1. **Melatih dan mengevaluasi model klasifikasi**

Klasifikasi adalah sejenis pembelajaran mesin yang digunakan untuk mengategorikan item ke dalam kelas.

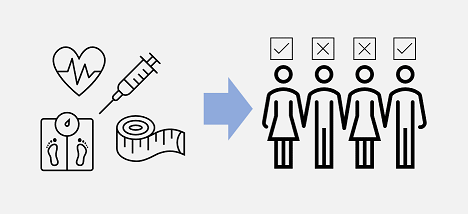
**Tujuan pembelajaran**

Dalam modul ini, Anda akan mempelajari:

* Kapan harus menggunakan klasifikasi
* Cara melatih dan mengevaluasi model klasifikasi menggunakan kerangka kerja Scikit-Learn

1. **Pengantar**

Klasifikasi adalah bentuk pembelajaran mesin tempat Anda melatih model untuk memprediksi kategori item. Misalnya, klinik kesehatan mungkin menggunakan data diagnostik seperti tinggi badan, berat badan, tekanan darah, kadar glukosa darah pasien untuk memprediksi apakah pasien menderita diabetes atau tidak.



Data Kategoris memiliki 'kelas' yang berbeda, bukan nilai numerik. Beberapa jenis data dapat berupa numerik atau kategoris: waktu untuk menjalankan perlombaan dapat berupa waktu dalam detik, atau kita dapat membagi waktu ke dalam kelas ‘cepat’, ‘sedang’ dan ‘lambat’ - kategoris. Sedangkan jenis data lainnya hanya dapat bersifat kategoris, seperti jenis bentuk - ‘lingkaran’, ‘segitiga’, atau ‘persegi’.

1. **Apa itu klasifikasi?**

Klasifikasi biner adalah klasifikasi dengan dua kategori. Misalnya, kita bisa memberi label pasien sebagai non-diabetes atau diabetes.

Prediksi kelas dibuat dengan menentukan probabilitas untuk setiap kemungkinan kelas dengan nilai antara 0 -tidak mungkin - dan 1 - pasti. Probabilitas total untuk semua kelas adalah 1, karena pasien pasti diabetes atau non-diabetes. Jadi, jika probabilitas prediksi pasien diabetes adalah 0,3, ada probabilitas yang sesuai 0,7 bahwa pasien non-diabetes.

Nilai ambang, biasanya 0,5, digunakan untuk menentukan kelas prediksi - jadi jika kelas positif (dalam hal ini, diabetes) memiliki probabilitas prediksi lebih besar dari ambang batas, maka klasifikasi diabetes diprediksi.

**Melatih dan mengevaluasi model klasifikasi**

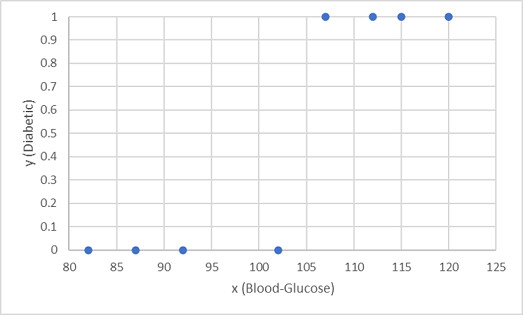
Klasifikasi adalah contoh teknik pembelajaran mesin terawasi, yang artinya bergantung pada data yang menyertakan nilai fitur yang diketahui (misalnya, pengukuran diagnostik untuk pasien) serta yang diketahui label nilai (misalnya, klasifikasi non-diabetes atau diabetes). Algoritma klasifikasi digunakan untuk menyesuaikan subhimpunan data ke fungsi yang dapat menghitung probabilitas untuk setiap label kelas dari nilai fitur. Data yang tersisa digunakan untuk mengevaluasi model dengan membandingkan prediksi yang dihasilkan dari fitur dengan label kelas yang diketahui.

**Contoh sederhana**

Mari kita telusuri contoh sederhana untuk membantu menjelaskan prinsip-prinsip kunci. Misalkan kita memiliki data pasien berikut, yang terdiri dari satu fitur (kadar glukosa darah) dan label kelas 0 untuk non-diabetes, 1 untuk diabetes.

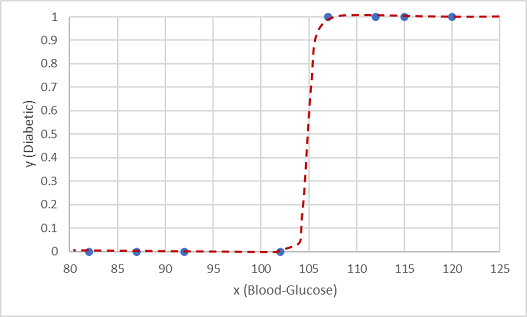
| **Gula darah** | **penderita diabetes** |
| --- | --- |
| 82 | 0 |
| 92 | 0 |
| 112 | 1 |
| 102 | 0 |
| 115 | 1 |
| 107 | 1 |
| 87 | 0 |
| 120 | 1 |
| 83 | 0 |
| 119 | 1 |
| 104 | 1 |
| 105 | 0 |
| 86 | 0 |
| 109 | 1 |

Kita akan menggunakan delapan pengamatan pertama untuk melatih model klasifikasi, dan kita akan mulai dengan memplot fitur glukosa darah (yang akan kami sebut x) dan prediksi diabetes label (yang akan kita sebut y).



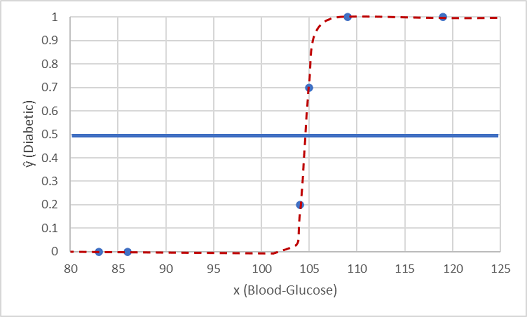
Yang kita butuhkan adalah fungsi yang menghitung nilai probabilitas untuk y berdasarkan x (dengan kata lain, kita membutuhkan fungsi f(x) = y). Anda dapat melihat dari grafik bahwa pasien dengan kadar glukosa darah rendah semuanya non-diabetes, sedangkan pasien dengan kadar glukosa darah yang lebih tinggi adalah diabetes. Sepertinya semakin tinggi kadar glukosa darah, semakin besar kemungkinan pasien menderita diabetes, dengan titik bengkok berada di antara 100 dan 110. Kita perlu menyesuaikan fungsi yang menghitung nilai antara 0 dan 1 untuk y ke nilai ini.

Salah satu fungsi tersebut adalah fungsi logistik, yang membentuk kurva sigmoidal (berbentuk S), seperti ini:



Sekarang kita dapat menggunakan fungsi tersebut untuk menghitung nilai probabilitas bahwa y positif, artinya pasien diabetes, dari sembarang nilai x dengan mencari titik pada garis fungsi untuk x. Kita dapat menetapkan nilai ambang 0,5 sebagai titik potong untuk prediksi label kelas.

Mari kita uji dengan nilai data yang kita tahan:



Poin yang diplot di bawah garis ambang batas akan menghasilkan kelas prediksi 0 - non-diabetes - dan poin di atas garis akan diprediksi sebagai 1 - diabetes.

Sekarang kita dapat membandingkan prediksi label berdasarkan fungsi logistik yang dienkapsulasi dalam model (yang akan kita sebut ŷ, atau "y-hat") dengan label kelas yang sebenarnya (y).

| **x** | **y** | **ŷ** |
| --- | --- | --- |
| 83 | 0 | 0 |
| 119 | 1 | 1 |
| 104 | 1 | 0 |
| 105 | 0 | 1 |
| 86 | 0 | 0 |
| 109 | 1 | 1 |

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

**Clasification Process:** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\3 - Melatih dan mengevaluasi model klasifikasi\01-binary.ipynb

1. **Mengevaluasi model klasifikasi**

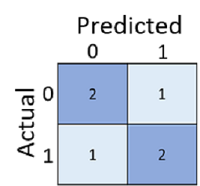
Akurasi pelatihan model klasifikasi jauh lebih penting daripada seberapa baik model itu akan bekerja ketika diberikan data baru yang tidak terlihat. Bagaimanapun juga, kita melatih model agar dapat digunakan pada data baru yang ditemukan di dunia nyata. Jadi, setelah kita melatih model klasifikasi, kita harus mengevaluasi performanya pada sekumpulan data baru yang tidak terlihat.

Pada pelajaran sebelumnya, kita membuat model yang akan memprediksi apakah pasien menderita diabetes atau tidak berdasarkan kadar glukosa darahnya. Sekarang, ketika diterapkan ke beberapa data yang bukan bagian dari set pelatihan, prediksi berikut diperoleh:

| **x** | **y** | **ŷ** |
| --- | --- | --- |
| 83 | 0 | 0 |
| 119 | 1 | 1 |
| 104 | 1 | 0 |
| 105 | 0 | 1 |
| 86 | 0 | 0 |
| 109 | 1 | 1 |

Ingat bahwa x mengacu pada tingkat glukosa darah, y mengacu pada apakah mereka benar-benar diabetes, dan ŷ mengacu pada prediksi model apakah mereka diabetes atau tidak.

Menghitung berapa banyak prediksi yang benar terkadang menyesatkan atau terlalu sederhana bagi kita untuk memahami jenis kesalahan yang akan terjadi di dunia nyata. Untuk mendapatkan informasi yang lebih detail, kita dapat mentabulasi hasil dalam struktur yang disebut confusion matrix, seperti ini:



Matriks kebingungan menunjukkan jumlah total kasus di mana:

* Model memperkirakan 0 dan label sebenarnya adalah 0 (negatif benar; kiri atas)
* Model memprediksi 1 dan label sebenarnya adalah 1 (positif benar; kanan bawah)
* Model memperkirakan 0 dan label sebenarnya adalah 1 (negatif palsu; kiri bawah)
* Model memprediksi 1 dan label sebenarnya adalah 0 (positif palsu; kanan atas)

Sel-sel dalam matriks kebingungan sering kali diarsir sehingga nilai yang lebih tinggi memiliki bayangan yang lebih dalam. Hal ini memudahkan untuk melihat tren diagonal yang kuat dari kiri atas ke kanan bawah, menyoroti sel di mana nilai prediksi dan nilai sebenarnya sama.

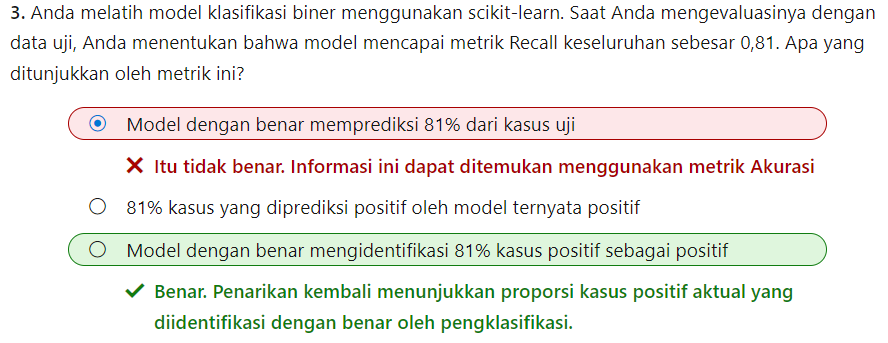
Dari nilai inti ini, Anda dapat menghitung rentang metrik lain yang dapat membantu Anda mengevaluasi performa model. Contohnya:

* Akurasi: (TP+TN)/(TP+TN+FP+FN) - dari semua prediksi, berapa banyak yang benar?
* Ingat: TP/(TP+FN) - dari semua kasus yang adalah positif, berapa banyak yang diidentifikasi model?
* Presisi: TP/(TP+FP) - dari semua kasus yang diprediksi model sebagai positif, berapa banyak yang sebenarnya yang positif?

1. **Latihan - Melakukan klasifikasi dengan metrik alternatif**

**Clasification alternatif process:** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\3 - Melatih dan mengevaluasi model klasifikasi\02-metrics.ipynb

**Note:**



1. **Membuat model klasifikasi multikelas**

Anda juga dapat membuat model klasifikasi multikelas, di mana terdapat lebih dari dua kemungkinan kelas. Misalnya, klinik kesehatan mungkin memperluas model diabetes untuk mengklasifikasikan pasien sebagai:

* Non-diabetes
* Diabetes tipe-1
* Diabetes tipe-2

Nilai probabilitas kelas individu masih akan bertambah hingga total 1 karena pasien pasti hanya berada di salah satu dari tiga kelas, dan kelas yang paling mungkin diprediksi oleh model.

**Menggunakan model klasifikasi Multikelas**

Klasifikasi multikelas dapat dianggap sebagai kombinasi dari beberapa pengklasifikasi biner. Ada dua cara Anda mendekati masalah:

* One vs Rest (OVR), saat pengklasifikasi dibuat untuk setiap nilai kelas yang mungkin, dengan hasil positif untuk kasus di mana prediksinya adalah kelas ini, dan prediksi negatif untuk kasus di mana prediksinya adalah kelas lain. Misalnya, masalah klasifikasi dengan empat kelas bentuk yang mungkin (persegi, lingkaran, segitiga, segi enam) akan membutuhkan empat pengklasifikasi yang memprediksi:
  + - persegi atau bukan
    - lingkaran atau bukan
    - segitiga atau bukan
    - segi enam atau bukan
* One vs One (OVO), di mana pengklasifikasi untuk setiap kemungkinan pasangan kelas dibuat. Masalah klasifikasi dengan empat kelas bentuk akan membutuhkan pengklasifikasi biner berikut:
  + - persegi atau lingkaran
    - persegi atau segitiga
    - persegi atau segi enam
    - lingkaran atau segitiga
    - lingkaran atau segi enam
    - segitiga atau segi enam

Dalam kedua pendekatan tersebut, model keseluruhan harus memperhitungkan semua prediksi ini untuk menentukan kategori tunggal mana yang dimiliki item tersebut.

Untungnya, di sebagian besar kerangka kerja pembelajaran mesin, termasuk scikit-learn, menerapkan model klasifikasi multikelas tidak jauh lebih kompleks daripada klasifikasi biner - dan dalam banyak kasus, penghitung yang digunakan untuk klasifikasi biner secara implisit mendukung klasifikasi multikelas dengan mengabstraksi algoritma OVR, sebuah OVO algoritma, atau dengan membiarkan pilihan salah satunya.

1. **. Latihan - Melatih dan mengevaluasi model klasifikasi multikelas**

**Multiclass clasification :** D:\Data Scientist Indonesia Cloud Skills Challenge\Jupyter Notebook - Data Scientist Indonesia Cloud Skills Challenge\3 - Melatih dan mengevaluasi model klasifikasi\03-multiclass.ipynb

1. **Ringkasan**

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

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

Tantangan: Mengklasifikasikan Anggur

Ingin menantang diri sendiri untuk melatih model klasifikasi? Coba tantangan di buku catatan /challenges/03 - Wine Classification Challenge.ipynb untuk melihat apakah Anda dapat mengklasifikasikan anggur ke dalam varietas anggurnya!

<https://github.com/MicrosoftDocs/ml-basics/blob/master/challenges/03%20-%20Wine%20Classification%20Challenge.ipynb>

