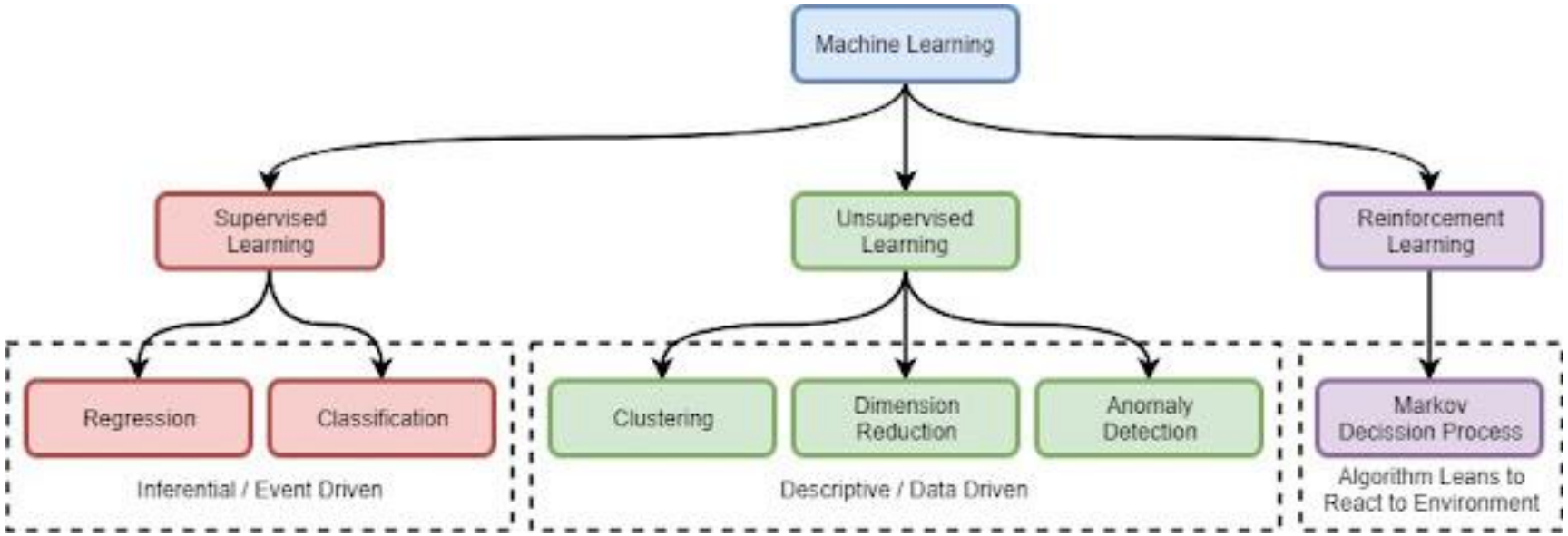
Ukuran Kinerja Model

Kuliah : 23-11-2023

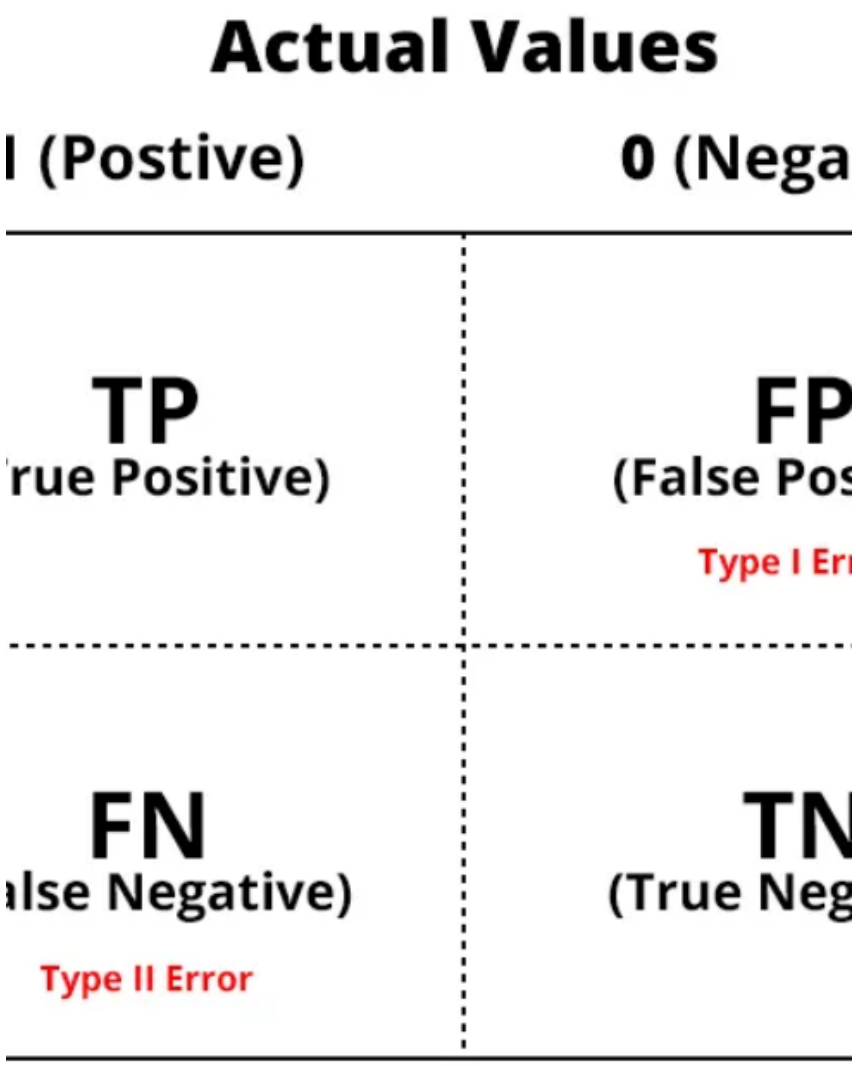

Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Regression
Classification
Clustering
Dimension Reduction
Anomaly Detection
Markov Decission Process
Inferential / Event Driven
Descriptive / Data Driven
Algorithm Learns to React to Environment

Ukuran Evaluasi

- Evaluasi terhadap suatu classifier umumnya dilakukan menggunakan sebuah data uji, yang tidak digunakan dalam pelatihan classifier tersebut.
- Ada sejumlah ukuran yang dapat digunakan untuk menilai atau mengevaluasi model klasifikasi, diantaranya :
  - Erraor rate
  - Recall
  - Sensitivity
  - Specificity
  - Precision
  - dll

Ukuran kinerja Klasifikas

- Apa itu confusion matrix dan mengapa kita memerlukan itu ?
- Confusion matrix juga sering disebut error matrix. Pada dasarnya confusion matrix memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya.
- Confusion matrix berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui.
- Gambar disamping merupakan confusion matrix dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda



Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix.

Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN).

Agar lebih mudah memahaminya, saya menggunakan contoh kasus sederhana untuk memprediksi seorang pasien menderita kanker atau tidak

True Positive (TP)

Merupakan data positif yang diprediksi benar. Contohnya, pasien menderita kanker (class 1) dan dari model yang dibuat memprediksi pasien tersebut menderita kanker (class 1).

True Negative (TN)

Merupakan data negatif yang diprediksi benar. Contohnya, pasien tidak menderita kanker (class 2) dan dari model yang dibuat memprediksi pasien tersebut tidak menderita kanker (class 2).

False Postive (FP) — Type I Error

Merupakan data negatif namun diprediksi sebagai data positif. Contohnya, pasien tidak menderita kanker (class 2) tetapi dari model yang telah memprediksi pasien tersebut menderita kanker (class 1).

False Negative (FN) — Type II Error

Merupakan data positif namun diprediksi sebagai data negatif. Contohnya, pasien menderita kanker (class 1) tetapi dari model yang dibuat memprediksi pasien tersebut tidak menderita kanker (class 2).

- Pada beberapa kasus “Type II Error” lebih berbahaya, kita dapat menghubungkan pernyataan itu dengan contoh prediksi kanker diatas.
- Jika pasien tidak menderita kanker tetapi diprediksi menderita kanker (FP), maka pada diagnosa selanjutnya pasien tersebut dapat mengetahui keadaan sebenarnya bahwa pasien tersebut benar tidak menderita kanker.
- Tetapi jika ada pasien yang sebenarnya menderita kanker tetapi diprediksi tidak menderita kanker (FN), maka pasien tersebut akan mengetahui keadaan sebenarnya dengan sangat terlambat dan pasien tersebut tidak segera mengambil tindakan pencegahan medis untuk kanker itu.
- Sehingga dapat menyebabkan kondisi pasien yang semakin memburuk setiap harinya bahkan kematian.
- Jadi dapat dikatakan bahwa “Type II Error” lebih berbahaya.

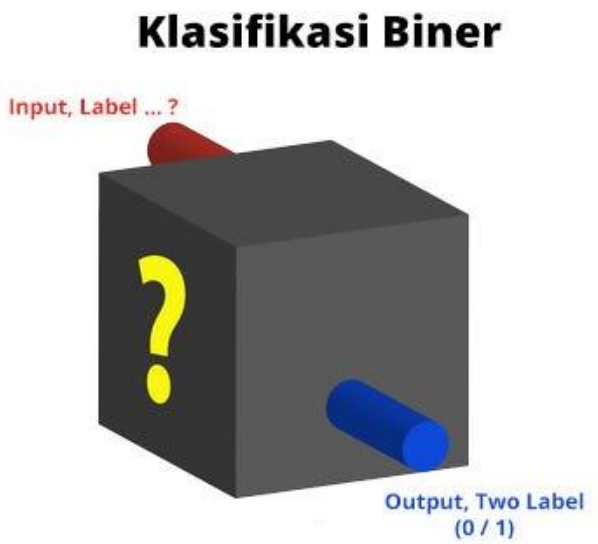
- Ada cara yang lebih mudah untuk mengingatnya, yaitu:
- Jika diawali dengan True maka prediksinya adalah benar, entah diprediksi terjadi atau tidak terjadi.
- Jika diawali dengan False maka prediksinya adalah salah.
- Positif dan negatif merupakan hasil prediksi dari model.
- Tentunya kita ingin model yang telah kita buat memberikan 0 false positive dan 0 false negative.
- Tetapi pada prakteknya hal tersebut tidak akan pernah terjadi karena model mana pun tidak akan memberikan keakuratan 100%.
- Jika model anda memberikan nilai 100% maka ada masalah pada model yang anda buat atau data yang anda gunakan.

Mengapa kita memerlukan confusion matrix ?

- Seperti yang telah dijelaskan diatas, confusion matrix akan memberi tahu seberapa baik model yang kita buat.
- Secara khusus confusion matrix juga memberikan informasi tentang TP, FP, TN, dan FN.
- Hal ini sangat berguna karena hasil dari klasifikasi umumnya tidak dapat diekspresikan dengan baik dalam satu angka saja.
- Dengan contoh yang sama untuk memprediksi kanker, anda akan mencoba memprediksi siapa yang akan mati karena kanker tahun ini berdasarkan perilaku seperti merokok dari seluruh populasi.
- Pada tahun tertentu, hanya 1% populasi yang mati karena kanker. Algoritma klasifikasi naif hanya akan memprediksi tidak ada yang mati karena kanker.
- Dengan confusion matrix memungkinkan kita untuk melihat dengan cepat, dari siapa yang akan diprediksi mati, berapa banyak yang mati dan yang tidak.

Manfaat confusion matrix

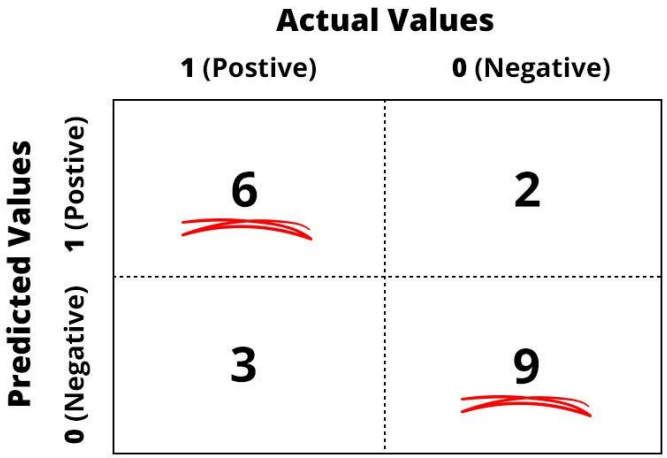
- Menunjukkan bagaimana model ketika membuat prediksi.
- Tidak hanya memberi informasi tentang kesalahan yang dibuat oleh model tetapi juga jenis kesalahan yang dibuat.
- Setiap kolom dari confusion matrix merepresentasikan instance dari kelas prediksi.
- Setiap baris dari confusion matrix mewakili instance dari kelas aktual.



Contoh klasifikasi biner

- Confusion matrix dapat digunakan untuk mengukur performa dalam permasalahan klasifikasi biner maupun permasalahan klasifikasi multiclass.
- Klasifikasi biner hanya menghasilkan dua ouput kelas (label), seperti “Ya” atau “Tidak”, “0” atau “1” untuk setiap data input yang diberikan Kelas utama biasanya dinotasikan sebagai data positif dan yang lainnya sebagai data negatif.

- Sebuah model akan dilatih untuk memprediksi apakah seorang pasien sedang menderita kanker atau tidak.
- Dengan asumsi terdapat 20 pasien dengan 9 pasien positif kanker dan 11 pasien negatif kanker, maka contoh confusion matrix yang dihasilkan model seperti disamping ini



- Jika dilihat dari confusion matrix diatas dari 9 pasien positif kanker, model memprediksi ada 3 pasien yang diprediksi negatif kanker (FN), dan dari 11 pasien negatif kanker, model memprediksi ada 2 pasien yang diprediksi positif kanker (FP).
- Prediksi yang benar terletak pada tabel diagonal (garis bawah merah), sehingga secara visual sangat mudah untuk melihat kesalahan prediksi karena kesalahan prediksi berada di luar tabel diagonal confusion matrix.

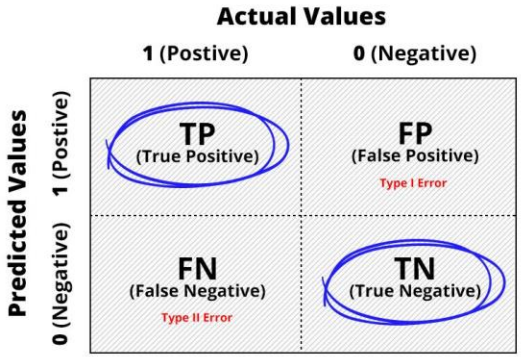
Bagaimana mengukur performance metrics dari confusion matrix ?

- Kita dapat menggunakan confusion matrix untuk menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat.
- Pada bagian ini mari kita pahami beberapa performance metrics populer yang umum dan sering digunakan: accuracy, precission, dan recall.

Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar.
- Maka, accuracy merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data.
- Dengan kata lain, accuracy merupakan tingkat kedekatan nilai prediksi dengan nilai aktual (sebenarnya).
- Nilai accuracy dapat diperoleh dengan persamaan



- Dari contoh confusion matrix klasifikasi biner diatas maka dengan menghitung nilai accuracy dapat menjawab pertanyaan “Berapa persen pasien yang benar diprediksi menderita kanker maupun yang tidak menderita kanker dari kesuluruhan pasien?”

$$accuracy = \frac{jumlah\ pasien\ dipredeksi\ benar\ (kanker\ +\ tidak\ kanker)}{jumlah\ pasien\ keseluruhan}$$

$$= \frac{6 + 9}{6 + 9 + 2 + 3} = \frac{15}{20} = 0{,}75$$

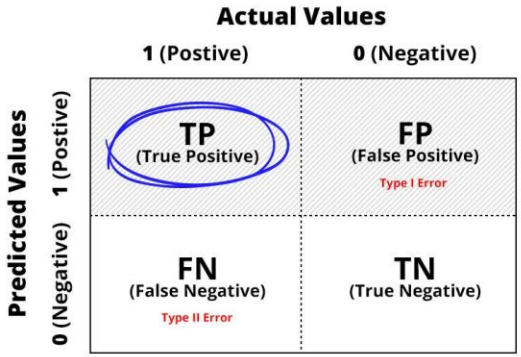
$$= 0{,}75 * 100\%$$

$$= 75\%$$

Precision (Positive Predictive Value)

$$precision = \frac{TP}{TP + FP} \quad (2)$$

- Precision menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model.
- Maka, precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf.
- Dari semua kelas positif yang telah di prediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai precision dapat diperoleh dengan persamaan



- Dari contoh confusion matrix klasifikasi biner diatas maka dengan menghitung nilai precision dapat menjawab pertanyaan “Berapa persen pasien yang benar menderita kanker dari keseluruhan pasien yang diprediksi menderita kanker?”

$$precision = \frac{jumlah\ pasien\ kanker\ diprediksi\ \boldsymbol{benar}}{jumlah\ pasien\ diprediksi\ kanker}$$

$$= \frac{6}{6+2} = \frac{6}{8} = 0{,}75$$

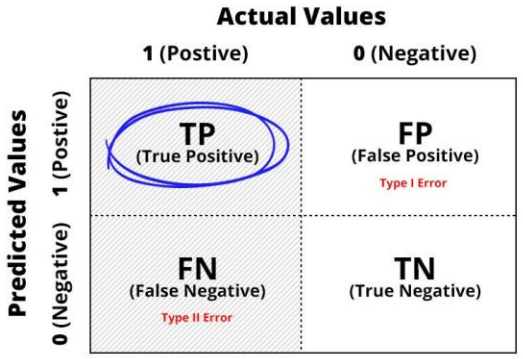
$$= 0{,}75 * 100\%$$

$$= 75\%$$

Recall atau Sensitivity (True Positive Rate)

$$recall = \frac{TP}{TP + FN} \quad (3)$$

- Recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Maka, recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Nilai recall dapat diperoleh dengan persamaan (3).



- Dari contoh confusion matrix klasifikasi biner diatas maka dengan menghitung nilai recall dapat menjawab pertanyaan “Berapa persen pasien yang diprediksi kanker dibandingkan keseluruhan pasien yang sebenarnya menderita kanker”.

$$recall = \frac{jumlah\ pasien\ kanker\ diprediksi\ \mathbf{benar}}{jumlah\ pasien\ kanker}$$

$$= \frac{6}{6+3} = \frac{6}{9} = 0{,}66$$

$$= 0{,}66 * 100\%$$

$$= 66\%$$

Model klasifikasi

- Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dapat dibagi menjadi 4 (empat) jenis yaitu klasifikasi binary, multi-class, multi-label dan hierarchical. Pada klasifikasi binary, data masukan dikelompokkan ke dalam salah satu dari dua kelas.
- Jenis klasifikasi ini merupakan bentuk klasifikasi yang paling sederhana dan banyak digunakan.
- Contoh penggunaannya antara lain dalam sistem yang melakukan deteksi orang atau bukan, sistem deteksi kendaraan atau bukan, dan sistem deteksi pergerakan atau bukan

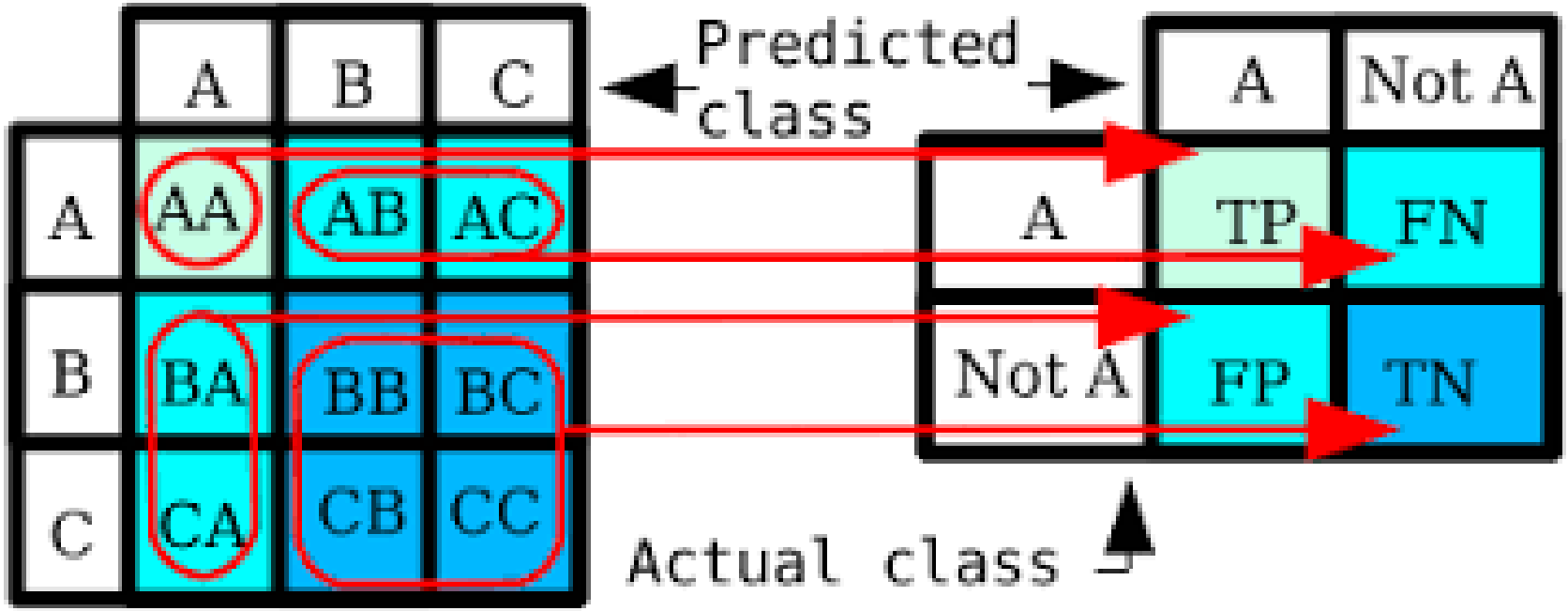
- Sementara itu, pada bentuk klasifikasi multi-class, data masukan diklasifikasikan menjadi beberapa kelas.
- Sebagai contoh sistem yang dapat mengklasifikasikan jenis kendaraan seperti sepeda, sepeda motor, mobil, bus, truk, dan sebagainya.
- Bentuk klasifikasi multi-label pada dasarnya sama dengan multi-class dimana data dikelompokkan menjadi beberapa kelas, namun pada klasifikasi multi-label, data dapat dimasukkan dalam beberapa kelas sekaligus.
- Bentuk klasifikasi yang terakhir adalah hierarchical. Data masukan dikelompokkan menjadi beberapa kelas, namun kelas tersebut dapat dikelompokkan kembali menjadi kelas-kelas yang lebih sederhana secara hirarkis.

Ukuran kinerja berbasis confusion matrix

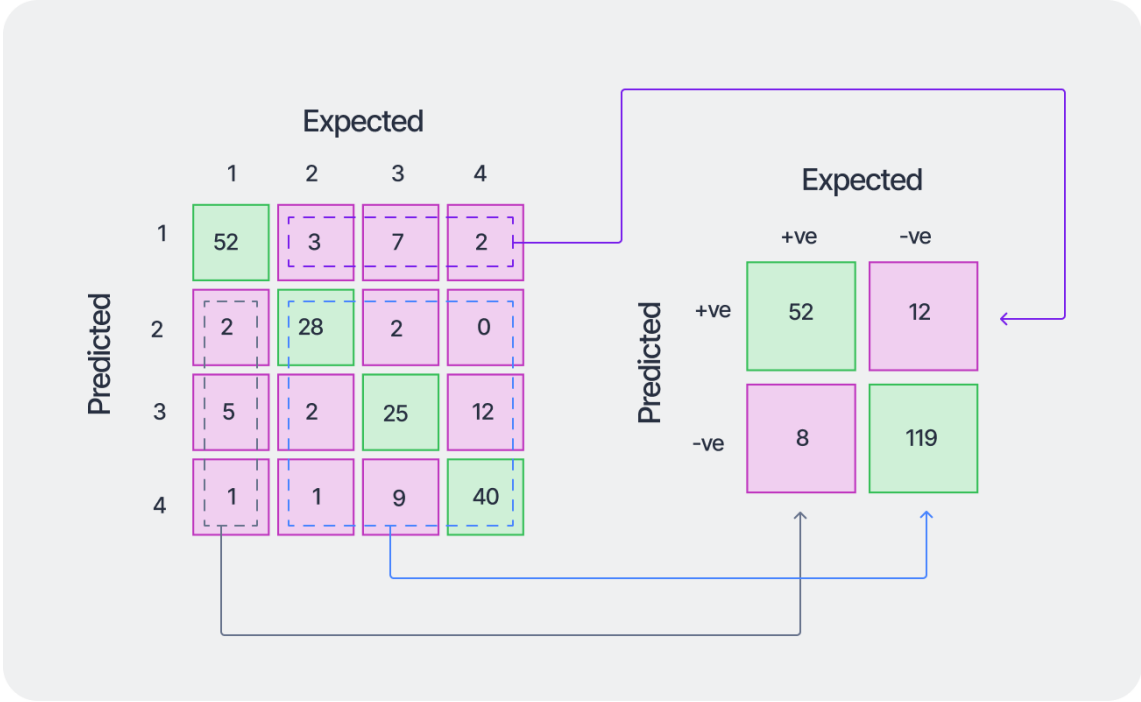
inpows.com

	Positive	**Negative**	
Positive	True Positive (TP)	False Negative (FN)	**Sensitivity** $\frac{TP}{(TP+FN)}$
Negative	False Positive (FP)	True Negative (TN)	**Specifity** $\frac{TN}{(TN+FP)}$
	Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Matrik 3x3


A
B
C
Predicted class
A
Not A
A
AA
AB
AC
A
TP
FN
B
BA
BB
BC
Not A
FP
TN
C
CA
CB
CC
Actual class

Matrix 4x4


Expected
1
2
3
4
Predicted
1
2
3
4
52
3
7
2
2
28
2
0
5
2
25
12
1
1
9
40
Expected
+ve
-ve
Predicted
+ve
-ve
52
12
8
119

Contoh

- TP = 970 TN= 40 FP = 960 FN = 30 P =1000 N =1000

$akurasi = \frac{TP+TN}{P+N} = \frac{970+40}{1000+1000} = 50{,}5\%$

$error = 1 - akurasi = 100\% - 50{,}5\% = 49{,}5$

$\text{Precission} = \frac{TP}{TP+FP} = \frac{970}{970+960} = 50{,}26\%$

$recall = \frac{TP}{TP+FN} = \frac{970}{970+30} = 97\%$

$F - one = \frac{2 \times precision \times recall}{precision+recall} = \frac{2\times TP}{2\times TP+FP+FN} = \frac{2\times 970}{(2\times 970)+960+30} = 66{,}21\%$

	Kelas = 'Ya'	Kelas = 'Tidak'	Jumlah
Kelas = 'ya'	970	30	1000
Kelas = 'tidak'	960	40	1000
Jumlah	1930	70	2000

Metode Validasi

Resubtitution

- Keuntungan: Sederhana
- Kelemahan: Paling lemah
- Kapan digunakan: Jika dirasa data latih cukup mewakili populasi.
- Langkah-langkahnya:

1. Melatih model dengan menggunakan data latih
2. Mengukur tingkat kesalahan berdasarkan keluaran dan nilai aktual dari seluruh objek data tersebut.

Hold-out (2 sub himpunan)

- Asumsi: data latih dan data uji dibangun dengan distribusi yang sama untuk setiap kelas. Agar proporsi setiap kelas sama.
- Langkah-langkah:

1. Membagi data set menjadi 2, biasanya 2/3 data latih dan 1/3 data uji(60/40, 70/30, 80/20 atau dengan pertimbangan tertentu)
2. Membangun model menggunakan data latih
3. Menguji model menggunakan data uji

Hold-out (3 sub himpunan)

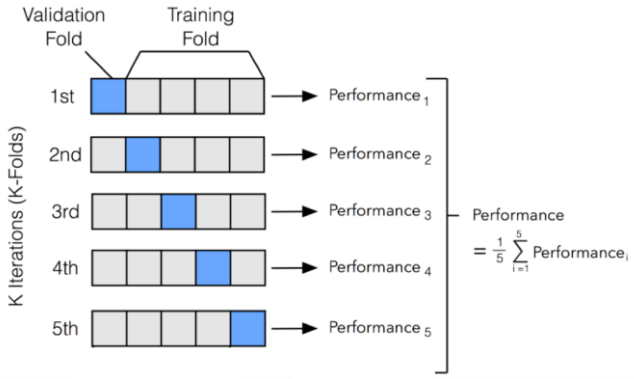
- Asumsi: data latih dan data uji dibangun dengan distribusi yang sama untuk setiap kelas. Agar proporsi setiap kelas sama.
- Langkah-langkah:

1. Membagi data set menjadi 3, data latih, data validasi, dan data uji.
2. Membangun model menggunakan data latih
3. Memvalidasi model menggunakan data validasi
4. Menguji model yang telah tervalidasi menggunakan data uji.

K-fold Cross Validation

Langkah-langkah:

1. Membagi dataset menjadi k sub himpunan (*fold*), sehingga setiap fold berisi 1/k, $D = \{d_1, d_2, \dots, d_k\}$
2. Menggunakan (k-1) fold untuk data latih Latih = d_i, i = 1, 2, k-1
3. Menguji model menggunakan dj, $j \neq i$
4. Menghitung akurasi: jumlah keseluruhan klasifikasi benar dalam k iterasi dibagi dengan jumlah tuple dalam himpunan data.

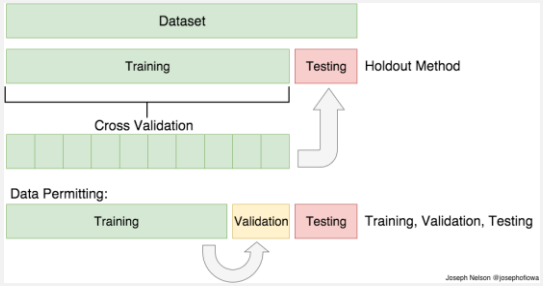


Sumber:
https://medium.com/@sebastiannorena/some-model-tuning-methods-bfef3e6544f0

K-fold Cross Validation (2)

Langkah-langkah:

1. Membagi dataset menjadi k sub himpunan (*fold*), sehingga setiap fold berisi 1/k, $D = \{d_1, d_2, \dots, d_k\}$
2. Menggunakan (k-2) fold untuk data latih Latih = d_i, i = 1, 2, k-2
3. Memvalidasi model menggunakan dj, $j \neq i$ (menaksir hyperparameter)
4. Menguji model hasil validasi menggunakan dk, $k \neq i \neq j$
5. Menghitung akurasi: jumlah keseluruhan klasifikasi benar dalam k iterasi dibagi dengan jumlah tuple dalam himpunan data.



Sumber: https://miro.medium.com/max/948/1*4G__SV580CxFj78o9yUXuQ.png

Leave-One-Out Cross Validation

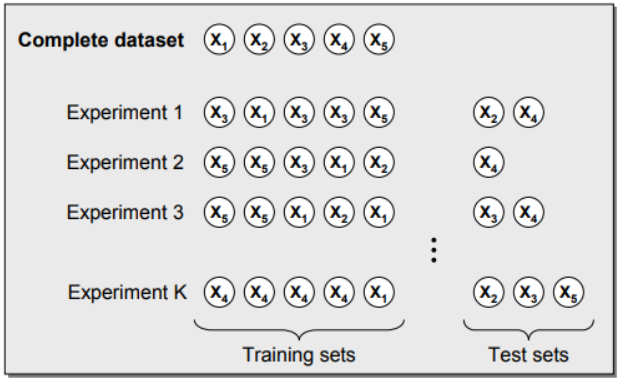
- Sama dengan K-fold hanya saja yang digunakan per data bukan per fold, sehingga setiap data pernah menjadi data latih dan data uji.

Random Subsampling

- Modifikasi Teknik hold-out.
- Menjalankan metode hold out beberapa kali, misal sejumlah k iterasi
- Mengevaluasi berdasarkan model klasifikasi berdasarkan rata-rata daris etiap iterasi tersebut.
- **Random:** pemilihan mana data latih dan data uji secara acak.

Bootstraping

Pemilihan data latih dilakukan dengan penyamplingan secara acak dengan distribusi seragam, sampel yang telah terambil boleh dimasukkan kembali ke sumber data



Sumber: https://vitalflux.com/wp-content/uploads/2018/02/bootstraping-validation-technique.png

Tugas Akhir Data Mining

- Tugas dikerjakan berkelompok dengan anggota maksimal 3 mahasiswa
- Membangun model data mining untuk mendeteksi jenis serangan (IDS) yang mempunyai kinerja yang baik untuk setiap jenis serangan yang ada
- Model dibangun dan diuji dengan menggunakan dataset :

Dataset	Description
KDD'99	It is generated using simulation of normal and attacks traffic in a military environment (US AirForce LAN). It contains nine weeks of simulation in rat tcpdump files. The dataset is characterized using 41 features related to intrinsic, content, and traffic. Four types of attacks are simulated: DoS, Prob, U2R, and R2L.
NSL-KDD	It is a modification to the KDD'99 dataset with solving the problems of redundancy, duplicates, the imbalance of data.
UNSW-Nb15	It was created using the IXIA PefectStorm tool to extract normal and attack network traffic based on 100 GB of raw network traffic. It is characterized using 49 features. It consists of around 175 thousand records for training and around 82 thousand records for testing. There are nine types of attacks: Fuzzers, Analysis, Backdoor, DoS, Exploit, Generic, Reconnaissance, Shellcode, Worm
CICIDS2017	It was created in an emulated environment in a 5 day period. It contains traffic in packet flow and bidirectional flow. 80 features are extracted. Attacks involve: Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS

Timeline pengerjaan

- Progres ke-1 : 30 Nopember 2023
- Progres ke-2 : 7 Desember 2023
- Progres ke-3 : 14 Desember 2023
- Presentasi Hasil Final Model : 21 Desember 2023
- Pengumpulan laporan akhir : 23 Desember 2023
  - Laporan dalam format Artikel dengan template IEEE dan sitasi IEEE.
  - Sistematikanya :
    - Judul
    - Abstrak
    - Pendahuluan
    - Metode
    - Hasil dan pembahasan
    - Kesimpulan
    - Referensi