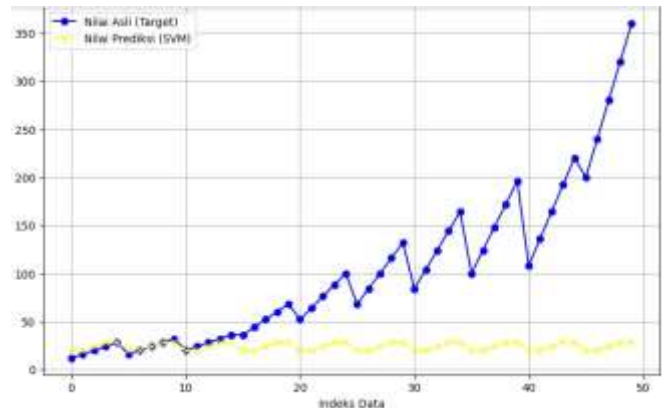


Praktikum Fisika Komputasi Senin, 2 Desember 2024

Support Vector Machine



Grafik di atas menunjukkan perbandingan antara nilai asli (garis biru) sama nilai prediksi dari model SVM (garis kuning). Kalau dilihat, nilai asli jelas punya pola yang terus naik, terutama setelah indeks data sekitar 20-an ke atas, bahkan lonjakannya makin lama makin tinggi. Tapi, prediksi dari model SVM kelihatan datar-datar aja, nggak bisa ngikutin kenaikan yang ada di nilai asli. Hasilnya, prediksi dan data aslinya jadi kelihatan jauh banget beda.

Masalah kayak gini biasanya muncul karena model yang dipakai belum maksimal atau nggak cocok buat data yang pola naik-turunnya kayak gitu. Kemungkinan, model ini ngalamin yang namanya underfitting, alias nggak cukup belajar buat ngerti pola data. Selain itu, bisa juga ada yang kurang pas di pengolahan datanya, misalnya fitur yang dipakai buat melatih model kurang sesuai sama data aslinya, jadi hasil prediksinya juga nggak akurat.

Dari grafik ini juga kelihatan kalau model SVM cuma lumayan pas buat ngikutin pola di bagian awal, waktu nilai asli masih rendah dan nggak terlalu ribet. Tapi pas nilai asli mulai naik drastis, model kayak nggak paham dan tetep prediksi di angka yang hampir sama terus. Ini berarti model kurang fleksibel buat nangkep pola data yang lebih kompleks atau non-linear.

Supaya hasil prediksinya lebih bagus, ada beberapa hal yang bisa dicoba. Misalnya, data yang mau dipakai dilatih dulu, kayak di-normalisasi atau di-standarisasi, biar model lebih gampang ngolahnya. Terus, parameter di model SVM juga bisa disesuaikan, kayak kernel, C , atau γ , supaya performanya lebih optimal. Kalau ternyata setelah diutak-atik masih belum bisa juga, mungkin perlu ganti ke model lain yang lebih canggih, kayak neural network, karena lebih bisa nangkep pola data yang naik-turun drastis.

Intinya, grafik ini nunjukin kalau model SVM yang sekarang belum bisa ngikutin pola dari data aslinya. Jadi, perlu ada langkah perbaikan, entah dari sisi data, model, atau parameternya, biar hasil prediksi lebih akurat dan mendekati nilai aslinya.

Algoritma program integral trapezoid itu dipakai buat ngitung nilai integral suatu fungsi secara pendekatan atau aproksimasi. Intinya, kita nggak menghitung integral secara eksak pakai rumus matematika langsung, tapi kita coba mendekati hasilnya dengan cara memecah interval integrasi jadi beberapa bagian kecil, lalu menghitung luas tiap bagian itu dengan metode trapezoid. Metode ini sederhana tapi cukup efektif buat dapetin nilai integral, terutama kalau fungsi yang mau diintegalkan susah dihitung secara langsung. Jadi, langkah-langkah utamanya dimulai dari menentukan batas bawah dan batas atas dari integral, yang di kode ini disebut sebagai a dan b . Nah, fungsi yang mau diintegalkan juga disiapkan, di sini diwakili dengan f . Interval antara batas atas dan bawah dibagi jadi beberapa bagian kecil, jumlahnya diatur sama variabel n (di sini $n = 100$). Semakin besar nilai n , biasanya hasil integral makin mendekati nilai sebenarnya, karena pembagian intervalnya jadi lebih halus. Setelah intervalnya dibagi, kita mulai menghitung nilai fungsi di tiap titik dalam interval itu. Jadi, untuk setiap titik, kita cari nilai $f(x)$, terus kita tambahkan semua nilai fungsi dari titik-titik itu. Nilai integral akhirnya dihitung pakai rumus trapezoid, yang intinya ngitung luas dari trapesium-trapesium kecil di bawah kurva fungsi. Nah, di kode itu, setelah integral dihitung untuk satu fungsi, programnya ngulang buat beberapa fungsi lain. Contohnya, ada fungsi $2x$, $2x+2$, $4x+6$, dan seterusnya. Looping ini dibuat untuk ngumpulin banyak data hasil integral dari berbagai fungsi dengan interval yang berbeda-beda. Semua hasil ini nanti dipakai buat bikin dataset, yang isinya adalah kombinasi antara input (kayak batas bawah, batas atas, sama fungsi yang diintegalkan) dan output (nilai integral yang udah dihitung). Setelah dataset ini siap, langkah berikutnya adalah memakai metode Support Vector Machine (SVM) buat bikin prediksi nilai integral. Jadi, alurnya kayak gini: pertama, data yang udah dihitung tadi dijadiin bahan buat "ngajarin" model SVM. Model ini bakal belajar pola dari hubungan antara input (kayak interval dan fungsi) dengan outputnya (nilai integral). Setelah model selesai dilatih, kita bisa pakai model ini buat prediksi nilai integral fungsi lain tanpa perlu ngitung manual lagi pakai metode trapezoid.