Data Science

Iris Flower Classification

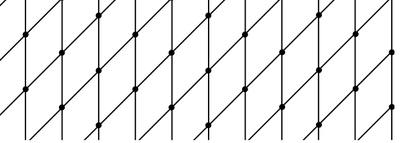
Presentation by

Nur Istiqama

Introduction

Tanaman Iris sering digunakan dalam penelitian botani karena memiliki bentuk dan ukuran yang beragam. Dataset Iris adalah salah satu dataset paling terkenal dalam dunia Data Science dan Machine Learning, yang sering digunakan untuk eksperimen dalam klasifikasi data. Dataset ini terdiri dari tiga jenis spesies: Setosa, Versicolor, dan Virginica. Setiap spesies memiliki karakteristik unik berdasarkan panjang dan lebar sepal serta petal. Machine Learning digunakan untuk mengotomatisasi proses klasifikasi ini, meningkatkan efisiensi serta akurasi dalam mengidentifikasi spesies tanaman Iris.

Tools




ibimbing


Visual Studio Code


python


Matplotlib


pandas


scikit
learn


seaborn

Dataset

```python
# Load dataset Iris dari scikit-learn
iris = datasets.load_iris()

# Konversi menjadi DataFrame agar lebih mudah dipahami
df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df['target'] = iris.target

# Mapping angka target ke nama spesies
df['target'] = df['target'].map({0: 'Setosa', 1: 'Versicolor', 2: 'Virginica'})

# Tampilkan 5 data pertama
print(df.head())

# Cek jumlah baris dan kolom
print("Jumlah baris dan kolom:", df.shape)
```

Sumber : Scikit-Learn Iris Dataset
Jumlah data : 150 sampel
Kelas : Setosa, Versicolor, Virginica
Model Machine Learning : K-Nearest Neighbors

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
0                5.1               3.5                1.4               0.2  Setosa
1                4.9               3.0                1.4               0.2  Setosa
2                4.7               3.2                1.3               0.2  Setosa
3                4.6               3.1                1.5               0.2  Setosa
4                5.0               3.6                1.4               0.2  Setosa
Jumlah baris dan kolom: (150, 5)
```

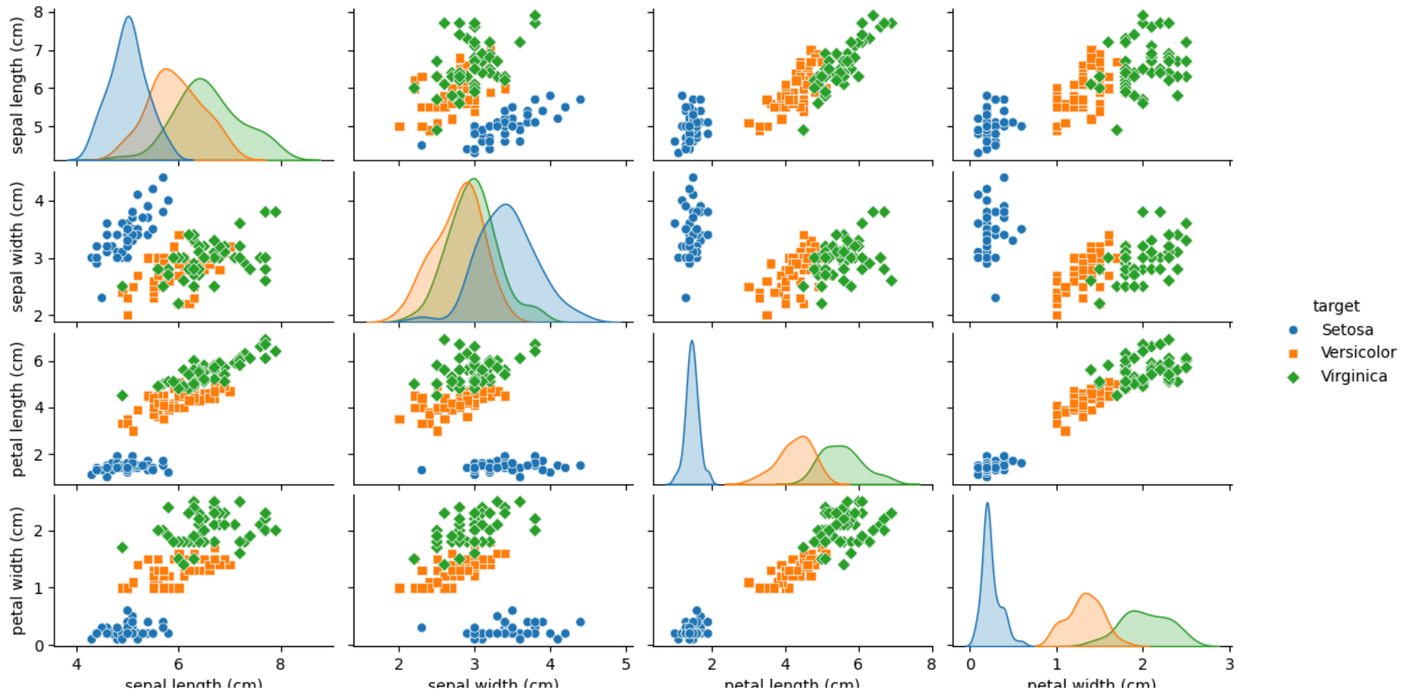
Exploratory Data Analysis (EDA)

- Dataset terdiri dari 150 sampel dan 5 kolom
- Memiliki tipe data numerik
- Tidak ada missing value
- Terdapat 50 sampel per kelas

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   sepal length (cm)  150 non-null    float64
 1   sepal width (cm)   150 non-null    float64
 2   petal length (cm)  150 non-null    float64
 3   petal width (cm)   150 non-null    float64
 4   target             150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
target
Setosa        50
Versicolor    50
Virginica     50
Name: count, dtype: int64
```

Exploratory Data Analysis (EDA)

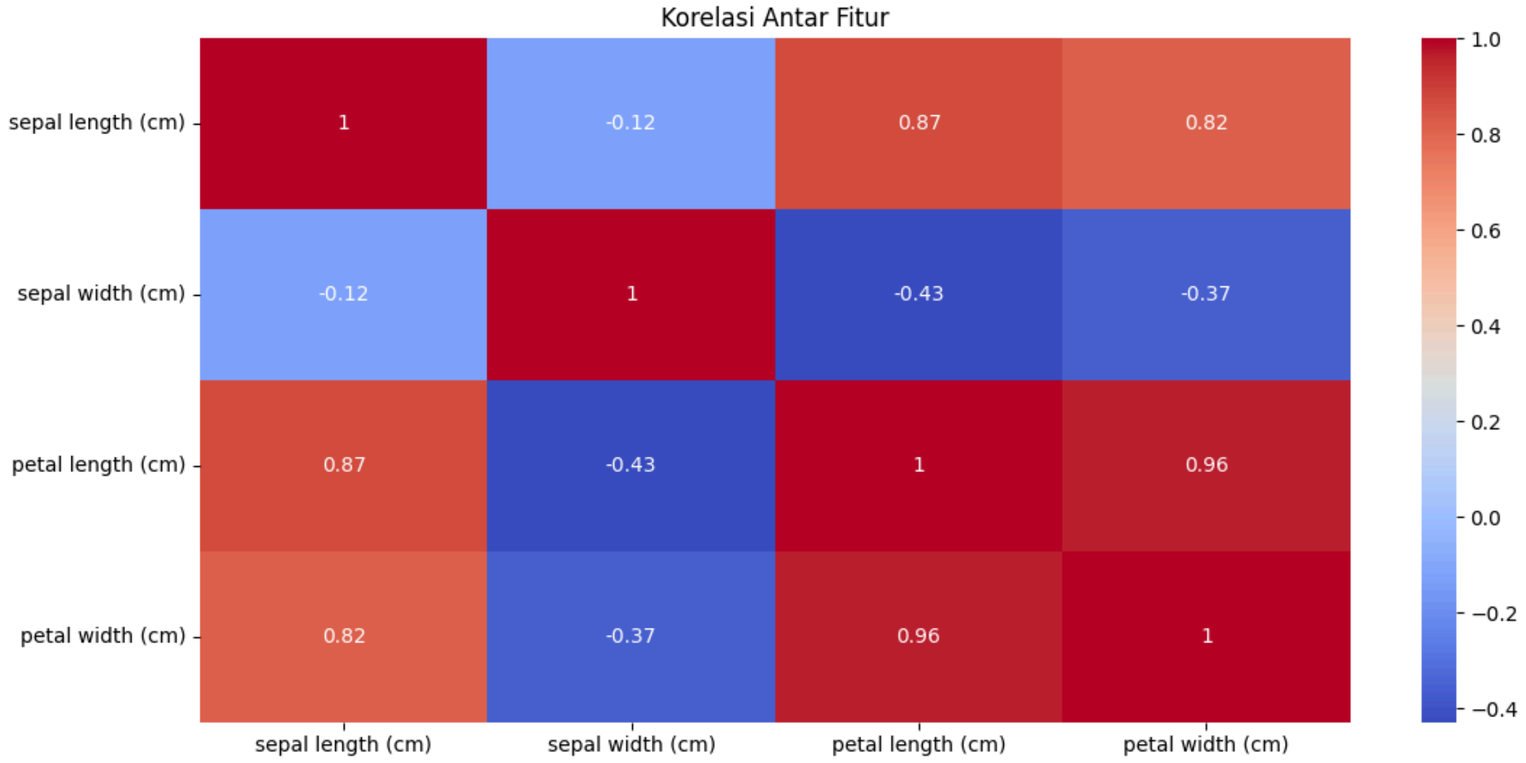
```
# Pairplot untuk melihat hubungan antar fitur
sns.pairplot(df, hue='target', markers=["o", "s", "D"])
plt.show()
```



Visualisasi Data (Pairplot)

Setiap titik dalam grafik mewakili satu sampel bunga. Diagonal atas dan bawah menunjukkan distribusi dari setiap fitur menggunakan grafik density plot. Scatter plot di bagian bawah menunjukkan hubungan antar fitur

Exploratory Data Analysis (EDA)



- Petal length dan petal width memiliki korelasi tinggi (0.96), menunjukkan hubungan yang sangat kuat.
- Sepal length dan petal length juga memiliki korelasi kuat (0.87), menandakan bahwa semakin panjang sepal, semakin panjang pula petalnya.
- Sepal width dan petal length memiliki korelasi negatif (-0.43), menunjukkan hubungan berlawanan yang lemah.

Machine Learning Model

Split Data

```
# Pisahkan fitur (X) dan target (y)
X = iris.data
y = iris.target

# Bagi data menjadi training (80%) dan testing (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Cek jumlah data training dan testing
print(f"Jumlah data training: {X_train.shape[0]}")
print(f"Jumlah data testing: {X_test.shape[0]}")
```

```
Jumlah data training: 120
Jumlah data testing: 30
```

- 80% data digunakan untuk pelatihan (train)
- 20% untuk pengujian (test)

Machine Learning Model

Model Training

```
# Buat model KNN dengan k=3
model = KNeighborsClassifier(n_neighbors=3)

# Latih model dengan data training
model.fit(X_train, y_train)

# Prediksi data testing
y_pred = model.predict(X_test)
```

- Model KNN dibuat dengan jumlah tetangga (k) = 3.
- Model dilatih menggunakan X_train (fitur) dan y_train (label).
- Model yang telah dilatih digunakan untuk memprediksi kelas dari X_test (data uji). Hasil prediksi disimpan dalam y_pred.

Evaluasi Model

Akurasi Model & Classification Report

```python
# Cek akurasi
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi model: {accuracy:.2f}")
```

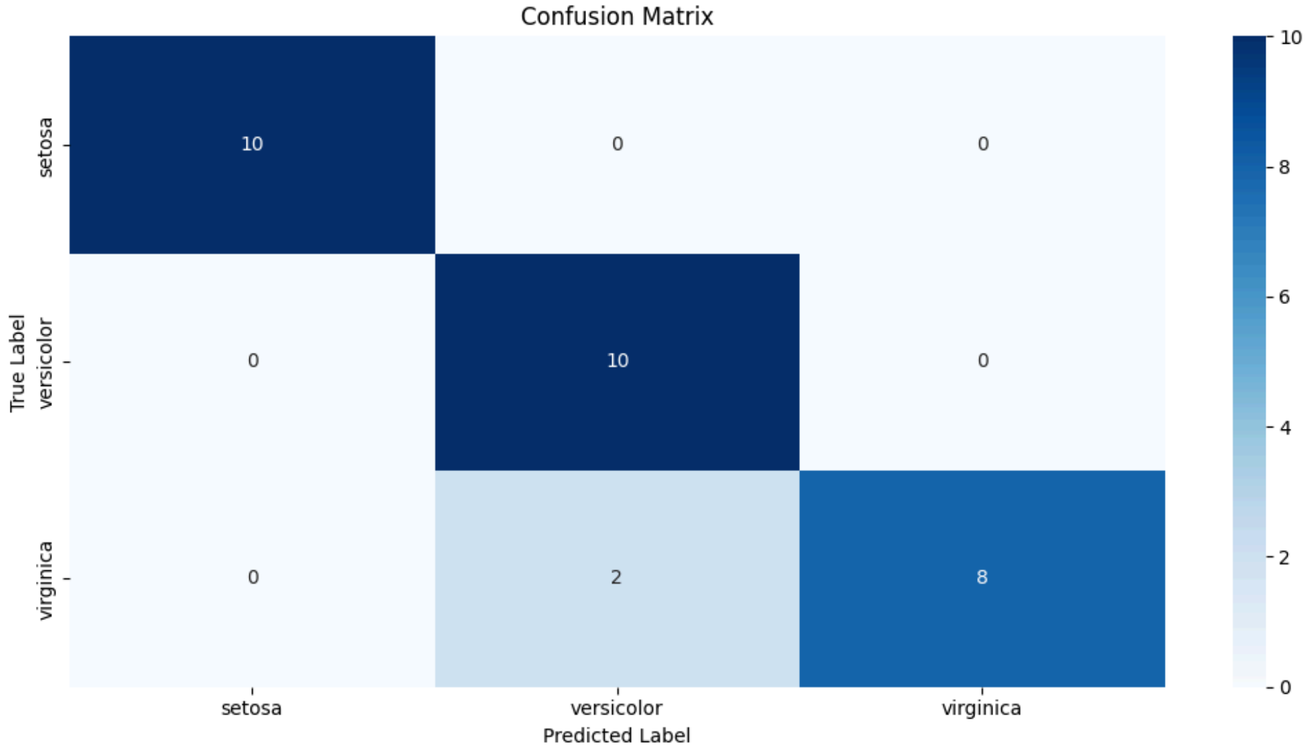
```python
# Tampilkan classification report
print("Classification Report:")
print(classification_report(y_test, y_pred,
                            target_names=iris.target_names))
```

```
Akurasi model: 0.93
Classification Report:
              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       0.83      1.00      0.91        10
   virginica       1.00      0.80      0.89        10

    accuracy                           0.93        30
   macro avg       0.94      0.93      0.93        30
weighted avg       0.94      0.93      0.93        30
```

Evaluasi Model

Confusion Matrix

```
# Tampilkan confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(confusion_matrix(y_test, y_pred),
            annot=True, fmt='d', cmap='Blues',
            xticklabels=iris.target_names,
            yticklabels=iris.target_names)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```



Thank You

+6285756850344

https://github.com/NurIstiqama

@nur_istyqamaa

www.linkedin.com/in/nur-istiqama-38b108275

nuristiqama091@gmail.com