Nama: Haidir Umar

NIM: 202231507

Ujian Tengah Semester Matkul Pembelajaran Mesin

DECISION REGRESI DATASET R02_rice_field

Penjelasan Dari Setiap Kodingan Yang Dibuat:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

Pandas (pd) untuk manipulasi data (tabel).

NumPy (np) untuk perhitungan numerik.

Matplotlib (plt) untuk membuat grafik dan visualisasi data.

Seaborn (sns) untuk membuat grafik yang lebih estetis dan kompleks.

R02_rice_field = pd.read_csv('R02_rice_field.csv', sep=';')

R02_rice_field

pd.read_csv('R02_rice_field.csv', sep=';') digunakan untuk membaca file CSV yang dipisahkan dengan titik koma (;) dan menyimpannya ke dalam variabel **R02_rice_field** dalam bentuk DataFrame.

R02_rice_field setelah itu akan berisi data dari file CSV yang dapat kita analisis atau manipulasi.

R02_rice_field.info()

- Fungsi info() digunakan untuk memberikan gambaran umum tentang dataset.

R02_rice_field.describe()

- Fungsi describe() digunakan untuk memberikan ringkasan statistik dari kolom-kolom numerik dalam dataset.

R02_rice_field.isna().sum()

- Kode isna() akan menghasilkan DataFrame boolean yang menandai setiap nilai NaN dalam dataset dengan True.
- Fungsi sum() yang digunakan setelahnya akan menghitung jumlah nilai NaN untuk setiap kolom.
- Kode ini berguna untuk melihat apakah ada kolom yang memiliki nilai NaN dan seberapa banyak jumlahnya.

DECISION KLASIFIKASI DATASET K04_bank_customers

x = K04_bank_customers[:, :-1]

- Pada baris ini, kita mengekstrak semua baris (:) dan semua kolom kecuali kolom terakhir (:-1) dari array K04_bank_customers.
- x di sini akan berisi fitur atau variabel independen dari data bank customer, tanpa kolom target atau label yang berada di kolom terakhir.

y = K04_bank_customers[:, -1]

- Baris ini mengambil semua baris (:) dan hanya kolom terakhir (-1) dari array K04_bank_customers.
- y akan berisi kolom target atau variabel dependen, yang biasanya merupakan label atau kelas yang akan diprediksi.

x = np.asarray(x, dtype=np.float64)

- Baris ini mengonversi x ke dalam array NumPy dengan tipe data float64.
- Tipe data float64 (angka desimal) digunakan untuk memastikan bahwa data fitur berada dalam bentuk numerik yang sesuai untuk analisis atau pemodelan, khususnya dalam algoritma machine learning yang memerlukan data numerik.

y = np.asarray(y, dtype=int)

- Baris ini mengonversi y menjadi array NumPy dengan tipe data int (integer atau bilangan bulat).
- Tipe data int pada y mengindikasikan bahwa variabel target mungkin berupa label atau kelas diskrit, yang seringkali lebih baik disimpan sebagai bilangan bulat.

import numpy as np

- Mengimpor pustaka NumPy dengan alias np. NumPy adalah pustaka Python yang digunakan untuk operasi matematika dan manipulasi array.

from sklearn.model_selection import train_test_split

- Mengimpor fungsi train_test_split dari pustaka scikit-learn, yang berguna untuk membagi dataset menjadi data pelatihan dan data pengujian secara acak.

x = np.random.rand(100, 5)

- Membuat array x berukuran (100, 5) yang berisi 100 sampel data dengan 5 fitur. Setiap nilai dalam array ini merupakan bilangan acak antara 0 dan 1, yang dihasilkan oleh fungsi np.random.rand().
- Misalnya, x mungkin berisi data numerik yang akan digunakan sebagai fitur dalam analisis atau pemodelan.

y = np.random.randint(0, 2, 100)

- Membuat array y yang berisi 100 nilai integer acak, di mana setiap nilai adalah 0 atau 1. Fungsi np.random.randint(0, 2, 100) menghasilkan bilangan bulat acak dari 0 (inklusif) sampai 2 (eksklusif).
- Di sini, y mungkin merupakan label atau target yang menunjukkan dua kelas, misalnya, klasifikasi biner (0 atau 1). **x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)**
- Fungsi train_test_split() digunakan untuk membagi dataset x dan y menjadi data pelatihan dan data pengujian.
- test_size=0.3 menentukan bahwa 30% dari data akan digunakan sebagai data pengujian, sedangkan 70% sisanya sebagai data pelatihan.
- random_state=0 memastikan bahwa pembagian dataset dilakukan secara acak namun tetap konsisten setiap kali kode dijalankan, sehingga hasilnya dapat direproduksi.

print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)

- Menampilkan ukuran dari data pelatihan dan data pengujian untuk memverifikasi bahwa pembagian data sudah benar.
- Karena data berukuran 100 sampel dan menggunakan test_size=0.3, ukuran x_train dan y_train adalah (70, 5) dan (70,), sedangkan ukuran x_test dan y_test adalah (30, 5) dan (30,).

from sklearn.tree import DecisionTreeClassifier

- Pada baris ini, Anda mengimpor DecisionTreeClassifier dari pustaka scikit-learn (atau sklearn).
- DecisionTreeClassifier adalah implementasi dari algoritma pohon keputusan untuk tugas klasifikasi, yang digunakan untuk memodelkan hubungan antara fitur (x) dan target (y) dalam bentuk pohon keputusan. Pohon keputusan digunakan untuk memprediksi label berdasarkan fitur-fitur yang diberikan.**model =**

DecisionTreeClassifier()

- Baris ini membuat objek model dari kelas DecisionTreeClassifier.
- Dengan baris ini, Anda membuat instance dari pohon keputusan yang belum dilatih. Anda dapat menambahkan parameter ke dalamnya untuk menyesuaikan model (misalnya, kedalaman pohon atau kriteria pemisahan), tetapi pada contoh ini menggunakan nilai default.

model.fit(x_train, y_train)

- Fungsi fit() digunakan untuk melatih model dengan data pelatihan.
- x_train berisi fitur (data masukan), dan y_train berisi target atau label yang diinginkan (output).
- Selama proses pelatihan ini, pohon keputusan akan mempelajari pola-pola dalam data untuk memetakan fitur-fitur ke target yang sesuai. Model ini akan menghasilkan struktur pohon keputusan yang dapat digunakan untuk membuat prediksi pada data baru.

import matplotlib.pyplot as plt

- Mengimpor pustaka matplotlib.pyplot dengan alias plt.
- matplotlib adalah pustaka Python yang digunakan untuk membuat visualisasi data. Modul pyplot memungkinkan pembuatan grafik secara mudah, termasuk grafik pohon keputusan.

from sklearn import tree

- Mengimpor modul tree dari pustaka scikit-learn.
- Modul ini menyediakan berbagai alat untuk bekerja dengan pohon keputusan, termasuk fungsi plot_tree yang digunakan untuk menggambarkan pohon keputusan dalam bentuk visual.

plt.subplots(figsize = (12,10))

- Membuat sub-plot atau area gambar dengan ukuran tertentu, di sini menggunakan ukuran (12, 10) inci.
- Fungsi subplots() menghasilkan objek figur dan sumbu untuk membuat grafik atau visualisasi, yang dalam hal ini digunakan untuk menggambar pohon keputusan.
- figsize = (12,10) menentukan ukuran area gambar untuk visualisasi, di mana 12 adalah lebar gambar dan 10 adalah tingginya, yang membuat pohon keputusan lebih mudah dibaca, terutama jika modelnya cukup besar.

tree.plot_tree(model, fontsize = 10)

- Fungsi plot_tree() digunakan untuk menggambarkan pohon keputusan yang telah dilatih, yaitu model model yang merupakan objek dari DecisionTreeClassifier.
- model di sini adalah model pohon keputusan yang sudah dilatih dengan data.
- Parameter fontsize = 10 digunakan untuk mengatur ukuran font dari teks yang ada di dalam visualisasi pohon, agar teks tersebut cukup jelas dan terbaca meskipun ada banyak detail dalam pohon keputusan.

plt.show()

- Fungsi show() digunakan untuk menampilkan visualisasi pohon keputusan yang telah digambar.
- Setelah memanggil plt.show(), jendela grafik akan muncul dengan tampilan pohon keputusan yang menggambarkan proses pembelahan dan keputusan di setiap simpul (node) pohon.**from sklearn.metrics import accuracy_score**
- Mengimpor fungsi accuracy_score dari pustaka scikit-learn.
- accuracy_score digunakan untuk menghitung tingkat akurasi dari model klasifikasi, yaitu persentase prediksi yang benar dibandingkan dengan total jumlah prediksi yang dilakukan.

y_pred = model.predict(x_test)

- Fungsi predict() digunakan untuk membuat prediksi berdasarkan data pengujian (x_test).
- Di sini, model.predict(x_test) menghasilkan prediksi untuk setiap sampel dalam x_test menggunakan model pohon keputusan yang telah dilatih.
- Hasil dari predict() disimpan dalam variabel y_pred, yang berisi label atau kelas yang diprediksi oleh model untuk setiap sampel di x_test.

accuracy_score(y_test, y_pred)

- Fungsi accuracy_score() digunakan untuk menghitung akurasi model dengan membandingkan nilai prediksi (y_pred) dengan nilai yang sebenarnya (y_test).
- y_test adalah label target yang benar dari data pengujian, sedangkan y_pred adalah label target yang diprediksi oleh model.

from sklearn.metrics import classification_report

- Mengimpor fungsi classification_report dari pustaka scikit-learn.
- classification_report digunakan untuk menghasilkan laporan yang mencakup berbagai metrik evaluasi klasifikasi, seperti precision, recall, f1-score, dan support, untuk setiap kelas dalam masalah klasifikasi.

print(classification_report(y_test, y_pred))

- Fungsi classification_report() digunakan untuk menghasilkan laporan yang memberikan beberapa metrik penting tentang kinerja model klasifikasi.
- y_test adalah label asli atau target yang benar dari data pengujian, dan y_pred adalah prediksi yang dihasilkan oleh model.
- Fungsi ini menghitung metrik berikut untuk setiap kelas (misalnya, kelas 0 dan 1 dalam kasus klasifikasi biner)

print(x_test[:5])

- x_test adalah data pengujian yang berisi fitur (misalnya, data masukan atau variabel independen).
- x_test[:5] akan menampilkan 5 sampel pertama dari data pengujian. Ini mengambil baris pertama hingga ke-5 dari array x_test.
- Output ini akan menunjukkan fitur-fitur yang digunakan oleh model untuk membuat prediksi. Biasanya ini berisi data numerik yang dimasukkan ke dalam model.

print(y_pred[:5])

- y_pred adalah prediksi yang dihasilkan oleh model untuk data pengujian (x_test).
- y_pred[:5] akan menampilkan 5 prediksi pertama yang dibuat oleh model untuk data pengujian.
- Output ini menunjukkan label atau kelas yang diprediksi oleh model untuk 5 sampel pertama dari data pengujian. **print(y_test[:5])**

- y_test adalah nilai target yang sebenarnya atau label dari data pengujian (x_test).
- y_test[:5] akan menampilkan 5 label pertama yang sebenarnya untuk data pengujian.
- Output ini menunjukkan kelas yang benar untuk 5 sampel pertama dari data pengujian. Ini digunakan untuk membandingkan hasil prediksi dengan nilai target yang sebenarnya.

K04_bank_customers = np.array([[7.7, 2.8, 6.7, 2.0, 1.0]])

- Di sini, Anda membuat sebuah array NumPy yang berisi satu sampel data baru dengan lima fitur. Data ini diwakili sebagai sebuah array dua dimensi (baris dan kolom).
- K04_bank_customers adalah data input baru yang berisi lima fitur, di mana setiap angka mungkin mewakili atribut tertentu dari data pelanggan bank (misalnya, usia, pendapatan, status, dll). Array ini memiliki ukuran (1, 5), yang berarti hanya ada satu sampel data dengan lima fitur.

print(model.predict(K04_bank_customers))

- model.predict(K04_bank_customers) digunakan untuk membuat prediksi berdasarkan model pohon keputusan (model) yang telah dilatih sebelumnya.
- Fungsi predict() mengambil input data baru (dalam hal ini, data pelanggan bank dalam K04_bank_customers) dan mengembalikan prediksi untuk kelas atau label target berdasarkan model yang sudah dilatih.
- Hasilnya, model akan memprediksi label target untuk data tersebut, misalnya kelas 0 atau 1 dalam kasus klasifikasi biner.
- print() akan menampilkan hasil prediksi tersebut ke layer

TERIMAKASIH