1. **Melatih dan mengevaluasi model regresi**

Regresi adalah jenis pembelajaran mesin yang umum digunakan untuk memprediksi nilai numerik.

**Tujuan pembelajaran**

Dalam modul ini, Anda akan mempelajari:

* Kapan model regresi digunakan.
* Cara melatih dan mengevaluasi model regresi menggunakan kerangka kerja Scikit-Learn.

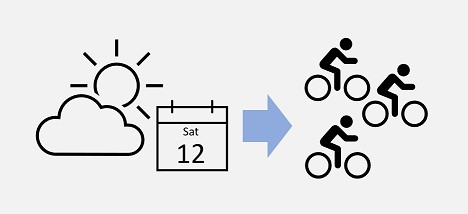
1. **Pengantar**

**Regresi** adalah tempat model memprediksi angka.

Dalam pembelajaran mesin, tujuan regresi adalah untuk membuat model yang dapat memprediksi nilai numerik yang dapat diukur, seperti harga, jumlah, ukuran, atau angka skalar lainnya.

Regresi adalah teknik statistik yang sangat penting bagi ilmu pengetahuan karena kemudahan interpretasi, ketahanan, dan kecepatan dalam perhitungan. Model regresi memberikan dasar yang sangat baik untuk memahami cara kerja teknik pembelajaran mesin yang lebih kompleks.

Dalam situasi dunia nyata, terutama ketika sedikit data yang tersedia, model regresi sangat berguna untuk membuat prediksi. Misalnya, jika sebuah perusahaan persewaan sepeda ingin memprediksi jumlah persewaan yang diharapkan pada hari tertentu di masa depan, model regresi dapat memprediksi angka ini. Sebuah model dapat dibuat dengan menggunakan data yang ada seperti jumlah sepeda yang disewa pada hari-hari di mana musim, hari dalam seminggu, dan sebagainya, juga direkam.



1. **Apa itu regresi?**

Regresi bekerja dengan membangun hubungan antara variabel dalam data yang mewakili karakteristik—dikenal sebagai fitur—dari hal yang diamati, dan variabel yang kita coba prediksi—dikenal sebagai label. Ingat perusahaan kami yang menyewakan sepeda dan ingin memprediksi jumlah sepeda yang disewakan yang diharapkan pada hari tertentu. Dalam hal ini, fitur meliputi hal-hal seperti hari dalam seminggu, bulan, dan sebagainya, sedangkan label adalah jumlah sepeda yang disewakan.

Untuk melatih model, kita mulai dengan sampel data yang berisi fitur, serta nilai label yang diketahui - jadi dalam hal ini kita memerlukan data historis yang mencakup tanggal, kondisi cuaca, dan jumlah sepeda yang disewakan.

Kemudian kita akan membagi sampel data ini menjadi dua subset:

* Himpunan data Latihan tempat kita menerapkan algoritma yang menentukan fungsi yang mencakup hubungan antara nilai fitur dan nilai label yang diketahui.
* Himpunan data validasi atau pengujian yang dapat kita gunakan untuk mengevaluasi model dengan memakainya guna menghasilkan prediksi untuk label dan membandingkannya dengan nilai label sebenarnya yang diketahui.

Penggunaan data riwayat dengan nilai label yang diketahui untuk melatih model menjadikan regresi sebagai contoh pembelajaran mesin dengan pengawasan.

**Contoh sederhana**

Mari kita ambil contoh sederhana untuk melihat bagaimana cara kerja proses pelatihan dan evaluasi pada dasarnya. Misalkan kita menyederhanakan skenario sehingga kita menggunakan satu fitur—suhu rata-rata harian—untuk memprediksi label persewaan sepeda.

Kami mulai dengan beberapa data yang mencakup nilai yang diketahui untuk fitur suhu harian rata-rata dan label penyewaan sepeda.

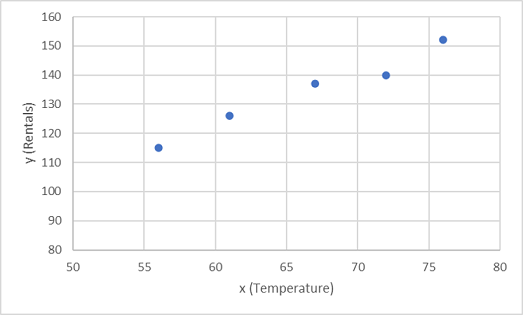
| **Suhu** | **Penyewaan** |
| --- | --- |
| 56 | 115 |
| 61 | 126 |
| 67 | 137 |
| 72 | 140 |
| 76 | 152 |
| 82 | 156 |
| 54 | 114 |
| 62 | 129 |

Sekarang kita akan secara acak memilih lima dari pengamatan ini dan menggunakannya untuk melatih model regresi. Ketika kita berbicara tentang 'melatih model', yang kita maksud adalah menemukan fungsi (persamaan matematika; sebut saja f) yang dapat menggunakan fitur suhu (yang akan kita sebut x) untuk menghitung jumlah persewaan (yang akan kita sebut y). Dengan kata lain, kita perlu mendefinisikan fungsi berikut: f(x) = y.

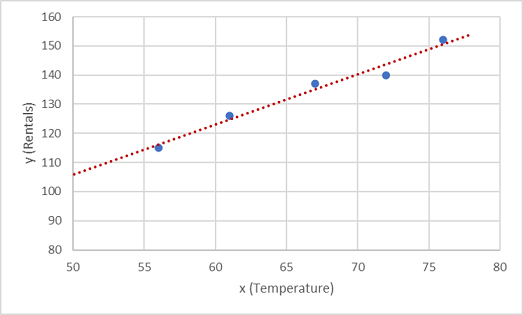
Kumpulan data pelatihan kita akan terlihat seperti ini:

| **x** | **y** |
| --- | --- |
| 56 | 115 |
| 61 | 126 |
| 67 | 137 |
| 72 | 140 |
| 76 | 152 |

Mari kita mulai dengan memplot nilai pelatihan untuk x dan y pada bagan:



Sekarang kita perlu menyesuaikan nilai-nilai ini dengan fungsi, yang memungkinkan untuk beberapa variasi acak. Anda mungkin dapat melihat bahwa titik-titik yang diplot membentuk garis diagonal yang hampir lurus - dengan kata lain, ada hubungan linier yang nyata antara x dan y, jadi kita harus menemukan fungsi linier yang paling cocok untuk sampel data. Ada berbagai algoritma yang dapat kita gunakan untuk menentukan fungsi ini, yang pada akhirnya akan menemukan garis lurus dengan varians keseluruhan minimal dari titik-titik yang diplot; seperti ini:

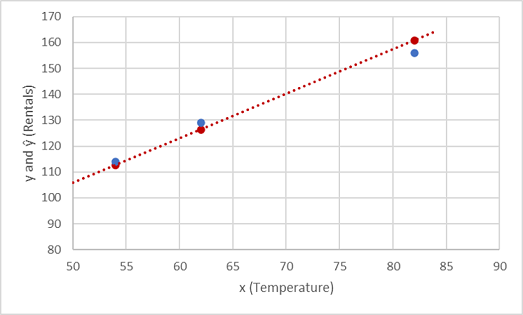


Garis mewakili fungsi linier yang dapat digunakan dengan nilai x apa pun untuk menerapkan kemiringan garis dan perpotongan (di mana garis memotong sumbu y ketika x adalah 0) untuk menghitung y. Dalam kasus ini, jika kita perpanjang garis ke kiri, kita akan menemukan bahwa ketika x adalah 0, y adalah sekitar 20, dan kemiringan garis sedemikian rupa sehingga untuk setiap satuan x yang Anda gerakkan ke kanan, y bertambah sekitar 1.7. Oleh karena itu, fungsi f kami dapat dihitung sebagai 20 + 1,7x.

Setelah mendefinisikan fungsi prediktif, kita dapat menggunakannya untuk memprediksi label untuk data validasi yang kita tahan dan membandingkan nilai prediksi (yang biasanya kita tunjukkan dengan simbol ŷ, atau "y-hat") dengan nilai y yang sebenarnya diketahui.

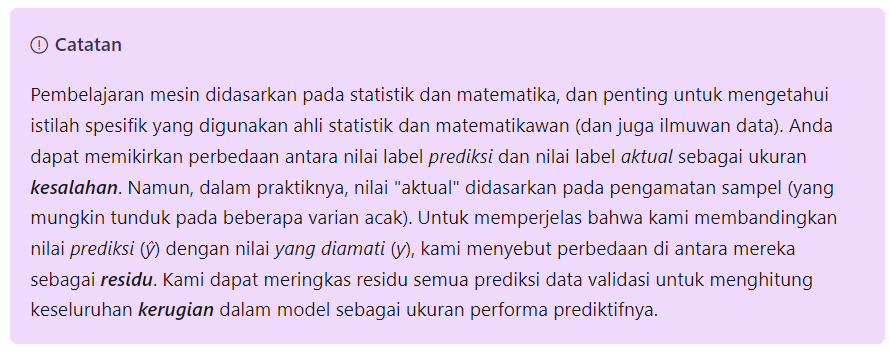
| **x** | **y** | **ŷ** |
| --- | --- | --- |
| 82 | 156 | 159,4 |
| 54 | 114 | 111,8 |
| 62 | 129 | 125.4 |

.Mari kita lihat bagaimana nilai y dan dibandingkan dalam sebuah plot:



Titik yang diplot pada garis fungsi adalah nilai ŷ yang diprediksi yang dihitung oleh fungsi, dan titik yang diplot lainnya adalah nilai aktual y.

Ada berbagai cara yang bisa digunakan untuk mengukur varian antara nilai yang diprediksi dan aktual, dan kita dapat menggunakan metrik ini untuk mengevaluasi seberapa baik model memprediksi.



Salah satu cara paling umum untuk mengukur kerugian adalah dengan mengkuadratkan residu individu, menjumlahkan kuadrat, dan menghitung rata-rata. Mengkuadratkan residu memiliki efek mendasarkan perhitungan pada nilai mutlak (mengabaikan apakah perbedaannya negatif atau positif) dan memberi bobot lebih pada perbedaan yang lebih besar. Metrik ini disebut **Kesalahan Rata-Rata Kuadrat.**

Untuk data validasi, perhitungannya terlihat seperti ini:

| **y** | **ŷ** | **y - ŷ** | **(y - ŷ)2** |
| --- | --- | --- | --- |
| 156 | 159,4 | -3,4 | 11,56 |
| 114 | 111,8 | 2.2 | 4.84 |
| 129 | 125.4 | 3.6+ | 12,96 |
|  | Jumlah total | ∑ | 29,36 |
|  | Rerata | x̄ | ****9,79**** |

Jadi kerugian untuk model kita berdasarkan metrik MSE adalah 9,79.

Jadi apakah itu hasil yang baik? Sulit untuk mengatakannya karena nilai MSE tidak dinyatakan dalam satuan pengukuran yang relevan. Kita tahu bahwa semakin rendah nilainya, semakin sedikit kerugian yang ada dalam model; dan karena itu, semakin baik prediksinya. Ini menjadikannya metrik yang berguna untuk membandingkan dua model dan menemukan yang berperforma terbaik.

Terkadang, ada baiknya untuk mengekspresikan kerugian dalam satuan pengukuran yang sama dengan nilai label yang diprediksi itu sendiri - dalam hal ini, jumlah penyewaan. Hal ini dapat dilakukan dengan menghitung akar kuadrat dari MSE, yang menghasilkan metrik yang secara mengejutkan dikenal sebagai Kesalahan **Rata-Rata Akar Kuadrat (RMSE)**.

√9,79 = 3,13

Jadi RMSE model kita menunjukkan bahwa kerugiannya hanya lebih dari 3, **yang dapat Anda tafsirkan secara longgar bahwa secara rata-rata, prediksi yang salah berkisar sekitar 3 penyewaan**.

Ada banyak metrik lain yang dapat digunakan untuk mengukur kerugian dalam regresi. Misalnya, **R2 (R-Squared)** (terkadang disebut sebagai koefisien determinasi) adalah korelasi antara x dan y kuadrat. Hal ini menghasilkan nilai antara 0 dan 1 yang mengukur jumlah varian yang dapat dijelaskan oleh model. Umumnya, semakin dekat nilai ini ke 1, semakin baik tingkat prediksi model.

1. **Latihan - Melatih dan mengevaluasi model regresi**

we used simple regression models to look at the relationship between features of a bike rentals datase

**First Regresion :** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\01-training

Membuat model awal - menarik kesimpulan : **bahwa diperlukan model yang lebih bagus**

1. **Bereksperimen dengan model**

Di Pelajaran 2, kita melihat pemasangan garis lurus ke poin data. Namun, regresi dapat cocok dengan banyak jenis hubungan, termasuk hubungan dengan banyak faktor, dan hubungan di mana pentingnya satu faktor bergantung pada faktor lainnya.

**Bereksperimen dengan model**

**Model regresi** sering dipilih karena model tersebut bekerja dengan sampel data kecil yang kuat, mudah diinterpretasikan, dan ada variasinya.

**Regresi linier** adalah bentuk regresi paling sederhana, tanpa batasan jumlah fitur yang digunakan. Regresi linier memiliki berbagai bentuk - sering diberi nama berdasarkan jumlah fitur yang digunakan dan bentuk kurva yang sesuai.

**Pohon keputusan** mengambil pendekatan langkah demi langkah untuk memprediksi variabel. Jika memikirkan contoh sepeda kita, pohon keputusan mungkin merupakan contoh bertahap pertama antara yang ada selama Musim Semi/Musim Panas dan Musim Gugur/Musim Dingin, membuat prediksi berdasarkan hari dalam seminggu. Musim Semi/Musim Panas-Senin mungkin memiliki tarif sewa sepeda 100 per hari, sedangkan Musim Gugur/Musim Dingin-Senin mungkin memiliki tarif sewa 20 per hari.

**Algoritma ansambel** membangun tidak hanya satu pohon keputusan, tetapi sejumlah besar pohon - memungkinkan prediksi yang lebih baik pada data yang lebih kompleks. Algoritma ansambel, seperti **Forest Acak**, banyak digunakan dalam pembelajaran mesin dan sains karena kemampuan prediksinya yang kuat.

Ilmuwan data sering bereksperimen menggunakan model yang berbeda. Dalam latihan berikut, kita akan bereksperimen dengan berbagai jenis model untuk membandingkan performanya pada data yang sama.

1. **Regression - Experimenting with additional models**

in the previous notebook, we used simple regression models to look at the relationship between features of a bike rentals dataset. In this notebook, we'll experiment with more complex models to improve our regression performance.

**Next Regresion :** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\02-experimenting

1. **Meningkatkan model dengan hyperparameter**

Model sederhana dengan himpunan data kecil sering kali dapat ditampung dalam satu langkah, sedangkan himpunan data yang lebih besar dan model yang lebih rumit pasti cocok dengan berulang kali menggunakan model dengan data pelatihan dan membandingkan output dengan label yang diharapkan. Jika prediksi cukup akurat, kami akan menganggap bahwa model tersebut terlatih. Jika tidak, kami akan menyesuaikan modelnya sedikit dan mengulanginya lagi.

**Hyperparameter** adalah nilai yang mengubah cara model cocok selama loop ini. Tingkat pembelajaran, misalnya, adalah hyperparameter yang menetapkan seberapa banyak model disesuaikan selama setiap siklus pelatihan. Tingkat pembelajaran yang tinggi berarti suatu model dapat dilatih lebih cepat, tetapi jika terlalu tinggi penyesuaiannya dapat menjadi sangat besar sehingga model tidak pernah 'disesuaikan' dan tidak optimal.

**Data praproses**

Prapemrosesan mengacu pada perubahan yang Anda buat pada data Anda sebelum diteruskan ke model. Kami sebelumnya telah membaca bahwa prapemrosesan dapat melibatkan pembersihan himpunan data Anda. Meskipun ini penting, prapemrosesan juga dapat mencakup mengubah format data Anda, sehingga model lebih mudah digunakan. Misalnya, data yang dideskripsikan sebagai 'merah', 'oranye', 'kuning', 'kapur', dan 'hijau', dapat bekerja lebih baik jika diubah ke dalam format yang lebih asli untuk komputer, seperti angka yang menyatakan jumlah warna merah dan jumlah warna hijau.

**Fitur penskalaan**

Langkah pra-pemrosesan yang paling umum adalah menskalakan fitur sehingga fitur tersebut berada di antara nol dan satu. Misalnya, berat sepeda dan jarak yang ditempuh seseorang dengan sepeda mungkin akan menampilkan dua angka yang sangat berbeda, tetapi dengan menskalakan kedua angka menjadi antara nol dan satu memungkinkan model belajar lebih efektif dari data.

**Menggunakan kategori sebagai fitur**

Dalam pembelajaran mesin, Anda juga dapat menggunakan fitur kategoris seperti 'sepeda', 'papan luncur', atau 'mobil'. Fitur ini diwakili oleh 0 atau 1 nilai dalam vektor satu panas - vektor yang memiliki 0 atau 1 untuk setiap nilai yang mungkin. Misalnya, sepeda, skateboard, dan mobil mungkin berturut-turut (1,0,0), (0,1,0), dan (0,0,1).

1. **Latihan - Mengoptimalkan dan menyimpan model**

In the previous notebook, we used complex regression models to look at the relationship between features of a bike rentals dataset. In this notebook, we'll see if we can improve the performance of these models even further.

**Optimasi model :** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\03-tuning

1. **Ringkasan**

Dalam modul ini, Anda mempelajari bagaimana regresi dapat digunakan untuk membuat model pembelajaran mesin yang memprediksi nilai numerik. Anda kemudian menggunakan kerangka kerja scikit-learn di Python untuk melatih dan mengevaluasi model regresi.

Meskipun scikit-learn adalah kerangka kerja populer untuk menulis kode untuk melatih model pengklusteran, Anda juga dapat membuat solusi pembelajaran mesin untuk pengklusteran menggunakan alat grafis di Pembelajaran Mesin Microsoft Azure. Anda dapat mempelajari lebih lanjut pengembangan model regresi tanpa kode menggunakan Azure Machine Learning di modul Membuat Model Regresi dengan perancang Pembelajaran Mesin Microsoft Azure.

**Tantangan: Prediksi Harga Real Estat**

Apakah Anda siap untuk membuat model regresi Anda sendiri? Coba tantangan memprediksi harga properti real estat di buku catatan 02 - Real Estate Regression Challenge.ipynb!

<https://github.com/MicrosoftDocs/ml-basics/blob/master/challenges/02%20-%20Real%20Estate%20Regression%20Challenge.ipynb>

