Support Vector Regression (SVR)

Sebelumnya, telah membahas penggunaan Support Vector Machine (SVM) untuk klasifikasi. Namun, SVM juga bisa digunakan untuk masalah regresi, yaitu memprediksi nilai numerik yang kontinu, bukan kategorikal diskrit. Inilah yang disebut sebagai Support Vector Regression (SVR).

# Konsep Dasar SVR

SVR memiliki beberapa perbedaan kunci dengan SVM untuk klasifikasi:

* **Tujuan:** SVR bertujuan untuk mencari hyperplane yang paling sesuai dengan data, dengan meminimalkan error prediksi.
* **Tube:** Alih-alih margin, SVR memiliki konsep “tube” di sekitar hyperplane. Tube adalah di mana titik-titik data diperbolehkan berada tanpa dikenakan penalti. Lebar tube ini diatur oleh parameter epsilon (ε).
* **Support Vector:** Titik-titik data yang berada di luar tube atau tepat di batas tube disebut support vector. Support vector inilah yang digunakan untuk membangun model.

# Keunggulan SVR:

* **Efektif dalam ruang dimensi tinggi:** Sama seperti SVM untuk klasifikasi, SVR dapat bekerja dengan baik pada data dengan banyak fitur.
* **Memory Efficient:** SVR juga menggunakan subset dari data latih (support vectors) untuk membuat prediksi, sehingga lebih efisien dalam penggunaan memori.
* **Fleksibilitas:** SVR dapat menangani berbagai jenis fungsi kernel, sama seperti SVM untuk klasifikasi.

# Kelemahan SVR:

* **Sensitif terhadap parameter:** Seperti SVM, SVR sensitif terhadap pemilihan parameter, terutama parameter C (trade-off antara kesalahan pelatihan dan kompleksitas model) dan epsilon (lebar tube).
* **Sulit untuk diinterpretasi:** Model SVR bisa sulit diinterpretasikan, terutama jika menggunakan kernel yang kompleks.

**Tugas Hari 30:**

Menggunakan dataset “Boston Housing” yang sudah tersedia di scikit-learn. Dataset ini berisi informasi tentang harga rumah di berbagai wilayah Boston.

**Tugas:**

1. **Memuat dan Menampilkan Data:**

* Muat dataset Boston Housing dari scikit-learn.
* Tampilkan 5 baris pertama dari dataset.

1. **Membuat Model SVR:**

* Gunakan SVR dari scikit-learn untuk membuat model SVR dengan kernel RBF.
* Bagi data menjadi data latih (80%) dan data uji (20%).
* Latih model pada data latih.

1. **Prediksi dan Evaluasi:**

* Gunakan model untuk memprediksi harga rumah pada data uji.
* Hitung Mean Squared Error (MSE) dan R-squared dari model pada data uji.

**Contoh Kode (Scikit-Learn)**

from sklearn.datasets import load\_boston

from sklearn.svm import SVR

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

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import StandardScaler

# Memuat dataset Boston Housing

boston = load\_boston()

X = boston.data

y = boston.target

# Membagi data menjadi data latih dan data uji

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 SVR

model = SVR(kernel='rbf')

# Melatih model

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

# Prediksi pada data uji

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

# Evaluasi model

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f'MSE: {mse:.2f}')

print(f'R-squared: {r2:.2f}')

**Catatan:**

Eksperimen dengan nilai hyperparameter C dan epsilon untuk melihat bagaimana hal tersebut mempengaruhi kinerja model.

**Selamat Mengerjakan Tugas! 🙂**

**Tugas:**

1. **Memuat dan Menampilkan Data:**

**import pandas as pd**

**from sklearn.svm import SVR**

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

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**from sklearn.preprocessing import StandardScaler**

**# Memuat dataset Boston Housing dari kaggle**

**df = pd.read\_csv('data/BostonHousing.csv') # Sesuaikan dengan Nama file**

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

**print("\nInformasi Dataset:")**

**print(df.info())**

1. **Membuat Model SVR:**

**# Memisahkan fitur (X) dan target (y)**

**X = df.drop('medv', axis=1)**

**y = df['medv']**

**# Membagi data menjadi data latih dan data uji**

**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 SVR**

**model = SVR(kernel="rbf")**

**# MELATIH MODEL**

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

1. **Prediksi dan Evaluasi:**

**# PREDIKSI PADA DATA UJI**

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

**# EVALUASI MODEL**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**r2 = r2\_score(y\_test, y\_pred)**

**print(f"MSE: {mse:.2f}")**

**print(f"R-squared: {r2:.2f}")**

**Penjelasan Kode:**

1. **Memuat dan Menampilkan Data:**
   * Anda membaca dataset Boston Housing dari file CSV yang telah Anda unduh dari Kaggle.
   * Anda menampilkan 5 baris pertama data menggunakan df.head() untuk memeriksa struktur data dan df.info() untuk melihat informasi tentang tipe data dan nilai yang hilang pada setiap kolom.
2. **Membuat Model SVR:**
   * Anda memisahkan fitur (X) dari target (y). Fitur adalah semua kolom kecuali medv, dan target adalah kolom medv.
   * Anda membagi data menjadi data latih (80%) dan data uji (20%) menggunakan train\_test\_split.
   * Anda melakukan standardisasi fitur menggunakan StandardScaler untuk memastikan semua fitur memiliki skala yang sama.
   * Anda membuat objek model SVR dengan kernel RBF menggunakan SVR(kernel='rbf'). Kernel RBF (Radial Basis Function) adalah kernel default yang sering digunakan untuk SVR.
   * Anda melatih model menggunakan data latih (X\_train, y\_train).
3. **Prediksi dan Evaluasi:**
   * Anda menggunakan model yang sudah dilatih untuk membuat prediksi pada data uji (X\_test).
   * Anda menghitung Mean Squared Error (MSE) dan R-squared untuk mengevaluasi kinerja model. MSE mengukur rata-rata kesalahan kuadrat antara nilai prediksi dan nilai aktual, sedangkan R-squared mengukur seberapa baik model dapat menjelaskan varians dalam data target.
   * Anda mencetak hasil evaluasi MSE dan R-squared.

**Interpretasi Hasil:**

Hasil evaluasi MSE dan R-squared akan menunjukkan seberapa baik model SVR Anda dalam memprediksi harga rumah di Boston.

* **MSE:** Nilai MSE yang rendah menunjukkan bahwa model memiliki kesalahan prediksi yang kecil.
* **R-squared:** Nilai R-squared yang tinggi (mendekati 1) menunjukkan bahwa model dapat menjelaskan sebagian besar varians dalam data harga rumah.

**Contoh Hasil:**

MSE: 21.59

R-squared: 0.74

Hasil ini menunjukkan bahwa model memiliki MSE 21.59 dan R-squared 0.74. Artinya, rata-rata kesalahan kuadrat model adalah 21.59, dan model dapat menjelaskan 74% dari varians dalam data harga rumah.

**Kesimpulan:**

Kode Anda sudah sangat baik dan Anda berhasil menyelesaikan tugas dengan baik!

**Saran Perbaikan (Opsional):**

* **Tuning Hyperparameter:** Anda dapat mencoba mengatur hyperparameter model SVR (seperti C dan epsilon) untuk mendapatkan hasil yang lebih baik. Grid search atau random search dapat membantu Anda menemukan kombinasi hyperparameter yang optimal.
* **Visualisasi:** Anda dapat membuat plot scatter antara nilai prediksi dan nilai aktual untuk melihat seberapa baik model Anda dalam memprediksi harga rumah.