Syifa Wanda Isnaini

1103201248

Machine Learning

"Halo semuanya! Selamat datang di tutorial hari ini tentang Pembelajaran Mesin dan Clustering. Dalam tutorial ini, kita akan membahas dasar-dasar eksplorasi data, pembersihan, visualisasi, dan clustering menggunakan Python. Mari kita mulai!"

[Langkah 1: Mengimpor Pustaka yang Diperlukan]

"Pertama-tama, mari impor pustaka yang diperlukan untuk analisis kita. Pustaka-pustaka ini termasuk NumPy, Pandas, Matplotlib, Seaborn, scikit-learn, Yellowbrick, dan Google Colab bagi mereka yang menggunakan Colab."

NumPy (np): Perpustakaan untuk operasi numerik di Python

Pandas (pd): Perpustakaan untuk manipulasi dan analisis data

Matplotlib (plt): Perpustakaan plotting populer untuk membuat visualisasi statis, interaktif, dan beranimasi di Python.

Seaborn (sns): Dibangun di atas Matplotlib, Seaborn adalah perpustakaan visualisasi data statistik. Ini menyederhanakan proses membuat grafik statistik yang informatif dan menarik.

scikit-learn (sklearn): Perpustakaan pembelajaran mesin yang komprehensif. Ini mencakup alat untuk pra-pemrosesan data, pemilihan model, metrik kinerja, dan berbagai algoritma pembelajaran mesin.

Yellowbrick (yellowbrick): Perpustakaan visualisasi untuk scikit-learn. Ini memperluas fungsionalitas Matplotlib dan Seaborn untuk menyediakan alat visual untuk tugas pembelajaran mesin, termasuk evaluasi dan pemilihan model.

Google Colab (google.colab): Perpustakaan untuk berinteraksi dengan Google Colaboratory, platform cloud gratis untuk mengembangkan dan menjalankan kode Python secara kolaboratif.

python

Copy code

Mengimpor pustaka yang diperlukan

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.preprocessing import LabelEncoder

from sklearn.cluster import KMeans

from yellowbrick.cluster import KElbowVisualizer

from google.colab import drive

Mounting Google Drive untuk mengakses dataset

drive.mount('/content/drive')

Memuat dataset ke dalam DataFrame Pandas

df = pd.read\_csv('/content/drive/MyDrive/College.csv')

[Langkah 2: Analisis Data Eksploratif (EDA)]

"Sekarang, mari lakukan Analisis Data Eksploratif (EDA) untuk memahami dataset kita dengan lebih baik. Kita akan menampilkan beberapa baris pertama, memberikan informasi tentang dataset, dan menghasilkan ringkasan statistik."

python

Copy code

Analisis Data Eksploratif (EDA)

Menampilkan beberapa baris pertama dari dataset

df.head()

Memberikan informasi tentang dataset

df.info()

Ringkasan statistik dataset

df.describe()

[Langkah 3: Visualisasi Data]

"Selanjutnya, mari visualisasikan data kita. Kita akan membuat scatter plot untuk memvisualisasikan 'Room.Board' terhadap 'Grad.Rate' dengan diferensiasi oleh 'Private.'"

python

Copy code

Visualisasi Data

Scatter plot untuk memvisualisasikan 'Room.Board' terhadap 'Grad.Rate' dengan diferensiasi oleh 'Private'

sns.set\_style('whitegrid')

sns.lmplot(x='Room.Board', y='Grad.Rate', data=df, hue='Private',

palette='coolwarm', height=6, aspect=1, fit\_reg=False)

plt.show()

[Langkah 4: Pembersihan Data]

"Sekarang, mari atasi pembersihan data yang mungkin diperlukan. Dalam kasus kita, kita akan menangani nilai 'Grad.Rate' yang lebih dari 100."

python

Copy code

Pembersihan dan Persiapan Data

Menangani nilai 'Grad.Rate' yang lebih dari 100

df[df['Grad.Rate'] > 100]

df['Grad.Rate']['Cazenovia College'] = 100

df[df['Grad.Rate'] > 100]

[Langkah 5: training dan evaluasi]

Menghasilkan laporan klasifikasi yang melibatkan precision, recall, dan f1-score.

1. non\_numeric\_cols = df.select\_dtypes(include=['object']).columns.tolist(): Mengidentifikasi kolom-kolom dalam DataFrame (df) yang memiliki tipe data objek (biasanya kategorikal). Kolom-kolom ini kemudian akan diabaikan atau diencode, tergantung pada kebutuhan.

2. X = df.drop(non\_numeric\_cols + ['Cluster'], axis=1): Membuang kolom-kolom non-numerik dan kolom target ('Cluster') dari dataset untuk menyiapkan fitur (features) untuk model.

3. y = df['Cluster']: Menentukan target yang ingin diprediksi.

4. label\_encoder = LabelEncoder(): Membuat objek LabelEncoder dari scikit-learn untuk mengkodekan label kategori menjadi nilai numerik.

5. y = label\_encoder.fit\_transform(y): Menggunakan LabelEncoder untuk mengkodekan label target ('Cluster') menjadi nilai numerik. Ini diperlukan karena model-machine learning pada umumnya memerlukan input yang berupa nilai numerik.

6. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42): Membagi dataset menjadi data latih (train) dan data uji (test) untuk melatih dan mengevaluasi model.

7. model = DecisionTreeClassifier(): Membuat model klasifikasi pohon keputusan sebagai contoh. Anda dapat menggantinya dengan model lain yang sesuai dengan tugas Anda.

8. model.fit(X\_train, y\_train): Melatih model menggunakan data latih.

9. predictions = model.predict(X\_test): Membuat prediksi menggunakan data uji.

10. accuracy = accuracy\_score(y\_test, predictions): Mengukur akurasi model.

11. report = classification\_report(y\_test, predictions): Menghasilkan laporan klasifikasi yang melibatkan precision, recall, dan f1-score.

[Langkah 6: Model Clustering - K-Means]

"Beralih ke clustering, mari implementasikan algoritma clustering K-Means. Kita akan mulai dengan menemukan jumlah kluster optimal menggunakan metode siku."

python

Copy code

Membuat instance KMeans

model = KMeans()

Menggunakan metode siku untuk menemukan jumlah kluster yang optimal

visualizer = KElbowVisualizer(model, k=(2,10))

visualizer.fit(X)

visualizer.show()

[Langkah 6: Mengaplikasikan Clustering K-Means]

"Dengan jumlah kluster optimal teridentifikasi, mari terapkan clustering K-Means pada dataset kita."

python

Copy code

Membuat model KMeans dengan, misalnya, 4 kluster

kmeans = KMeans(n\_clusters=4)

kmeans.fit(X)

cluster\_labels = kmeans.labels\_

print("Cluster Labels:", cluster\_labels)

[Langkah 7: Visualisasi Hasil Clustering]

"Terakhir, mari visualisasikan hasil clustering dengan scatter plot."

python

Copy code

Visualisasikan hasil clustering

plt.scatter(X['Room.Board'], X['Grad.Rate'], c=cluster\_labels, cmap='viridis')

plt.title('KMeans Clustering')

plt.xlabel('Room & Board')

plt.ylabel('Graduation Rate')

plt.show()

Kesimpulan :

Data telah dijelajahi dan dibersihkan dengan beberapa visualisasi untuk pemahaman awal.

Sebuah model Decision Tree telah dibuat dan dievaluasi untuk klasifikasi.

Selain itu, dilakukan clustering menggunakan algoritma KMeans untuk memahami pola grup dalam data.

Namun, pada penggunaan KMeans, hasil visualisasi cluster yang dihasilkan mungkin belum memberikan pemahaman yang mendalam tentang pola di dalam data. Evaluasi lebih lanjut dan interpretasi tergantung pada kebutuhan analisis yang lebih spesifik.

"Dan itulah semuanya! Kita telah mencakup dasar-dasar eksplorasi data, pembersihan, visualisasi, dan clustering. Silakan sesuaikan langkah-langkah ini dengan dataset dan kebutuhan analisis spesifik Anda. Jika Anda menemukan tutorial ini membantu, jangan lupa untuk suka dan berlangganan untuk konten lebih lanjut. Selamat coding!"