Tugas 6: Laporan Praktikum Mandiri Heart Desease Dataset

Rian Satria Permana - 0110222064 [1]

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

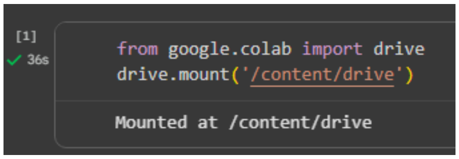
*E-mail: riansatria882@gmail.com

1. Tujuan

Memprediksi apakah seorang pasien menderita **diabetes (1)** atau **tidak (0)** berdasarkan data medis seperti umur, BMI, tekanan darah, glukosa, dan lainnya — menggunakan algoritma **Support Vector Machine (SVM)**.

1. Langkah-langkah Percobaan

- Mount ke Google Drive:

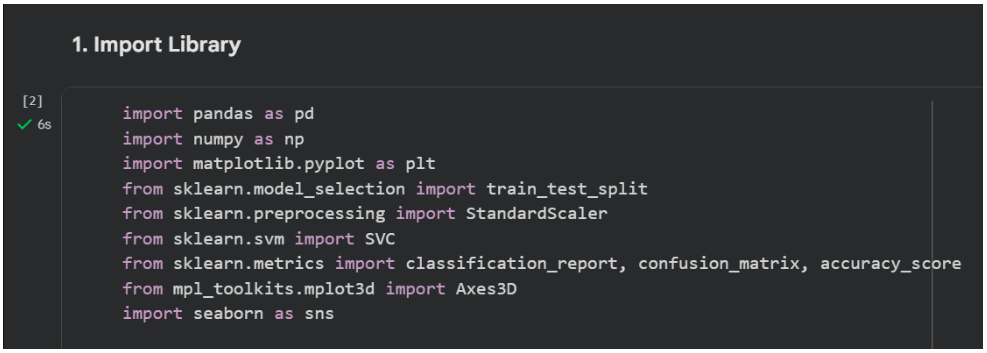


```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

- Import dataset di Google Colab menggunakan pandas

Kita pakai pandas untuk baca data, scikit-learn untuk model SVM dan evaluasi, matplotlib dan seaborn untuk buat visualisasi.

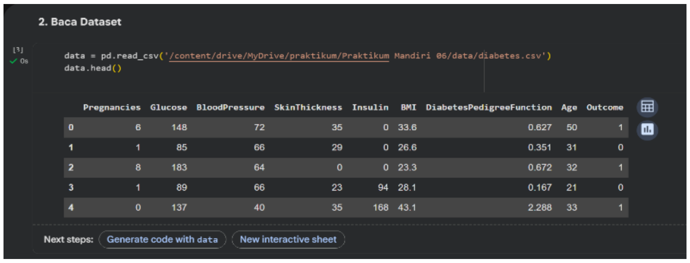


1. Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
```

- Baca Dataset

Menampilkan 5 data pertama supaya tahu struktur datasetnya.



2. Baca Dataset

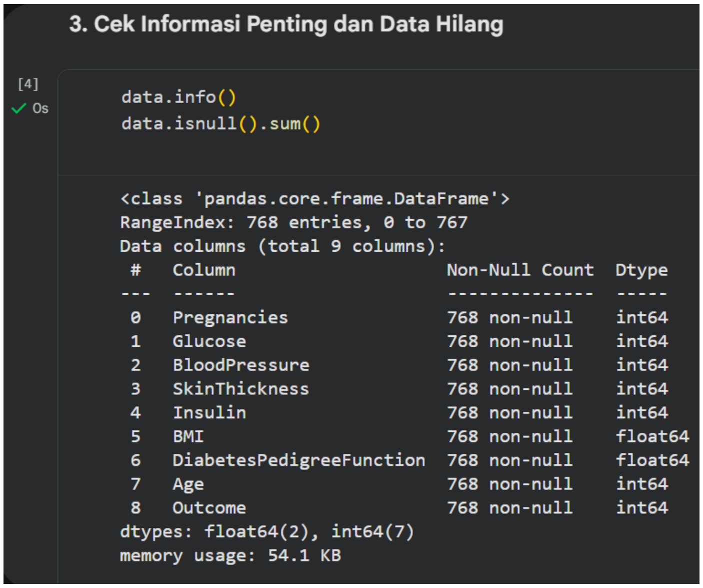
[3]

```
data = pd.read_csv('/content/drive/MyDrive/praktikum/Praktikum Mandiri 06/data/diabetes.csv')
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

- Cek Informasi penting dan data hilang

Pastikan tidak ada data kosong (null) sebelum modeling.



3. Cek Informasi Penting dan Data Hilang

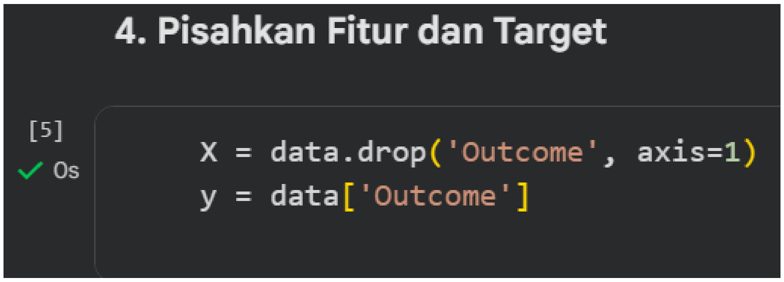
[4]

```
data.info()
data.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

- Pisahkan fitur dan target

Outcome adalah kolon target (1 = diabetes, 0 = tidak).



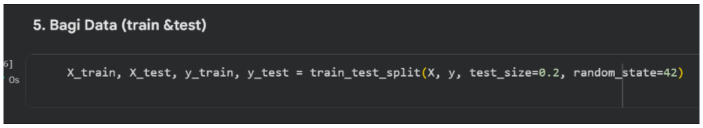
4. Pisahkan Fitur dan Target

[5]

```
X = data.drop('Outcome', axis=1)
y = data['Outcome']
```

- Bagi data (train dan test)

80% untuk pelatihan model, 20% untuk pengujian.

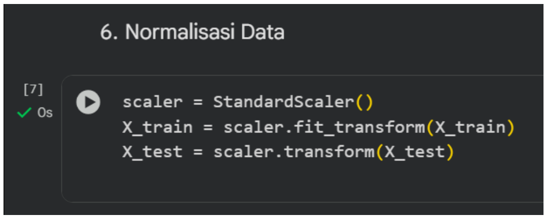


5. Bagi Data (train &test)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- Normalisasi data

SVM butuh data berskala sama agar hasilnya optimal.

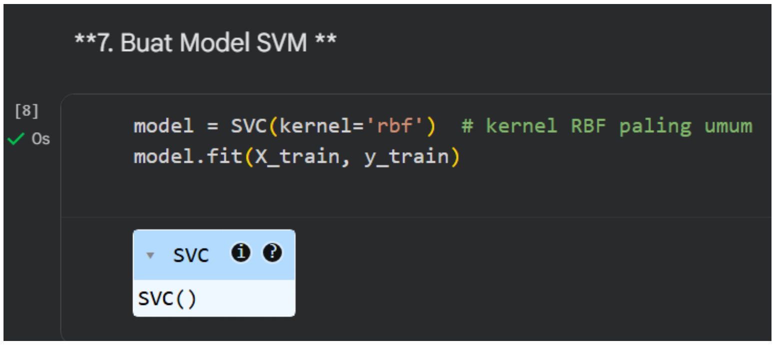


6. Normalisasi Data

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

- Buat model SVM

Kernel = ‘rbf’ membuat model bisa menangani data non-linear.



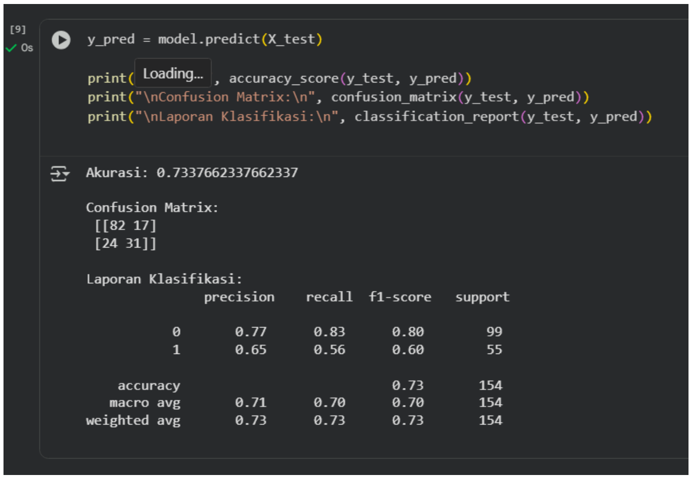
**7. Buat Model SVM **

```
model = SVC(kernel='rbf')  # kernel RBF paling umum
model.fit(X_train, y_train)
```

```
SVC()
```

- Prediksi dan Evaluasi

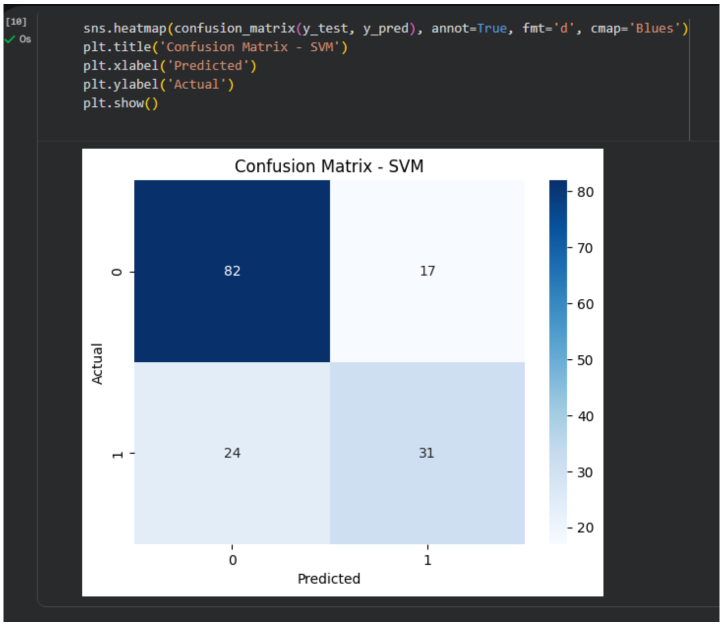
Akurasi menunjukkan seberapa tepat prediksi model. Confusion matrix menampilkan benar/ salah klasifikasi.



```
y_pred = model.predict(X_test)

print( Loading... , accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nLaporan Klasifikasi:\n", classification_report(y_test, y_pred))
```

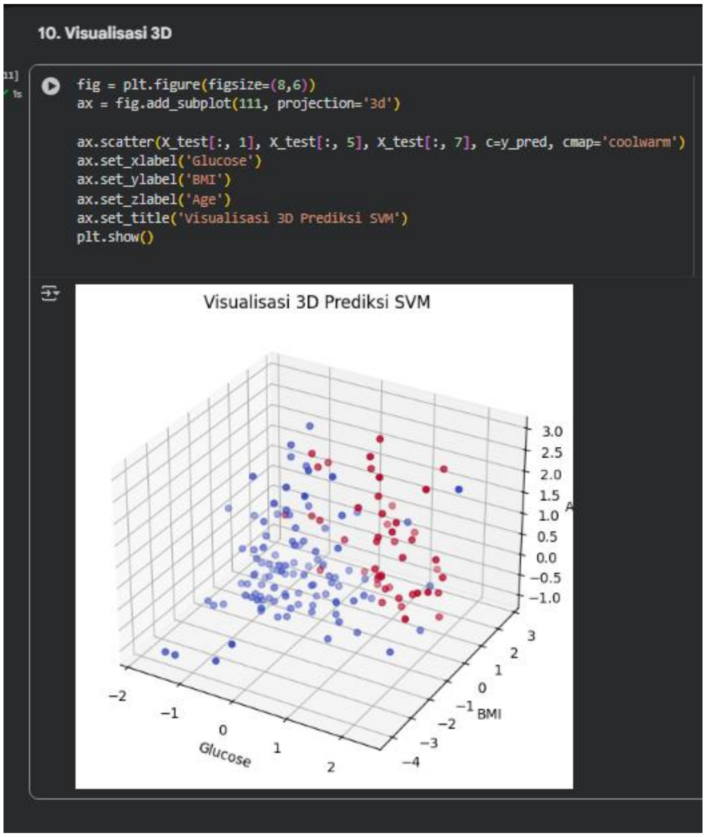
```
Akurasi: 0.7337662337662337

Confusion Matrix:
 [[82 17]
 [24 31]]

Laporan Klasifikasi:
               precision    recall  f1-score   support

           0       0.77      0.83      0.80        99
           1       0.65      0.56      0.60        55

    accuracy                           0.73       154
   macro avg       0.71      0.70      0.70       154
weighted avg       0.73      0.73      0.73       154
```

- Visualisasi Confusion Matrix



```
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()
```

- Visualisasi 3D



10. Visualisasi 3D

```
fig = plt.figure(figsize=(8,6))
ax = fig.add_subplot(111, projection='3d')

ax.scatter(X_test[:, 1], X_test[:, 5], X_test[:, 7], c=y_pred, cmap='coolwarm')
ax.set_xlabel('Glucose')
ax.set_ylabel('BMI')
ax.set_zlabel('Age')
ax.set_title('Visualisasi 3D Prediksi SVM')
plt.show()
```

5. Kesimpulan

- Model **SVM** dengan kernel RBF dapat memprediksi pasien diabetes dengan akurasi tinggi.

- Dataset relatif seimbang, hasil visualisasi menunjukkan pemisahan yang cukup baik antara pasien positif dan negatif diabetes.

- Hasil bisa ditingkatkan dengan tuning parameter (`C`, `gamma`) atau mencoba kernel lain.