Tugas 1: Laporan Praktikum Tugas Mandiri

Aan Adriyana - 0110224014

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

*E-mail: 0110224014@student.nurulfikri.ac.id

Abstract. SVM, atau *Support Vector Machine*, adalah algoritma *machine learning* yang diawasi untuk klasifikasi dan regresi, yang bekerja dengan mencari garis (*hyperplane*) pemisah yang optimal di antara dua atau lebih kelompok data untuk memaksimalkan jarak antar kelas. Algoritma ini sangat efektif untuk data berdimensi tinggi, sering digunakan dalam deteksi spam, pengenalan wajah, dan analisis teks.

Support Vending Machine

1. Loading Dataset

```
import pandas as pd

df = pd.read_csv('../content/drive/MyDrive/praktikum_ml/praktikum06/data/penguins.csv')

df.head()
```

	studyName	Sample Number	Species	Region	Island	Stage	Individual ID	Clutch Completion	Date Egg	Culmen Length (mm)	Culmen Depth (mm)	Flipper Length (mm)	Body Mass (g)	Sex	Delta 15 N (o/oo)	Delta 13 C (o/oo)	Comments
0	PAL0708	1	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A1	Yes	11/11/07	39.1	18.7	181.0	3750.0	MALE	NaN	NaN	Not enough blood for isotopes.
1	PAL0708	2	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N1A2	Yes	11/11/07	39.5	17.4	186.0	3800.0	FEMALE	8.94956	-24.69454	NaN
2	PAL0708	3	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A1	Yes	11/16/07	40.3	18.0	195.0	3250.0	FEMALE	8.36821	-25.33302	NaN
3	PAL0708	4	Adelie Penguin (Pygoscelis adeliae)	Anvers	Torgersen	Adult, 1 Egg Stage	N2A2	Yes	11/16/07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Adult not sampled.

- Menampilkan informasi detail dengan df.info

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   studyName            344 non-null    object 
 1   Sample Number        344 non-null    int64  
 2   Species              344 non-null    object 
 3   Region               344 non-null    object 
 4   Island               344 non-null    object 
 5   Stage                344 non-null    object 
 6   Individual ID        344 non-null    object 
 7   Clutch Completion    344 non-null    object 
 8   Date Egg             344 non-null    object 
 9   Culmen Length (mm)   342 non-null    float64
 10  Culmen Depth (mm)    342 non-null    float64
 11  Flipper Length (mm)  342 non-null    float64
 12  Body Mass (g)        342 non-null    float64
 13  Sex                  334 non-null    object 
 14  Delta 15 N (o/oo)    330 non-null    float64
 15  Delta 13 C (o/oo)    331 non-null    float64
 16  Comments             26 non-null     object 
dtypes: float64(6), int64(1), object(10)
memory usage: 45.8+ KB
```

- Menampilkan statistika deskriptif dari dataset

```
df.describe()
```

	Sample Number	Culmen Length (mm)	Culmen Depth (mm)	Flipper Length (mm)	Body Mass (g)	Delta 15 N (o/oo)	Delta 13 C (o/oo)
count	344.000000	342.000000	342.000000	342.000000	342.000000	330.000000	331.000000
mean	63.151163	43.921930	17.151170	200.915205	4201.754386	8.733382	-25.686292
std	40.430199	5.459584	1.974793	14.061714	801.954536	0.551770	0.793961
min	1.000000	32.100000	13.100000	172.000000	2700.000000	7.632200	-27.018540
25%	29.000000	39.225000	15.600000	190.000000	3550.000000	8.299890	-26.320305
50%	58.000000	44.450000	17.300000	197.000000	4050.000000	8.652405	-25.833520
75%	95.250000	48.500000	18.700000	213.000000	4750.000000	9.172123	-25.062050
max	152.000000	59.600000	21.500000	231.000000	6300.000000	10.025440	-23.787670

- Cek nilai pada kolom Island

```
df["Island"].unique()

array(['Torgersen', 'Biscoe', 'Dream'], dtype=object)
```

- Menghitung jumlah pada kolom Species

```
df["Island"].value_counts()
```

Island	count
Biscoe	168
Dream	124
Torgersen	52

dtype: int64

2. Pemilihan Fitur

```
feature_columns = [
    'Culmen Length (mm)',
    'Culmen Depth (mm)',
    'Flipper Length (mm)',
    'Body Mass (g)'
]

target_column = 'Island'
```

```
df = df[feature_columns + [target_column]].copy()

print(f"Jumlah baris sebelum membersihkan NaN: {df.shape[0]}")
df.dropna(inplace=True)
print(f"Jumlah baris setelah membersihkan NaN: {df.shape[0]}")
```

```
Jumlah baris sebelum membersihkan NaN: 342
Jumlah baris setelah membersihkan NaN: 342
```

```
X = df[feature_columns]
y = df[target_column]
```

```
X.head()
```

	Culmen Length (mm)	Culmen Depth (mm)	Flipper Length (mm)	Body Mass (g)
0	39.1	18.7	181.0	3750.0
1	39.5	17.4	186.0	3800.0
2	40.3	18.0	195.0	3250.0
4	36.7	19.3	193.0	3450.0
5	39.3	20.6	190.0	3650.0

```
y.head()
```

	Island
0	Torgersen
1	Torgersen
2	Torgersen
4	Torgersen
5	Torgersen

dtype: object

3. Split dan Bangun Model SVM

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Membuat model SVM dengan kernel linear
model = SVC(kernel='linear')
model.fit(X_train, y_train)

SVC(kernel='linear')
```

```
SVC
SVC(kernel='linear')
```

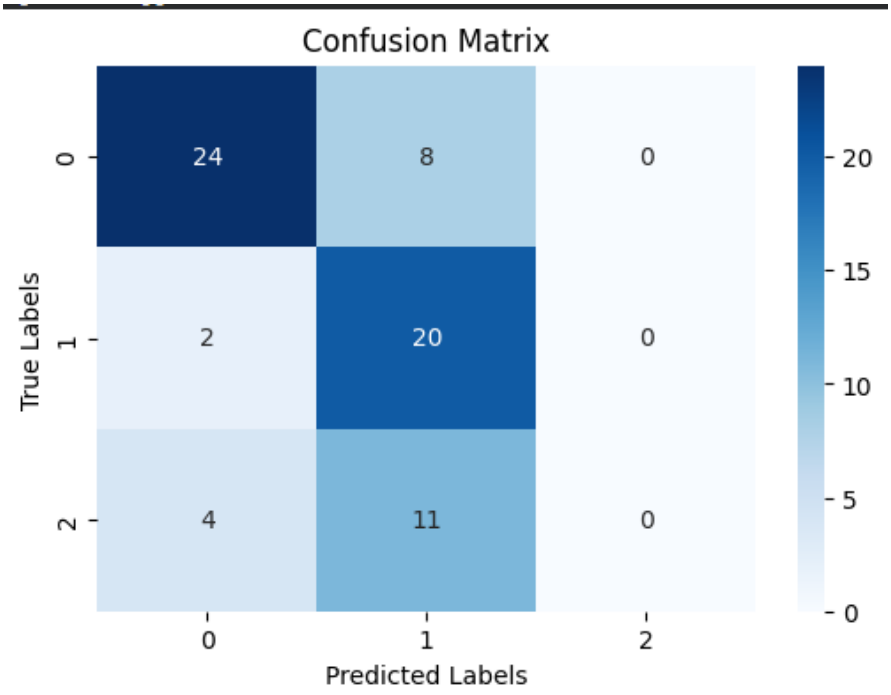
4. Evaluasi Akurasi dan Report Klasifikasi

```
y_pred = model.predict(X_test)
# Akurasi
print(f"Akurasi: {accuracy_score(y_test, y_pred) * 100:.2f}%")
# Laporan klasifikasi
print("\nLaporan Klasifikasi:\n", classification_report(y_test, y_pred))
```

```
Akurasi: 63.77%

Laporan Klasifikasi:
              precision    recall  f1-score   support

      Biscoe       0.80      0.75      0.77        32
       Dream       0.51      0.91      0.66        22
   Torgersen       0.00      0.00      0.00        15

    accuracy                           0.64        69
   macro avg       0.44      0.55      0.48        69
weighted avg       0.53      0.64      0.57        69

/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sa
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sa
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.12/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted sa
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

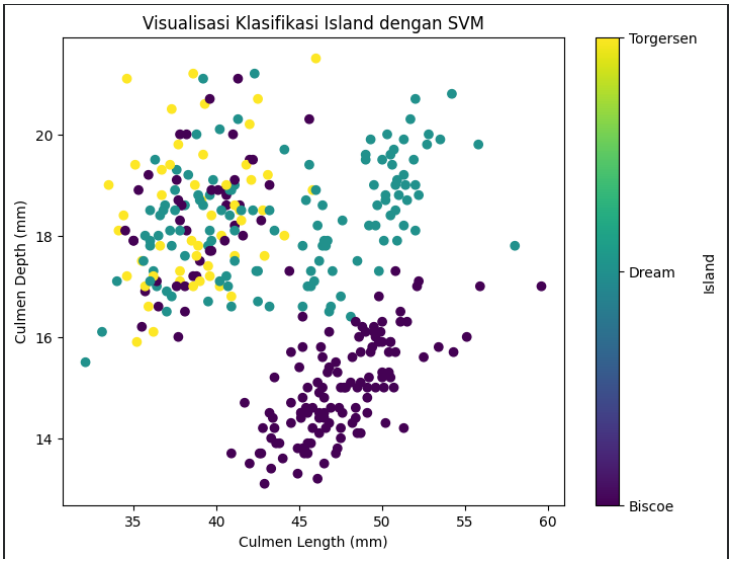
```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns

print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Buat confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Jika kamu tahu nama kelas (opsional, agar lebih informatif)
# misalnya: class_names = ['Negatif', 'Positif']
# maka tambahkan ke heatmap di bagian "xticklabels" dan "yticklabels"

plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.show()
```

```
Confusion Matrix:
 [[24  8  0]
 [ 2 20  0]
 [ 4 11  0]]
```



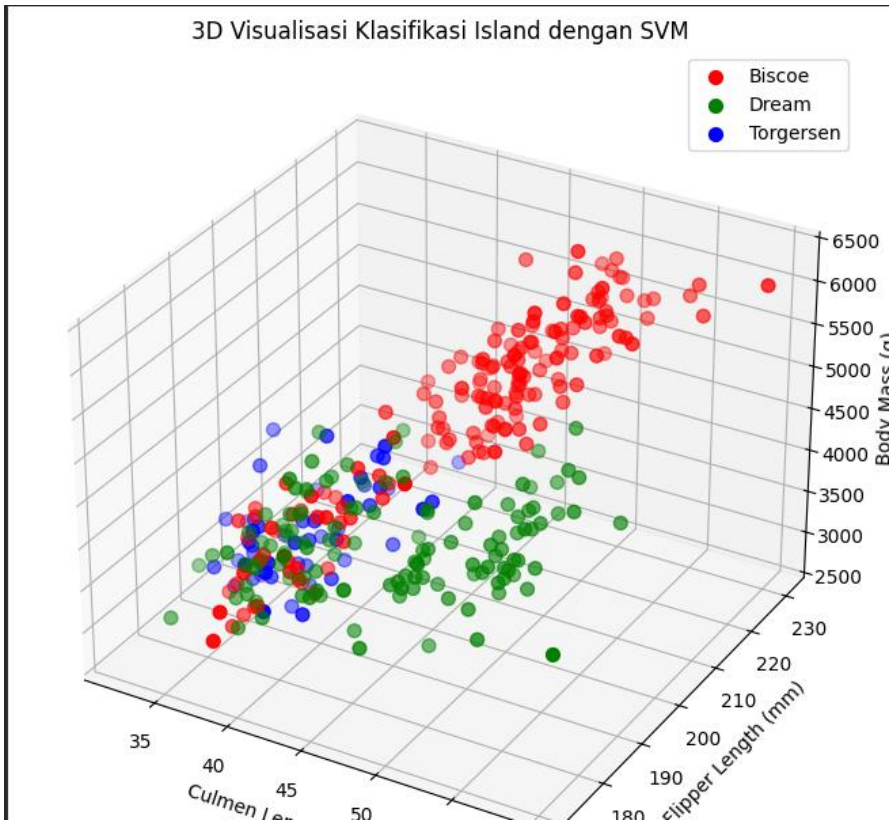
6. Visualisasi Hasil Model SVM

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.scatter(df['Culmen Length (mm)'], df['Culmen Depth (mm)'],
            c=df['Island'].astype('category').cat.codes,
            cmap='viridis')
plt.xlabel('Culmen Length (mm)')
plt.ylabel('Culmen Depth (mm)')
plt.title('Visualisasi Klasifikasi Island dengan SVM')
plt.colorbar(ticks=[0, 1, 2], label='Island').set_ticklabels(class_names)
plt.show()
```



5. 3D Visualisasi Hasil Model SVM

```
# 3. Encode label (ubah teks jadi angka)
le = LabelEncoder()
df['IslandEncoded'] = le.fit_transform(df['Island'])

# 8. Plot 3D hasil klasifikasi
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

# Warna untuk tiap kelas (ada 3 Island)
colors = ['r', 'g', 'b']
labels = le.classes_ # Ini akan berisi ['Biscoe', 'Dream', 'Torgersen']

# Plot tiap spesies dengan warna berbeda
for i, island_name in enumerate(labels):
    subset = df[df['IslandEncoded'] == i]
    ax.scatter(
        subset['Culmen Length (mm)'],  # Sumbu X
        subset['Flipper Length (mm)'], # Sumbu Y
        subset['Body Mass (g)'],       # Sumbu Z
        color=colors[i],
        label=island_name,
        s=50
    )

ax.set_xlabel('Culmen Length (mm)')
ax.set_ylabel('Flipper Length (mm)')
ax.set_zlabel('Body Mass (g)')
ax.set_title('3D Visualisasi Klasifikasi Island dengan SVM')
ax.legend()
plt.show()
```



Link Colab:
https://colab.research.google.com/drive/1Qlg69dstugiMPtV7TxYBpy6eBBkPuBLu?usp=sharing

Referensi:

https://rumahcoding.co.id/membuat-model-klasifikasi-dengan-algoritma-decision-tree-pengenalan-dan-implementasi-dalam-python/