Random Forest

Random Forest adalah model ensemble learning yang menggabungkan banyak pohon keputusan (decision trees) untuk membuat prediksi. Ensemble learning adalah teknik di mana kita menggabungkan beberapa model yang lebih lemah untuk mendapatkan model yang lebih kuat dan akurat.

# Cara Kerja Random Forest:

1. **Bootstrap Sampling:** Random Forest mengambil sampel acak dari data latih dengan penggatian (bootstrap). Setiap sampel ini digunakan untuk melatih satu pohon keputusan.
2. **Random Subspace Method:** Pada setiap node pohon keputusan, hanya subset fitur acak yang dipertimbangkan untuk pemisahan. Ini membantu mengurangi korelasi antar pohon dan mencegah overfitting.
3. **Voting atau Averaging:** Prediksi akhir dari Random Forest adalah hasil voting (untuk klasifikasi) atau averaging (untuk regresi) dari prediksi setiap pohon keputusan.

# Keunggulan Random Forest:

* **Mengurangi Overfitting:** Mengurangi overfitting karena menggabungkan prediksi dari banyak pohon.
* **Kinerja yang Baik:** Seringkali menghasilkan akurasi yang tinggi dan bekerja dengan baik pada berbagai jenis data.
* **Dapat Menangani Fitur Kategorikal dan Numerikal:** Tidak perlu melakukan one-hot encoding untuk fitur kategorikal.
* **Dapat Menangani Data yang Hilang:** Dapat mengatasi nilai yang hilang dengan baik.
* **Memberikan Informasi Pentingnya Fitur:** Dapat menunjukkan fitur mana yang paling penting dalam membuat prediksi.

# Kelemahan Random Forest:

* **Lebih Sulit Diinterpretasikan:** Lebih sulit untuk memahami bagaimana model membuat keputusan dibandingkan dengan pohon keputusan tunggal.
* **Lebih Lambat untuk Dilatih:** Membutuhkan waktu pelatihan yang lebih lama karena melatih banyak pohon.

**Tugas Hari 33:**

Menggunakan Dataset “Social Network Ads”

**Tugas**

1. **Membuat Model Random Forest:**

* Gunakan RandomForestClassifier dari scikit-learn untuk membuat model Random Forest.
* Atur jumlah pohon (n\_estimators) menjadi 100 (atau nilai lain yang Anda inginkan).
* Bagi data menjadi data latih (80%) dan data uji (20%).
* Latih model pada data latih.

1. **Prediksi dan Evaluasi:**

* Gunakan model untuk memprediksi label kelas pada data uji.
* Hitung akurasi, presisi, recall, F1-score dari model pada data uji.

1. **Pentingnya Fitur (Opsional):**

* Jika tertarik, dapat melihat pentingnya fitur menggunakan atribut feature\_importances\_ dari model.

**Contoh Kode (Scikit-Learn)**

from sklearn.ensemble import RandomForestClassifier

# ... (kode untuk membaca dan mempersiapkan dataset Social Network Ads)

# Membagi data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardisasi Fitur

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Membuat model Random Forest

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Melatih model

model.fit(X\_train, y\_train)

# Prediksi pada data uji

y\_pred = model.predict(X\_test)

# Evaluasi

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print(f'Akurasi: {accuracy:.2f}')

print(f'Presisi: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print(f'F1-Score: {f1:.2f}')

# Pentingnya Fitur (Opsional)

importances = model.feature\_importances\_

for feature, importance in zip(X.columns, importances):

print(f'{feature}: {importance:.3f}')

**Selamat Mengerjakan Tugas! 🙂**

**Tugas:**

1. **Membuat Model Random Forest**

**from sklearn.ensemble import RandomForestClassifier**

**# ...(Kode membaca data dan persiapan dataset)**

**# Membagi Data**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**# Standardisasi Fitur**

**scaler = StandardScaler()**

**X\_train = scaler.fit\_transform(X\_train)**

**X\_test = scaler.transform(X\_test)**

**# Membuat Model Random Forest**

**model\_rf = RandomForestClassifier(n\_estimators=100, random\_state=42)**

1. **Prediksi dan Evaluasi**

**# Melatih Model**

**model\_rf.fit(X\_train, y\_train)**

**# Prediksi Data uji**

**y\_pred = model\_rf.predict(X\_test)**

**# Evaluasi**

**accuracy\_rf = accuracy\_score(y\_test, y\_pred)**

**precision\_rf = precision\_score(y\_test, y\_pred)**

**recall\_rf = recall\_score(y\_test, y\_pred)**

**f1\_rf = f1\_score(y\_test, y\_pred)**

**print(f'Akurasi: {accuracy\_rf:.2f}')**

**print(f'Akurasi: {precision\_rf:.2f}')**

**print(f'Akurasi: {recall\_rf:.2f}')**

**print(f'Akurasi: {f1\_rf:.2f}')**

1. **Pentingnya Fitur (Opsional)**

**importtances = model\_rf.feature\_importances\_**

**for feature, importance in zip(X.columns, importtances):**

**print(f'{feature}: {importance:.3f}')**

# Kode Full

Python

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

# 1. Membaca dan Menampilkan Data

df = pd.read\_csv('Social\_Network\_Ads.csv') # Pastikan nama file CSV sudah benar

# Menampilkan 5 baris pertama

print(df.head().to\_markdown(index=False, numalign="left", stralign="left"))

print(df.info())

# 2. Memisahkan Fitur (X) dan Target (y)

X = df[['Age', 'EstimatedSalary', 'Gender']]

y = df['Purchased']

# 3. Encoding Fitur Kategorikal Gender

X = pd.get\_dummies(X, columns=['Gender'], drop\_first=True)

# 4. Membagi Data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 5. Standardisasi Fitur

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# 6. Membuat Model Random Forest

model\_rf = RandomForestClassifier(n\_estimators=100, random\_state=42)

# 7. Melatih Model

model\_rf.fit(X\_train, y\_train)

# 8. Prediksi Data uji

y\_pred = model\_rf.predict(X\_test)

# 9. Evaluasi

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print(f'\nAkurasi: {accuracy:.2f}')

print(f'Presisi: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print(f'F1-Score: {f1:.2f}')

# 10. Pentingnya Fitur (Opsional)

importances = model\_rf.feature\_importances\_

for feature, importance in zip(X.columns, importances):

print(f'{feature}: {importance:.3f}')

Gunakan kode dengan hati-hati.

play\_circleeditcontent\_copy

**Perubahan:**

* Bagian "Pentingnya Fitur":
  + Memperbaiki typo dari importtances menjadi importances.

**Penjelasan Tambahan:**

* **Pentingnya Fitur (feature\_importances\_)**: Atribut ini dalam model Random Forest menunjukkan seberapa penting setiap fitur dalam membuat prediksi. Nilai yang lebih tinggi menunjukkan fitur yang lebih penting.

Dengan perbaikan ini, Anda sekarang dapat melihat pentingnya setiap fitur dalam model Random Forest yang Anda buat. Informasi ini dapat membantu Anda dalam memahami faktor-faktor mana yang paling berpengaruh dalam memprediksi pembelian produk dalam dataset "Social Network Ads".