DOLab

live class

Supervised Classification #1

Outline

1. Model Klasifikasi dan Penggunaannya
2. Tipe Klasifikasi
3. Algoritma Klasifikasi #1 (SVM, DT, RF)
4. Evaluasi Model Klasifikasi
5. *Hands-on*

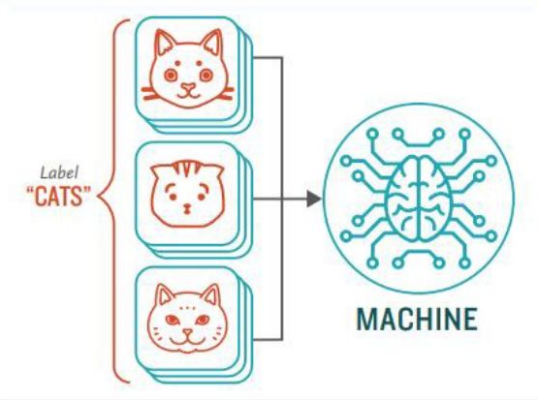
Apa Itu Model Klasifikasi?

MACHINE LEARNING TYPES

Supervised	• Menggunakan dataset **memiliki label** (E) untuk memprediksi varible target (T)
Unsupervised	• Menggunakan dataset **tanpa label** (E) untuk melihat/mempelajari pola (T)
Semi-supervised	• Menggunakan data **dg label** dan **tanpa label** (E) untuk memprediksi / mempelajari pola (T)
Reinforced Learning	• Menggunakan data hasil simulasi secara iterative (E) untuk mencapai tujuan (T) (memperbesar **reward** / mengurangi error)

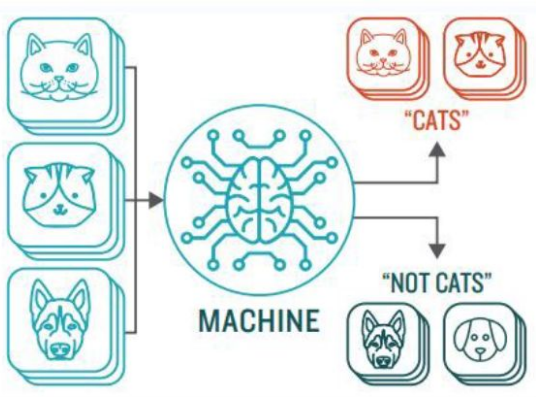
HOW SUPERVISED LEARNING WORKS

STEP 1: Training

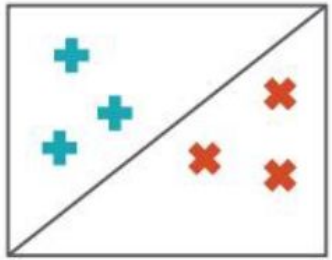


Studying patterns from previously labeled data that are then used to predict labels for new data.

STEP 2: Predicting

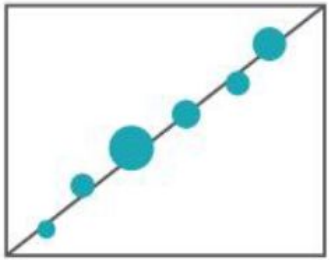


Different Types Based on Target Variable



CLASSIFICATION

Sorting items into categories

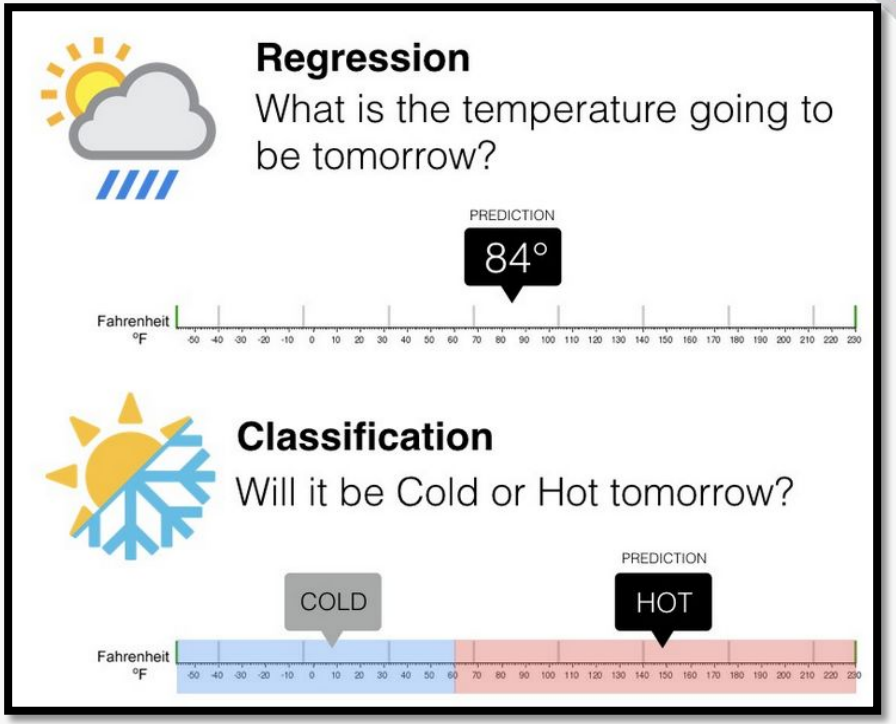


REGRESSION

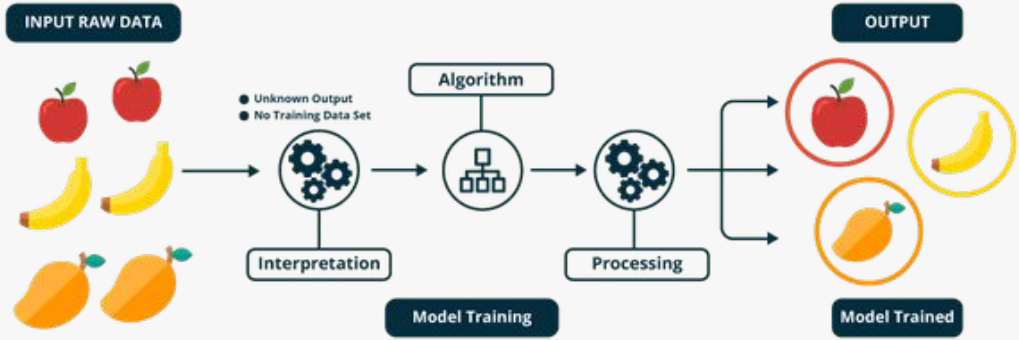
Identifying real values (dollars, weight, etc.)

CLASSIFICATION VS REGRESSION

Regression	Classification
Digunakan untuk memprediksi data continuos (*continuos quantity*)	Digunakan untuk memprediksi label dikret pada suatu kelas (*discrete class label*)
Regresi dengan multiple input biasa disebut multivariate regression	Klasifikasi dengan 2 label kelas disebut binary dan lebih dari 2 kelas disebut dengan multi-class
Scoring yang umum digunakan : RMSE, R^2, MAE, MAPE	Scoring yang umum digunakan : Accuracy, F1-score, ROC-AUC
Contoh : prediksi harga rumah, prediksi GDP, prediksi pertumbuhan penduduk.	Contoh : *fraud-detection, email spam filter,image classification*.



CLASSIFICATION USE CASES



Handwriting recognition : digunakan untuk menginterpretasikan masukan tulisan tangan yang dapat dimengerti dari sumber seperti dokumen kertas, foto, layar sentuh, dan perangkat lainnya.

Web search engine: digunakan untuk mengklasifikasikan informasi di World Wide Web.

Speech recognition: digunakan untuk pengenalan dan terjemahan bahasa lisan menjadi teks oleh komputer.

Biological classification: digunakan untuk mengklasifikasikan organisme biologis berdasarkan karakteristik bersama (taksonomi).

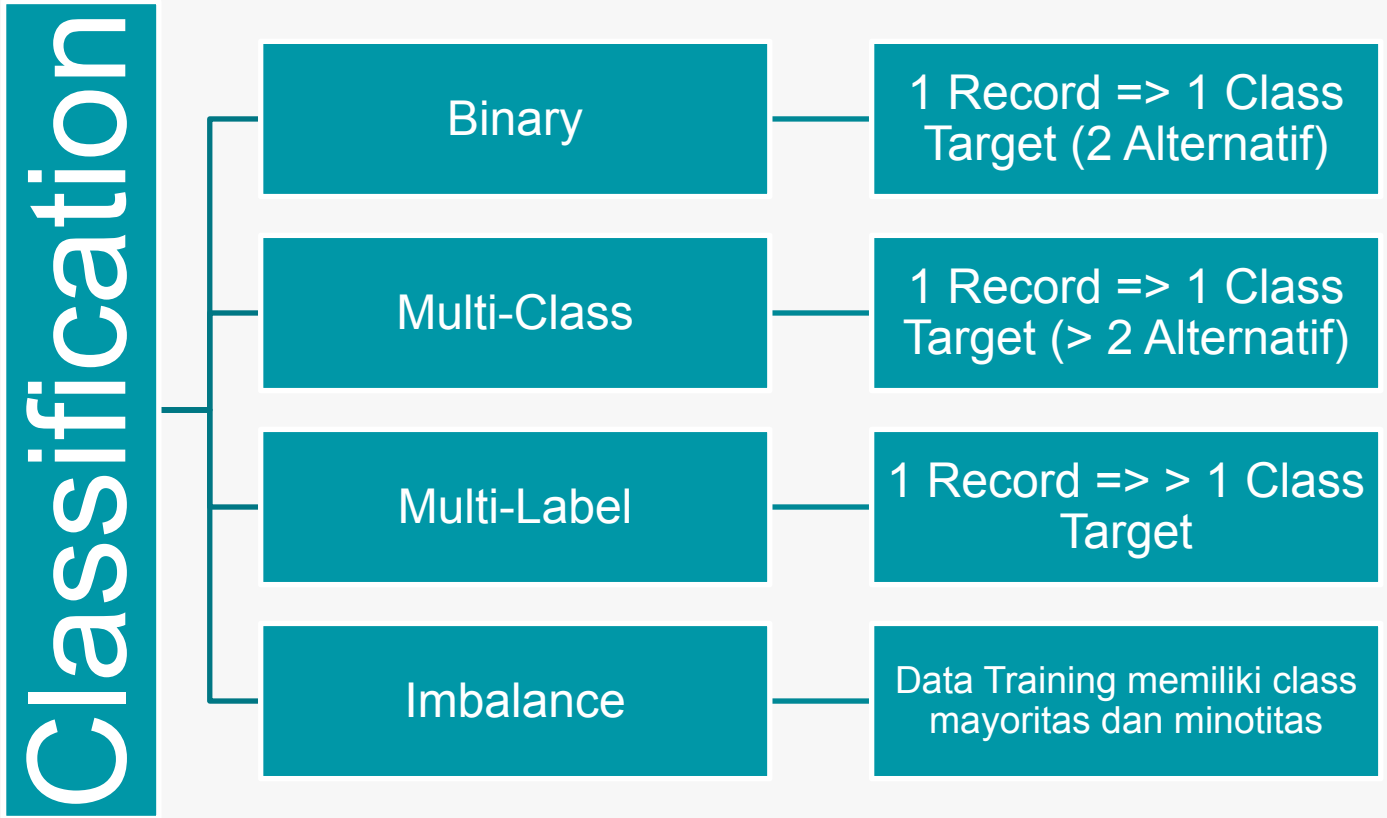
Credit scores : digunakan untuk menentukan siapa yang memenuhi syarat untuk pinjaman, dengan suku bunga berapa, dan batasan kredit apa.

Spam Detection - Mengklasifikasikan email sebagai spam atau bukan spam.

Sentiment Analysis - Memprediksi apakah suatu teks atau komentar memiliki sentimen positif, negatif atau netral.

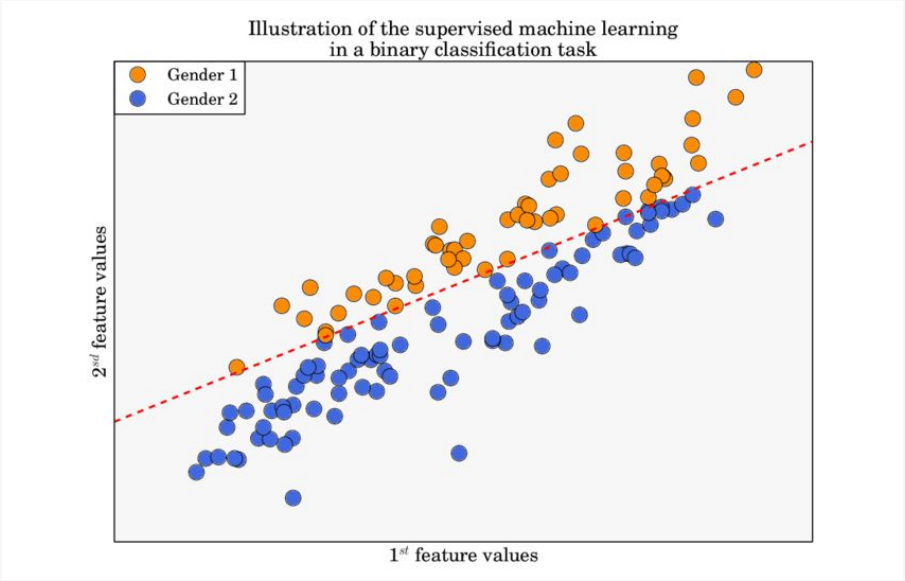
Tipe Kasus Klasifikasi

CLASSIFICATION CASE TYPES



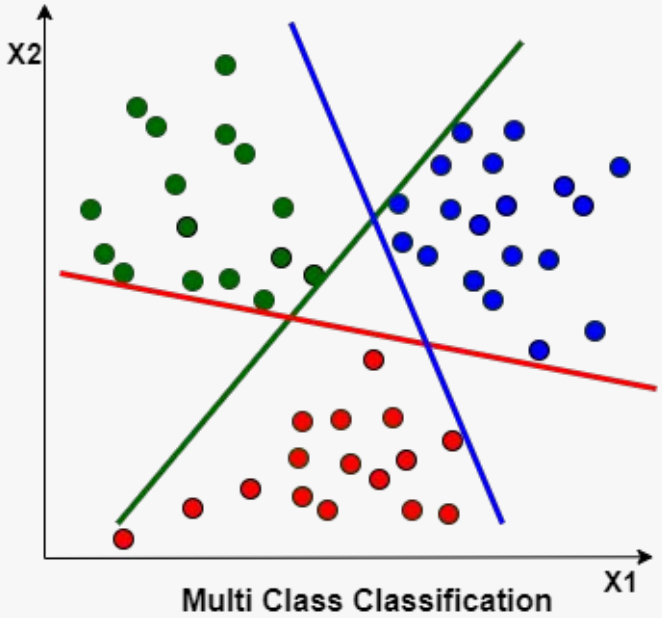
Binary Classification

- Klasifikasi biner melibatkan prediksi suatu data input sebagai salah **satu dari dua kelas** yang mungkin.
- Artinya, kategori dalam target variabel hanya ada dua, misalnya Ya atau Tidak, 1 atau 0.
- Contoh problem klasifikasi biner:
    - Deteksi *spam* (*spam* atau bukan *spam*)
    - Prediksi *churn* (*churn* atau tidak *churn*)



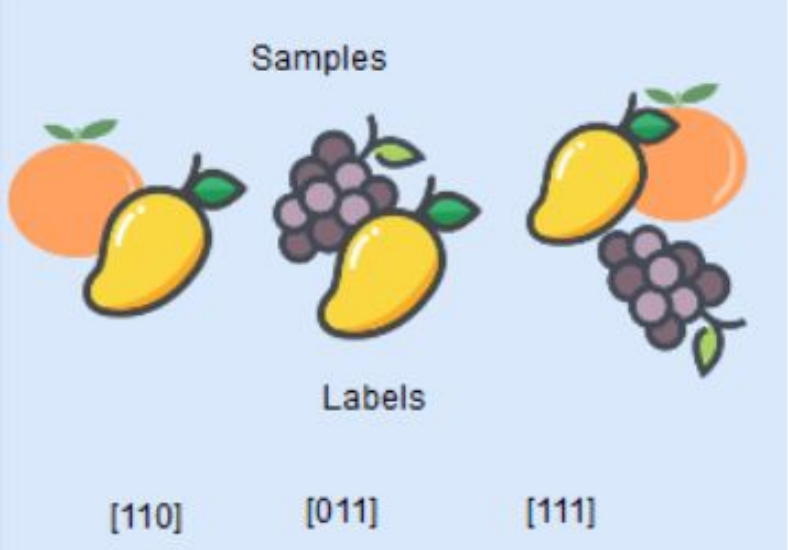
Multi-Class Classification

- Klasifikasi multi-kelas melibatkan prediksi suatu data input sebagai **salah satu dari tiga atau lebih kelas** yang mungkin.
- Artinya, jumlah kategori dalam target variabel lebih dari dua.
- Contoh problem klasifikasi multi-kelas:
  - Klasifikasi gambar
  - Klasifikasi teks/dokumen



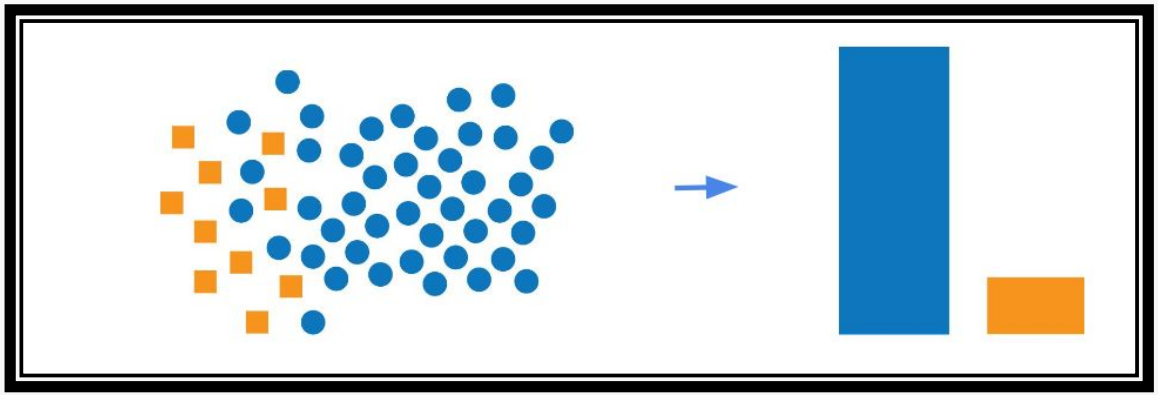
Multi-Label Classification

- Klasifikasi multi-label melibatkan prediksi suatu data input ke dalam **lebih dari satu kelas atau label secara bersamaan**.
- Contoh problem klasifikasi multi-label:
  - Klasifikasi gambar (beberapa objek)
  - Klasifikasi dokumen



Imbalance Classification

- Imbalanced classification mengacu pada problem klasifikasi di mana distribusi kelas dalam dataset tidak merata.
- Contoh problem klasifikasi data tidak seimbang:
  - Deteksi penipuan kartu kredit
  - Identifikasi kanker



Algoritma Klasifikasi #1

Support Vector Machine, Decision Tree, Random Forest

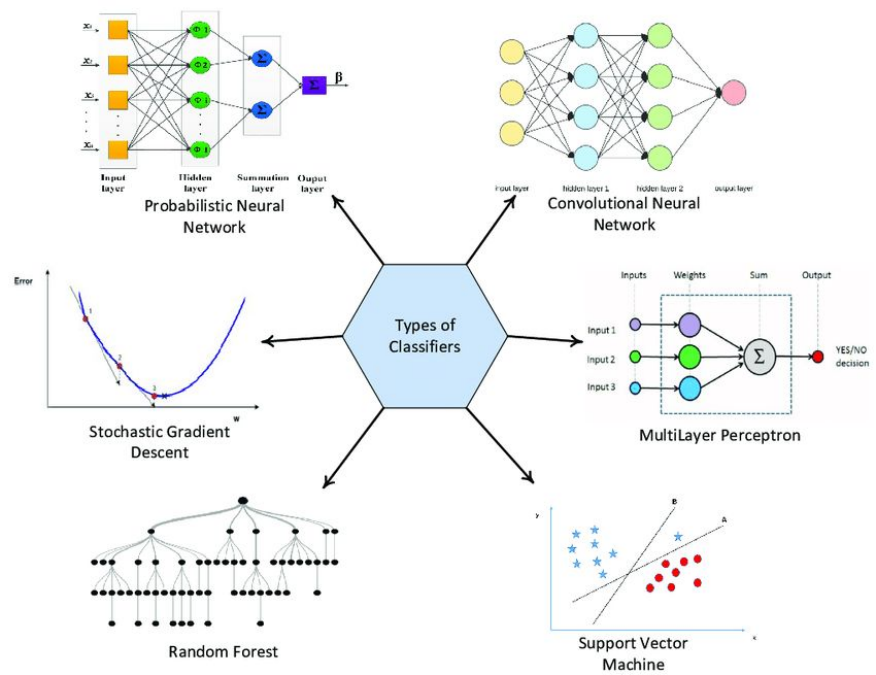
Alternative For Classification Case

Live Class:

1. **Support Vector Machine (Binary only)**
2. **Decision Tree**
3. **Bagging : Random Forest**
4. K Nearest Neighbor
5. Naïve Bayes

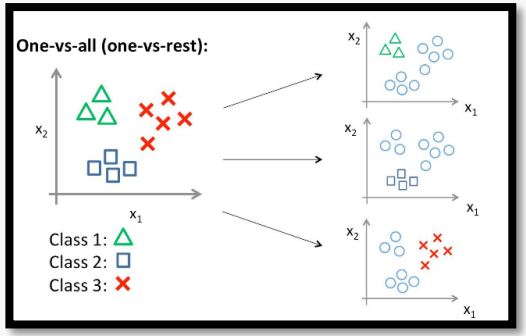
Supplementary:

1. Logistic Regression (Binary only)
2. Boosting : AdaBoost, XGBoost, LGBM
3. Stacking : Voting
4. Artificial Neural Network



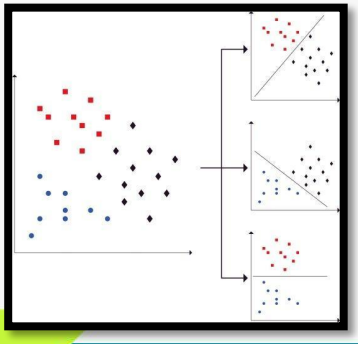
Binary only Algo for Multi-class Classification

Algoritma yang secara khusus dirancang untuk binary classification dapat diadaptasi untuk digunakan dalam masalah multi-class classification. Terdapat dua metode:



1. One-vs-Rest (OVR)

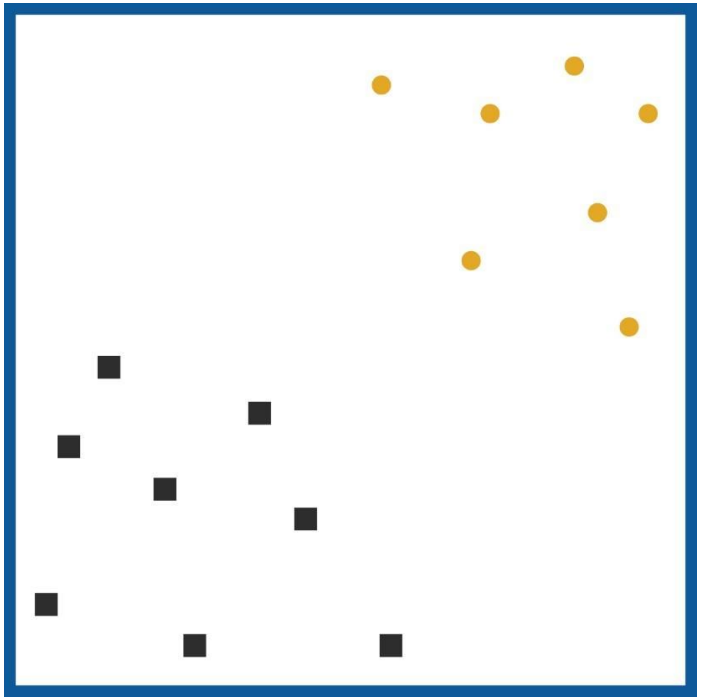
- Setiap kelas diperlakukan sebagai masalah klasifikasi biner terpisah.
- Untuk setiap kelas, klasifikasi biner tersebut dilatih untuk membedakan dari semua kelas lainnya.
- Klasifikasi dengan skor tertinggi dipilih sebagai kelas yang diprediksi.
- Dalam hal ini, OVR menghasilkan k klasifikasi biner problem untuk k kelas.



2. One-vs-One (OVO)

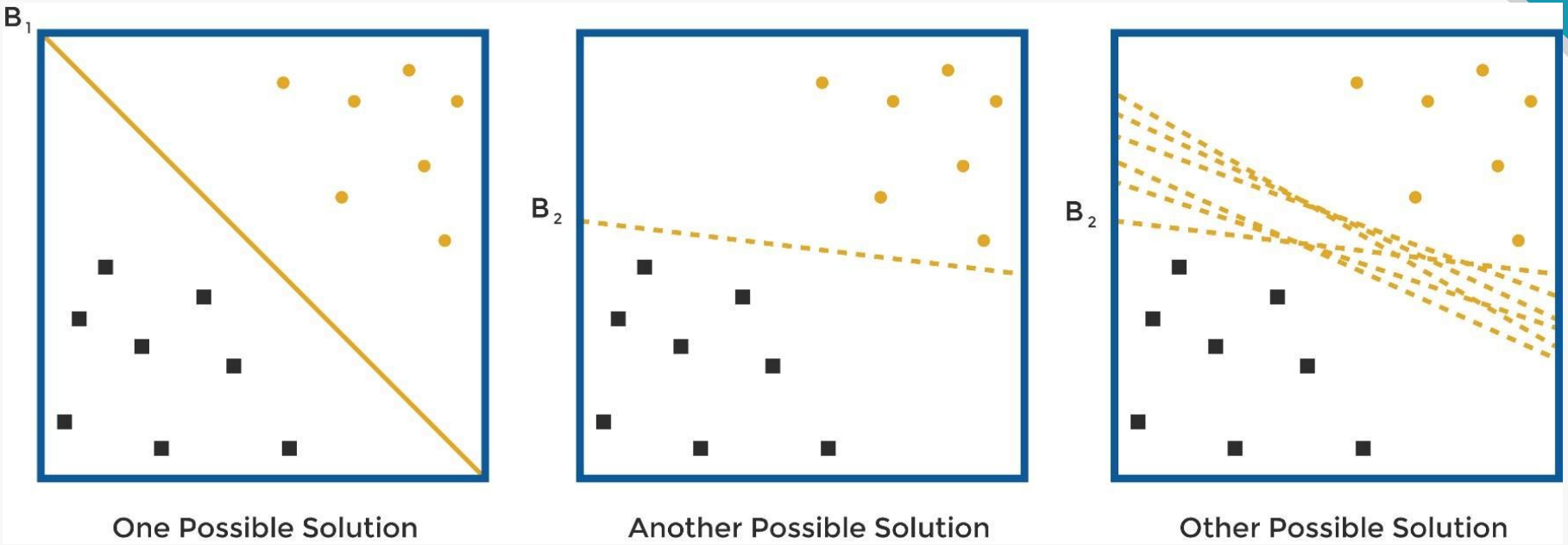
- Semua pasangan kemungkinan dari kelas dilatih dengan klasifikasi biner untuk membedakan antara setiap pasangan.
- Jika terdapat k kelas, maka k(k-1)/2 klasifikasi biner dilatih.
- Selama prediksi, setiap klasifikasi biner menghasilkan skor untuk setiap pasangan kelas yang mungkin.
- Kelas yang menang dalam sebagian besar kontes klasifikasi biner dipilih sebagai kelas yang diprediksi.

Support Vector Machine



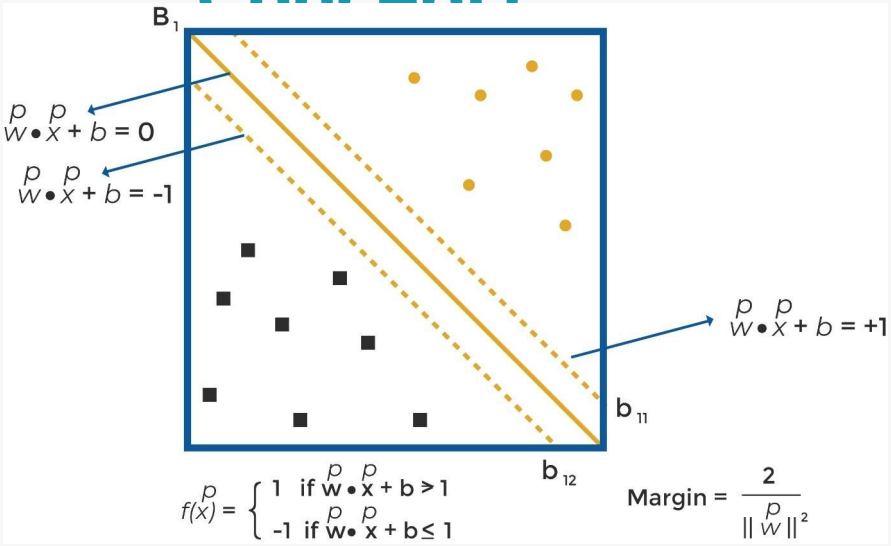
Tentukan hyperplane dalam bentuk garis (decision boundary yang dapat memisahkan data

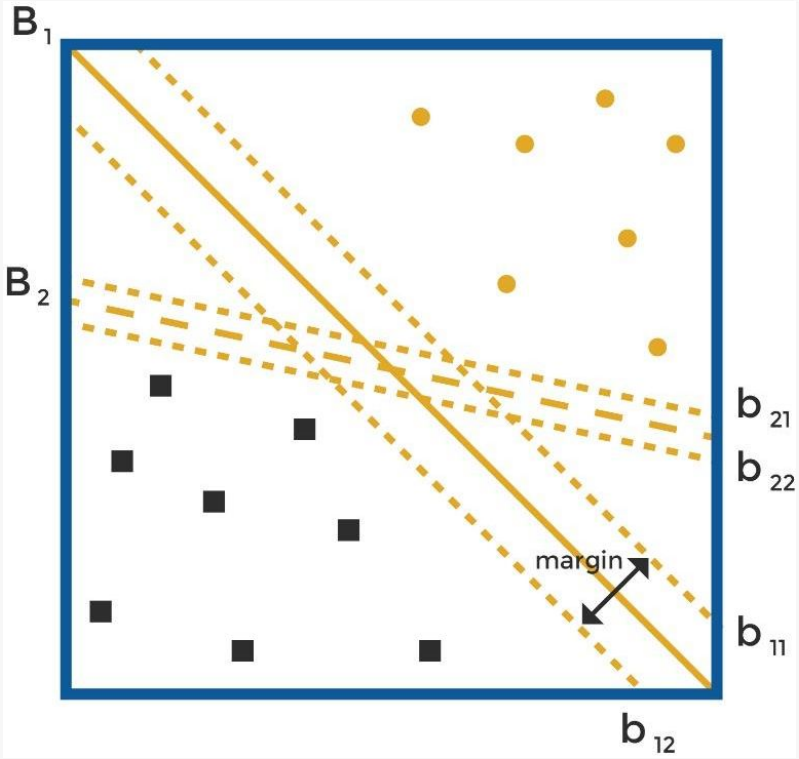
Support Vector Machine



- **Mana yang lebih baik? B1? B2?**
- **Bagaimana kita membandingkan kedua decision boundary?**

Support Vector Machine (Core Concept)





Salah satu caranya adalah mencari garis yang memiliki jarak margin (ke titik terdekat) paling maksimal

Sehingga B1 > B2

Support Vector Machine (Issue #1)

Bagaimana jika masalahnya tidak dapat dipisahkan dengan menggunakan garis?

- Bisa tetap menggunakan garis sebagai *boundary*, tetapi mengijinkan beberapa data yang salah klasifikasi (*soft margin*)
- Caranya adalah menambahkan slack variable / penalty (ξ) pada *objective function*-nya:

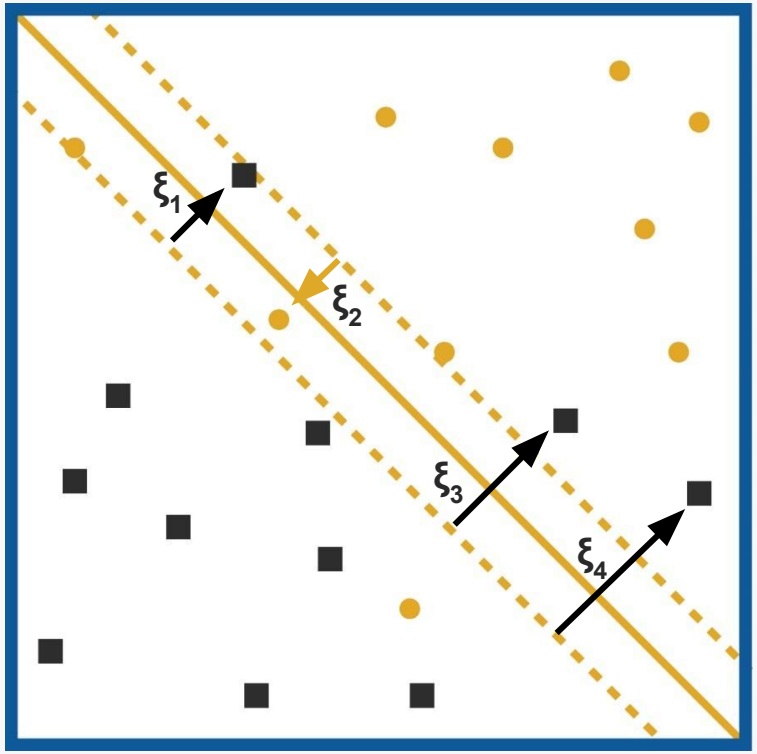
$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

Such that: $(w^T x_i + b)y_i \geq 1 - \xi_i$

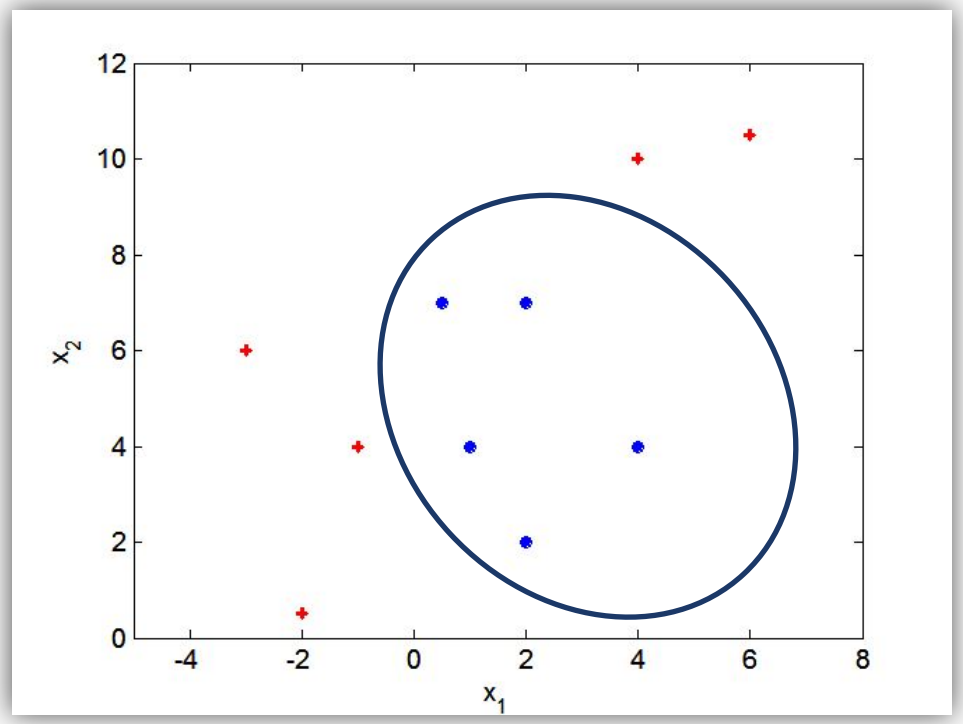
$\xi_i \geq 0$

for i = 1, … , m

- Pada saat proses training model, variable penalty ini akan di-*minimize*



Support Vector Machine (Issue #2)



Bagaimana jika boundary yang seharusnya tidak linear?

- Pergunakan apa yang disebut *kernel trick*
- Gunakan fungsi kernel yang mengubah data kita ke dalam ruang fitur / variable x sehingga data lebih mungkin dapat dipisahkan secara linear.
- Contoh *kernel function* yang dapat digunakan:
  - Linear
  - Polynomial
  - Gaussian radial basis function (RBF)

Support Vector Machine (Summary)

Advantages

- SVM sangat baik digunakan ketika kita tidak memiliki ide tentang data tersebut.
- Bekerja dengan baik bahkan dengan data yang tidak terstruktur dan semi-terstruktur seperti teks, gambar, dan pohon.
- Kernel trick adalah kekuatan sebenarnya dari SVM. Dengan fungsi kernel yang sesuai, kita dapat memecahkan masalah kompleks apa pun.
- Berbeda dengan neural network, SVM tidak memperhatikan optimum lokal.
- SVM relatif baik dalam menangani data dengan dimensi yang tinggi.
- Model SVM memiliki generalisasi dalam praktiknya, sehingga risiko overfitting lebih rendah dalam SVM.

Disadvantages

- Memilih fungsi kernel yang "baik" tidak mudah.
- Waktu pelatihan yang lama untuk dataset yang besar.
- Sulit untuk memahami dan menginterpretasikan model akhir, bobot variabel dan dampak individunya.
- Karena model akhirnya tidak mudah terlihat, kita tidak dapat melakukan kalibrasi kecil pada model sehingga sulit untuk menggabungkan logika bisnis kita.

DQLab

Decision Tree

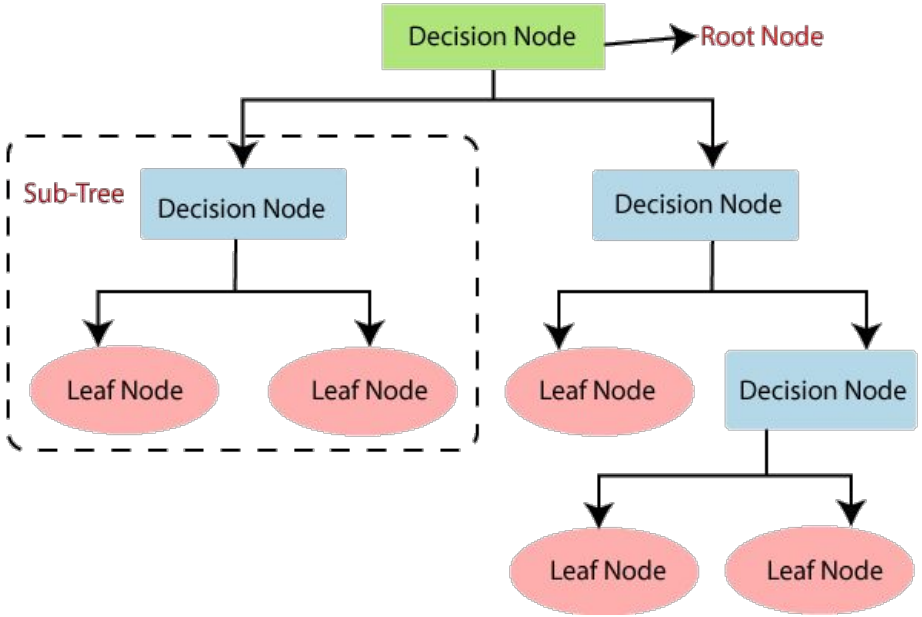


diagram berbentuk pohon yang digunakan untuk menentukan suatu tindakan. Setiap cabang pohon mewakili keputusan, kejadian, atau reaksi yang mungkin terjadi.

Decision Tree – Classification Task

Important Terms

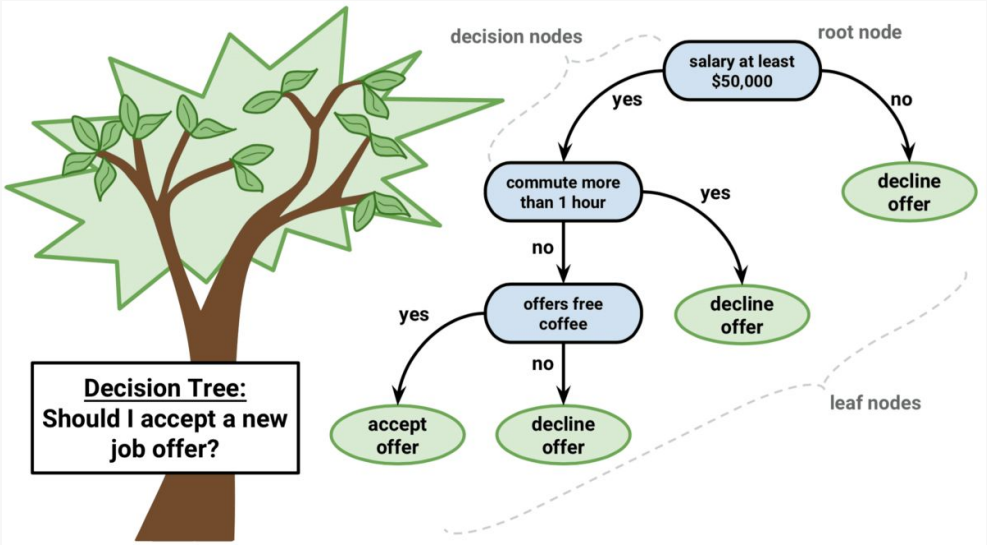
- **Root Node:**

  Node yang tidak mempunyai edge yang masuk dan 0 atau banyak edge yang keluar.
- **Splitting**

  Proses membagi node ke sub-nodes.
- **Decision Node**

  Node yang mempunyai satu edge yang masuk dan dua atau lebih edge yang keluar.
- **Leaf Node**

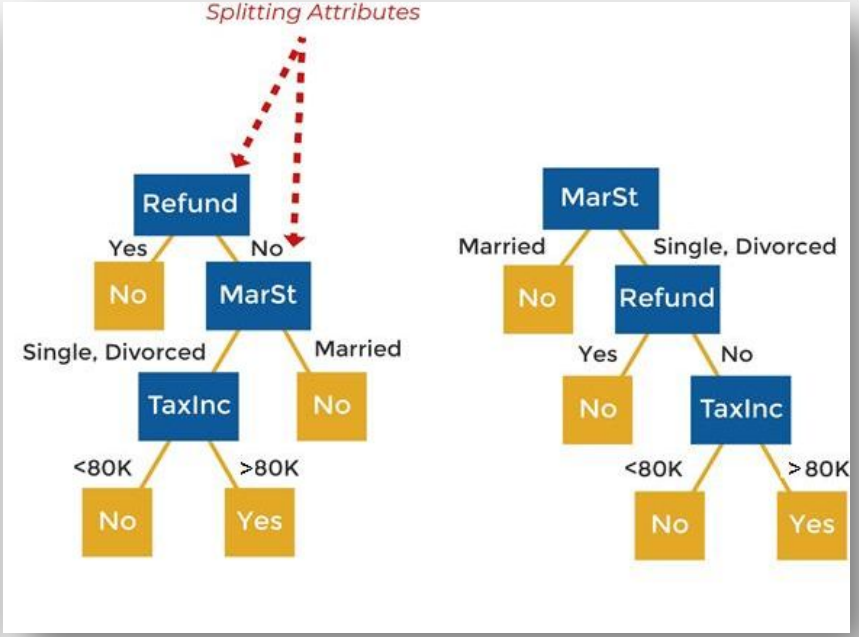
  Node yang mempunyai satu edge yang masuk dan tidak ada edge keluar. Leaf Node merepresentasikan prediksi kelas yang dihasilkan struktur tree tersebut.



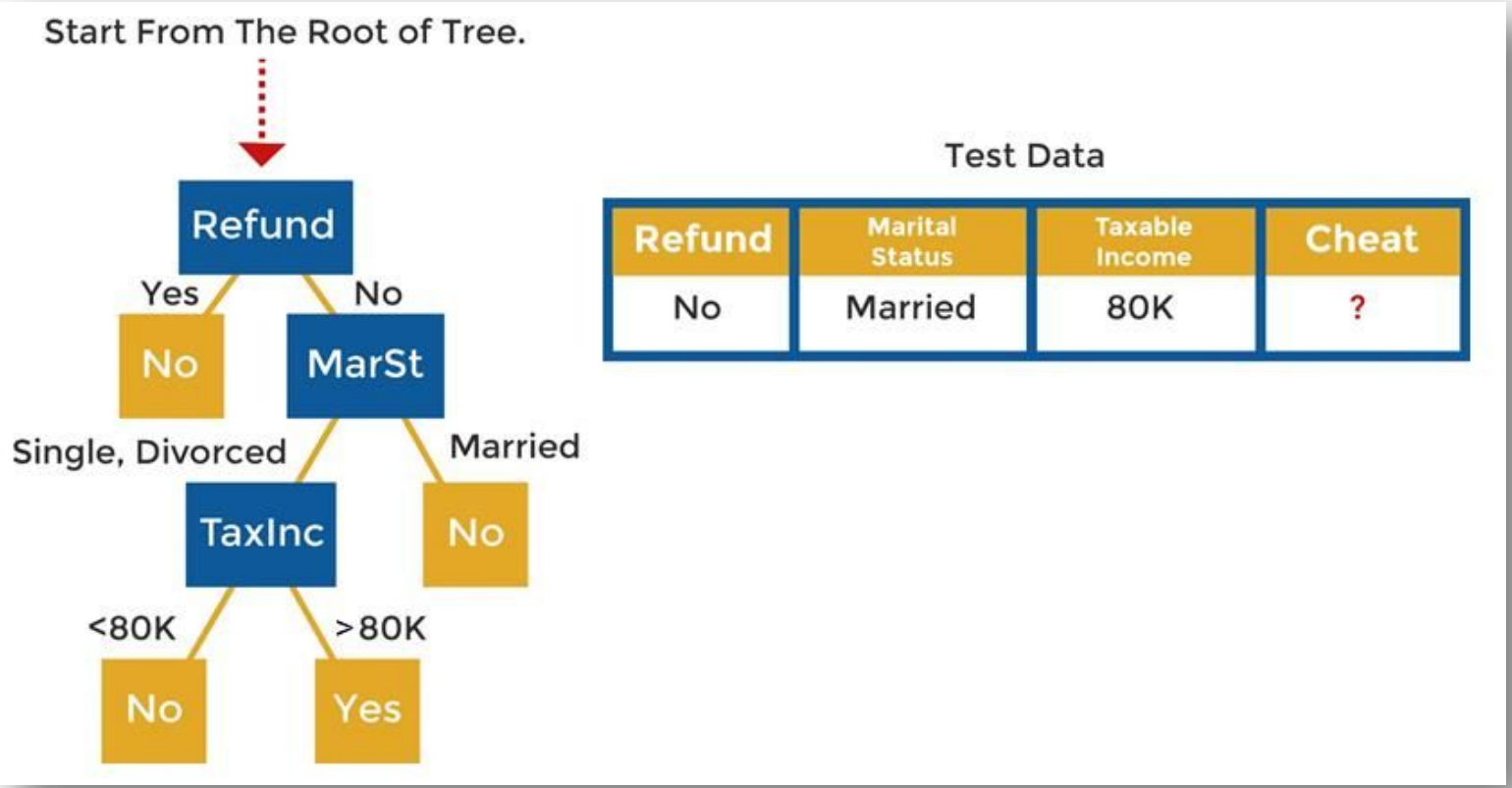
Decision Tree – Example - Training

	Categorical	Categorical	Continuous	Class
TID	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Decision Tree – Example - Testing

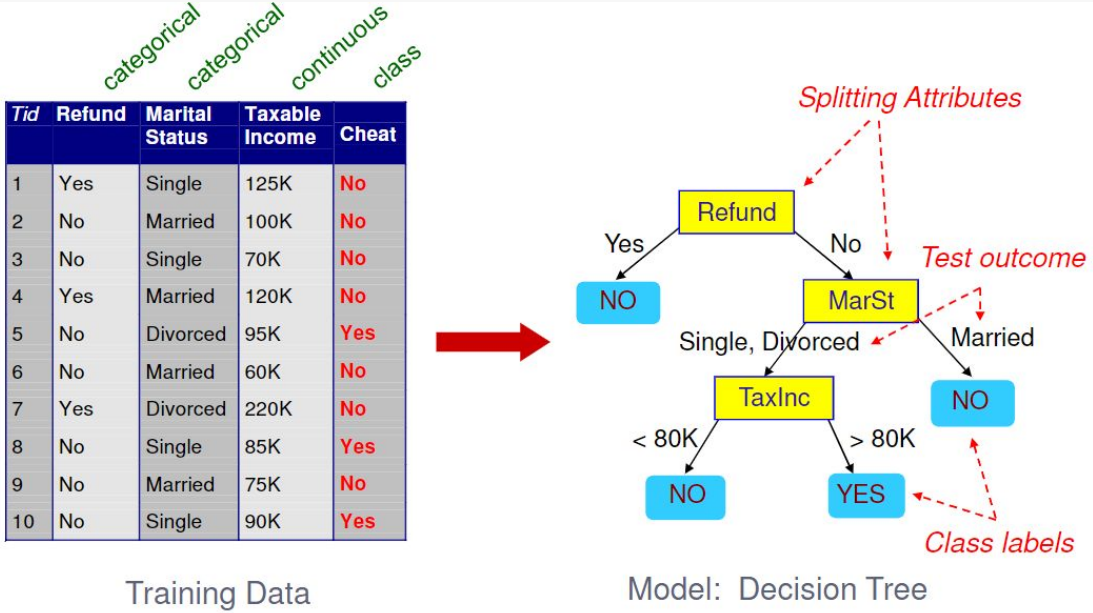


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Impurity Measurements

Tantangan utama: memilih attribute sebagai dasar splitting



Pemilihan attribute berdasarkan kriteria yang disebut "*impurity measure*"

Impurity measure adalah nilai yang menunjukan seberapa banyak kelas atau nilai target yang berbeda dalam kumpulan data tersebut

Contoh measurements:

- *Entropy*
- *Information Gain*
- *Gini Impurity* atau *Gini Index*

Attribute dengan nilai impurity terendah atau information gain tertinggi akan menjadi attribute splitting

Impurity Measurements

Entropy

ukuran ketidakpastian atau keacakan dalam sebuah himpunan data atau kelas

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy Target

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

Entropy Atribut

Information Gain

seberapa banyak informasi baru yang diperoleh dengan membagi data pada atribut tertentu

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Gini Index

seberapa seragam distribusi kelas dalam suatu kumpulan data

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Jenis Algoritma Decision Tree

ID3

- ID3 (*Iterative Dichotomiser 3*) menggunakan metode ***Information Gain*** untuk memilih fitur terbaik pada setiap node dan membangun pohon keputusan secara iterative
- Algoritma ini biasanya digunakan untuk memproses data kategorikal

C4.5

- Pengembangan dari ID3
- Algoritma ini menggunakan metode ***Gain Ratio*** untuk memilih fitur terbaik pada setiap node dan dapat memproses data kategorikal dan numerik.

CART

- Algoritma CART (*Classification and Regression Trees*) dapat digunakan untuk membangun pohon keputusan untuk masalah klasifikasi dan regresi.
- Algoritma ini menggunakan metode ***Gini Index*** untuk memilih fitur terbaik pada setiap node.

Decision Tree – Summary

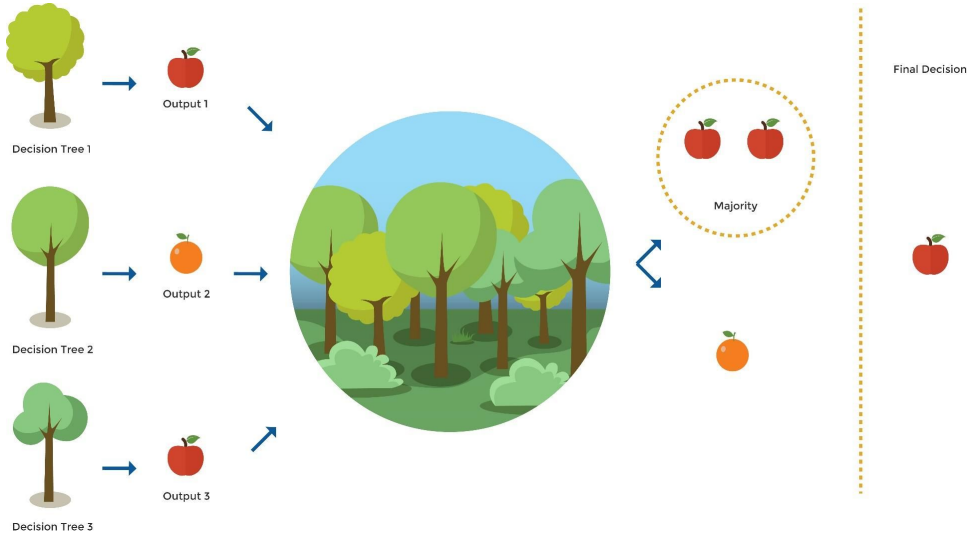
Advantages

- Tidak memerlukan sumber daya yang banyak untuk dibangun
- Sangat cepat dalam mengklasifikasikan data yang tidak diketahui
- Mudah diinterpretasikan untuk pohon dengan ukuran kecil
- Akurasi yang sebanding dengan teknik klasifikasi lainnya untuk banyak set data sederhana

Disadvantages

- Over-fitting ketika algoritma menangkap noise dalam data
- Model dapat menjadi tidak stabil karena variasi kecil dari data
- Pohon dengan bias rendah: sulit bagi model untuk bekerja dengan data baru

Random Forest



Metode yang beroperasi dengan membangun beberapa pohon keputusan selama tahap pelatihan. Keputusan final akan diambil dari keputusan mayoritas decision tree dalam forest.

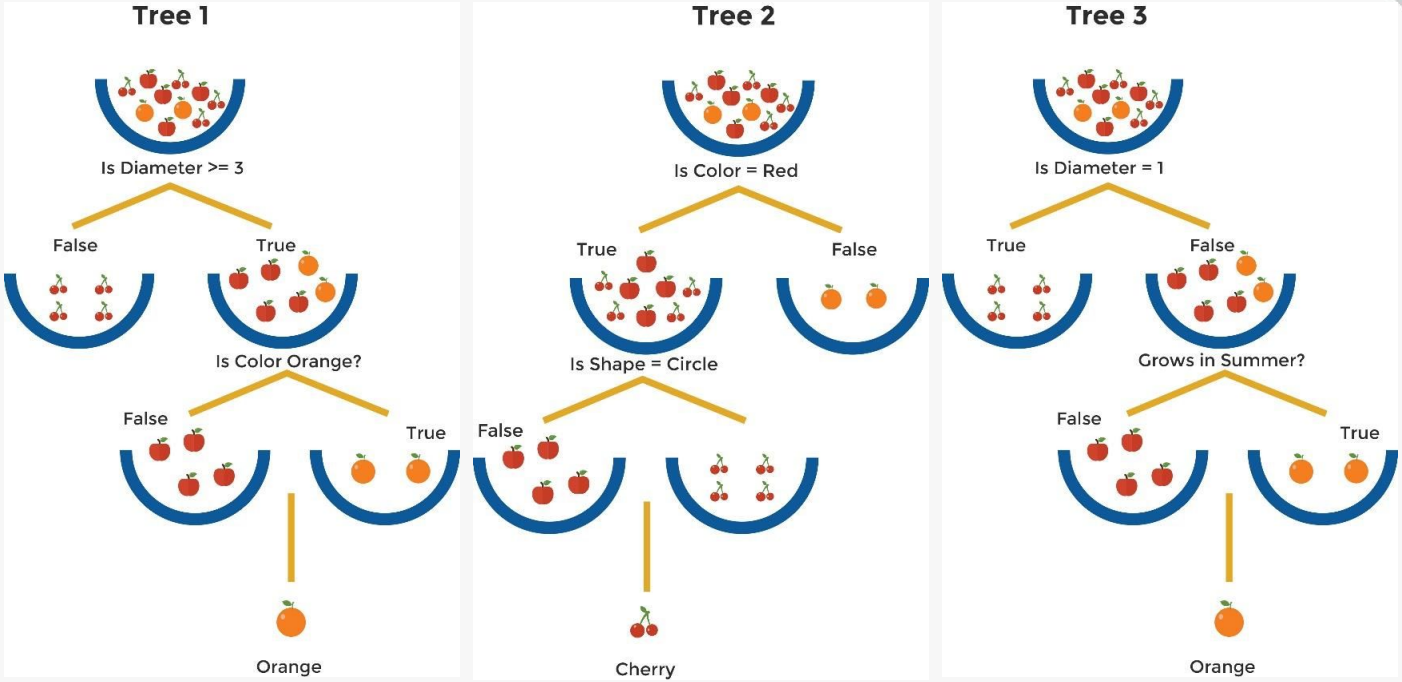
Random Forest = Lots of Decision Trees

Randomized training data:

- Bootstrap samples with replacements
- Select K variables which less than or equal to n variable

Bagging Many Decision Tree:

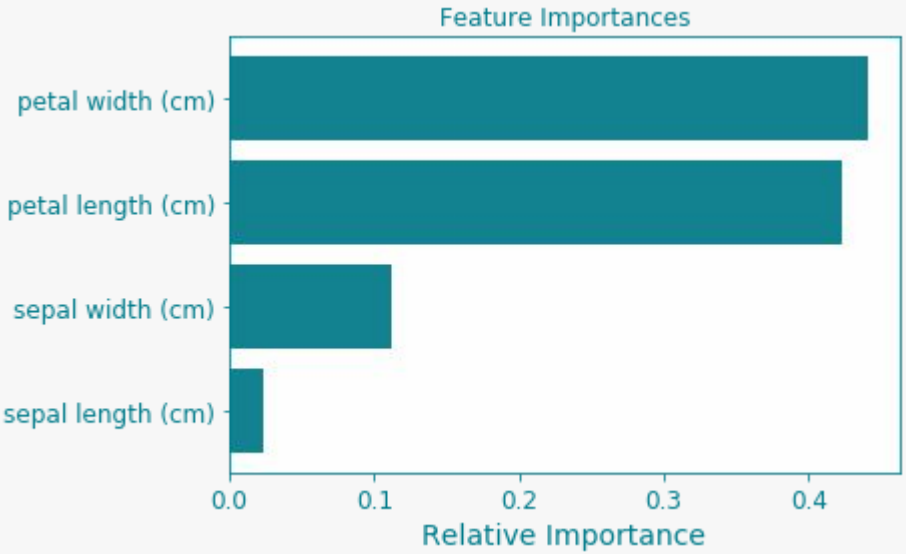
- Randomized data to train DT Model
- Voting from each DT decision



Major Vote is **ORANGE**

Random Forest – Feature Importance

fitur-fitur mana yang paling penting dalam menentukan prediksi.



Manfaat

- Membantu dalam memahami logika model
- Dapat digunakan untuk pemilihan variabel
- Dalam beberapa kasus bisnis, ada keuntungan dalam mengorbankan sedikit akurasi demi interpretabilitas.

Random Forest – Summary

Advantages

- dapat digunakan untuk kasus regresi maupun klasifikasi
- mudah untuk melihat tingkat peranan masing-masing fitur secara relatif atas prediksi target
- Dianggap sebagai algoritma yang sangat berguna dan mudah digunakan, karena hyperparameter default-nya sering menghasilkan hasil prediksi yang baik.

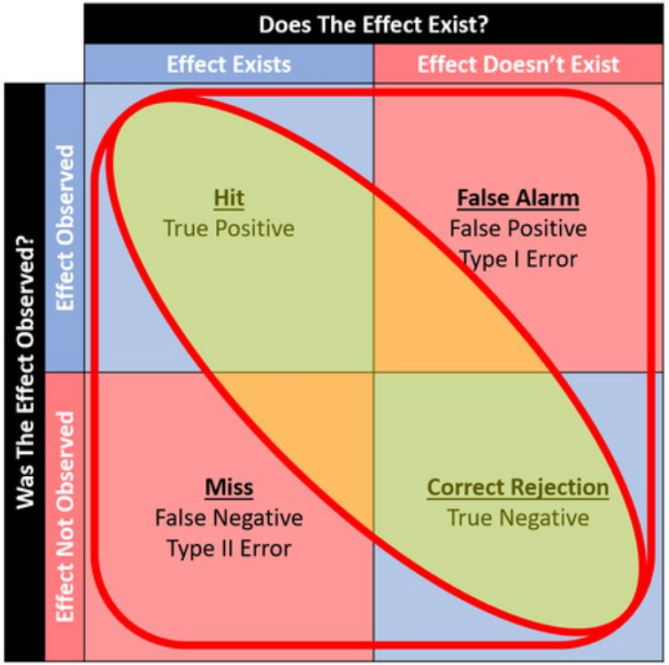
Disadvantages

- Prediksi yang lebih akurat membutuhkan lebih banyak pohon
- Pohon dalam jumlah besar dapat membuat algoritma menjadi lambat dan tidak efektif untuk prediksi real-time
- Ini adalah alat pemodelan prediktif dan bukan alat deskriptif

Evaluasi Model Klasifikasi

Prediksi Benar dan Salah

- **True Positive (TP):** Jumlah data yang bernilai Positif dan diprediksi benar sebagai Positif.
- **False Positive (FP):** Jumlah data yang bernilai Negatif tetapi diprediksi sebagai Positif.
- **False Negative (FN):** Jumlah data yang bernilai Positif tetapi diprediksi sebagai Negatif.
- **True Negative (TN):** Jumlah data yang bernilai Negatif dan diprediksi benar sebagai Negatif.



DQLab

Accuracy

Accuracy adalah metode evaluasi yang menghitung rasio antara jumlah data yang diprediksi benar dengan total jumlah data yang diuji.

$$Accuracy = \frac{Correct\ prediction}{Total\ cases} * 100\%$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100\%$$

Image Source

Precision

Precision adalah metode evaluasi yang menghitung rasio antara jumlah data yang diprediksi benar positif dengan total data yang diprediksi positif.

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

Image Source

Recall

Recall adalah metode evaluasi yang menghitung rasio antara jumlah data yang diprediksi benar positif dengan total data yang sebenarnya positif.

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

Image Source

F1-Score

F1-Score adalah metode evaluasi yang menghitung nilai rata-rata harmonis (*harmonic mean*) antara *precision* dan *recall*.

Nilai F1-score berkisar dari 0 hingga 1, dimana nilai 1 menunjukkan kinerja model yang sangat baik dan nilai 0 menunjukkan kinerja model yang buruk.

Dalam prakteknya, F1-score sangat penting digunakan ketika data tidak seimbang atau ketika kedua nilai presisi dan recall memiliki arti yang sama pentingnya. Contoh penggunaan F1-score dalam kehidupan sehari-hari adalah ketika kita ingin membandingkan kinerja dari beberapa model mesin pembelajaran untuk memilih model terbaik.

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

Image Source


Thanks!