DigitalSkola

LEARNING PROGRESS REVIEW

Week 11

Entropy Team

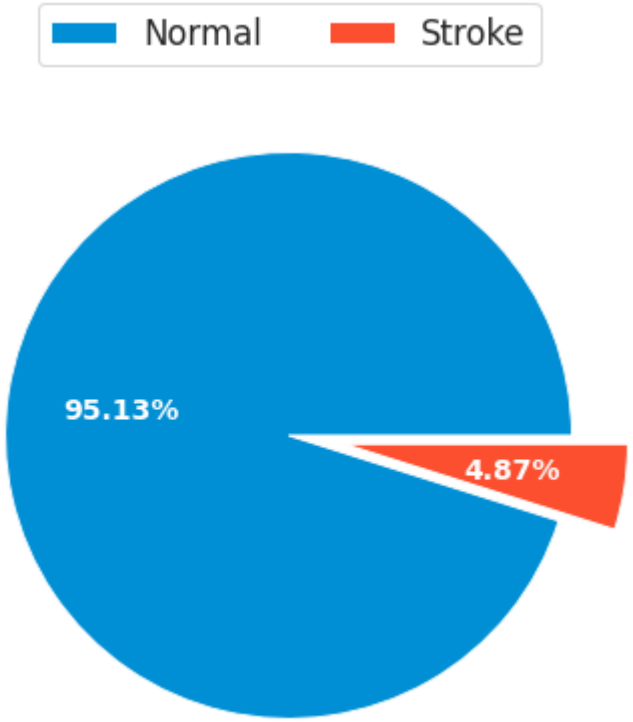
DAFTAR ISI

01

ADVANCED DATA PREPROCESSING

Imbalanced dataset &
text data

Imbalanced Dataset

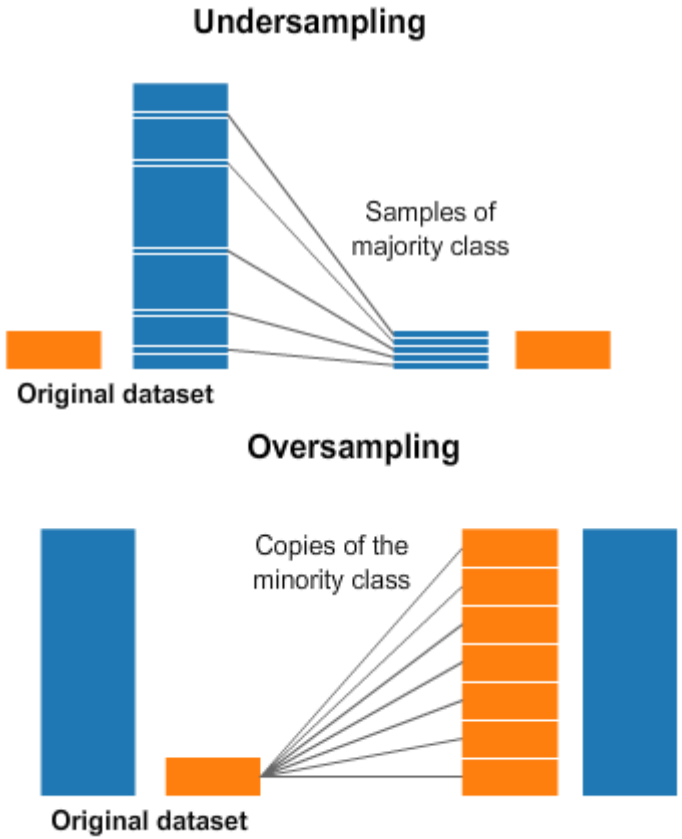


Imbalanced dataset yaitu suatu kondisi di mana **distribusi kelasnya tidak seimbang**

Sumber: https://www.kaggle.com/code/adhang/stroke-logistic-regression-smote-adasyn

DigitalSkola

Resampling



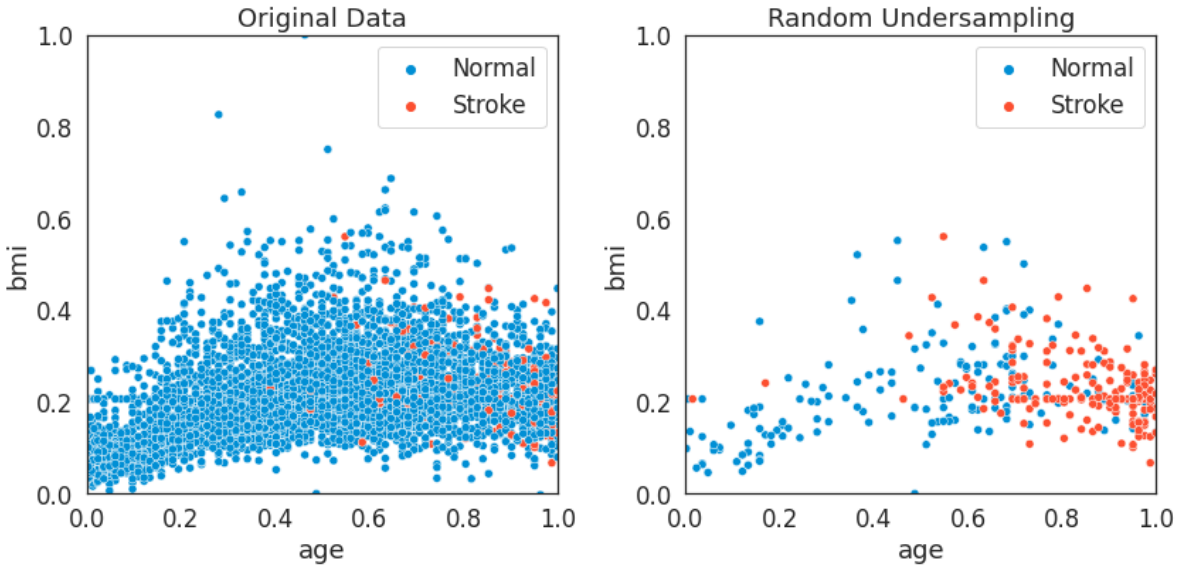
- *Resampling* merupakan salah satu metode yang dapat digunakan untuk **mengatasi *imbalanced dataset***

- Secara umum, *resampling* dibagi menjadi 2 jenis, yaitu
  - **Undersampling**
  - **Oversampling**

Sumber: https://www.kaggle.com/code/rafjaa/resampling-strategies-for-imbalanced-datasets

DigitalSkola

Random Undersampling

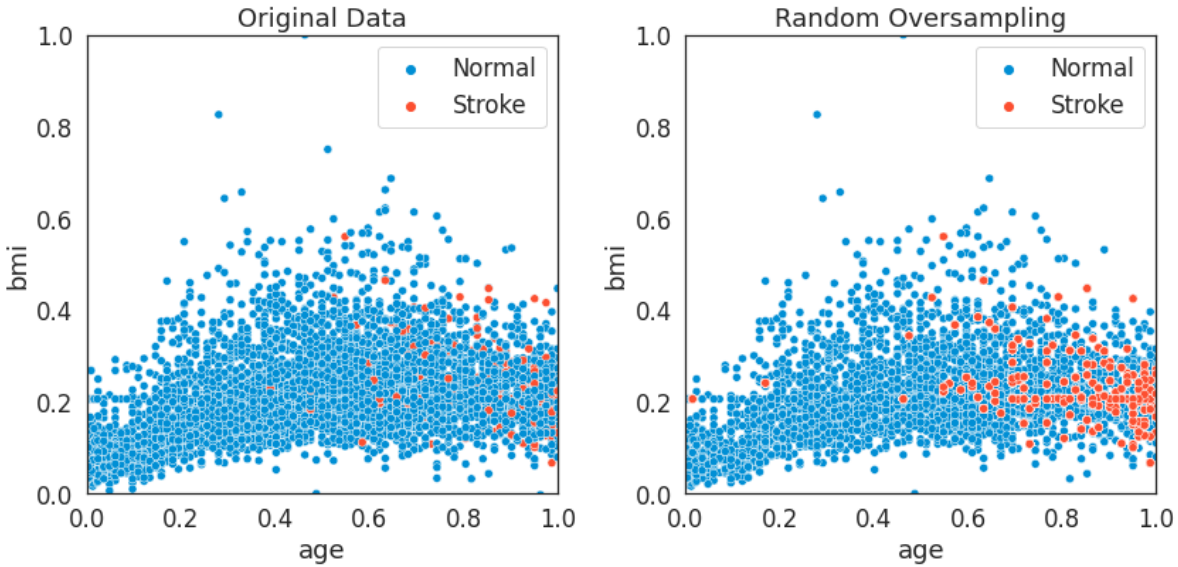


Menyeimbangkan distribusi kelas dengan **menghapus** sebagian dari **kelas mayoritas** secara acak

Sumber: https://www.kaggle.com/code/adhang/stroke-logistic-regression-smote-adasyn

DigitalSkola

Random Oversampling

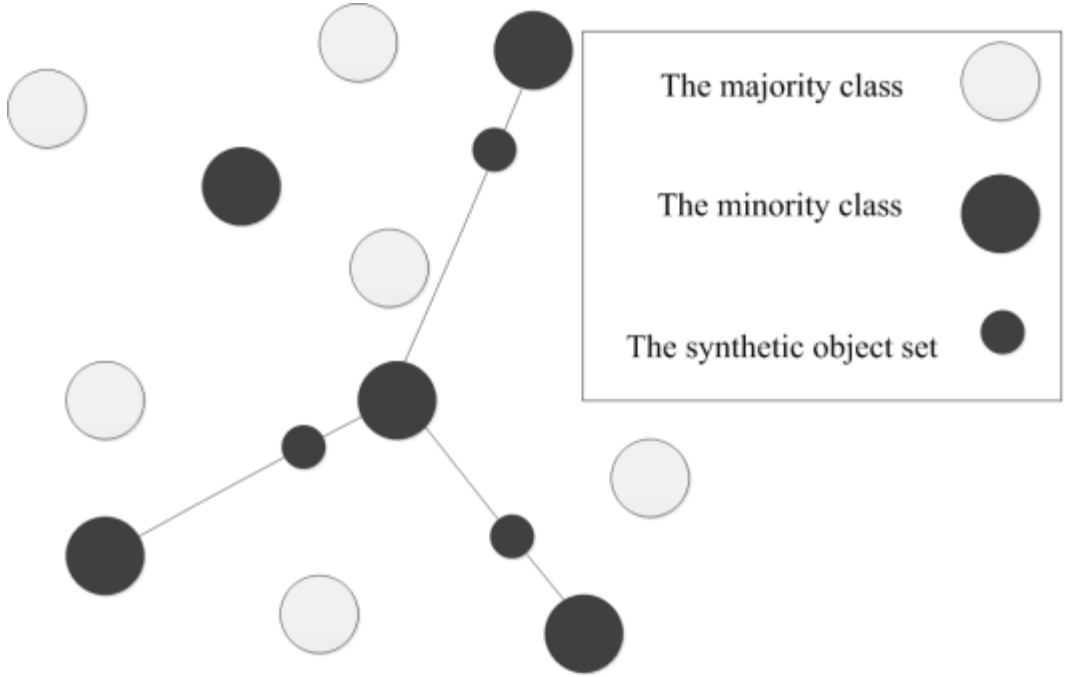


Menyeimbangkan distribusi kelas dengan **menduplikasi** sebagian dari **kelas minoritas** secara acak

Sumber: https://www.kaggle.com/code/adhang/stroke-logistic-regression-smote-adasyn

DigitalSkola

SMOTE

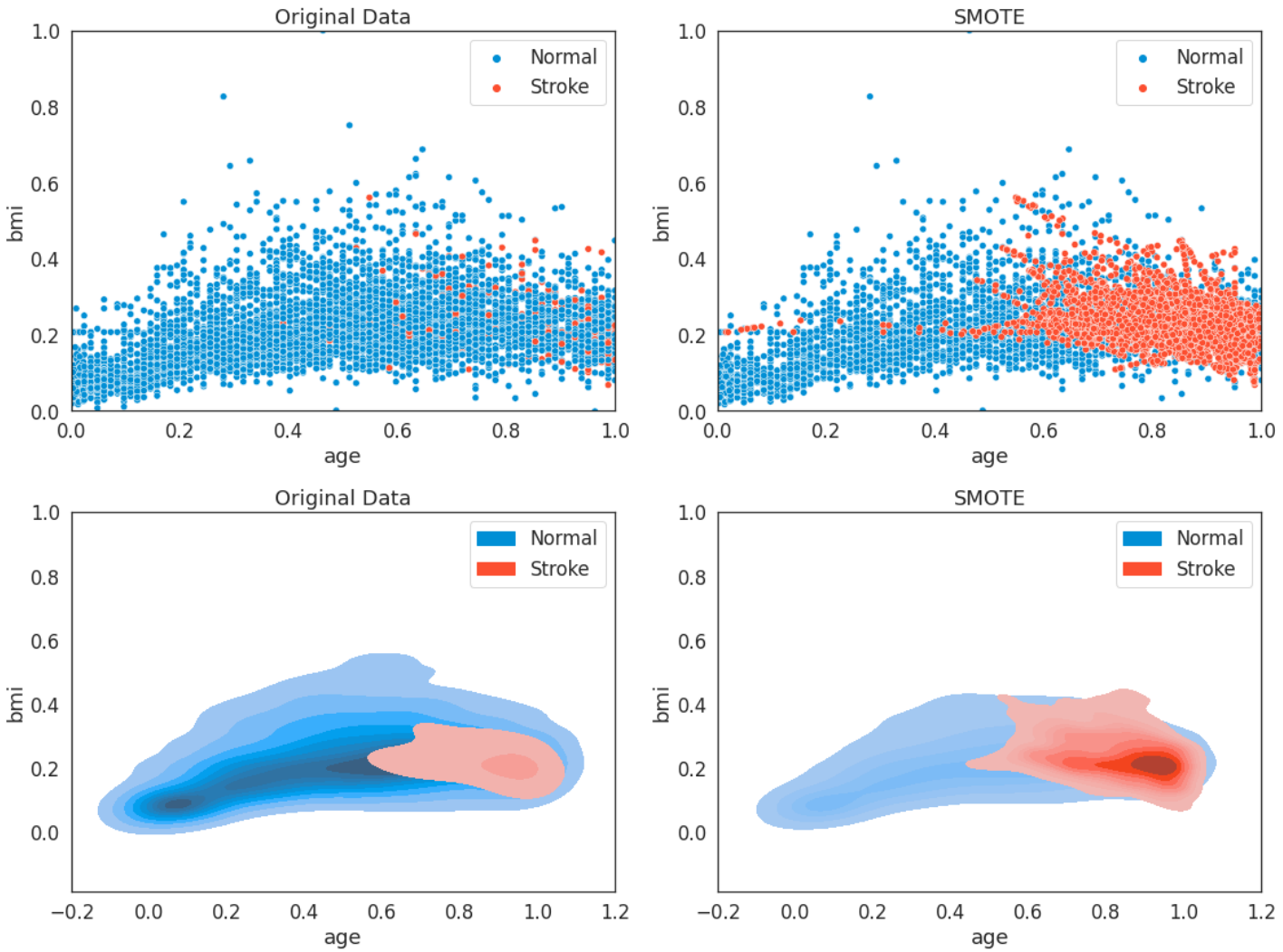


- *Synthetic Minority Oversampling Technique* (SMOTE)
- Secara garis besar, SMOTE membuat **sampel sintetis** yang berada **di antara sampel kelas minoritas** yang dipilih secara **acak**

Sumber: https://www.researchgate.net/figure/The-diagram-of-SMOTE-algorithm_fig1_315302210

DigitalSkola

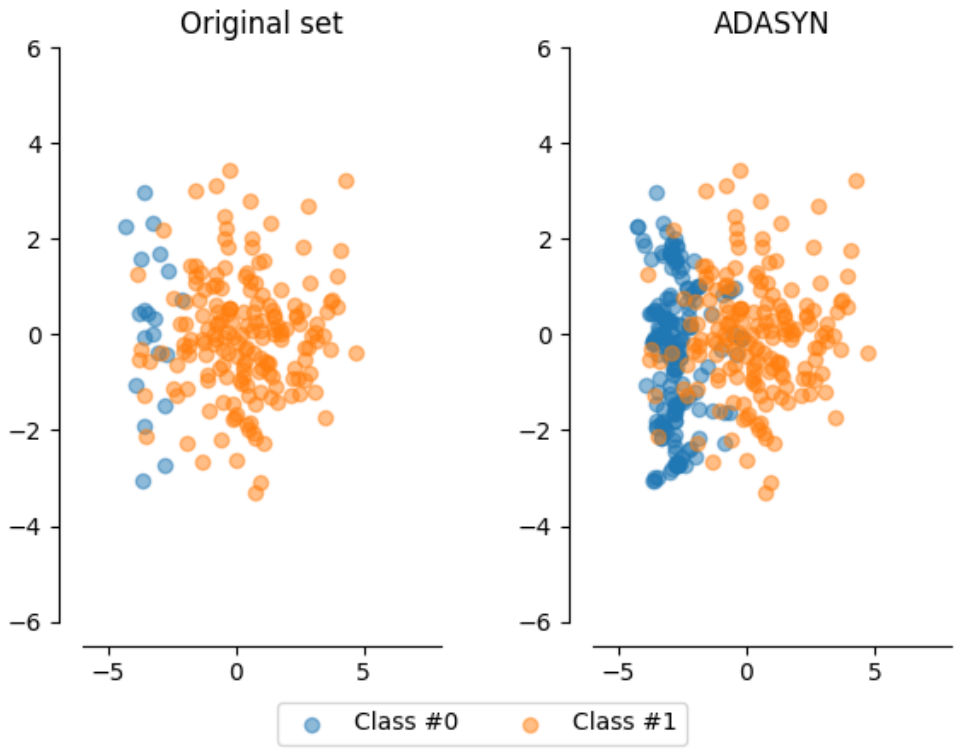
Contoh Hasil SMOTE



Sumber: https://www.kaggle.com/code/adhang/stroke-logistic-regression-smote-adasyn

DigitalSkola

ADASYN

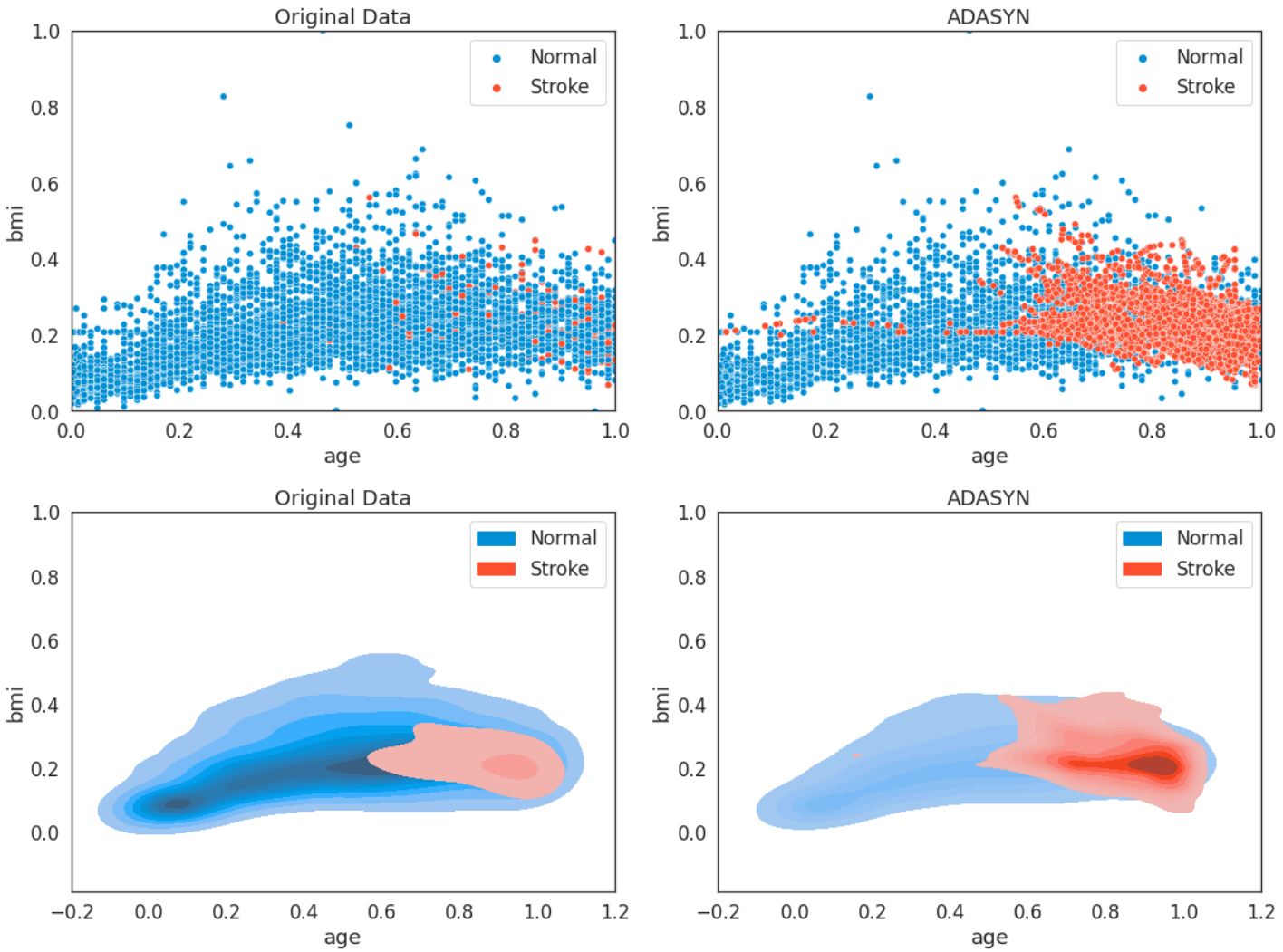


- *Adaptive Synthetic* (ADASYN)
- Secara garis besar, ADASYN membuat **sampel sintetis** yang berada **di antara sampel kelas minoritas** yang dipilih berdasarkan **bobot**
- ADASYN **berfokus** pada daerah yang **sulit diklasifikasikan**, yaitu daerah di mana sampel dari kelas minoritas dan mayoritas saling berdekatan

Sumber: http://glemaitre.github.io/imbalanced-learn/auto_examples/over-sampling/plot_adasyn.html

DigitalSkola

Contoh Hasil ADASYN



Sumber: https://www.kaggle.com/code/adhang/stroke-logistic-regression-smote-adasyn

DigitalSkola

Data Teks

airline_sentiment		airline		text
negative neutral Other (2363)	63% 21% 16%	United US Airways Other (7905)	26% 20% 54%	14427 unique values
neutral		Virgin America		@VirginAmerica What @dhepburn said.
positive		Virgin America		@VirginAmerica plus you've added commercials to the experience... tacky.
neutral		Virgin America		@VirginAmerica I didn't today... Must mean I need to take another trip!
negative		Virgin America		@VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces &...

- Selain data numerik, *dataset* juga dapat berisikan data teks
- ***Text classification*** berfungsi untuk mengelompokkan data teks ke dalam kelompok tertentu
- **Contoh pemanfaatannya** yaitu untuk *sentiment analysis*, *topic labeling*, *spam detection*, dll

Sumber: https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment

DigitalSkola

Pengolahan Data Teks

Secara umum, tahapan dalam pengolahan teks yaitu:

- ***Tokenization***, pemecahan teks menjadi potongan kecil (kata-kata)

- ***Text preprocessing***
  - Penghapusan kata “tidak penting” (*stop words*)
  - Pengubahan kata ke bentuk dasar (*stemming* dan *lemmatization*)

- ***Feature extractions***, pengubahan data teks menjadi numerik
  - *Bag of words*
  - TF-IDF
  - *Word embedding*

- **Pembuatan model**

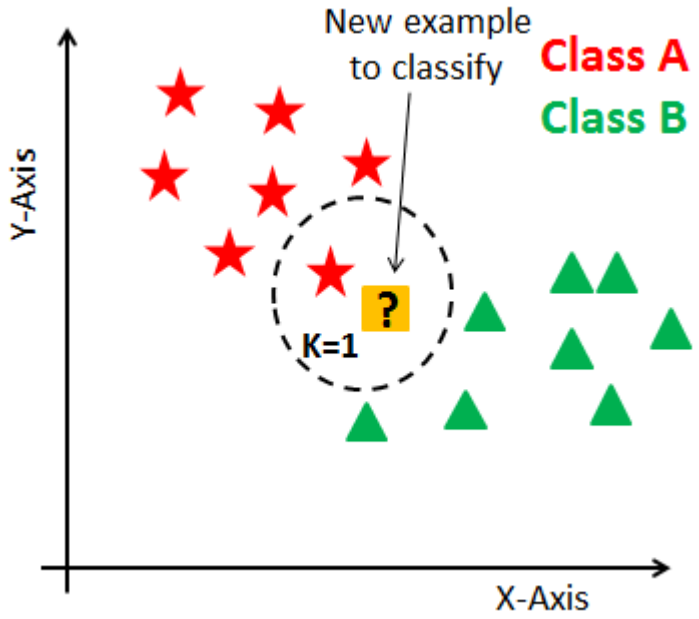
- **Evaluasi model**

02

CLASSIFICATION (PART I)

KNN, decision tree, ensemble method

K-Nearest Neighbors (KNN)

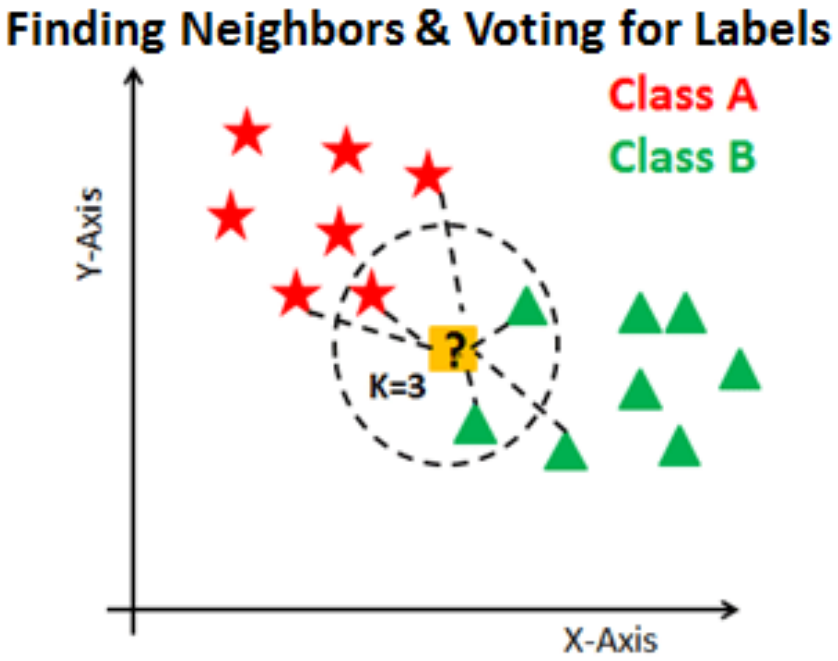


- Digunakan untuk klasifikasi suatu sampel berdasarkan **sejumlah (k) tetangga terdekatnya**
- Jika k=1, maka kelas dari sampel baru ditentukan oleh kelas dari 1 tetangga terdekatnya

Sumber: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

DigitalSkola

KNN – Penentuan Kelas

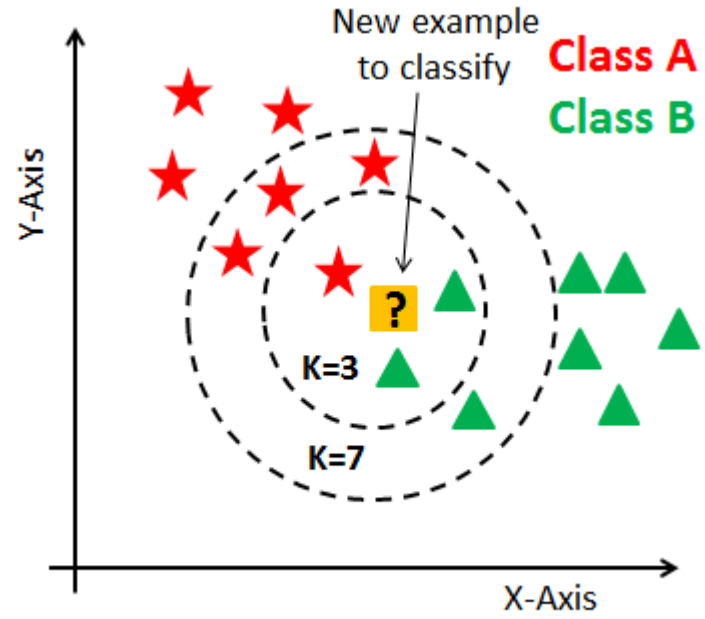


Penentuan kelas dari sampel baru:

- **Kelas mayoritas** dari tetangga terdekat
- **Memberi bobot** pada tiap tetangga berdasarkan jarak

Sumber: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

KNN – Beberapa Tips

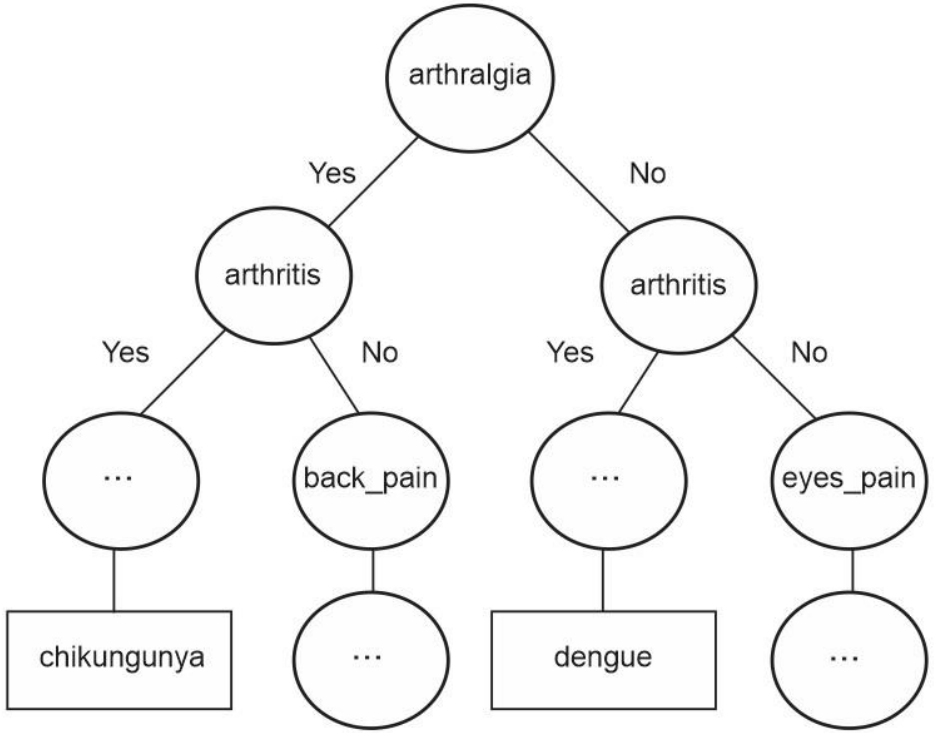


- Gunakan nilai **k** yang **ganjil**
- **Jangan** gunakan nilai **k** yang **terlalu kecil** atau terlalu **besar**
- Lakukan ***feature scaling*** agar rentangnya sama
- Buat **beberapa model** dengan **variasi nilai k** kemudian bandingkan performanya

Sumber: https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

DigitalSkola

Decision Tree



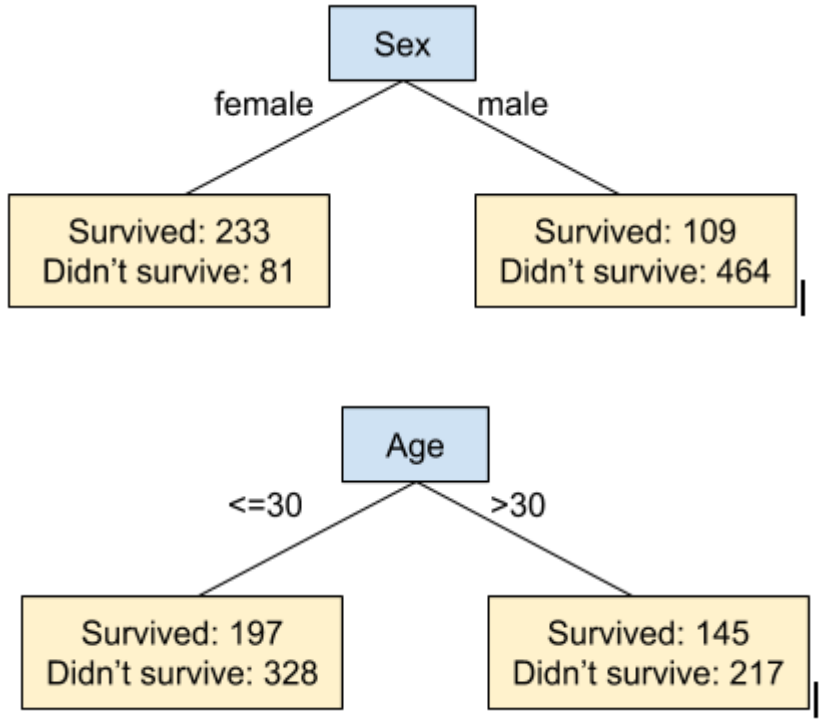
- *Decision tree* merupakan algoritma yang melakukan **partisi secara rekursif** dari *feature space*

- Struktur *decision tree* mirip seperti pohon yang terbalik
  - ***Root***, bagian paling atas yang menjadi **titik awal** proses prediksi
  - ***Leaf***, bagian paling bawah yang menjadi **hasil prediksi**
  - ***Node***, bagian tengah yang menjadi penentu **partisi**

Sumber: https://www.researchgate.net/figure/Example-of-a-Decision-Tree-Graph_fig2_316889388

DigitalSkola

Penentuan Node



- Setiap *node* merupakan atribut yang digunakan untuk **proses partisi**
- Pemilihan atribut berdasarkan **nilai *impurity***, yaitu kemampuan untuk mendapatkan kelas yang homogen
- Perhitungan nilai *impurity* tersebut dapat menggunakan ***gini index*** atau ***entropy***

Sumber: https://www.sololearn.com/learning/1094

DigitalSkola

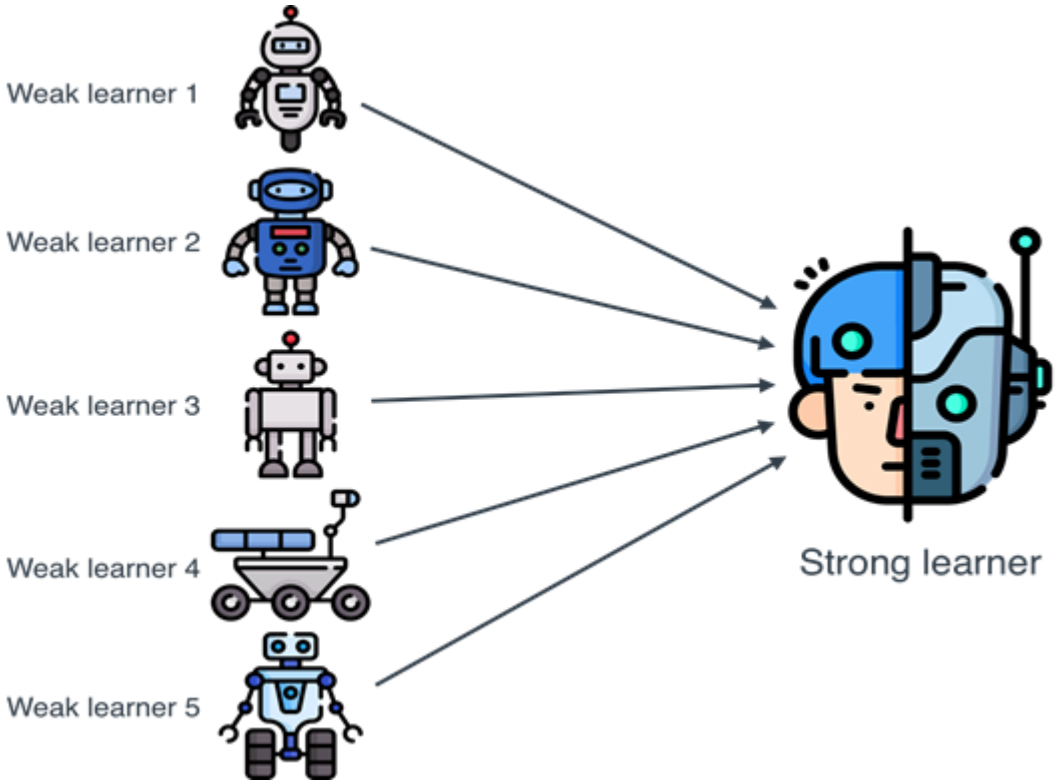
Pros – Cons dari Decision Tree

- **Kelebihan**
  - Mudah dipahami
  - Algoritma ringan dan cepat

- **Kekurangan**
  - Sensitif terhadap *noise*
  - Sangat rentan terhadap *overfitting*

- ***Pruning*** dilakukan untuk memangkas struktur *tree* agar model dapat menghasilkan prediksi yang lebih *general* (tidak *overfitting*)
  - **Pre-pruning** : mengatur *rules* yang menentukan **kapan** *tree* **berhenti** tumbuh
  - **Post-pruning** : membuat *tree* secara utuh, kemudian menentukan **bagian** mana yang **dihapus**

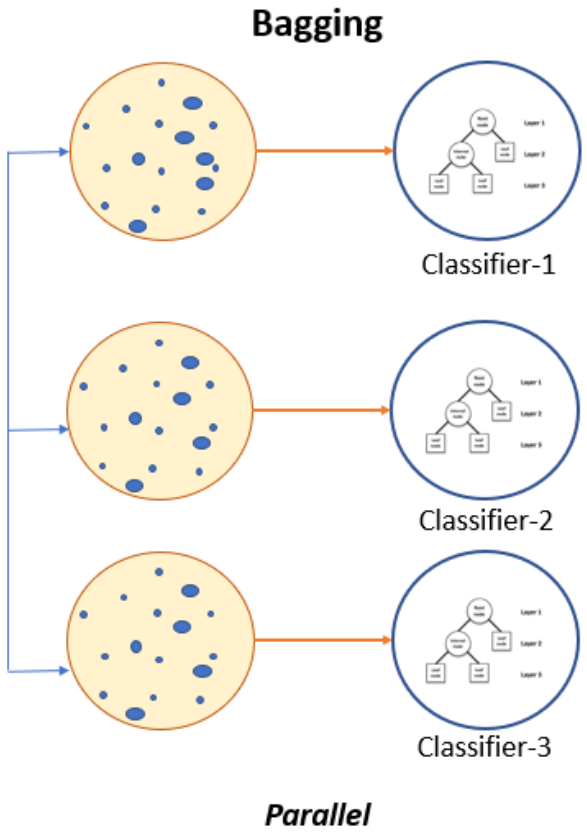
Ensemble Method



- *Ensemble method* **menggunakan beberapa model** untuk membuat prediksi
- Tujuannya yaitu untuk **mendapatkan akurasi** yang lebih baik dibandingkan hanya menggunakan 1 model

Sumber: https://livebook.manning.com/concept/machine-learning/ensemble-method

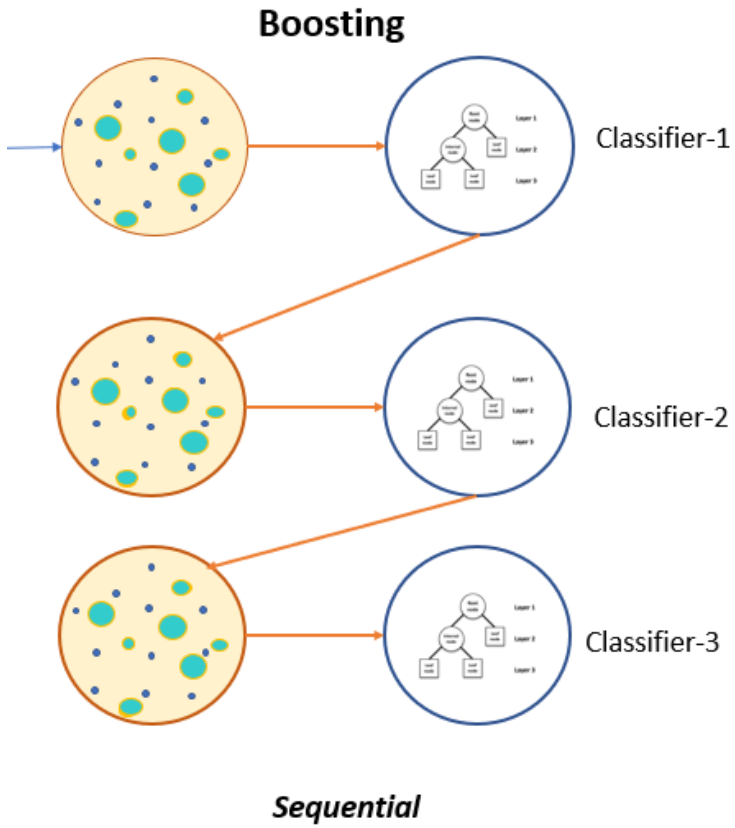
Bagging



- Bagging (***bootstrap aggregating***) akan melakukan sampling dengan metode *bootstrapping*
- **Setiap sampel** akan digunakan untuk **model yang berbeda**
- Hasil prediksi dari setiap model akan dikumpulkan (*aggregating*) kemudian:
  - Dipilih suara **mayoritas** (klasifikasi)
  - Dipilih nilai **rata-rata** (regresi)

Sumber: https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

Boosting



- Membuat prediksi dengan menggunakan **sampel** yang telah **diberi bobot**
- Sampel yang **sulit** untuk **diprediksi** akan memiliki **bobot** yang **besar**, sehingga kemungkinan muncul di iterasi berikutnya juga besar
- Tujuannya yaitu agar model dapat **lebih mempelajari sampel** tersebut

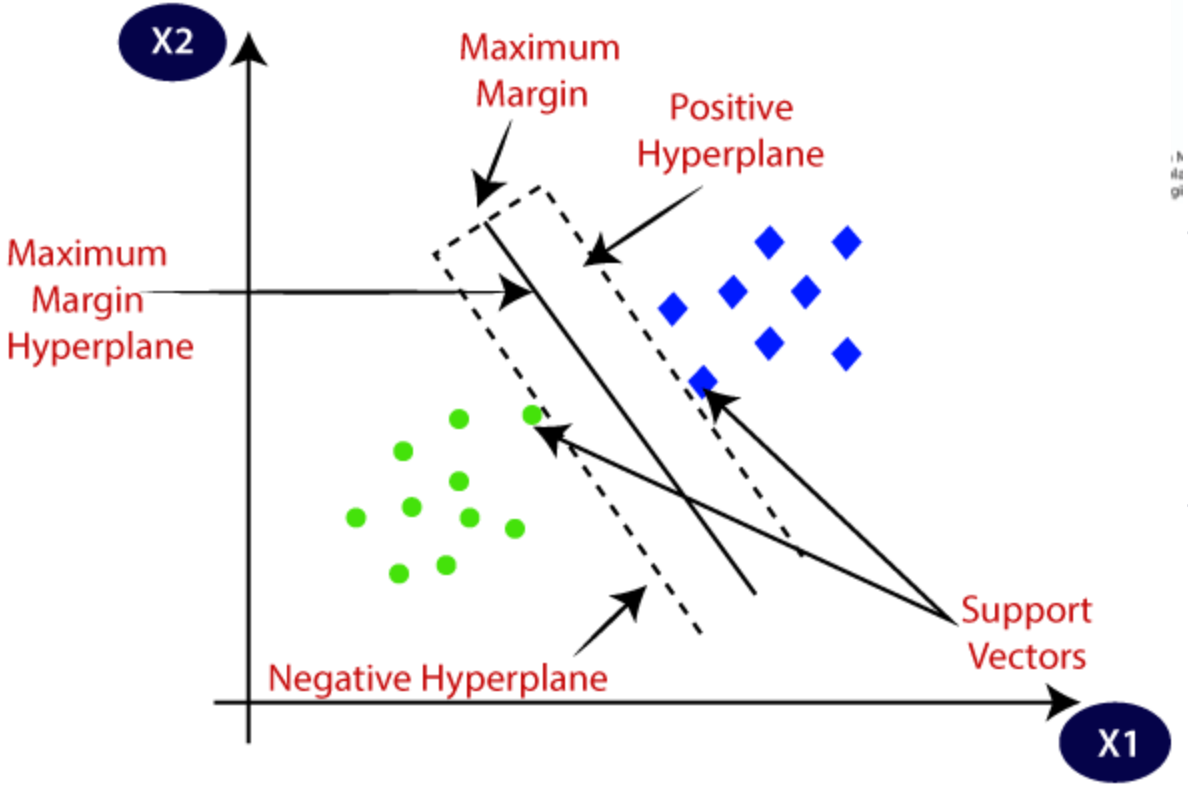
Sumber: https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

03

CLASSIFICATION (PART II)

SVM for binary & multiclass classification

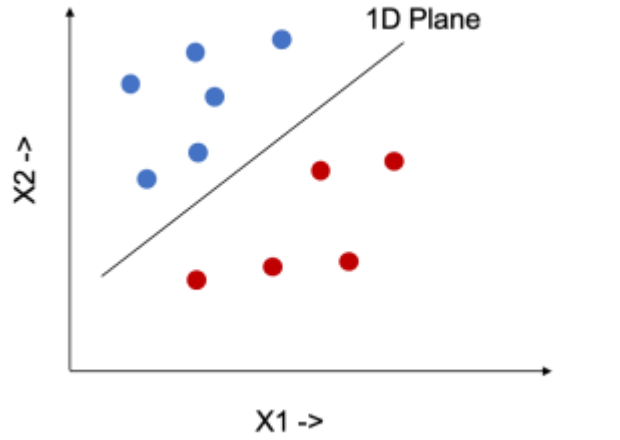
Support Vector Machine

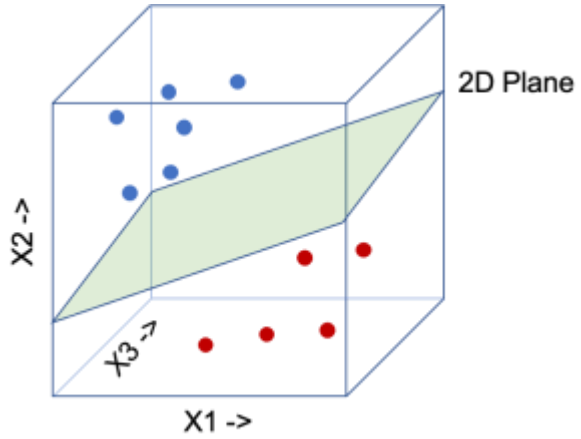


- *Support vector machine* (SVM) menggunakan ***hyperplane*** untuk **memisahkan 2 kelas**
- SVM digunakan untuk menentukan *hyperplane* dengan ***margin*** antara *support vector* yang **paling besar**

Sumber: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

Hyperplane



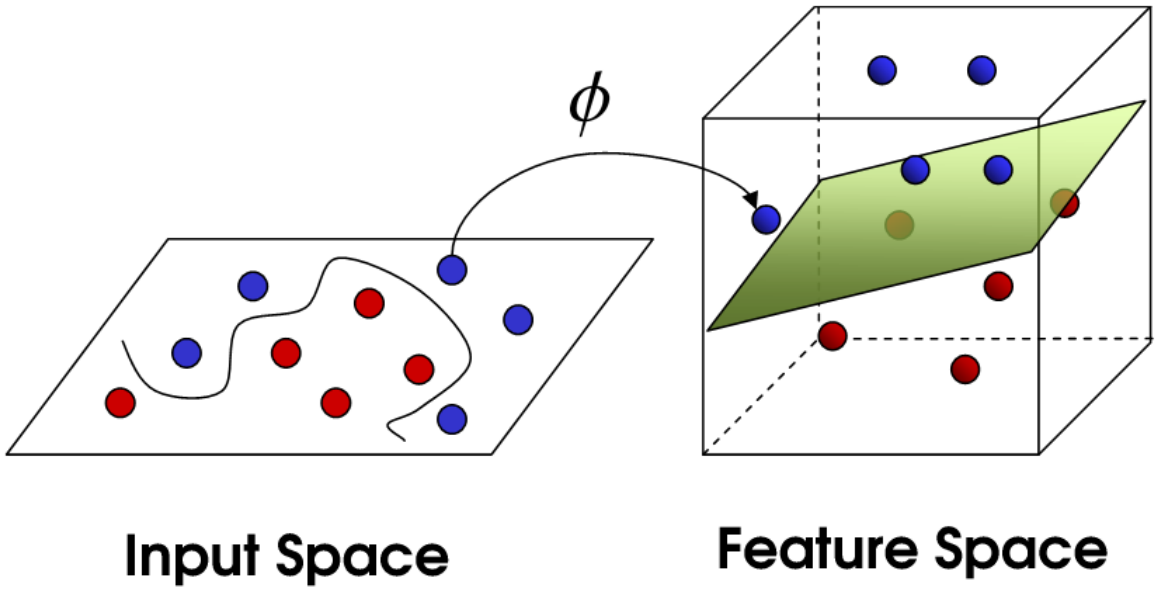


- *Hyperplane* yaitu **“bidang”** yang digunakan untuk memisahkan 2 kelas
- **Dimensi bidang** tersebut tergantung dari **dimensi data** (jumlah fitur yang digunakan)
- Misal, **sebuah garis** (bidang 1 dimensi) dapat digunakan untuk memisahkan **data 2 dimensi** (memiliki 2 fitur)

Sumber: http://www.adeveloperdiary.com/data-science/machine-learning/support-vector-machines-for-beginners-linear-svm/

DigitalSkola

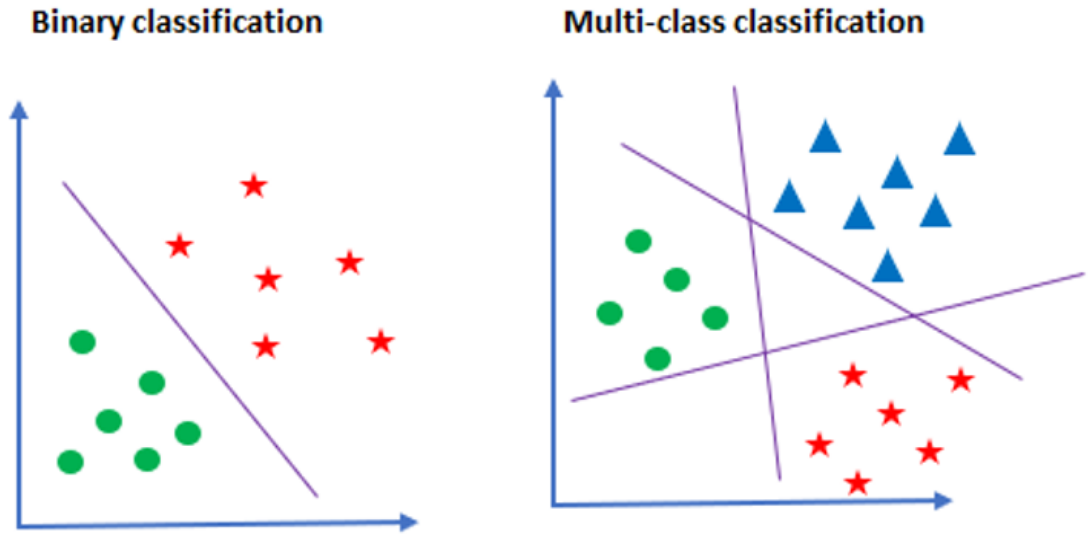
Kernel Trick



- *Kernel trick* dapat digunakan jika data **sulit dipisahkan secara linier**
- Secara umum, *kernel trick* akan **mengubah dimensi fitur** ke dimensi yang **lebih tinggi**

Sumber: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

Binary ke Multiclass Classification

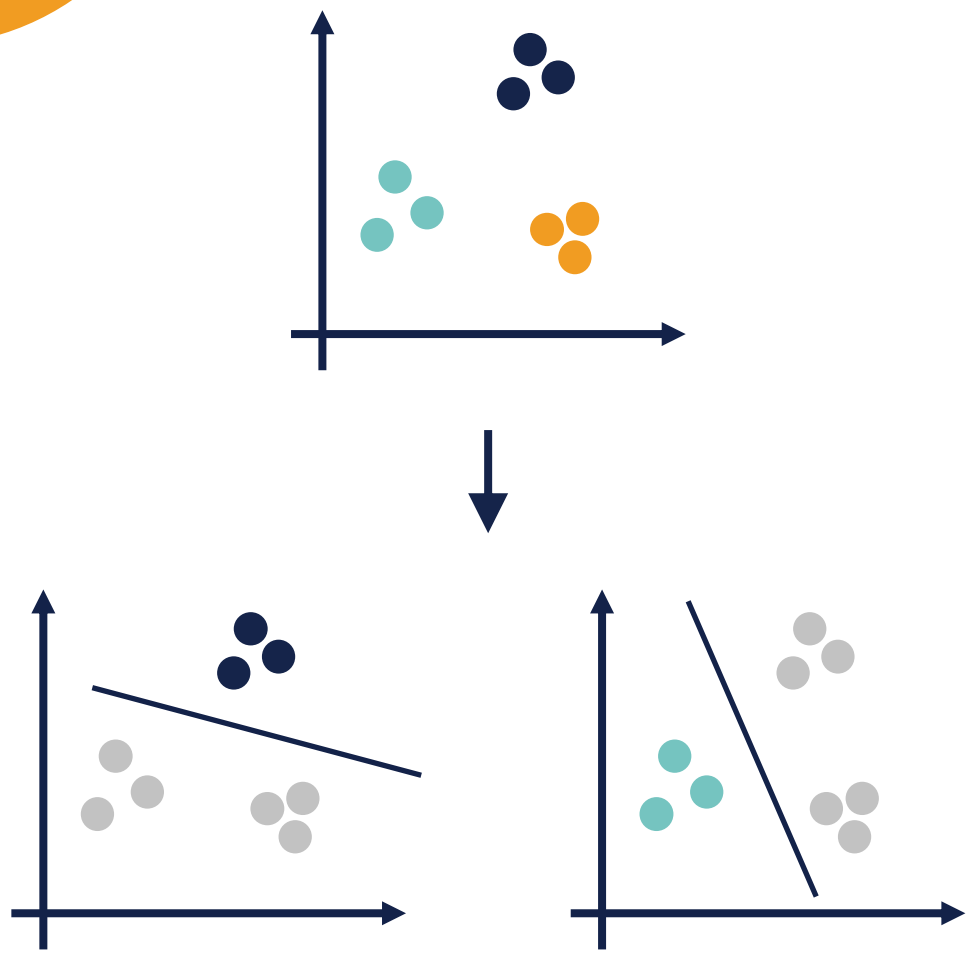


- *Binary classifier* dapat digunakan untuk kasus *multiclass classification*
- Caranya yaitu dengan membuat **beberapa *binary classifier*** dengan metode:
  - One vs Rest (OvR)
  - One vs One (OvO)
- Jika menggunakan SVM, berarti kita akan membuat beberapa bidang (*hyperplane*) pembatas

Sumber: https://michael-fuchs-python.netlify.app/2019/11/13/ovo-and-ovr-classifier/

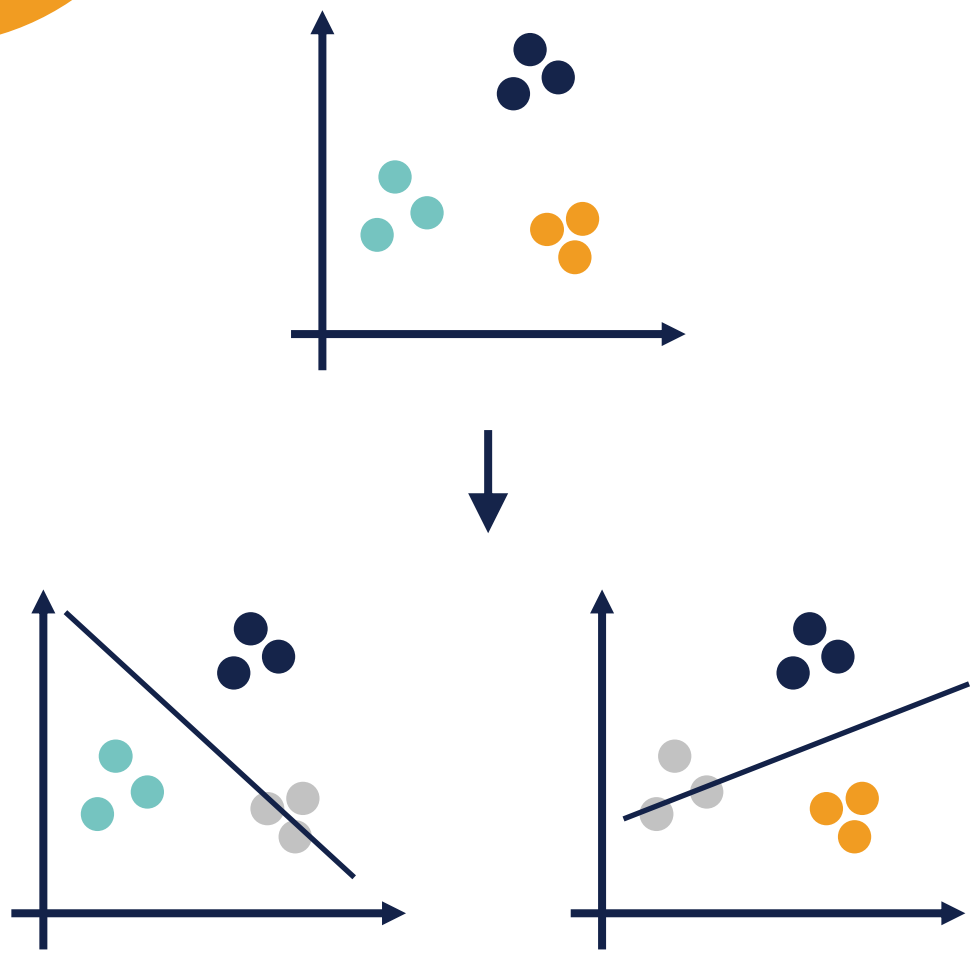
DigitalSkola

One vs Rest (OvR)



- Setiap *classifier* digunakan untuk klasifikasi **1 kelas terhadap seluruh kelas** lainnya
- Setiap *classifier* akan menghasilkan nilai **probabilitas** suatu sampel untuk masuk ke dalam kelas tertentu
- **Hasil prediksi** ditentukan oleh *classifier* yang menghasilkan **probabilitas tertinggi** untuk sampel tersebut

One vs One (OvO)



- Setiap *classifier* digunakan untuk klasifikasi **1 kelas terhadap 1 kelas** lainnya secara bergantian
- Setiap *classifier* akan menghasilkan **prediksi kelas** dari suatu sampel
- **Hasil prediksi akhir** ditentukan oleh voting, yaitu kelas yang **paling banyak muncul** dari hasil prediksi seluruh *classifier*

THANKS

Entropy Team