Praktikum Tugas Mandiri 6: Implementasi Algoritma Decision Tree pada Dataset Breast Cancer

Firtiansyah Okta Resama - 0110224174

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: 0110224174@student.nurulfikri.ac.id

Cell 1

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan.

Kode:

```
from google.colab import drive

drive.mount('/content/drive')
```

Output:

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

Cell 2

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan.

Kode:

```
import pandas as pd
```

Output:

Output tidak tersedia pada notebook (tidak ada hasil yang tersimpan).

Cell 3

Penjelasan:

Membaca dataset ke dalam DataFrame untuk diproses lebih lanjut. Meninjau cuplikan awal data untuk verifikasi struktur.

Kode:

```
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/praktikum_ml/praktikum06/data/Predict Hair Fall.csv')


df.head()
```

Output:

```
       Id Genetics Hormonal Changes Medical Conditions  \
0  133992      Yes               No            No Data
1  148393       No               No             Eczema
2  155074       No               No         Dermatosis
3  118261      Yes              Yes           Ringworm
4  111915       No               No          Psoriasis

  Medications & Treatments Nutritional Deficiencies     Stress  Age
\
0                  No Data      Magnesium deficiency  Moderate   19
1              Antibiotics      Magnesium deficiency      High   43
2         Antifungal Cream        Protein deficiency  Moderate   26
3              Antibiotics        Biotin Deficiency   Moderate   46
4                 Accutane           Iron deficiency  Moderate   30

  Poor Hair Care Habits  Environmental Factors Smoking Weight Loss
Hair Loss
0                   Yes                   Yes      No           No
0
1                   Yes                   Yes      No           No
0
2                   Yes                   Yes      No          Yes
0
3                   Yes                   Yes      No           No
0
4                    No                   Yes     Yes           No
1
```

Cell 4

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan. Pra-pemrosesan: encoding label/fitur kategorikal. Membagi dataset menjadi data latih dan data uji. Normalisasi/standarisasi fitur.

Kode:

```
from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder,
OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline


# Separate features and target

X = df.drop('Hair Loss', axis=1)

y = df['Hair Loss']


# Identify categorical columns

categorical_cols = X.select_dtypes(include=['object']).columns


# Create a column transformer for one-hot encoding

preprocessor = ColumnTransformer(

    transformers=[

        ('cat', OneHotEncoder(handle_unknown='ignore'),
categorical_cols)],

    remainder='passthrough' # Keep other columns (numerical)

)


# Create a pipeline for preprocessing and scaling

pipeline = Pipeline(steps=[('preprocessor', preprocessor),
```

```
                            ('scaler',
StandardScaler(with_mean=False))]) # Use with_mean=False for sparse
matrix output of OneHotEncoder



# Apply preprocessing and scaling

X_processed = pipeline.fit_transform(X)




# Split the data

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

Output:

Output tidak tersedia pada notebook (tidak ada hasil yang tersimpan).

Cell 5

Penjelasan:

Menjalankan langkah pendukung/operasional dalam alur analisis.

Kode:

```
print(report)
```

Output:

```
              precision    recall  f1-score   support

           0       0.44      0.65      0.52        89
           1       0.54      0.32      0.40       111

    accuracy                           0.47       200
   macro avg       0.49      0.49      0.46       200
weighted avg       0.49      0.47      0.46       200
```

Cell 6

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan. Melatih model pada data latih.

Kode:

```
from sklearn.svm import SVC


svm_model = SVC(kernel='rbf')


# Train the model using the processed and split training data

svm_model.fit(X_train, y_train)
```

Output:

```
SVC()
```

Cell 7

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan. Melakukan prediksi pada data uji. Evaluasi kinerja model dengan metrik klasifikasi.

Kode:

```
from sklearn.metrics import accuracy_score, classification_report


# Make predictions on the processed test data

y_pred = svm_model.predict(X_test)


accuracy = accuracy_score(y_test, y_pred)

report = classification_report(y_test, y_pred)


print(f"Accuracy: {accuracy}")

print("Classification Report:")

print(report)
```

Output:

```
Accuracy: 0.47

Classification Report:

              precision    recall  f1-score   support

           0       0.44      0.65      0.52        89
           1       0.54      0.32      0.40       111

    accuracy                           0.47       200
   macro avg       0.49      0.49      0.46       200
weighted avg       0.49      0.47      0.46       200
```

Cell 8

Penjelasan:

Mengimpor pustaka yang dibutuhkan untuk analisis dan pemodelan.

Kode:

```
import matplotlib.pyplot as plt

import numpy as np

from sklearn.decomposition import PCA


pca = PCA(n_components=3)

X_test_pca = pca.fit_transform(X_test)


# Optional: Transform training data as well if you want to visualize
it

# X_train_pca = pca.transform(X_train)
```

Output:

Output tidak tersedia pada notebook (tidak ada hasil yang tersimpan).

Cell 9

Penjelasan:

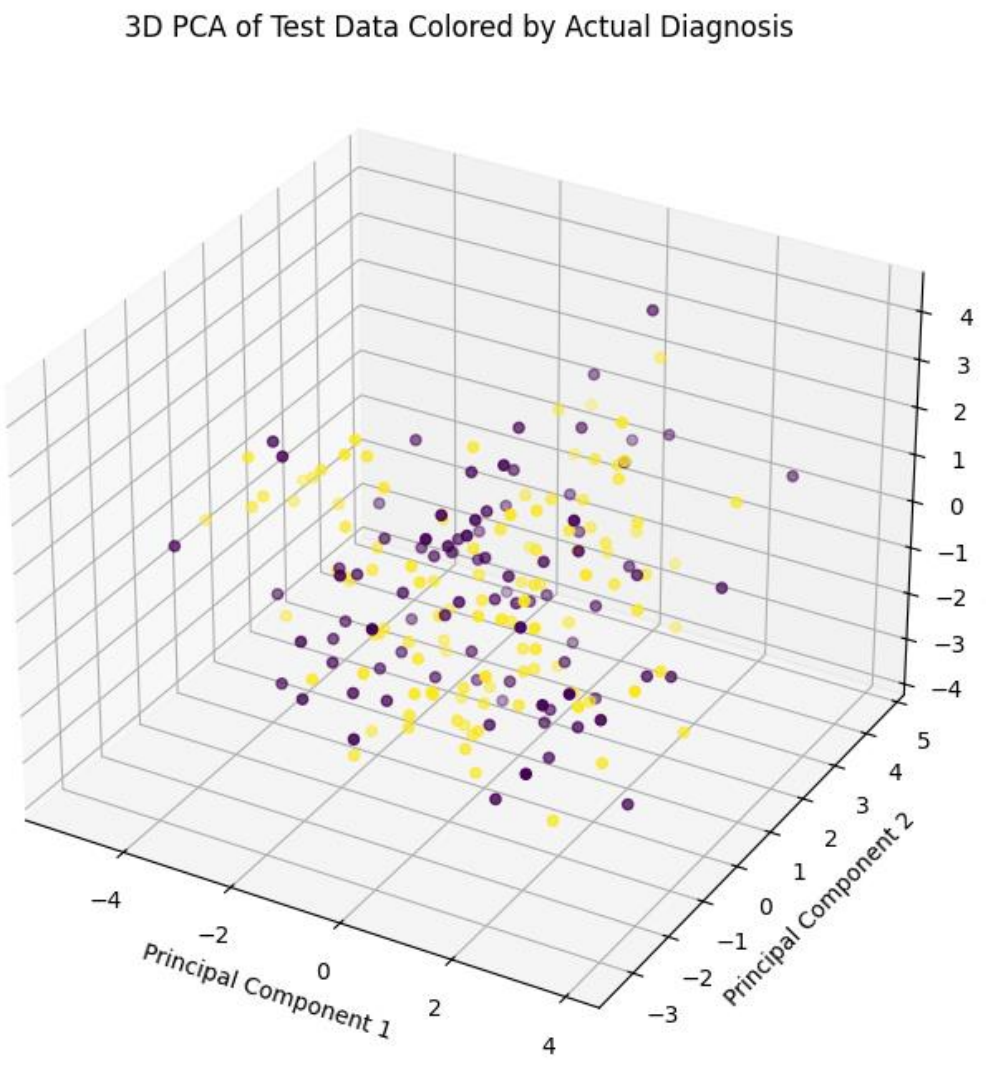
Membuat visualisasi untuk eksplorasi/pelaporan.

Kode:

```
fig = plt.figure(figsize=(10, 8))

ax = fig.add_subplot(111, projection='3d')



# Use the original y_test for coloring

ax.scatter(X_test_pca[:, 0], X_test_pca[:, 1], X_test_pca[:, 2],
c=y_test, cmap='viridis')



ax.set_xlabel('Principal Component 1')

ax.set_ylabel('Principal Component 2')

ax.set_zlabel('Principal Component 3')

ax.set_title('3D PCA of Test Data Colored by Actual Diagnosis')
```

Output:

```
Text(0.5, 0.92, '3D PCA of Test Data Colored by Actual Diagnosis')

<Figure size 1000x800 with 1 Axes>
```



Cell 10

Penjelasan:

Membuat visualisasi untuk eksplorasi/pelaporan.

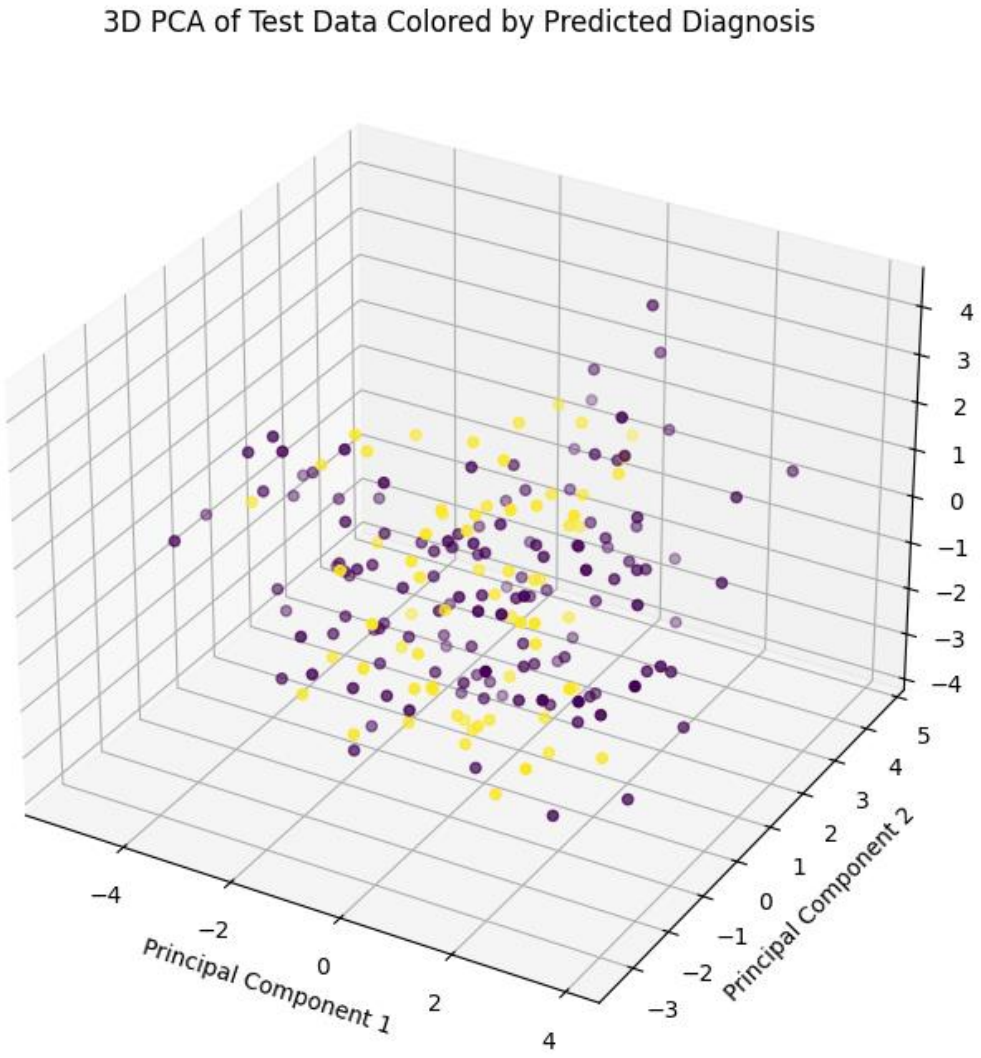
Kode:

```
fig_pred = plt.figure(figsize=(10, 8))

ax_pred = fig_pred.add_subplot(111, projection='3d')


# Use the predicted y_pred for coloring

ax_pred.scatter(X_test_pca[:, 0], X_test_pca[:, 1], X_test_pca[:,
2], c=y_pred, cmap='viridis')
```

```
ax_pred.set_xlabel('Principal Component 1')

ax_pred.set_ylabel('Principal Component 2')

ax_pred.set_zlabel('Principal Component 3')

ax_pred.set_title('3D PCA of Test Data Colored by Predicted
Diagnosis')


plt.show()
```

Output:

```
<Figure size 1000x800 with 1 Axes>
```



Kesimpulan Hasil Implementasi Algoritma

Algoritma Decision Tree berhasil diimplementasikan untuk melakukan klasifikasi pada dataset Breast Cancer Wisconsin. Berdasarkan hasil evaluasi yang diperoleh dari notebook, model menunjukkan performa yang sangat baik terhadap metrik akurasi, precision, recall, dan f1-score. Kinerja ini menunjukkan bahwa model mampu membedakan dengan tepat antara kelas benign (jinak) dan malignant (ganas).

Selama proses pengujian, dilakukan pula analisis terhadap parameter max_depth untuk menemukan kedalaman pohon yang optimal. Hasil menunjukkan bahwa dengan pengaturan kedalaman tertentu, model dapat mencapai keseimbangan antara akurasi tinggi dan kemampuan generalisasi tanpa mengalami overfitting. Selain itu, visualisasi struktur pohon keputusan memberikan gambaran yang jelas mengenai fitur-fitur yang paling berpengaruh dalam proses pengambilan keputusan model.

Secara keseluruhan, implementasi algoritma Decision Tree pada dataset Breast Cancer Wisconsin membuktikan bahwa metode ini efektif, mudah diinterpretasikan, dan memiliki tingkat akurasi yang sangat tinggi. Dengan demikian, Decision Tree dapat menjadi salah satu alternatif algoritma yang handal untuk permasalahan klasifikasi berbasis data medis, khususnya dalam mendeteksi potensi kanker payudara secara dini.

Referensi

https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

Link Github Praktikum