Supervised Learning - Regression

by Thio Perdana

OUTLINE

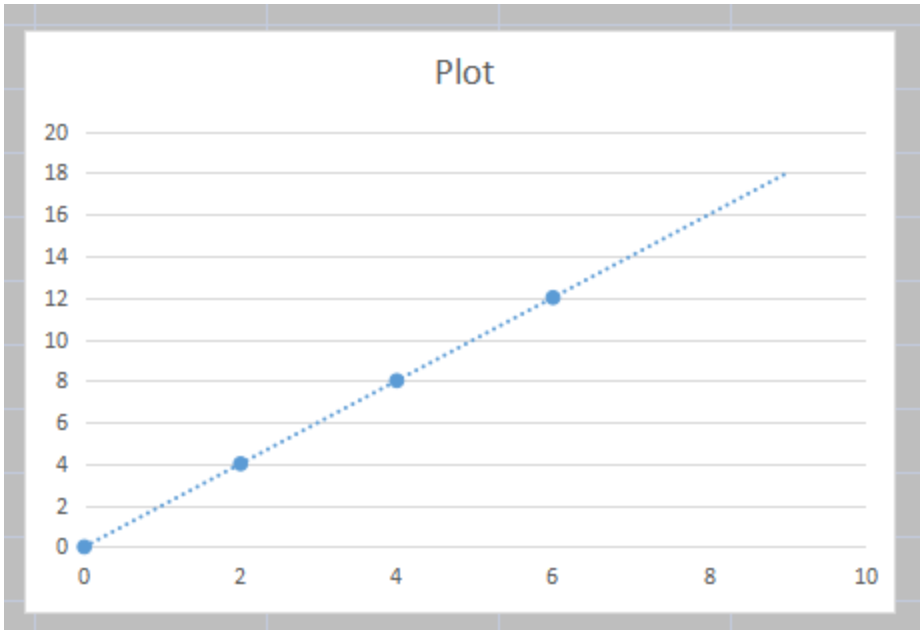
Pendahuluan

Supervised learning adalah salah satu paradigma dalam machine learning di mana algoritma belajar dari data yang telah diberi label. Artinya, kita memberi tahu algoritma tentang hubungan antara input dan output dengan menyediakan contoh-contoh yang sudah diketahui. Misalnya, jika kita ingin mengajarkan komputer untuk mengenali gambar kucing, kita akan memberikan kumpulan data gambar kucing yang sudah diberi label "kucing".

Pada saat kita mempelajari matematika dasar, tentu kita sudah tahu dengan apa yang dinamakan persamaan garis. Terdapat berbagai macam persamaan garis yang kita pelajari, mulai dari persamaan garis lurus, hingga polynomial. Tujuan dari mempelajari persamaan garis manapun sama, yaitu agar kita dapat memetakan data yang sudah ada ke dalam suatu persamaan lalu persamaan tersebut kita bisa pakai untuk mengetahui letak data-data yang bahkan belum ada. Sebagai contoh, misalkan kita mempunya dua buah titik seperti terlihat di bawah ini.

x	y
0	0
2	4
4	8
6	12

maka data tersebut bisa kita plot lalu kita cari persamaan garis yang bersesuaian



Dengan adanya persamaan garis tersebut, maka kita pun jadi dapat menebak nilai titik y jika kita mengetahui nilai x nya. Inilah gambaran sederhana tentang regresi. Regresi adalah sebuah metode yang digunakan untuk menemukan relasi antara variabel/fitur (pada contoh di atas, hubungan x dan y). Dengan Regresi kita dapat menemukan persamaan regresi yang paling cocok atau best to fit, yang dapat digunakan untuk melakukan prediksi atau *forecasting.* Data yang digunakan untuk regresi biasanya adalah data yang bersifat continu dan/atau *time series*. Secara definisi yang dimaksud dengan regresi yaitu,

The process of finding a model or function for distinguishing the data into continunous real values instead of using classes. Mathematically, with a regression problem, one is trying to find the fucntion approximation wtih the minimum error deviation. In regression, the data numeric dependency is predicted to distinguish it.

Contoh kasus penggunaan regresi lainnya misalkan kita ingin memprediksi harga rumah berdasarkan beberapa fitur tertentu. Kita memiliki data yang berisi informasi tentang luas tanah, jumlah kamar, lokasi

geografis, dan harga rumah yang sudah tercatat. Dengan menggunakan regresi, kita dapat mengembangkan model yang dapat mempelajari pola dari data ini dan kemudian digunakan untuk memprediksi harga rumah untuk data baru yang belum terlihat sebelumnya. Dengan kata lain, regresi membantu kita memahami hubungan antara variabel input (fitur) dan variabel output (harga rumah) sehingga kita dapat membuat prediksi yang masuk akal berdasarkan data yang telah kita miliki.

Tujuan utama dari regresi adalah memprediksi atau menjelaskan nilai variabel dependen berdasarkan nilai variabel independen. Pada dasarnya, kita ingin memahami pola atau hubungan matematis di antara variabel-variabel ini. Namun, dalam konteks "memprediksi nilai kontinu," ada beberapa hal yang perlu dicatat:

1. **Nilai Kontinu:** Regresi digunakan ketika variabel dependen adalah variabel kontinu, artinya nilainya dapat berupa bilangan riil atau sepanjang rentang tertentu.
2. **Prediksi:** Regresi memberikan kita alat untuk membuat prediksi tentang nilai variabel dependen berdasarkan nilai variabel independen yang kita miliki. Ini membantu dalam membuat estimasi atau proyeksi untuk keperluan bisnis atau penelitian.

Perbedaan antara Klasifikasi dan Regresi

Klasifikasi: Digunakan ketika output yang diinginkan adalah kategori atau label diskrit. Contohnya, memprediksi apakah email masuk ke dalam kotak masuk atau spam.

Regresi: Digunakan ketika output yang diinginkan adalah nilai kontinu. Contohnya, memprediksi harga rumah berdasarkan berbagai fitur seperti luas tanah, jumlah kamar, dll.

Perbedaan utama terletak pada jenis output yang dihasilkan: klasifikasi untuk kategori dan regresi untuk nilai yang dapat beragam.

Konsep Variabel Dependan dan Independen

Dalam konteks regresi, kita berurusan dengan dua jenis variabel utama: variabel dependen dan variabel independen. Mari kita bahas keduanya dengan lebih rinci.

Variabel Dependan

Variabel dependen adalah variabel yang ingin kita prediksi atau jelaskan. Dalam beberapa kasus, ini juga disebut sebagai variabel respons atau target. Dalam hubungan regresi, variabel dependen adalah fokus utama analisis kita. Misalnya, jika kita ingin memprediksi harga rumah (variabel dependen), kita akan mencari faktor-faktor yang memengaruhi harga tersebut.

Contoh:

Harga Rumah: Variabel dependen dalam kasus ini adalah harga rumah yang ingin kita prediksi berdasarkan variabel independen tertentu seperti luas tanah, jumlah kamar, dan lokasi.

Variabel Independen

Variabel independen adalah variabel yang digunakan untuk memprediksi atau menjelaskan variabel dependen. Jumlah dan jenis variabel independen dapat bervariasi tergantung pada kompleksitas model. Variabel independen juga dikenal sebagai prediktor atau fitur.

Contoh:

Luas Tanah, Jumlah Kamar, Lokasi: Variabel independen dalam kasus harga rumah mungkin mencakup luas tanah, jumlah kamar, dan lokasi. Variabel ini digunakan untuk membentuk model yang dapat memprediksi harga rumah.

Hubungan Antara Keduanya

Tujuan Utama: Tujuan utama regresi adalah memahami dan mendefinisikan hubungan antara variabel dependen dan independen. Dengan kata lain, bagaimana perubahan dalam variabel independen dapat memengaruhi variabel dependen.

Contoh Rumus Sederhana: Dalam regresi sederhana, kita dapat memiliki rumus umum seperti: `Y = a + bX`, di mana Y adalah variabel dependen, X adalah variabel independen, a adalah intercept, dan b adalah koefisien regresi.

Pahami bahwa variabel dependen dan independen bervariasi tergantung pada konteks masalah yang sedang dihadapi. Dengan konsep ini, kita dapat membangun model regresi untuk menganalisis dan memprediksi hubungan antar variabel dalam berbagai situasi.

Jenis Algoritma Regresi

secara garis besar, ada 5 buah regresi yang sering dipakai dan digunakan dengan tujuan *forecasting*

Linear Regression

Polynomial Regression

Ridge & Lasso Regression

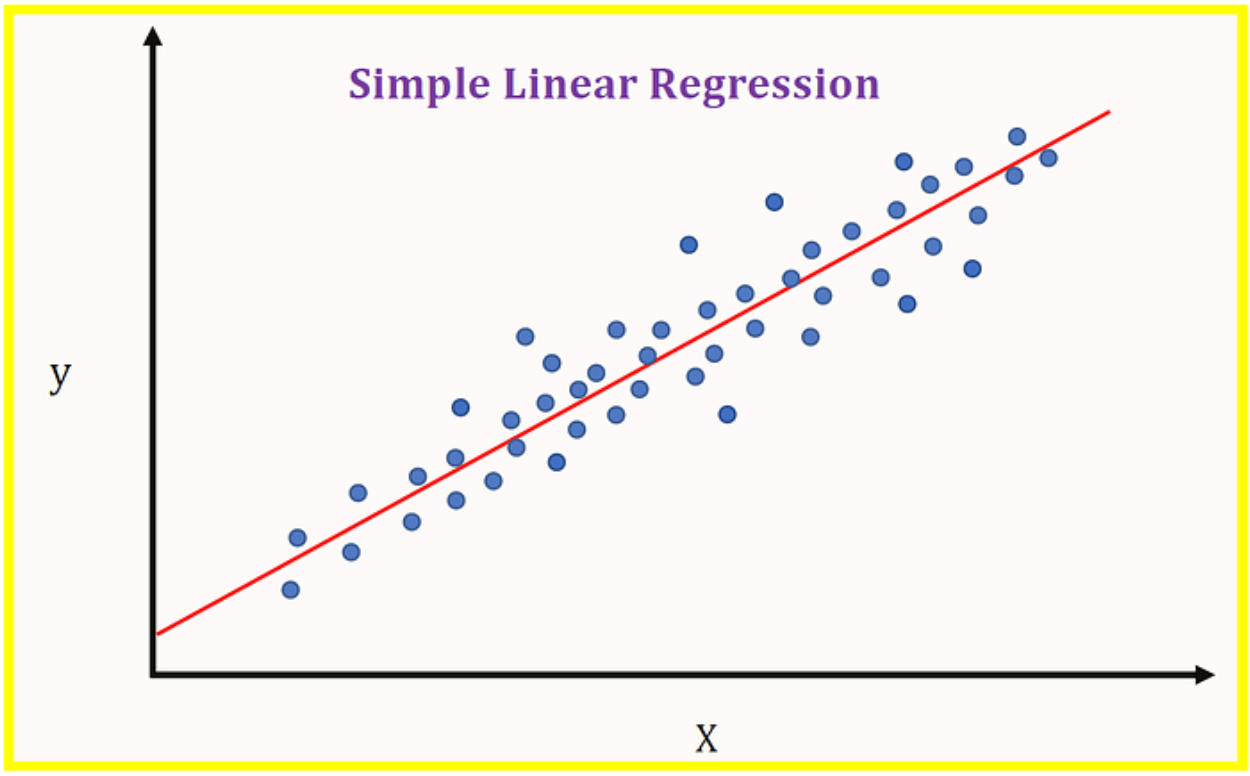
Support Vector Regression

Decision Tree & Random Forest Regression

Linear Regression

Linear regression adalah jenis regresi yang paling sederhana. Setiap poin-poin data akan berusaha di cocokkan ke dalam sebuah persamaan garis lurus. Linear regression digunakan untuk sembarang permasalahan yang berkaitan dengan prediksi, seperti:

- memprediksi stock price atau ekonomi
- memprediksi perkiraan gaji
- memprediksi curah hujan di suatu daerah
- memprediksi kenaikan harga tanah
- dan sebagainya



Kelebihan dari linear regression

- Mudah diimplementasikan
- Sederhana dan tidak terlalu kompleks.

Kekurangan dari linear regression

- Outlier sangat mempengaruhi model
- Terlalu menyederhanakan permasalahan dengan menganggap hubungan antar variabel/fitur adalah linear.

Persamaan Umum

Persamaan Umum (*Cost Function)* dari Linear Regression adalah

$$y = mx + c$$

Dengan.

Y = variabel hasil / prediksi

X = variabel bebas. biasanya fitur seperti *time-series* dimasukkan disini

m = gradien/kemiringan

c = titik perpotongan dengan sumbu y (intercept)

Atau untuk regresi linear multiple parameter

$$Y_i = b_0 + b_1X_{1i} + b_2X_{2i} + b_nX_{ni} + u_i$$

Y_i = dependent variable
b_0 = Intercept
$b_1 \ldots b_n$ = Coefficient of Regression
$X_{1i} \ldots X_{ni}$ = independent variable
u_i = disturbance error

Polynomial Regression

Regresi Polinomial adalah teknik yang menggunakan persamaan berpangkat (degree) untuk memetakan setiap poin dari data kita. Regresi polinomial adalah turunan dari regresi linear menggunakan konsep yang hampir mirip dengan linear regression akan tetapi dapat lebih baik dalam merepresentasikan data yang lebih acak.

Untuk mengimplementasikan regresi polinomial, langkah yang perlu ditambahkan pada model regresi linear.

1. Lakukan pemisahan data training dan data testing.

2. Gunakan `PolynomialFeatures` untuk mentransformasi fitur menjadi polinomial.

```
degree = 2  # derajat polinomial

poly = PolynomialFeatures(degree)

X_train_poly = poly.fit_transform(X_train)

X_test_poly = poly.transform(X_test)
```

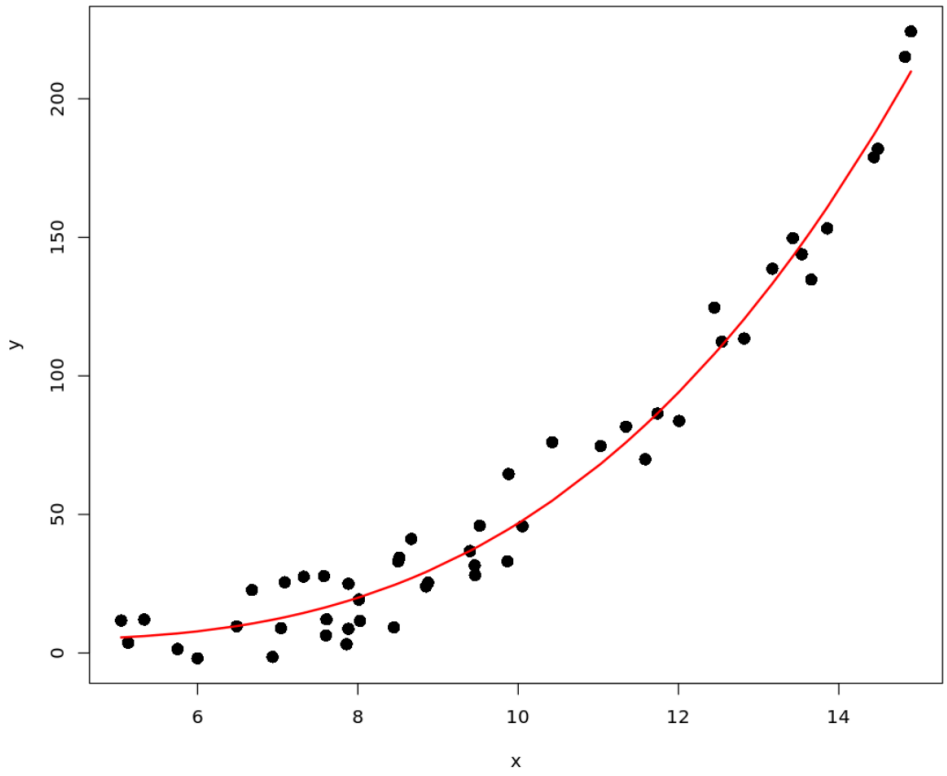
3. Inisialisasi model regresi linear dan latih model dengan data yang telah diubah.

```
model = LinearRegression()

model.fit(X_train_poly, y_train)
```

Penggunaannya hampir sama dengan linear, yaitu:

- memprediksi perkiraan gaji
- memprediksi curah hujan di suatu daerah
- memprediksi kenaikan harga tanah
- dan sebagainya



Kelebihan:

- Lebih baik dalam memetakan data kompleks dibanding linear regression

Kekurangan:

- Penentuan nilai degree (pangkat harus tepat)
- Implementasi syntax lebih rumit

Persamaan Umum

Persamaan Umum (*Cost Function)* dari Polynomial Regression adalah

Dengan.

Y = variabel hasil / prediksi

X = variabel bebas. biasanya fitur seperti *time-series* dimasukkan disini

b1,b2 = gradien/kemiringan

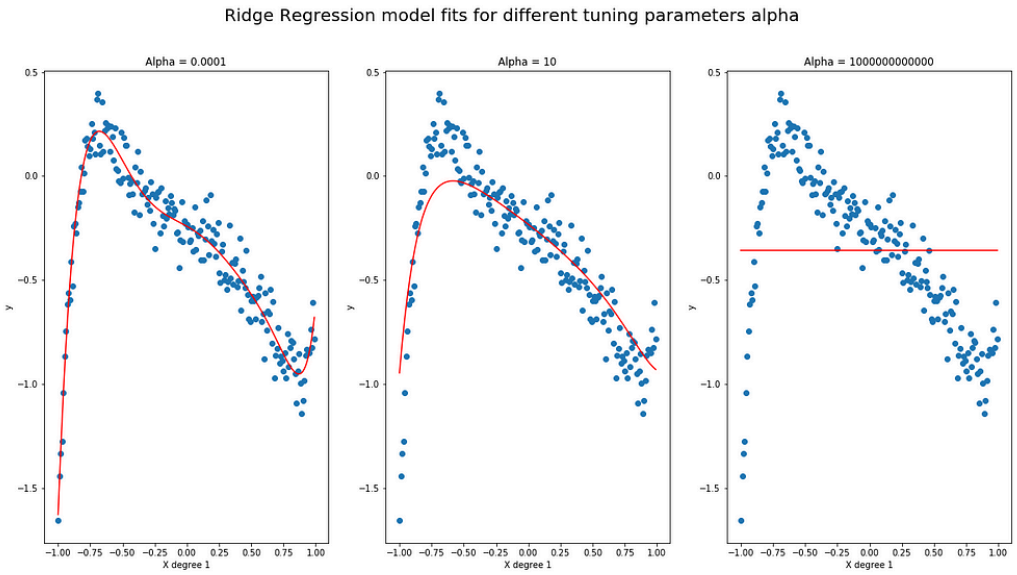
b0 = titik perpotongan dengan sumbu y

n = degree

Ridge & Lasso Regression

Ridge regression adalah metode penyetelan model yang digunakan untuk menganalisis data yang mengalami multicollinearity. Metode ini melakukan regularisasi L2. Ketika masalah multicollinearity terjadi, least-squares tidak bias, dan variansnya besar, yang mengakibatkan nilai yang diprediksi jauh dari nilai aktual.

Lambda adalah istilah hukuman. λ yang diberikan di sini ditandai dengan parameter alpha dalam fungsi ridge. Jadi, dengan mengubah nilai alpha, kita mengendalikan istilah hukuman. Semakin tinggi nilai alpha, semakin besar hukuman dan, oleh karena itu, magnitudo koefisien berkurang.



Ini mengecilkan parameter. Oleh karena itu, digunakan untuk mencegah multicollinearity Mengurangi kompleksitas model dengan penyusutan koefisien Lihat kursus gratis tentang analisis regresi. Model Regresi Ridge Untuk jenis model regresi mesin pembelajaran apa pun, persamaan regresi biasa membentuk dasar yang ditulis sebagai:

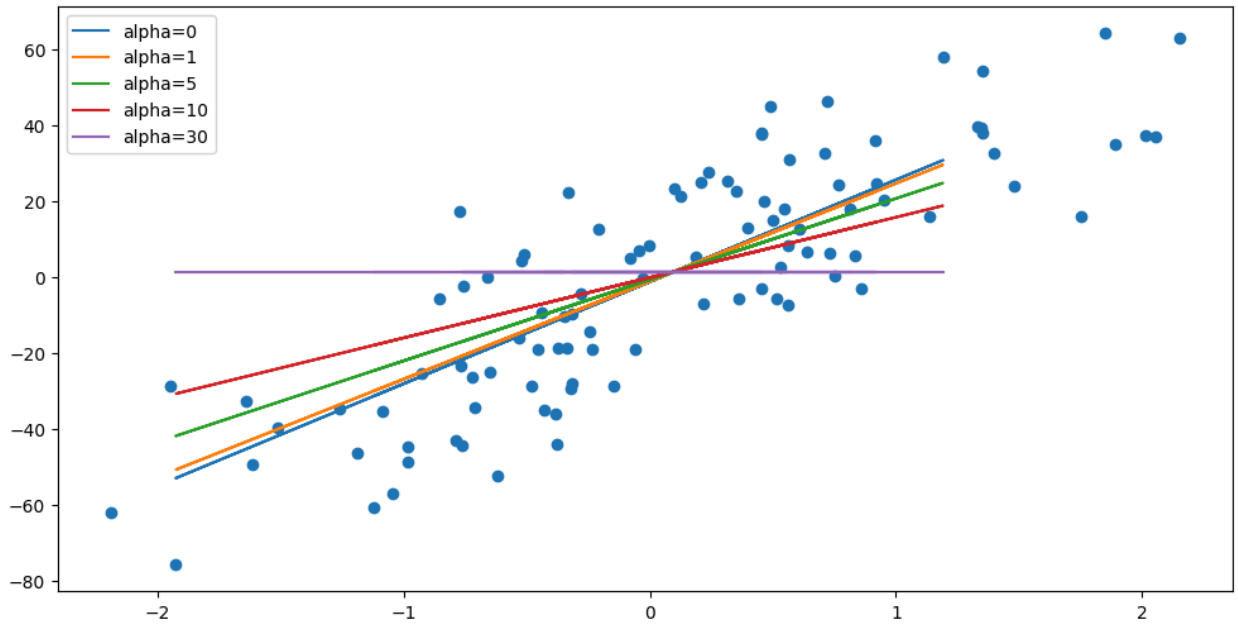
$$Y = XB + e$$

Di mana Y adalah variabel dependen, X mewakili variabel independen, B adalah koefisien regresi yang akan diestimasi, dan e mewakili kesalahan atau residual. Setelah kita menambahkan fungsi lambda ke

dalam persamaan ini, varians yang tidak dievaluasi oleh model umum dipertimbangkan. Setelah data siap dan diidentifikasi sebagai bagian dari regularisasi L2, ada langkah-langkah yang dapat diambil.

Regresi LASSO, juga dikenal sebagai regularisasi L1, adalah teknik populer yang digunakan dalam pemodelan statistik dan machine learning untuk memperkirakan hubungan antara variabel dan membuat prediksi. LASSO singkatan dari Least Absolute Shrinkage and Selection Operator.

Tujuan utama dari regresi LASSO adalah menemukan keseimbangan antara kesederhanaan model dan akurasi. Ini dicapai dengan menambahkan istilah hukuman ke model regresi linear tradisional, yang mendorong solusi yang jarang di mana beberapa koefisien dipaksa menjadi nol. Fitur ini membuat LASSO sangat berguna untuk seleksi fitur, karena dapat secara otomatis mengidentifikasi dan menyingkirkan variabel yang tidak relevan atau berlebih.



Regresi LASSO adalah teknik regularisasi. Ini digunakan di atas metode regresi untuk prediksi yang lebih akurat. Model ini menggunakan penyusutan. Penyusutan adalah di mana nilai data menyusut ke titik

sentral seperti rata-rata. Prosedur lasso mendorong model yang sederhana dan jarang (yaitu model dengan parameter lebih sedikit). Jenis regresi ini sangat cocok untuk model yang menunjukkan tingkat multicollinearity yang tinggi atau ketika Anda ingin mengotomatisasi beberapa bagian dari pemilihan model, seperti seleksi variabel/eliminasi parameter.

Linear vs Ridge vs Lasso

Regresi linear (di scikit-learn) merupakan bentuk paling dasar di mana model tidak dikenakan hukuman atas pilihan bobotnya sama sekali. Artinya, selama tahap pelatihan, jika model merasa bahwa satu fitur tertentu sangat penting, model dapat memberikan bobot besar pada fitur tersebut. Hal ini kadang-kadang dapat menyebabkan overfitting pada dataset kecil. Oleh karena itu, metode-metode berikut ini dikembangkan.

Lasso adalah modifikasi dari regresi linear, di mana model dikenai hukuman untuk jumlah nilai mutlak dari bobotnya. Dengan demikian, nilai mutlak bobot akan (secara umum) berkurang, dan banyak yang cenderung menjadi nol.

Ridge mengambil langkah lebih jauh dan memberikan hukuman pada model untuk jumlah nilai kuadrat dari bobotnya. Dengan demikian, bobot tidak hanya cenderung memiliki nilai mutlak yang lebih kecil, tetapi juga cenderung memberikan hukuman pada ekstrem dari bobot, menghasilkan sekelompok bobot yang lebih merata

Pentingnya Pemilihan Model

Regresi Ridge: Cocok ketika kita percaya bahwa sebagian besar variabel memiliki dampak pada output.

LASSO: Cocok ketika kita percaya bahwa hanya sejumlah kecil variabel yang benar-benar penting.

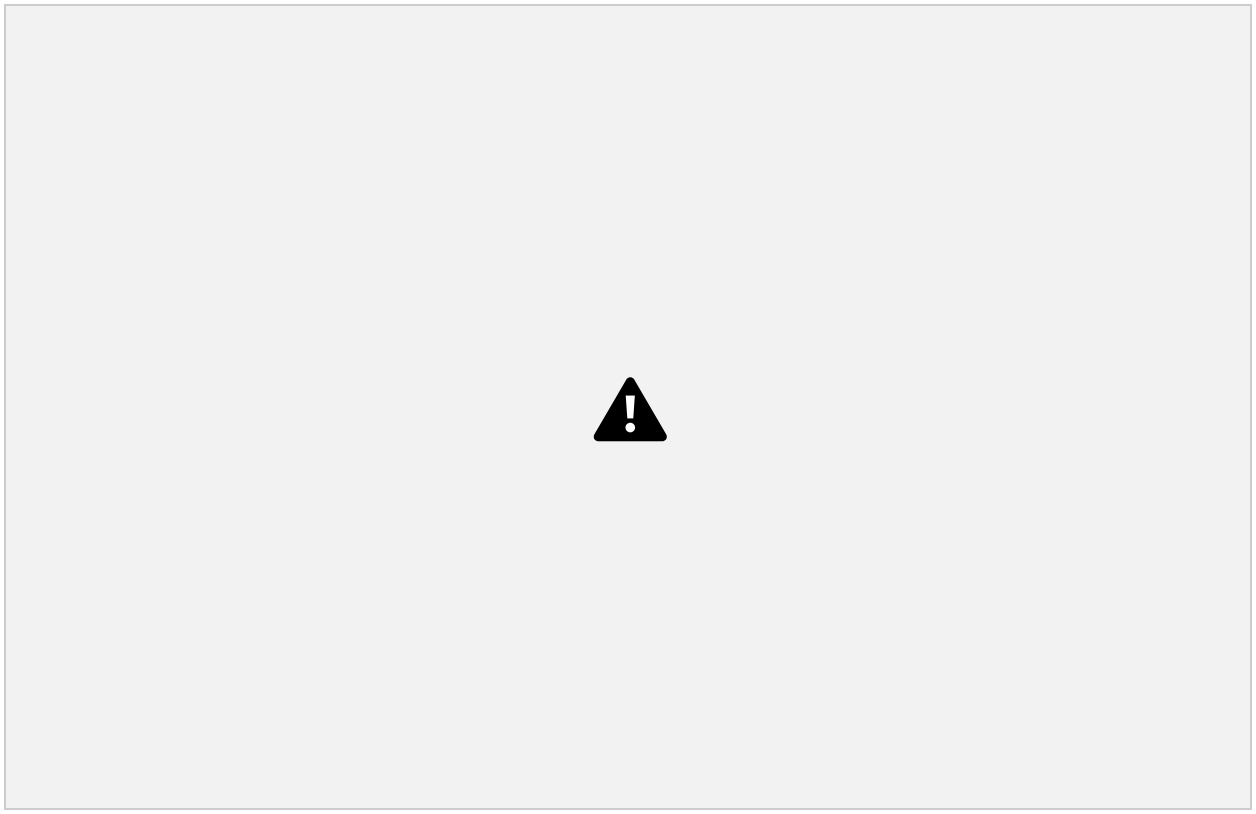
Manfaat

Mencegah Overfitting: Regresi Ridge dan LASSO membantu mencegah overfitting dengan membatasi kompleksitas model.

Seleksi Variabel: LASSO dapat berperan dalam seleksi variabel, menghasilkan model yang lebih sederhana dengan hanya mempertahankan variabel yang paling informatif.

Support Vector Regression

SVR adalah aplikasi regresi dari konsep SVM. Digunakan dengan menarik batas maksimum dan minimum dari suatu hyperplane. Garis pusat dari hyperplane inilah yang nanti akan dijadikan acuan untuk forecasting.



Kelebihan:

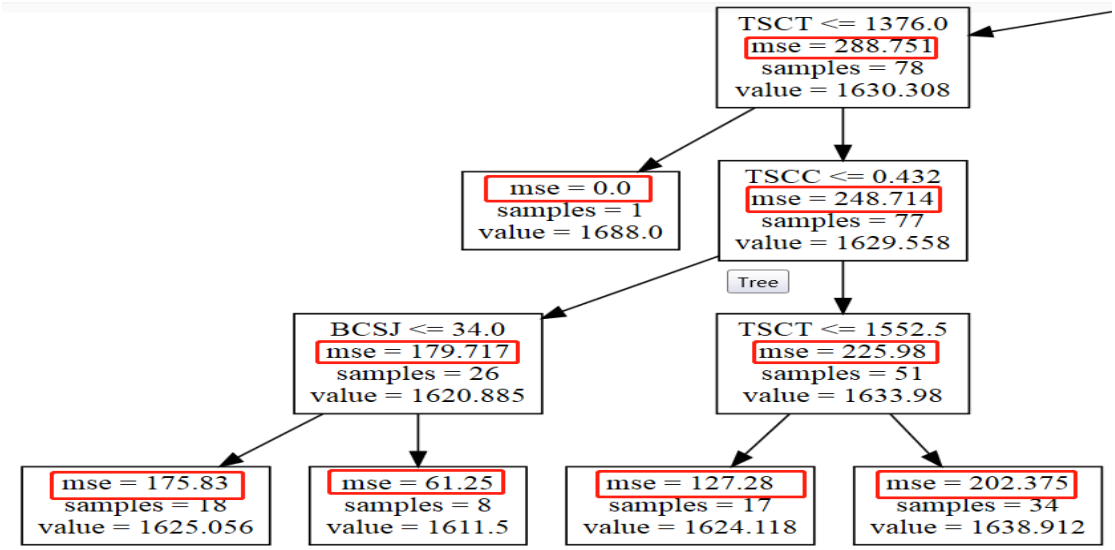
- Dapat diandalkan untuk menghadapi outlier
- Kemampuan generalisasi yang baik
- Akurasi prediksi tinggi

Kekurangan:

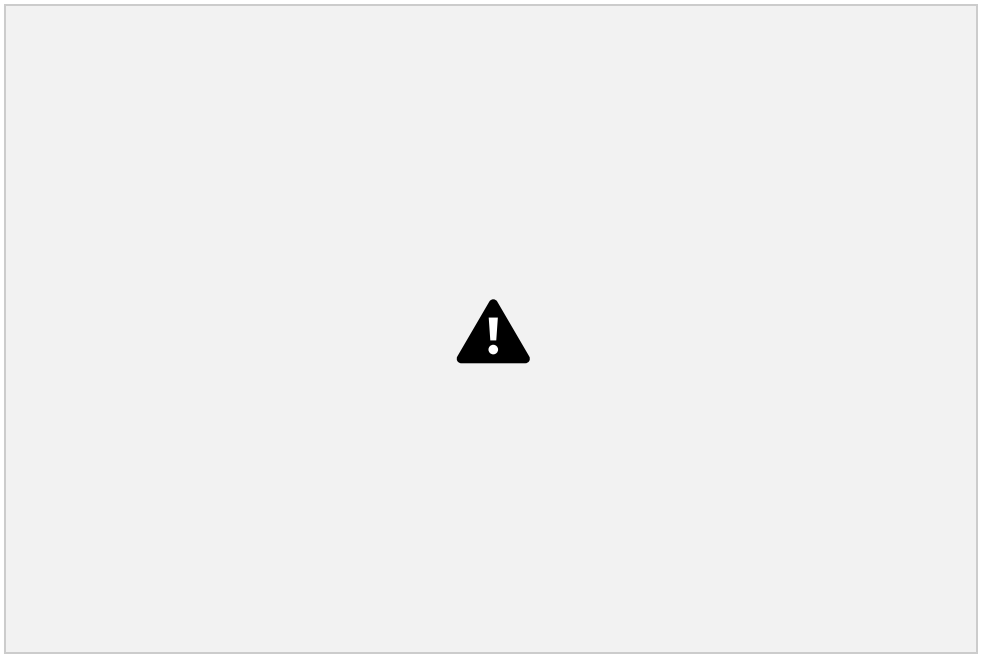
- Tidak cocok untuk dataset yang banyak
- Performa kurang baik bila terlalu banyak noise pada dataset

Decision Tree & Random Forest Regression

Decision Tree memiliki dua konsep yang bisa diaplikasikan, yaitu regression dan classification. Decision Tree sendiri menggunakan konsep pohon bercabang yang setiap *branch* nya akan menjadi percabangan keputusan/fitur dari dataset kita.



setiap node yang berwarna biru adalah fitur dari data kita, sedangkan tulisan dalam percabangannya adalah nilai dari fitur tersebut. sedangkan yang berwarna hijau adalah label dari setiap data. Agar lebih mudah memahami maka bisa coba lihat tabeldi bawah ini. .



Random Forest sendiri adalah kumpulan dari algoritma decision tree yang diproses bersama, sama seperti namanya, forest yang merupakan kumpulan dari tree.

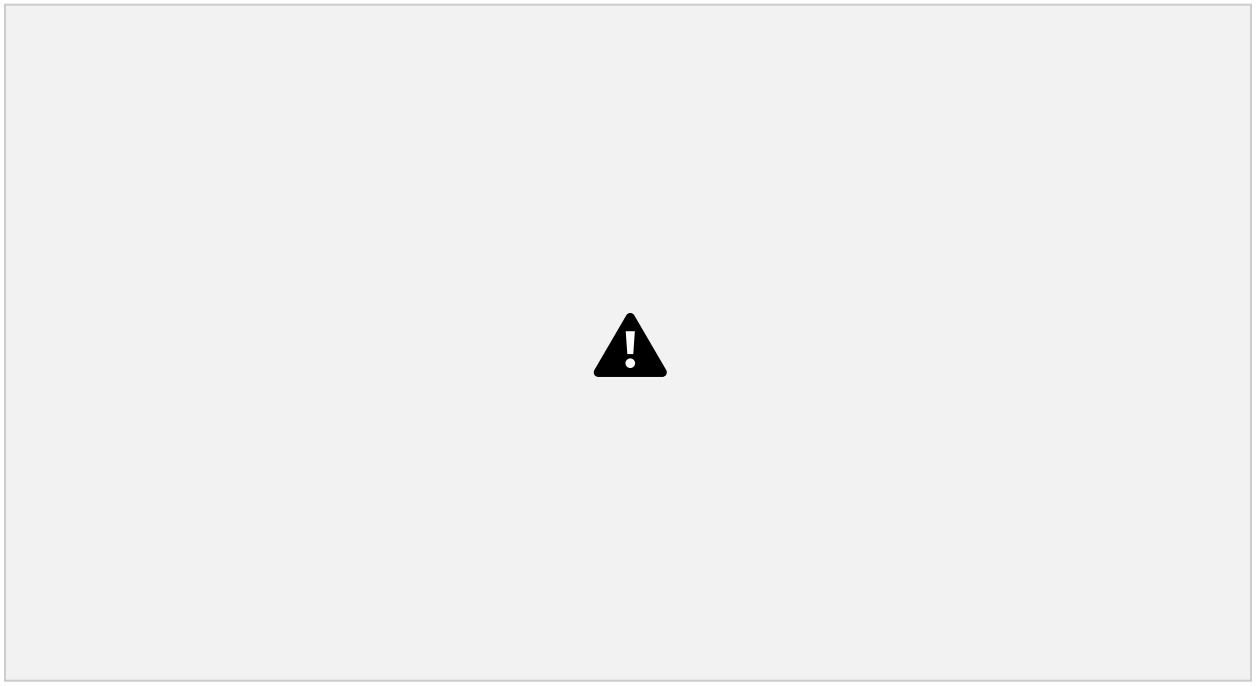
Kelebihan:

- Lebih mudah dimengerti, visualisasi sederhana
- Bekerja dengan baik tidak hanya untuk data numerik tapi juga kategorikal
- Memerlukan persiapan data yang minimum

Cons:

- Sering terjadi overfit
- Perubahan sedikit pada data akan mengubah struktur cukup signifikan.

How To Choose



Regresi dengan Python

Sekarang kita akan coba untuk melakukan forecasting sederhana dengan menggunakan Linear Regression.

Silahkan unduh data yang diperlukan

unduh dataset

langkah pertama yang akan kita lakukan adalah dengan membuka data tersebut dan kita masukkan pada dataframe panda

```
import pandas as pd

raw_data = pd.read_csv("bensin.csv")
print(raw_data.head())
```

```
   Liter  Kilometer
0     20      142.0
1     25      177.0
2     20      144.0
3     30      203.0
4     40      273.0
```

Untuk mengetahui informasi tentang dataset, kita kan menggunakan describe()

```
raw_data.describe()
```
✓ 0.7s

	Liter	Kilometer
count	65.000000	65.000000
mean	26.446154	181.064615
std	7.424686	49.741763
min	6.000000	32.000000
25%	23.000000	144.000000
50%	25.000000	177.000000
75%	30.000000	212.000000
max	45.000000	278.000000

Dilihat sekilas dari laporan di atas, data terdistribusi secara normal. Karena rentang data berada di puluhan hingga ratusan. kita akan coba tanpa menggunakan transformasi pada kesempatan ini.

Sekarang kita coba untuk membagi dataset kita. Agar lebih mudah dipahami dan lebih sederhana, dataset kita akan kita bagi hanya menjadi dua jenis, yaitu training & test. Training 80% dan Test 20%.

```
import numpy as np
from sklearn.model_selection import train_test_split as tts
```

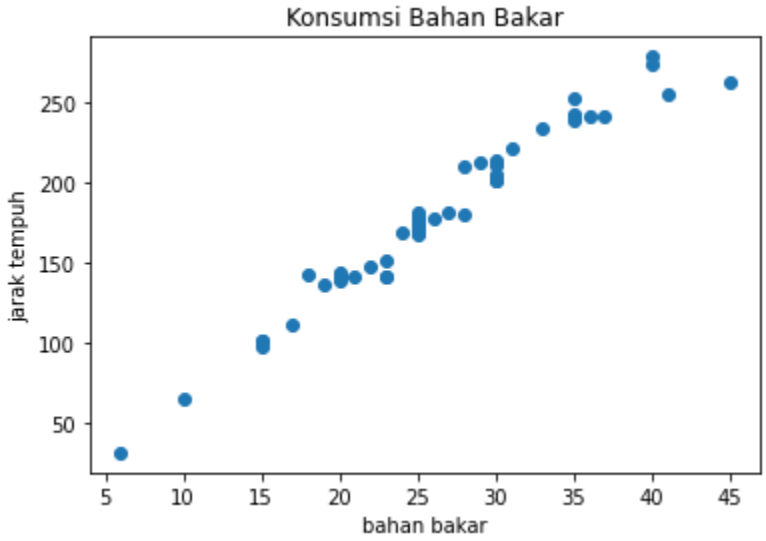
```
liter = raw_data[['Liter']]
jarak = raw_data[['Kilometer']]

X_train, X_test, y_train, y_test = tts(liter, jarak, random_state=21, test_size = 0.2)
```

Untuk melihat bagaimana struktur data training yang kita miliki, kita bisa melakukan plot terhadap data tersebut

```
import matplotlib.pyplot as plt

plt.scatter(X_train, y_train)
plt.xlabel("bahan bakar")
plt.ylabel("jarak tempuh")
plt.title("Konsumsi Bahan Bakar")
plt.show()
```



Terlihat bentuk data akan cukup baik bila dilakukan *forecasting* dengan menggunakan linear regression.

Sekarang mari kita buat model machine learning kita

```
from sklearn.linear_model import LinearRegression as lr
model_l = lr()
model_l.fit(X_train, y_train)
```

pertama kita import dari package sklear model yang ingin kita gunakan. Pada bagian inilah yang harus diganti jika kita ingin menggunakan algoritma lain, misalnya SVR, maka yang kita panggil adalah model SVR.

setelah kita panggil modul yang dibutuhkan, kita buat objek dari model yang kita pilih.

Dengan menggunakan metode fit, kita masukkan data training kita untuk melatih model yang kita miliki.

Setelah selesai melatih model kita, yang perlu kita lakukan adalah dengan mengevaluasi model kita.

Evaluasi dilakukan dengan cara melakukan prediksi terhadap data x_test dan hasilnya dibandingkan dengan hasil sebenarnya dari y_test.

```
y_pred = model_l.predict(X_test)
```

```
from sklearn import metrics



r2 = metrics.r2_score(y_test, y_pred)

print("The model performance for testing set")
print("--------------------------------------")
print('R2 score is {}'.format(r2))
```

modul metrics dibutuhkan untuk melakukan evaluasi nilai R_2

Bisa dilihat pada gambar di atas, kita melakukan perbandingan nilai antara y_test & y_pred

hasilnya,

The model performance for testing set

--

R2 score is 0.9541070903109727

score 0.954 merupakan hasil yang baik. Dimana prediksi y_test yang sempurna akan menghasilkan nilai $R_2 = 1$.

Referensi Eksternal

Pustaka

- dokumentasi LR
- dokumentasi Polynomial Regression
- dokumentasi SVR
- dokumentasi DTR