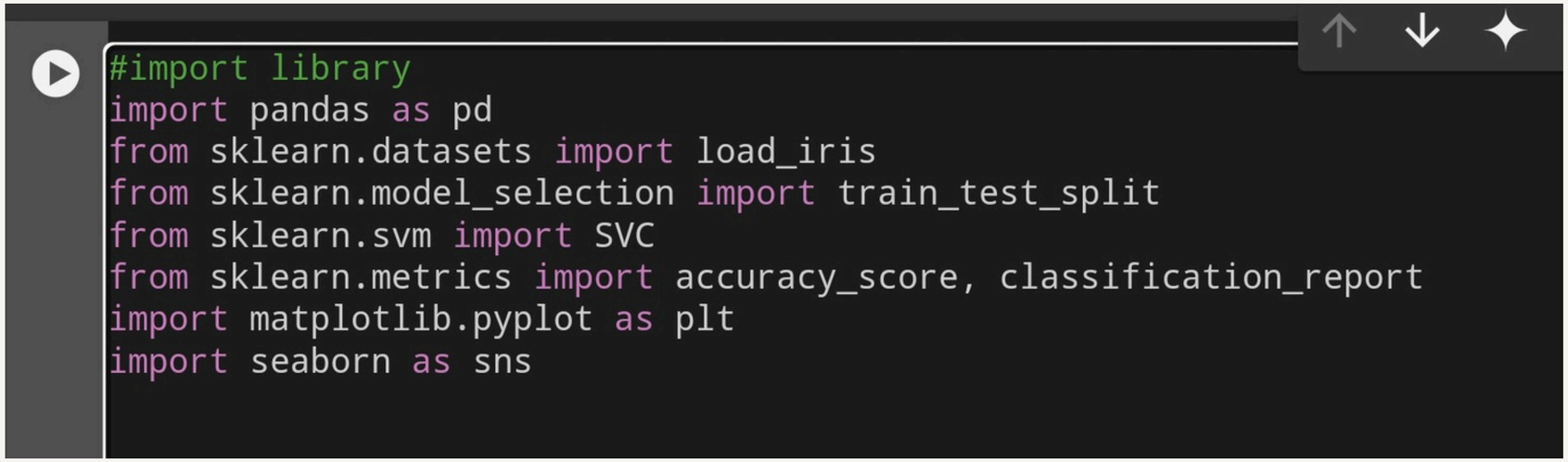
By
FARA FIROZA

DESKRIPSI PROYEK

Proyek ini bertujuan untuk mengklasifikasikan spesies bunga iris berdasarkan fitur-fiturnya (seperti panjang dan lebar sepal/petals). Saya menggunakan dataset Iris dan algoritma SVM untuk memodelkan dan menguji akurasi model.

IMPORT LIBRARY

Untuk memulai proyek ini, beberapa pustaka penting diimpor untuk pengolahan data, pelatihan model, dan evaluasi hasil.



```
#import library
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
```

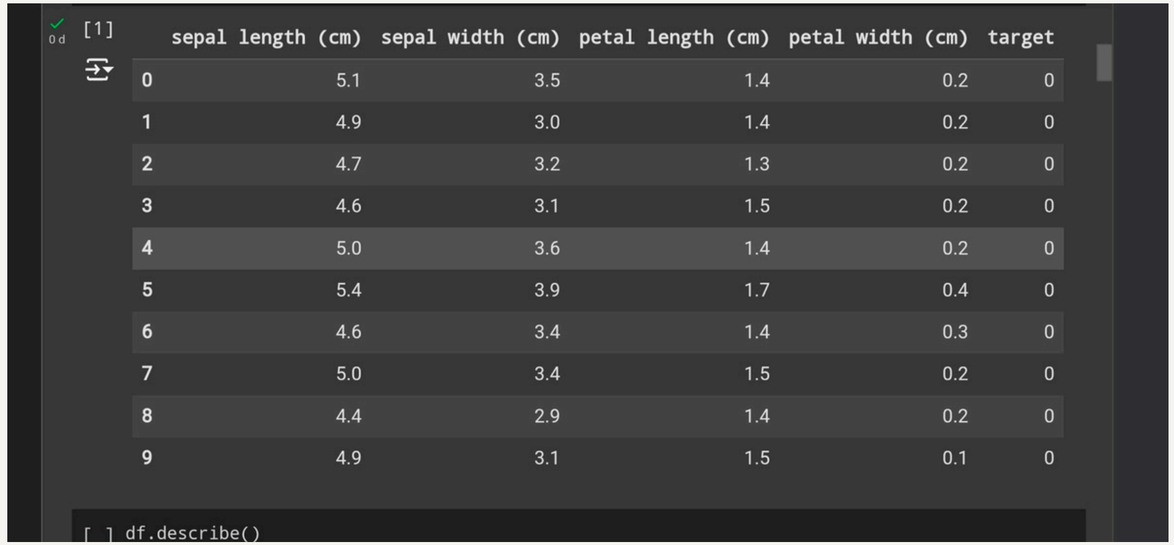
LOAD DATASET

```
#Load dataset and convert to DataFrame
data=load_iris()

#mengubah data Load Iris menjadi dataframe
df=pd.DataFrame(data.data,columns=data.feature_names)

df['target']=data.target

df.head(10)
```



[1]

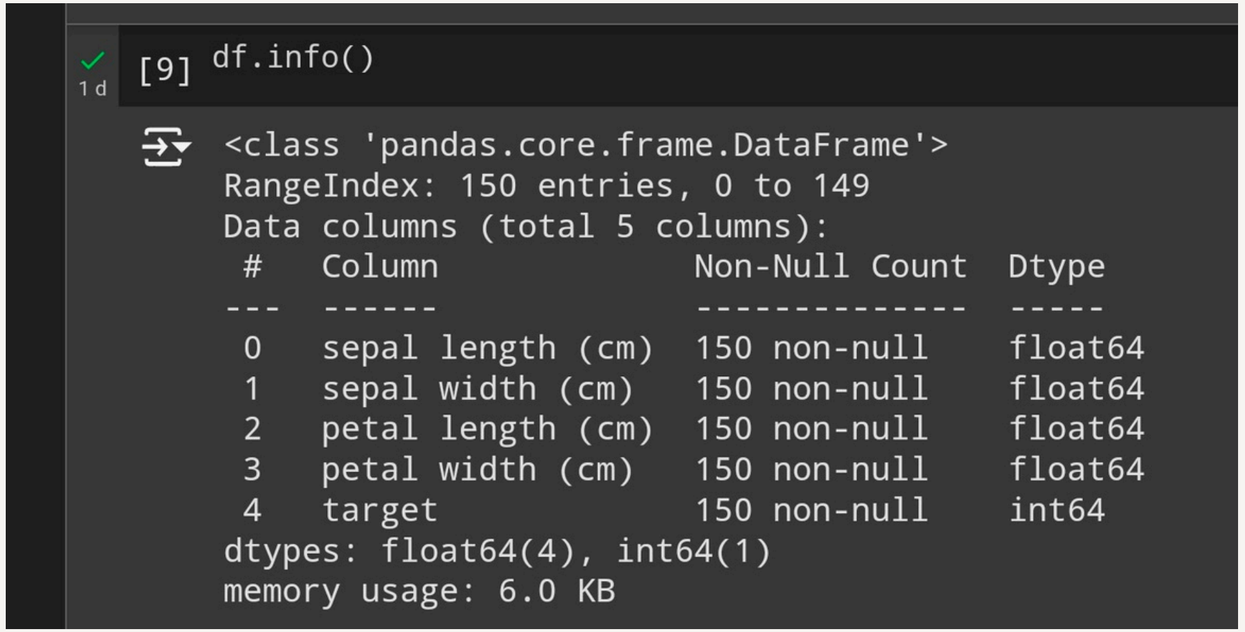
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
5	5.4	3.9	1.7	0.4	0
6	4.6	3.4	1.4	0.3	0
7	5.0	3.4	1.5	0.2	0
8	4.4	2.9	1.4	0.2	0
9	4.9	3.1	1.5	0.1	0

[ ] df.describe()

Dataset Iris dimuat menggunakan pustaka sklearn.datasets. Dataset ini memiliki 150 sampel bunga iris dengan 4 fitur (panjang sepal, lebar sepal, panjang petal, lebar petal) dan label target (spesies iris), namun disini saya hanya memggunakan 10 sampel.

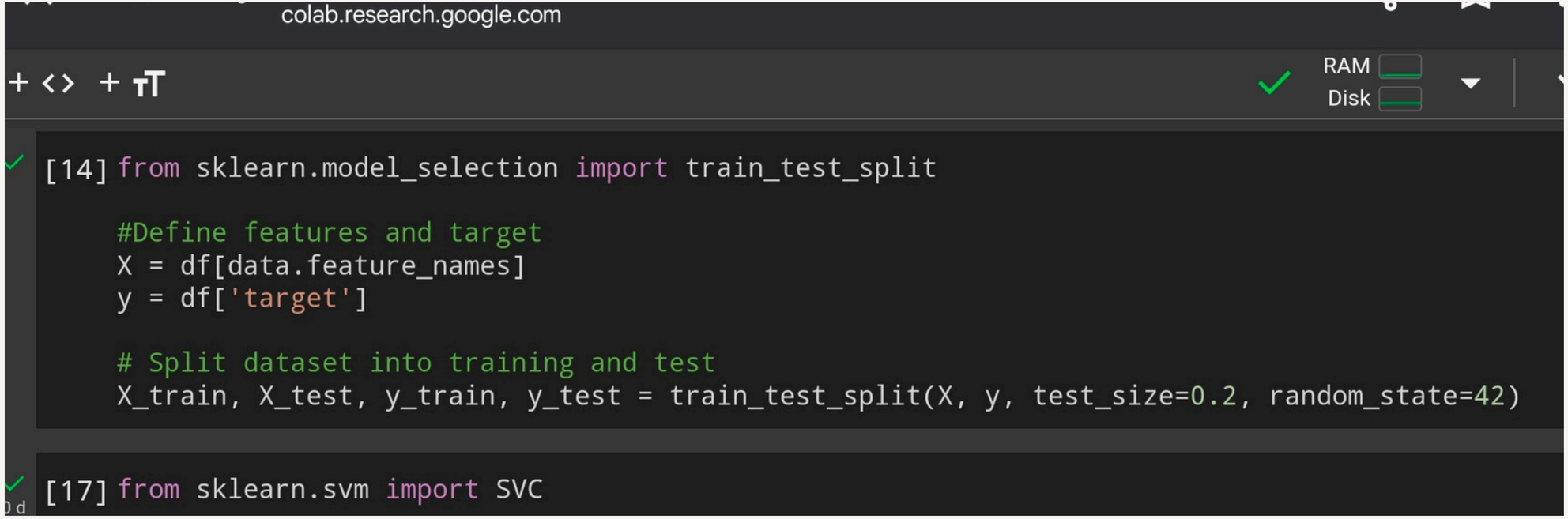
NULL DATA

Memeriksa apakah ada data yang hilang (null) dalam dataset.Jika ada nilai yang hilang, kita akan menangani data tersebut dengan metode yang sesuai (misalnya imputasi atau penghapusan).



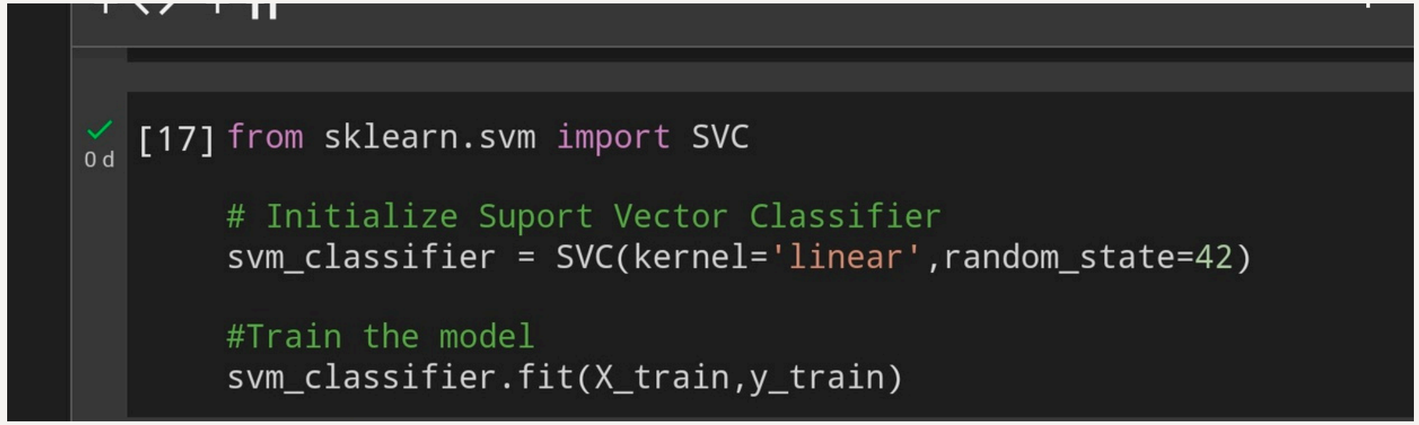
PREPROCESSING DATA

Menyiapkan data untuk pemodelan dengan memisahkan fitur (X) dan target (y).



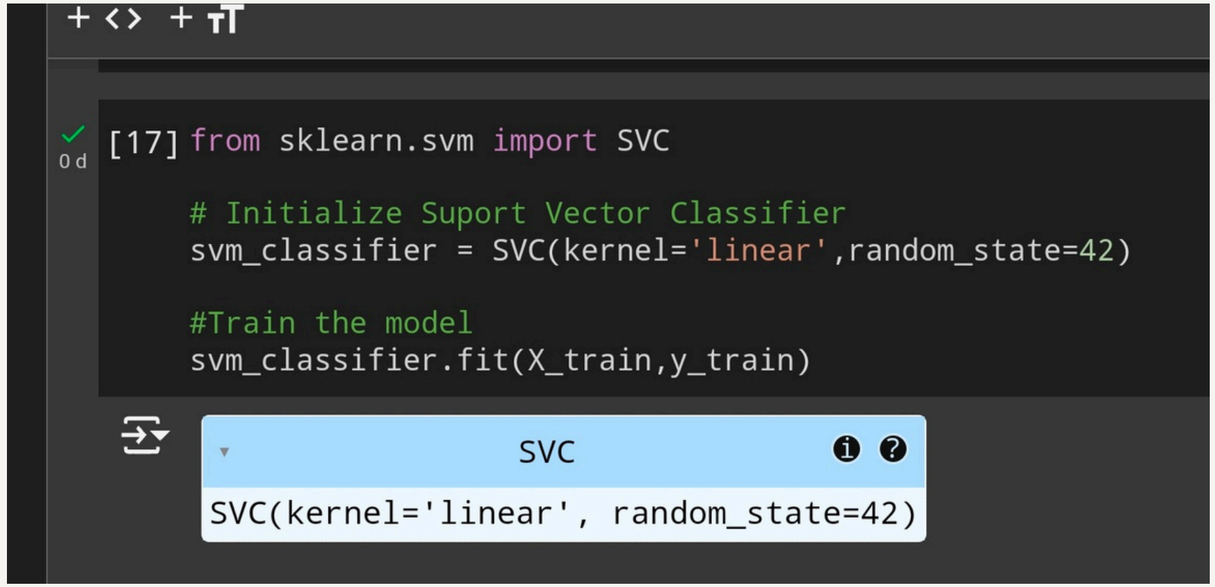
SPLITTING DATA

Memisahkan dataset menjadi data latih dan data uji. Data latih digunakan untuk melatih model, dan data uji digunakan untuk mengevaluasi akurasi model.



MODELING

Membangun model menggunakan Support Vector Machine (SVM) dengan kernel linear.



MODEL AKURASI

- Mengevaluasi model dengan menghitung akurasi menggunakan data uji.
- Akurasi model menunjukkan seberapa baik model dapat mengklasifikasikan data uji

```
#Predict on the test set
y_pred = svm_classifier.predict(X_test)

# Calculate accuracy
accuracy =  accuracy_score(y_test,y_pred)
print(f"Akurasi: {accuracy * 100:.2f}%")

#print classification report for details
print(classification_report(y_test,y_pred,
target_names=data.target_names))
```

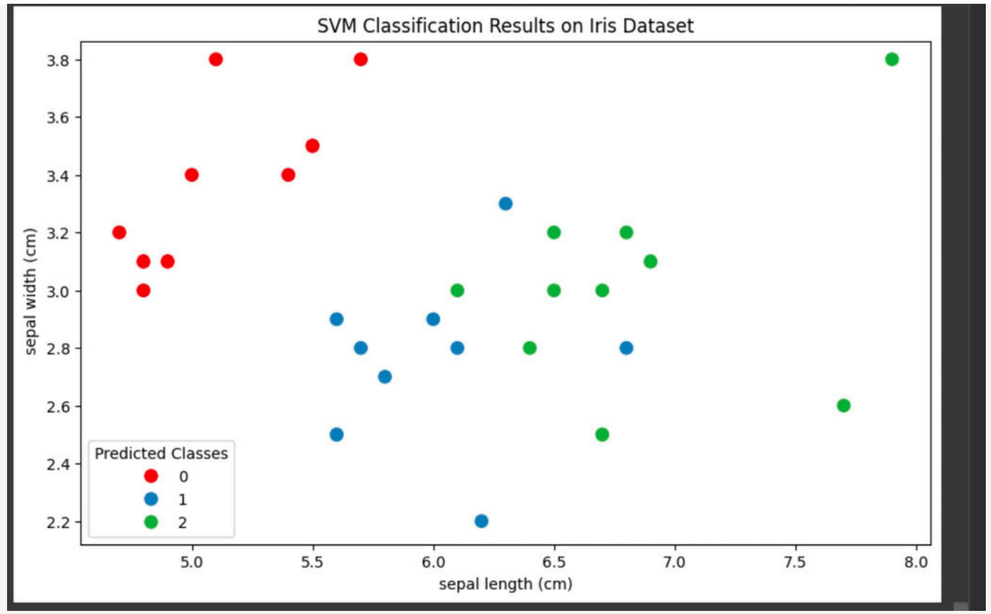
```
[20]
0d
Akurasi: 100.00%
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   virginica       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

VISUALISASI HASIL KLASIFIKASI

```
[21]
# Visualize the results
plt.figure(figsize=(10,6))
sns.scatterplot(x=X_test.iloc[:, 0], y=X_test.iloc[:, 1], hue=y_pred, palette='Set1', s=100
plt.title("SVM Classification Results on Iris Dataset")
plt.xlabel(data.feature_names[0])  # Corrected to use the actual feature name
plt.ylabel(data.feature_names[1])  # Corrected to use the actual feature name
plt.legend(title='Predicted Classes')  # Corrected 'tittle' to 'title'
plt.show()
```



Scatterplot menunjukkan hasil klasifikasi dataset Iris menggunakan algoritma SVM.

Setiap titik mewakili sampel bunga pada data uji.

Warna menunjukkan kelas bunga (Setosa, Versicolor, Virginica).

Sumbu X dan Y merepresentasikan fitur utama dataset, seperti panjang dan lebar sepal.

KESIMPULAN

- Model SVM dengan kernel linear berhasil mengklasifikasikan data Iris dengan akurasi tinggi
- Hasil klasifikasi menunjukkan pemisahan yang jelas antara spesies bunga iris.
- Proyek ini dapat diperluas dengan mencoba kernel SVM lainnya atau algoritma lain untuk membandingkan performanya.

Terima kasih