

Machine Learning

Hendrik Santoso Sugiarto

IBDA2032 – *Artificial Intelligence*

Capaian Pembelajaran

- Konsep ML
- Masalah dalam ML
- End-to-end ML

Machine Learning

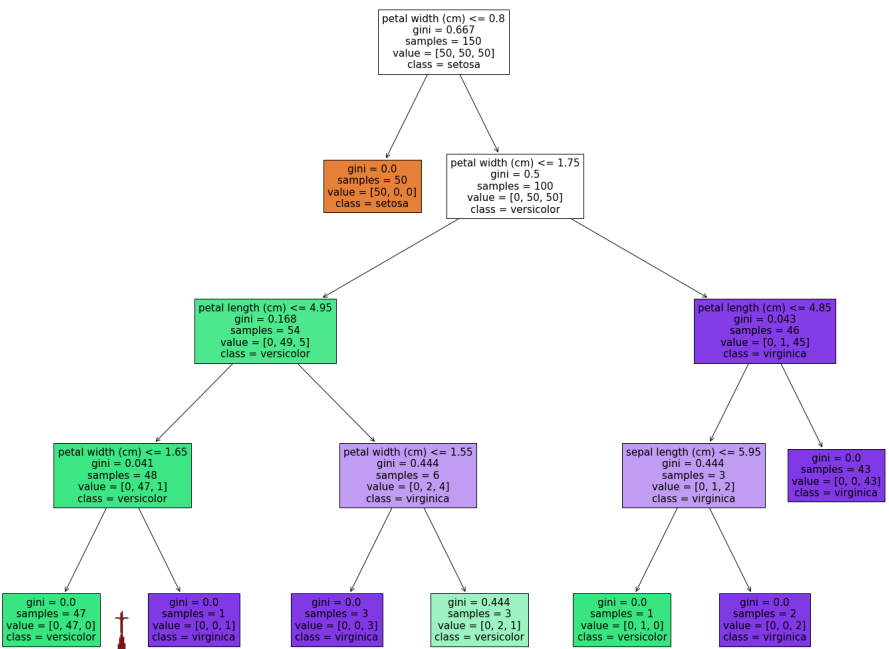
Framework Umum

- Machine Learning = Statistik (pemodelan matematis) + Optimisasi (menemukan parameter optimal)
- Setiap algoritma machine learning mempunyai 3 komponen:
 - Representasi (contoh: decision tree, rules, graphical model, svm, neural net, etc)
 - Evaluasi (contoh: accuracy, precision, recall, F1, likelihood, entropy, MSE, etc)
 - Optimisasi (contoh: greedy search, linear programming, gradient descent, adam, newton, etc)

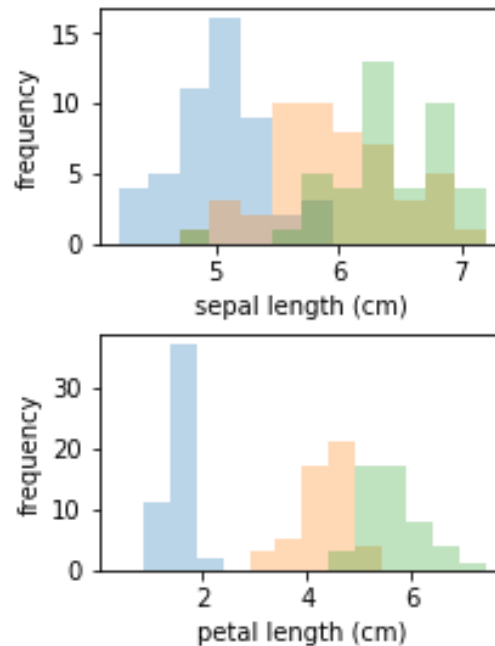
Representasi

- Data yang sama dapat direpresentasikan dengan model yang berbeda
- Tiap model memiliki asumsinya masing-masing

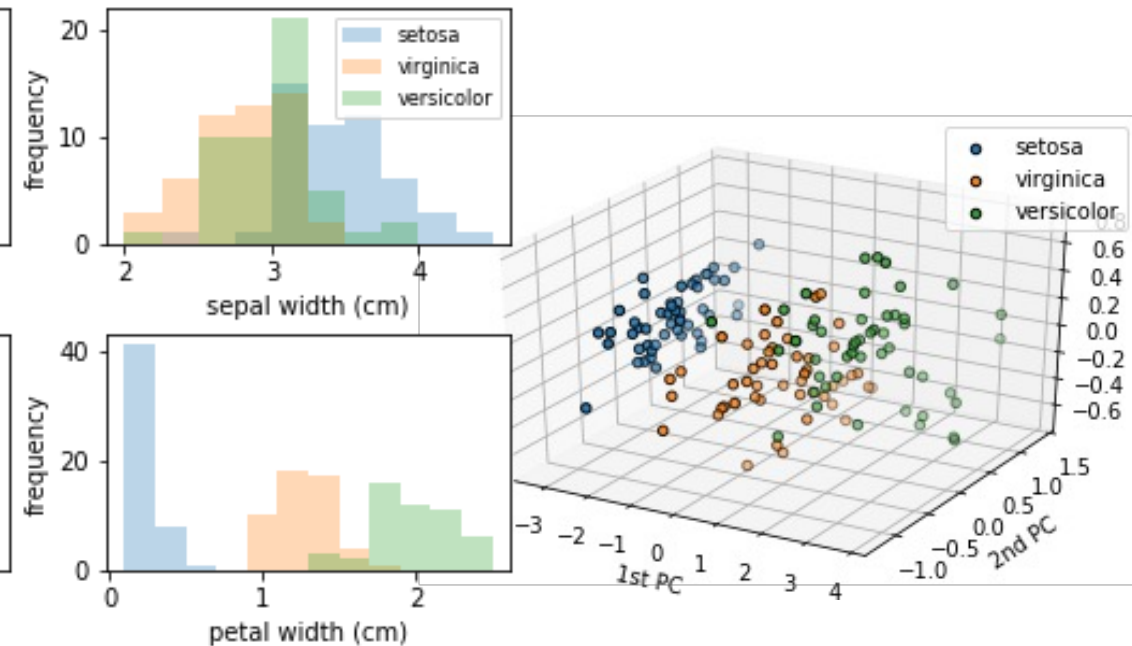
Paling masuk akal



Peluang terbesar

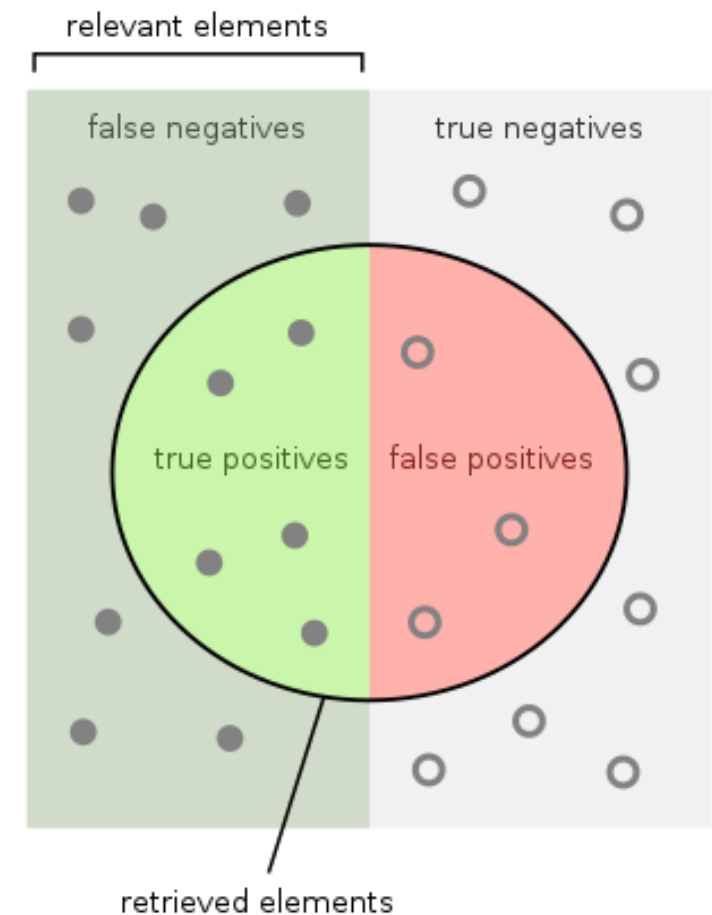


Pilihan terdekat



Evaluasi

- Performa dari model akan diuji dengan metrik yang berbeda tergantung tujuan
- Tiap metrik mengukur hal yang berbeda-beda



How many retrieved items are relevant?

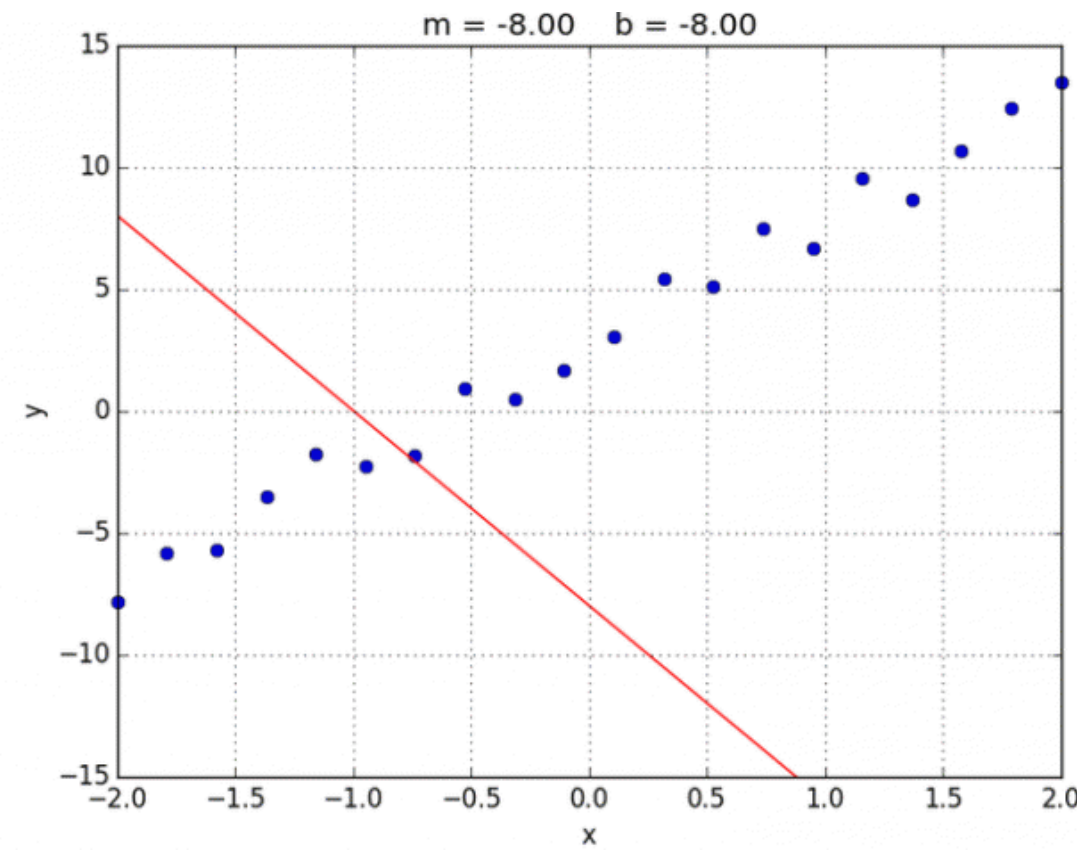
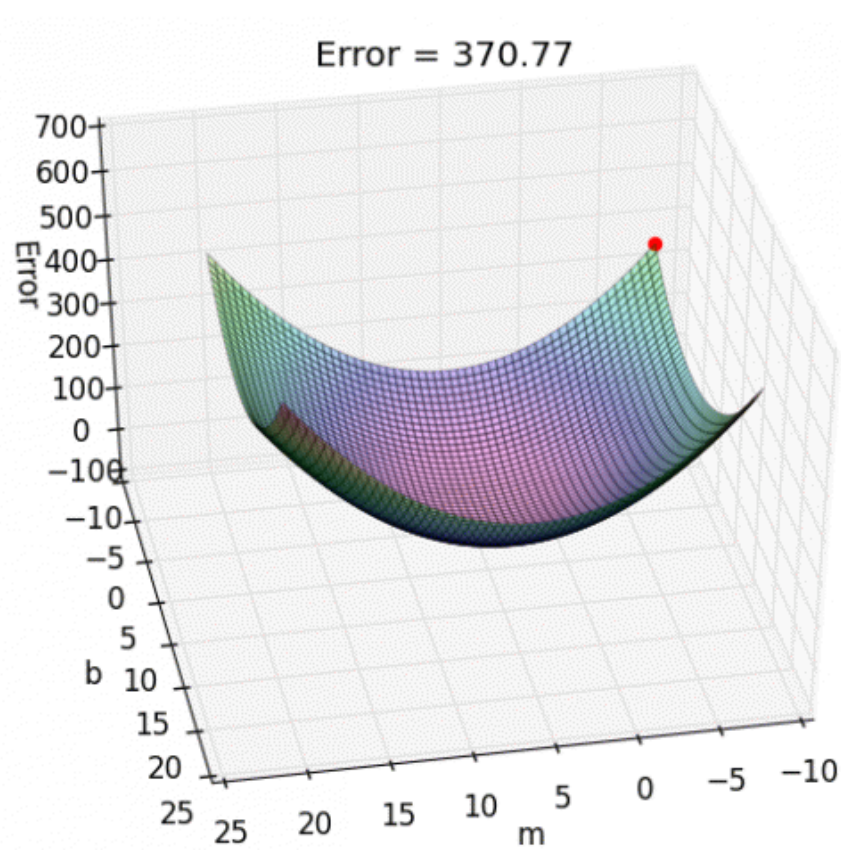
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Optimisasi

- Tiap model memiliki parameter yang diatur untuk menghasilkan performa optimal
- Terdapat berbagai teknik optimisasi parameter (tergantung kesulitan optimisasi)
- Teknik paling umum adalah gradient descent



Parameter vs Hyperparameter

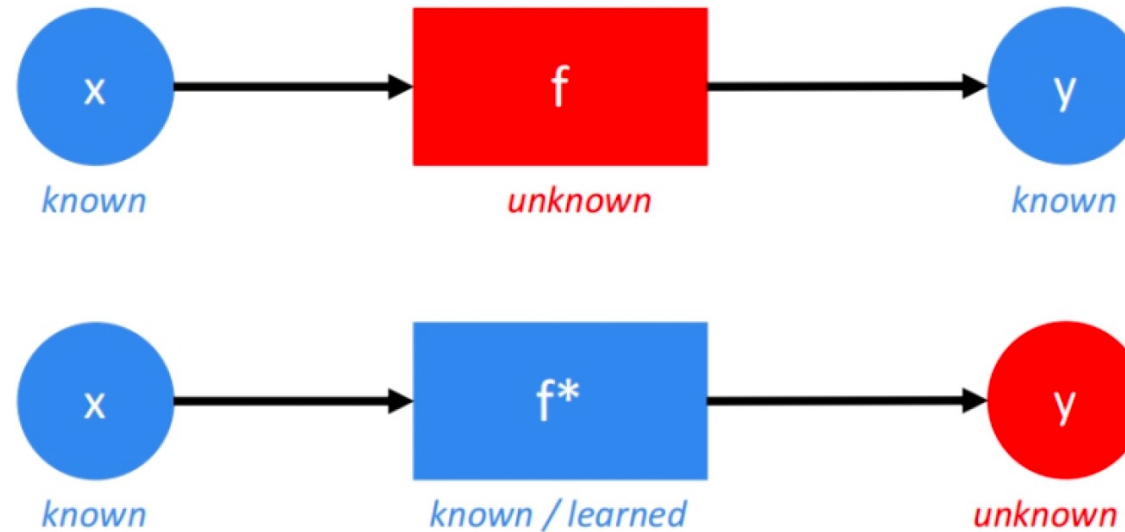
- Parameter:
 - variable yang intrinsic terhadap model
 - nilai variable ini akan difit terhadap data latih
 - proses adjustment nilai parameter terjadi pada optimisasi
- Hyperparameter:
 - variable yang extrinsic terhadap model
 - merubah hyperparameter berarti merubah kerumitan model
 - proses adjustment nilai hyperparameter terjadi pada finetuning

Jenis pembelajaran

- Supervised (inductive) learning: data latih memiliki ekspektasi output
- Unsupervised learning: data latih tidak memiliki ekspektasi output
- Semi-supervised learning: data latih memiliki beberapa ekspektasi output
- Reinforcement learning: output dari setiap aksi berbentuk rewards

Supervised Learning

- Given (input, correct output), predict (input,?)



- Klasifikasi: output diskrit

- Klasifikasi biner: input x , temukan $y \in \{-1, +1\}$
- Klasifikasi multi kelas: input x , temukan $y \in \{1, \dots, k\}$

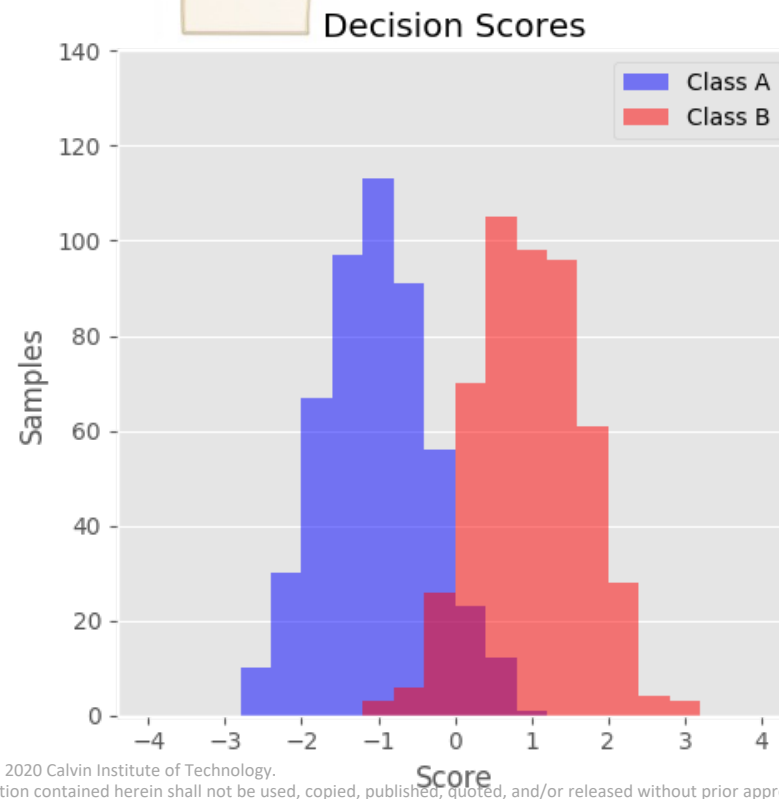
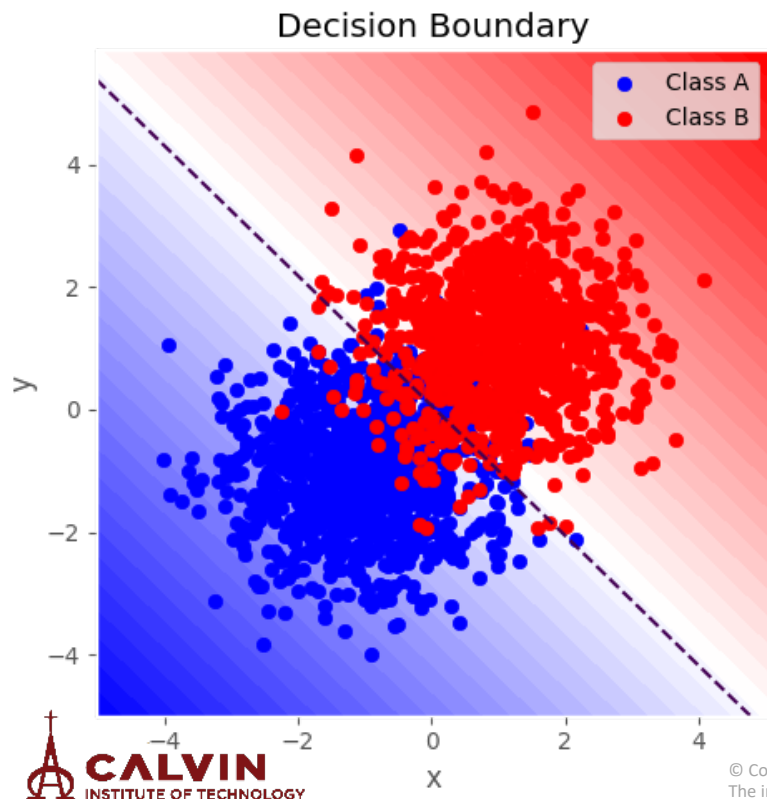
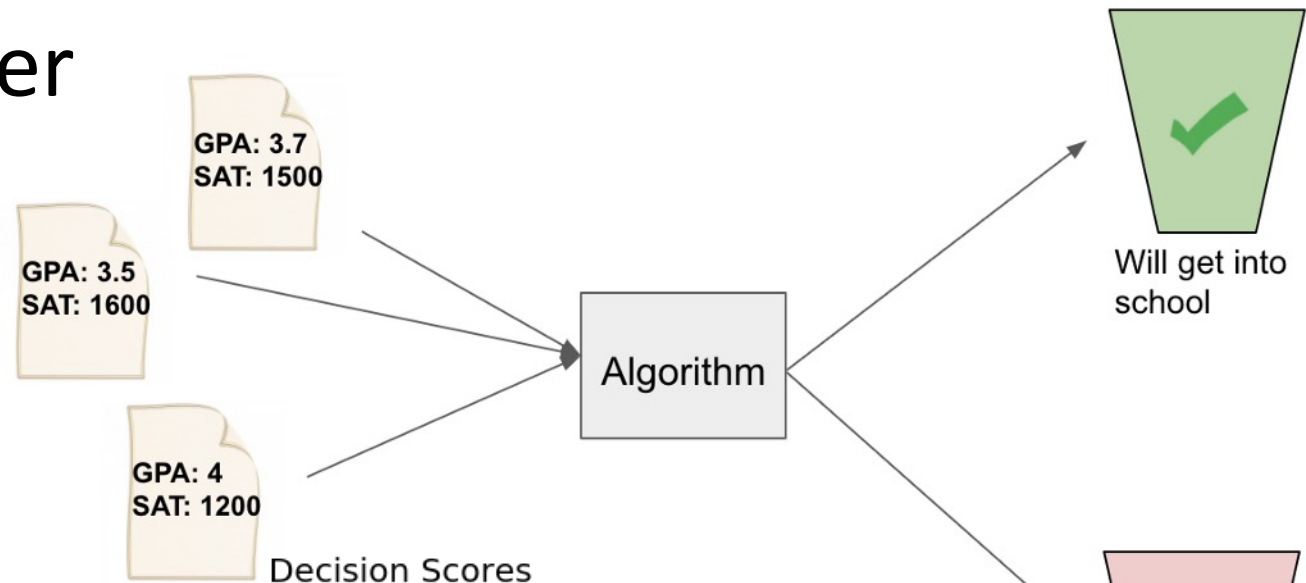
$y \in \{0, 1\}$

- Regresi: output kontinu

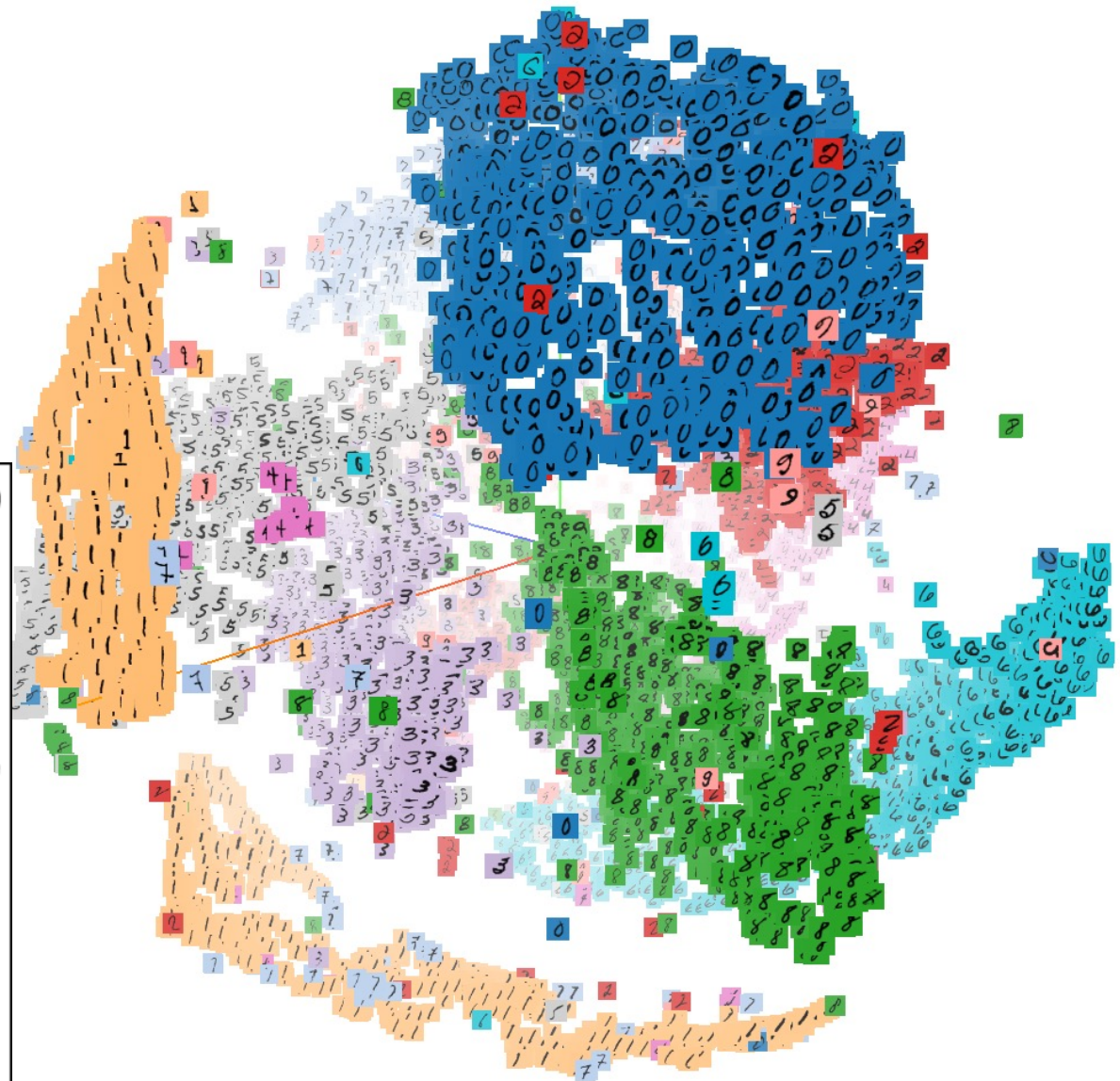
- Jika input x , temukan $y \in \mathbf{R}^d$

$\rightarrow \text{Real}$

Contoh Klasifikasi Biner

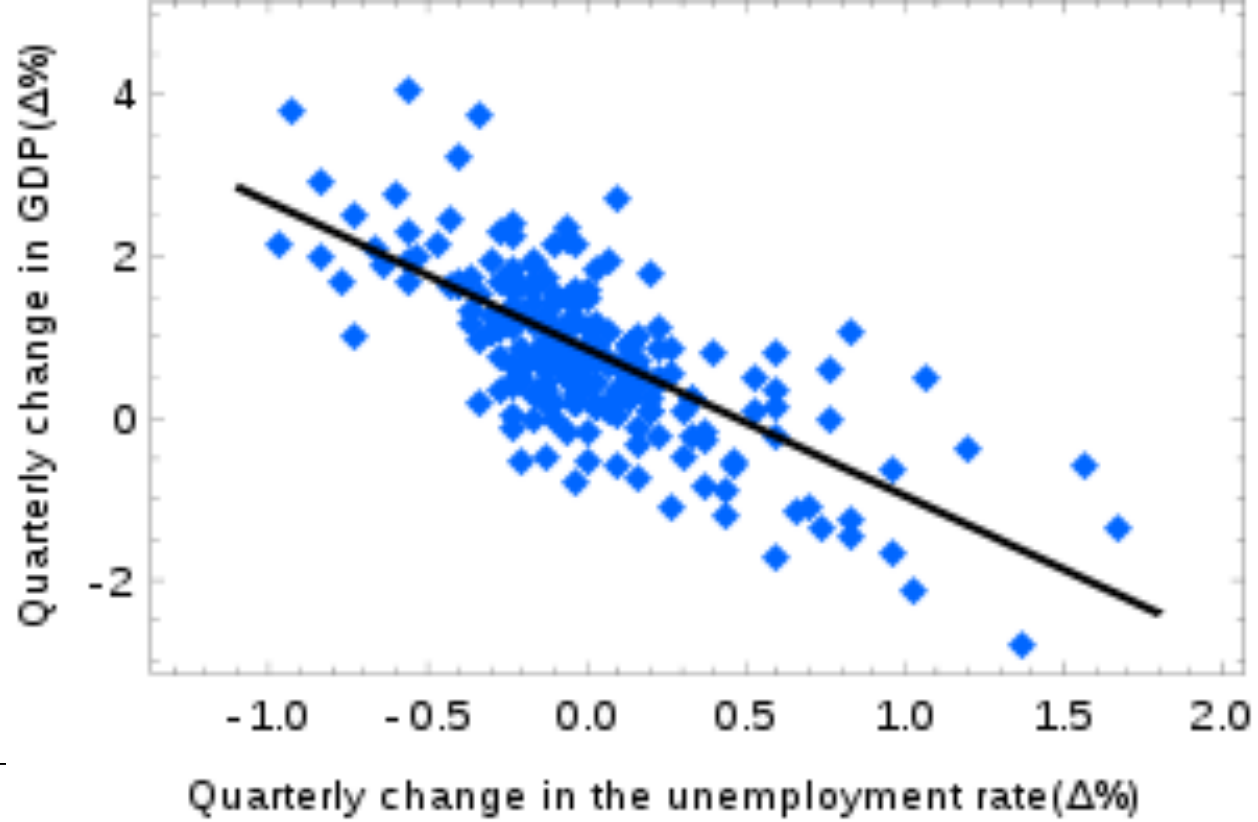


Klasifikasi Multi Kelas

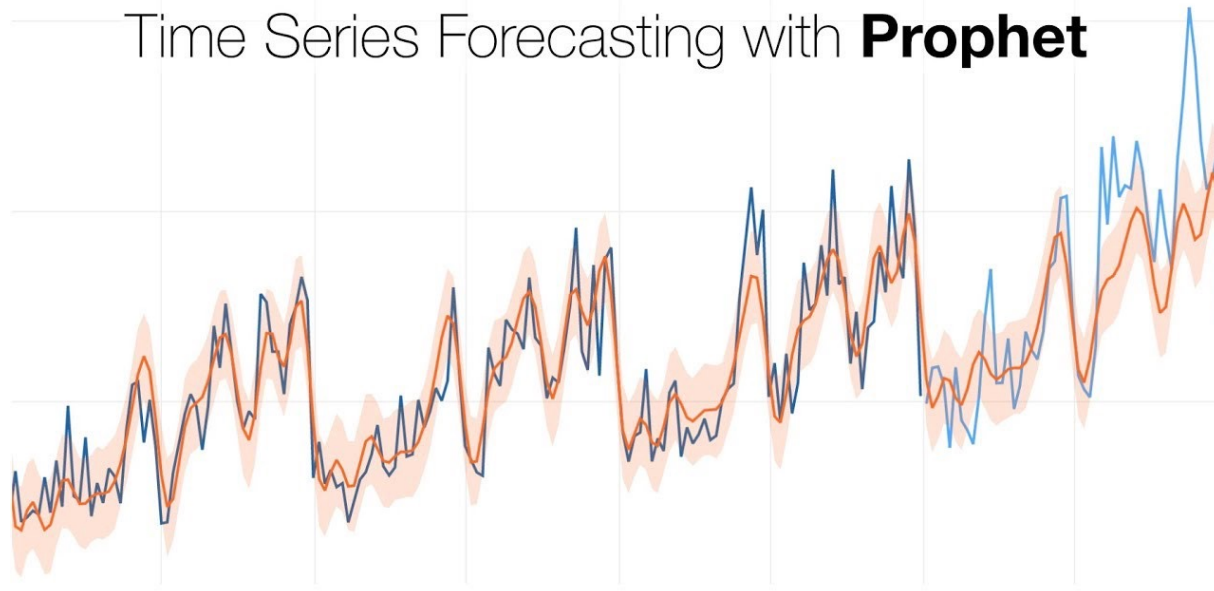


Contoh Regresi

- Linear
- Non Linear

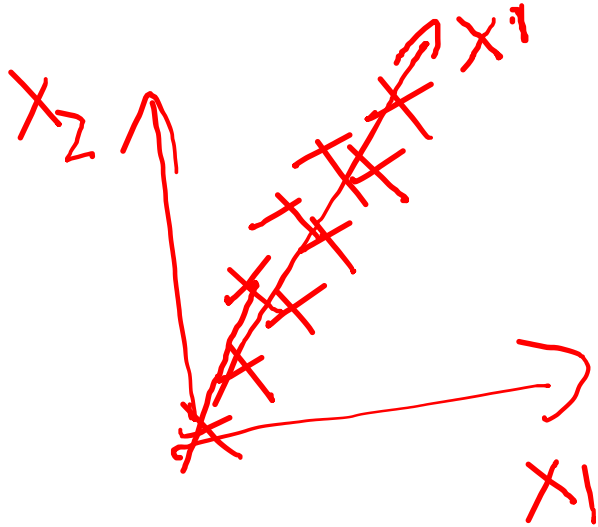


Time Series Forecasting with **Prophet**



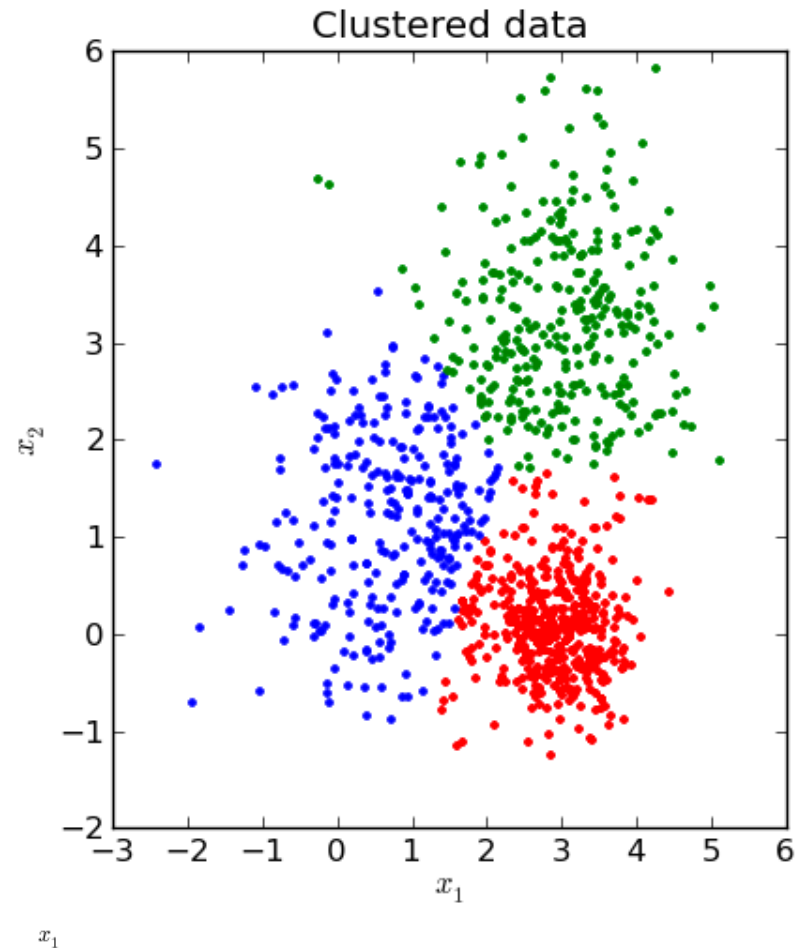
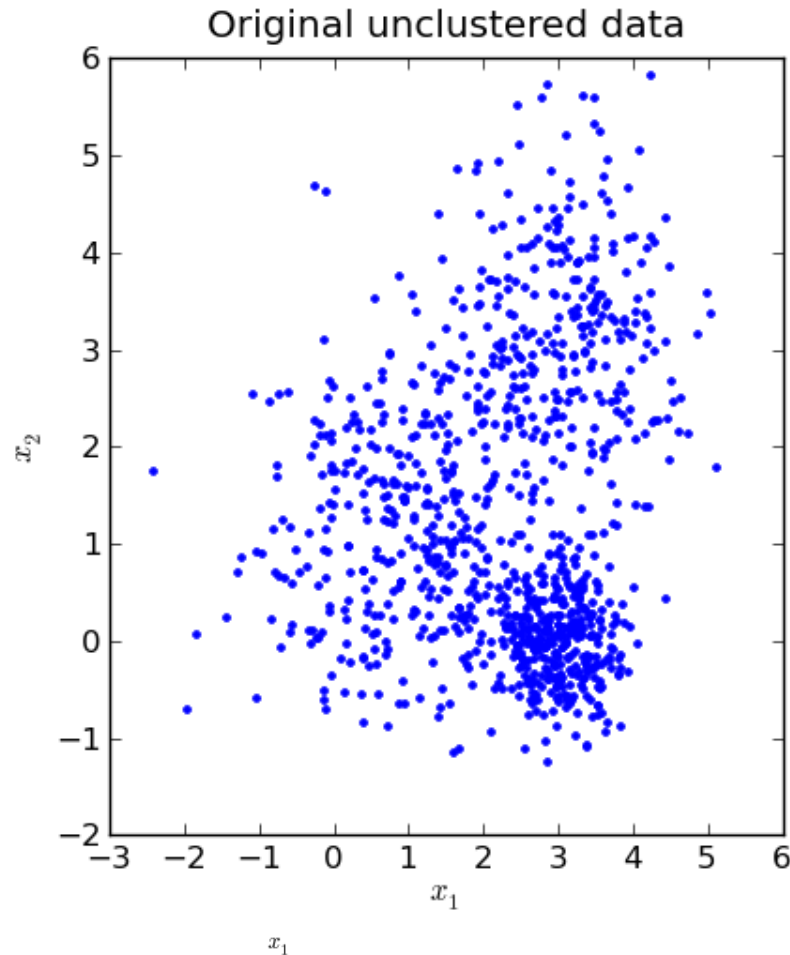
Unsupervised Learning

- Given (input, output), predict (input relationship)
- Clustering: menemukan pengkelompokan data
- Dimension reduction: menemukan subspace untuk merepresentasikan data
- Independent component: menemukan beberapa factor observasi
- Anomaly detection: menemukan data yang aneh



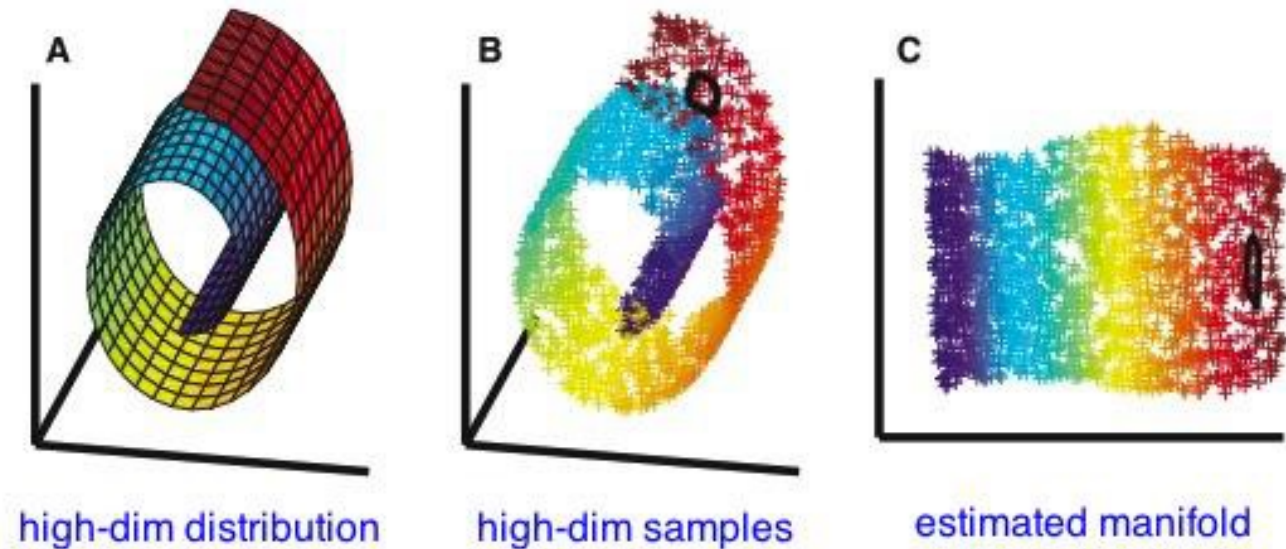
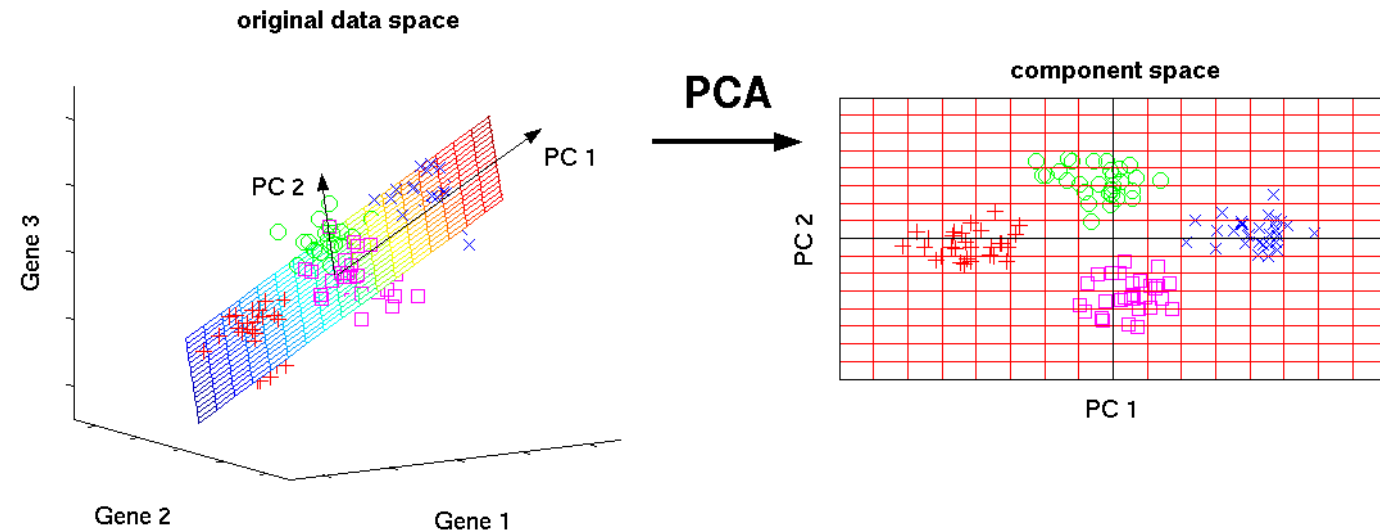
Contoh Clustering

- Segmentasi user, grouping market, dll

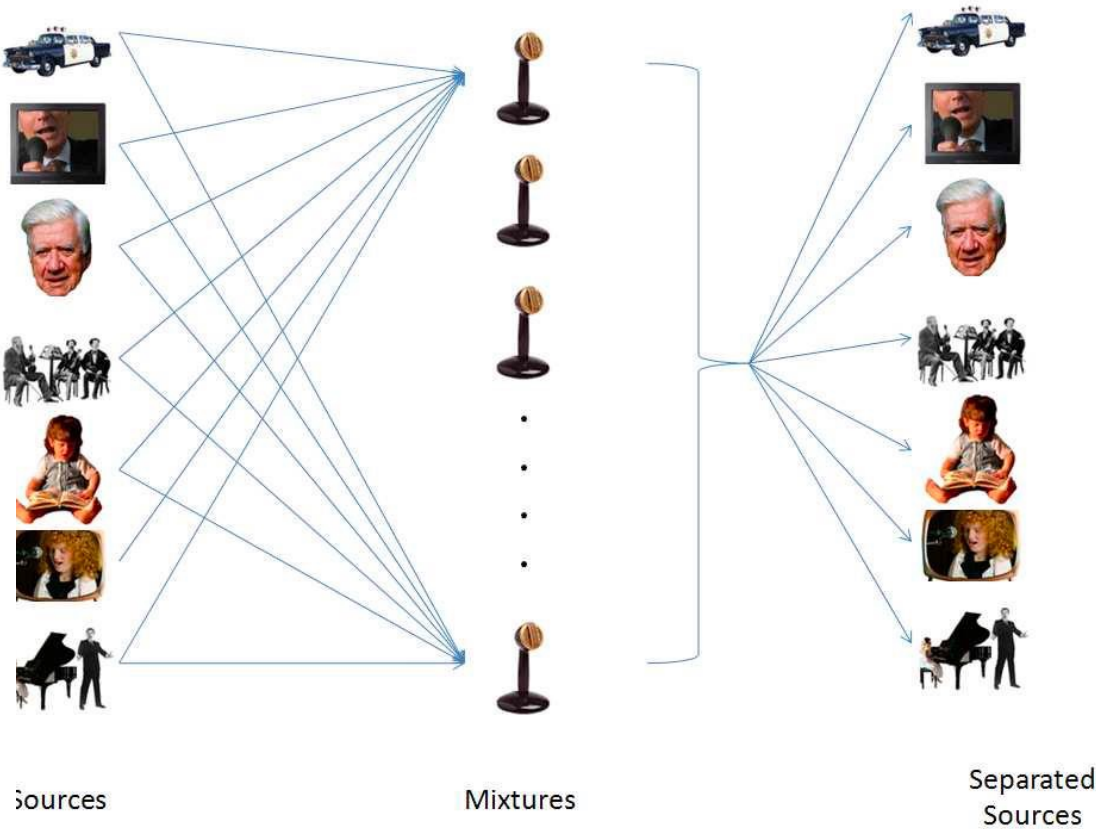
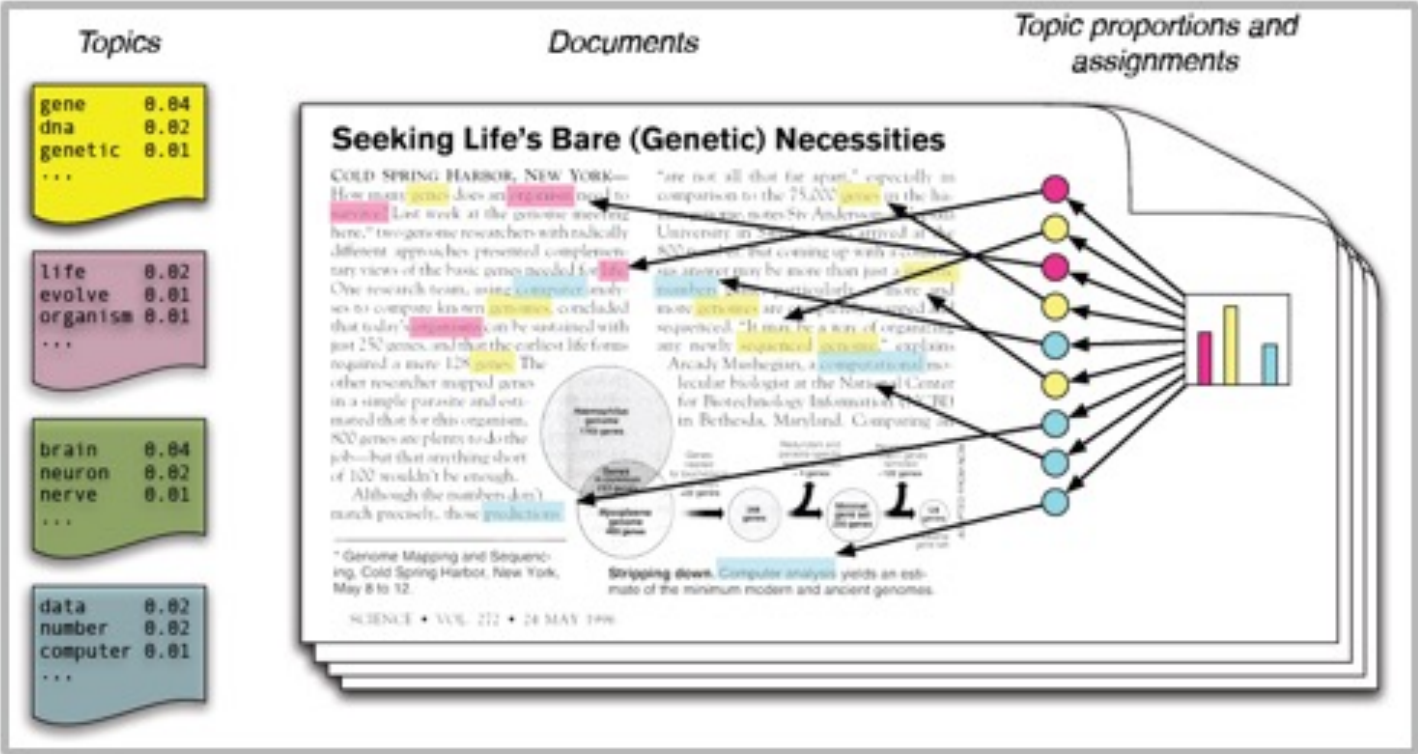


Contoh Reduksi Dimensi

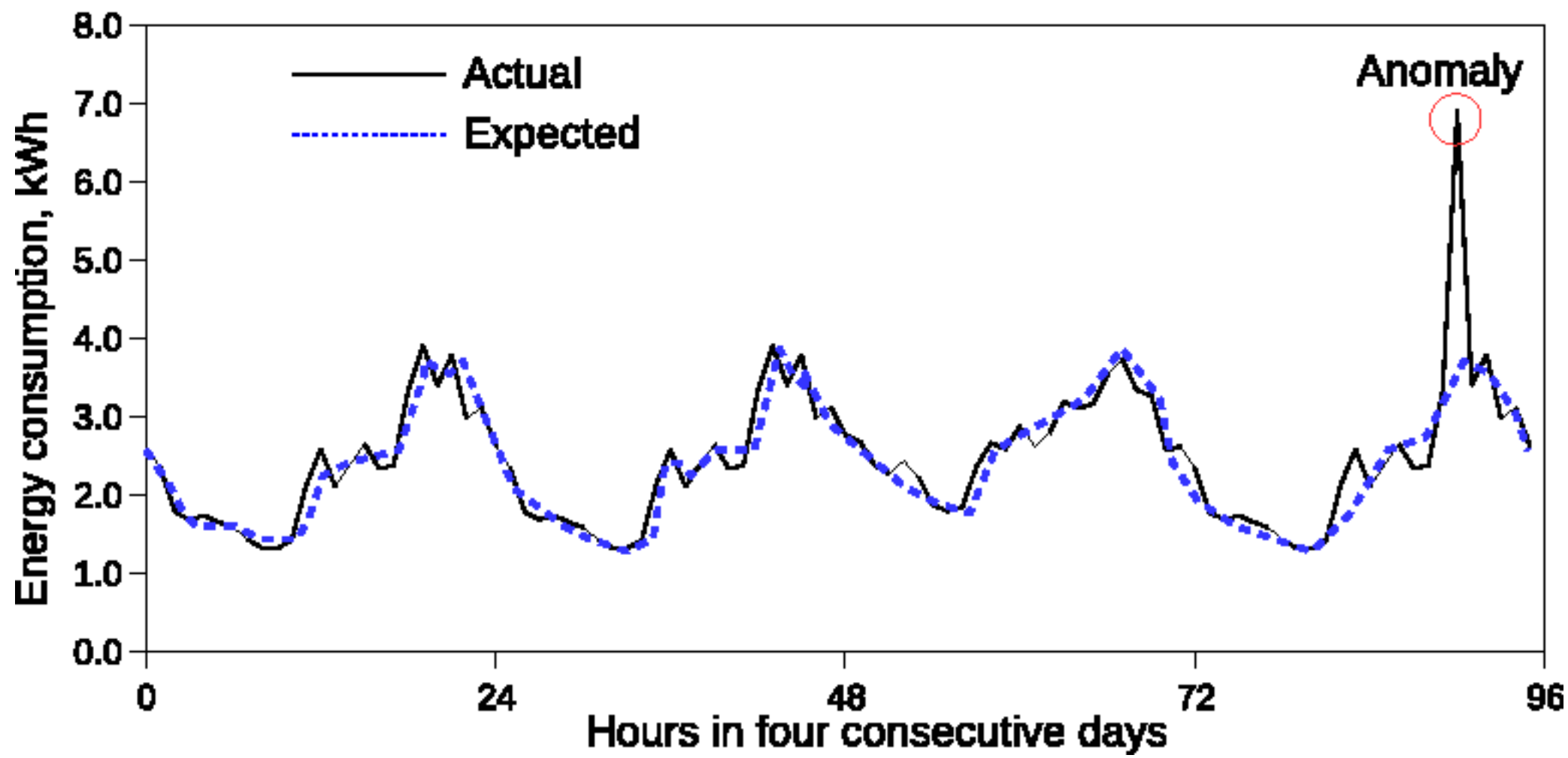
- Linear
- Nonlinear



Contoh Komponen Independen

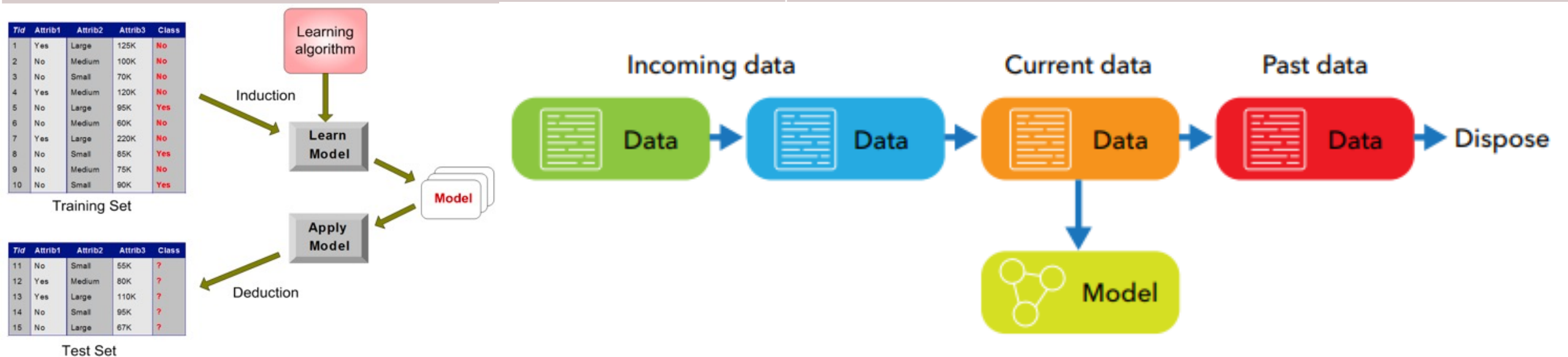


Contoh Deteksi Anomali



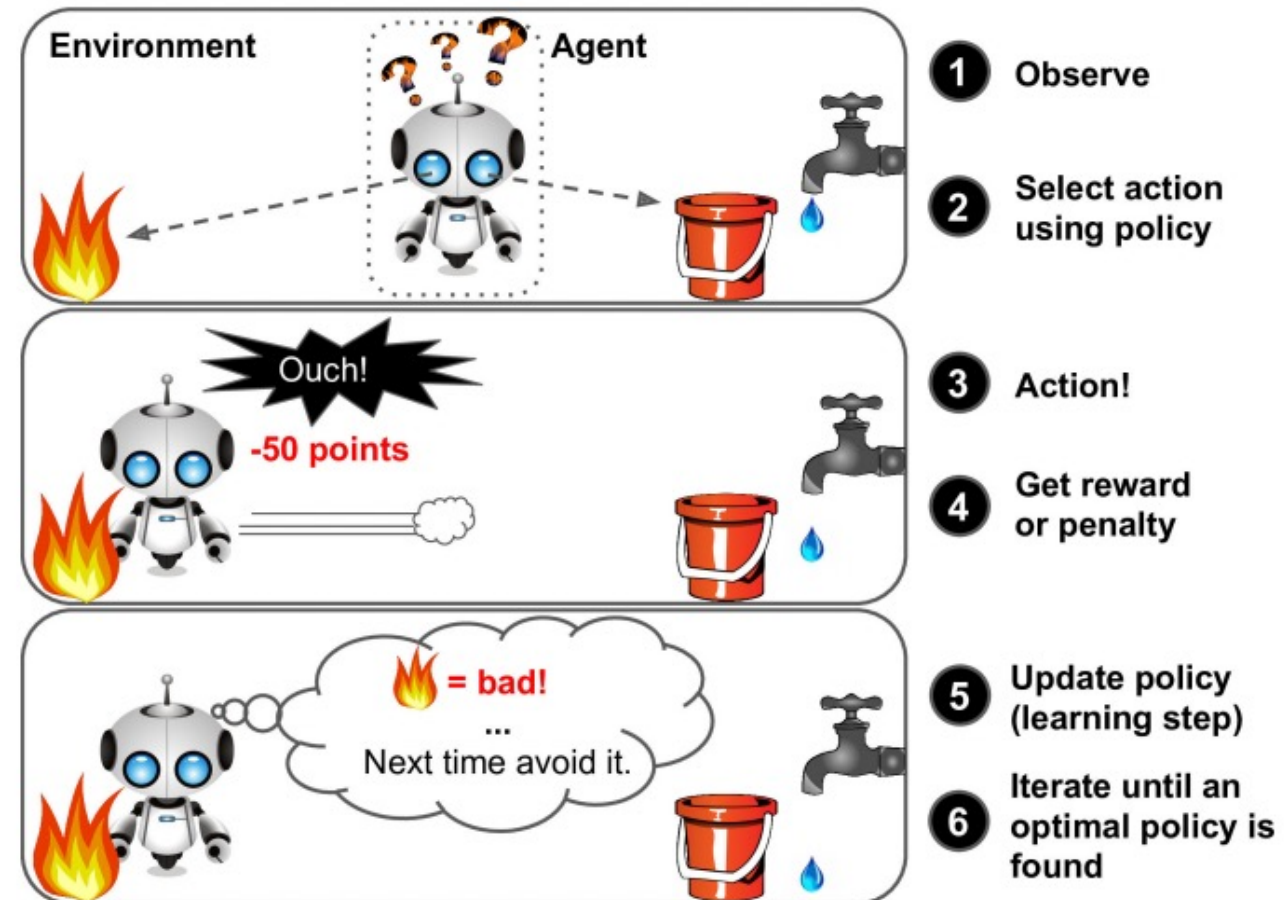
Batch vs online

Batch learning	Online learning
gunakan data latih $(x_1, y_1) \dots (x_l, y_l)$ lalu deploy	pelatihan sekuensial \rightarrow latih x_l , prediksi $f(x_1)$, latih x_l , prediksi $f(x_1) \dots$
- retraining mahal	+ menghindari retraining
- tidak scalable	+ scalable
- efisiensi rendah	+ efisien
+ optimisasi terpisah	+ algoritma simpel
+ konvergen cepat	- Sulit memilih model
+ generalisasi lebih baik	- generalisasi tidak terjamin

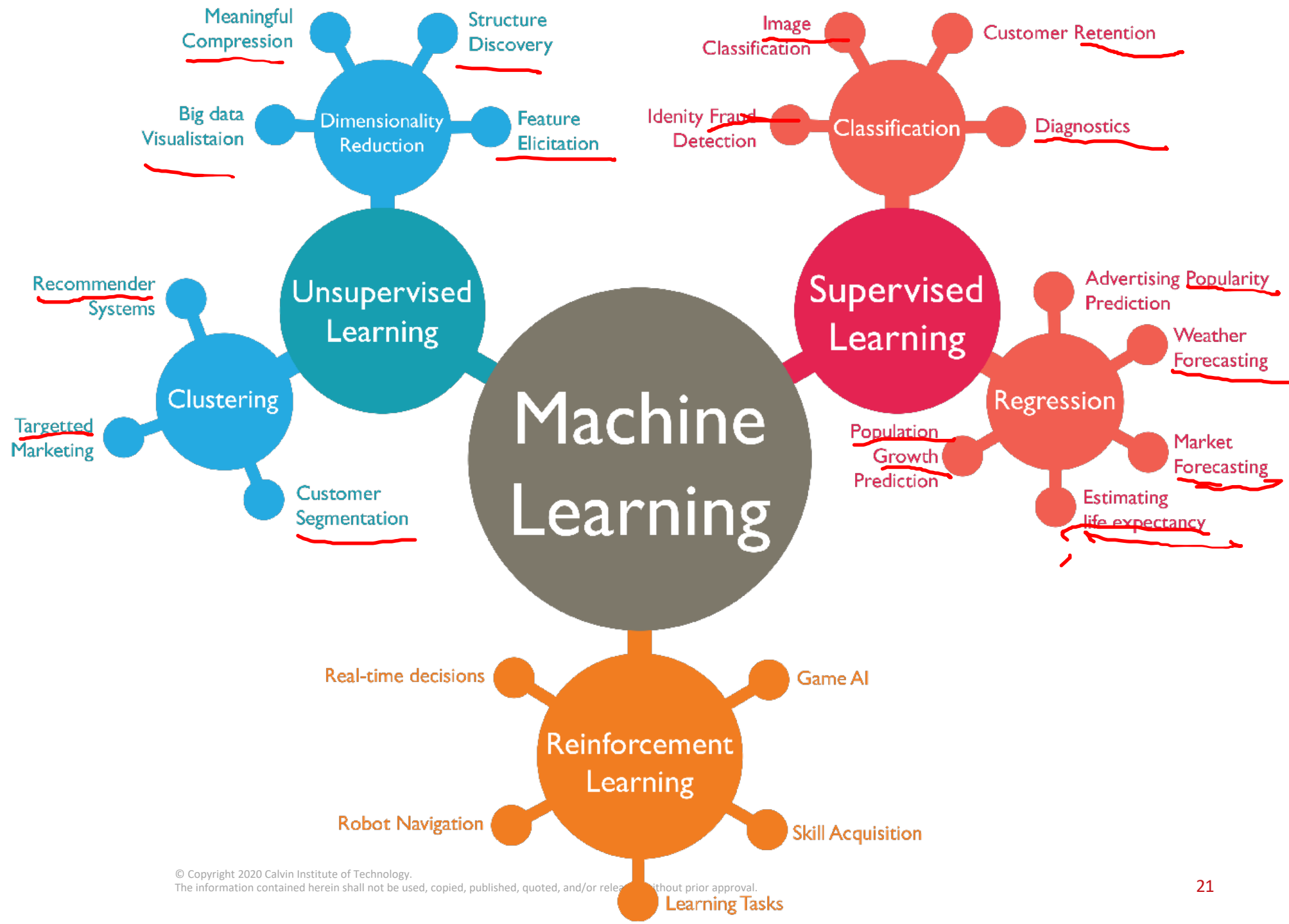


Interaksi dengan Lingkungan

- Active learning
 - Prediksi y untuk x , perbaiki model, pilih x yang baru
 - Contoh: bertanya di kelas
- Reinforcement learning
 - Ambil tindakan, lingkungan berespon, ambil tindakan baru
 - Contoh: menyetir mobil



Peta ML



Masalah dalam ML

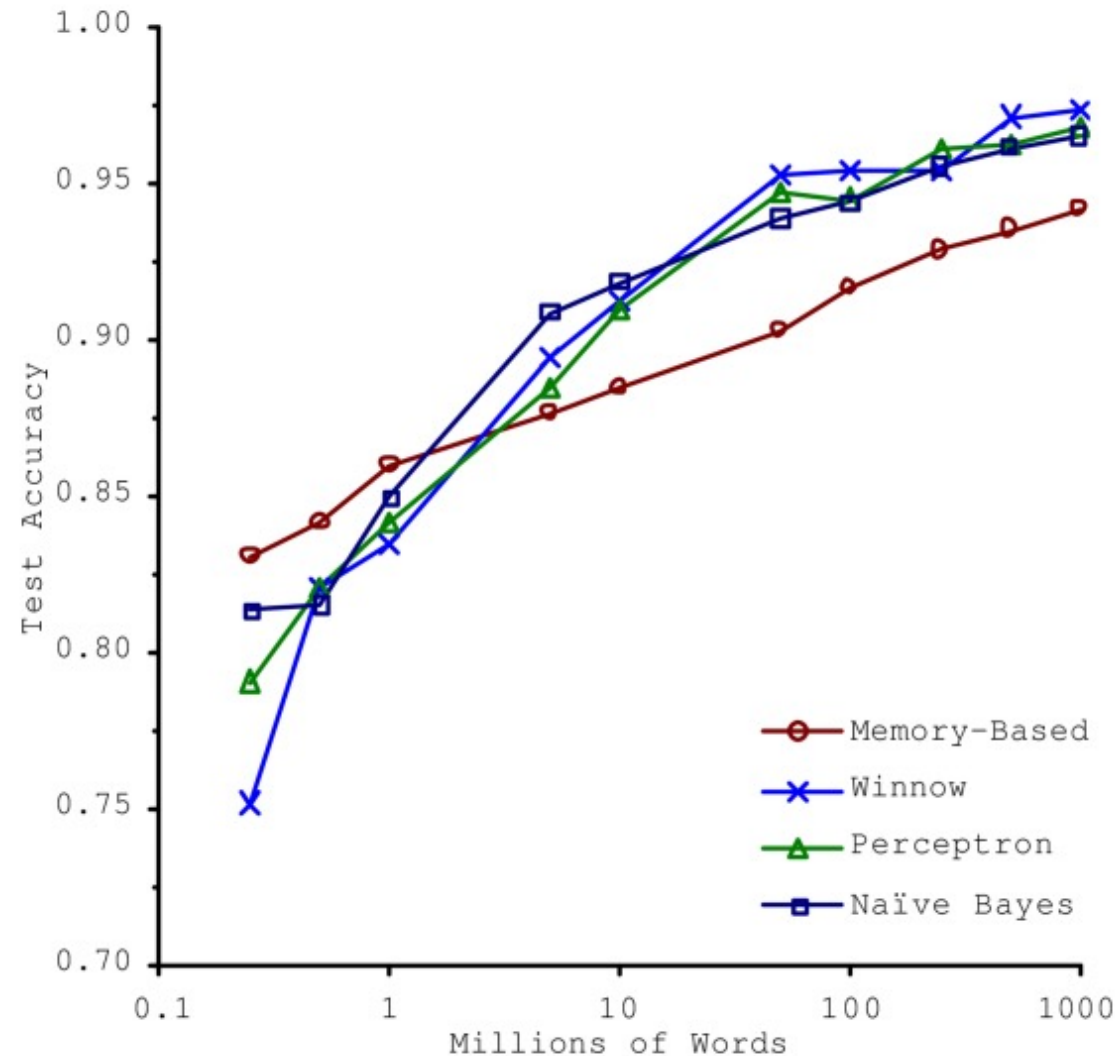
Masalah dalam ML

- Bad Data
 - Data terlalu sedikit
 - Data tidak representatif
 - Kualitas data jelek
 - Fitur yang irelevan
- Bad Algorithm / *model*
 - Overfitting
 - Underfitting

Data terlalu sedikit

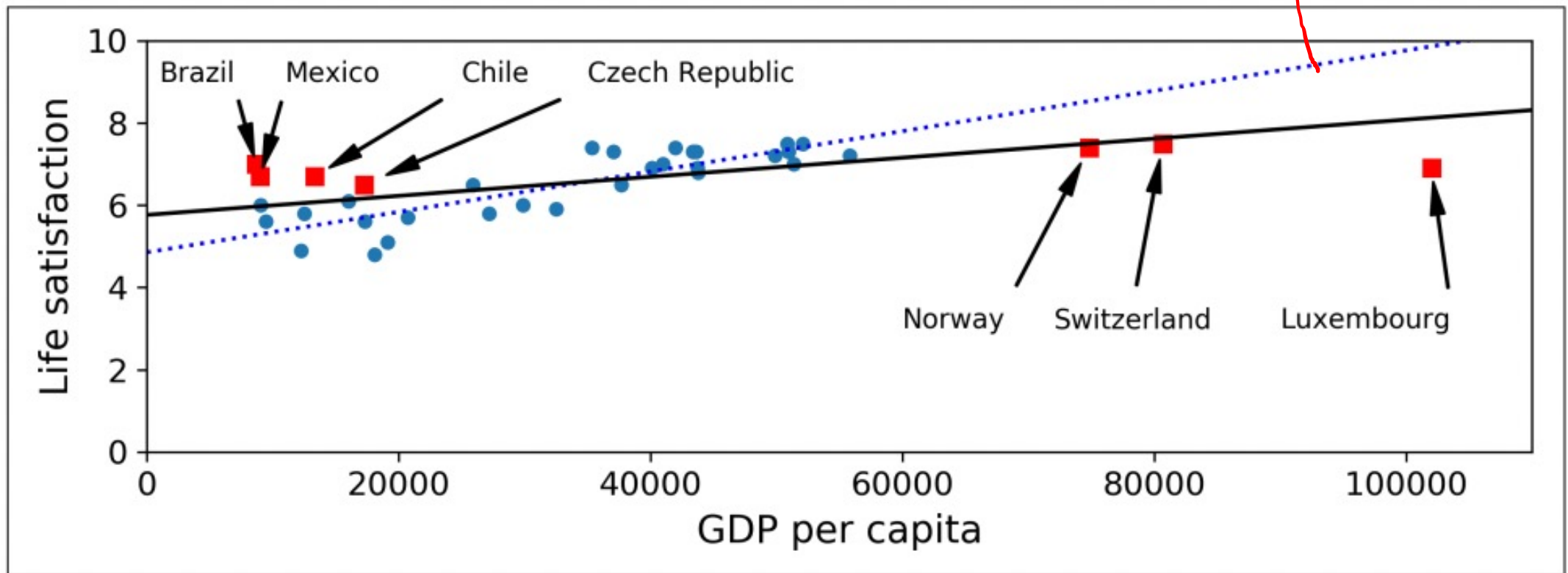
No Free Lunch Theorem

- <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/35179.pdf>



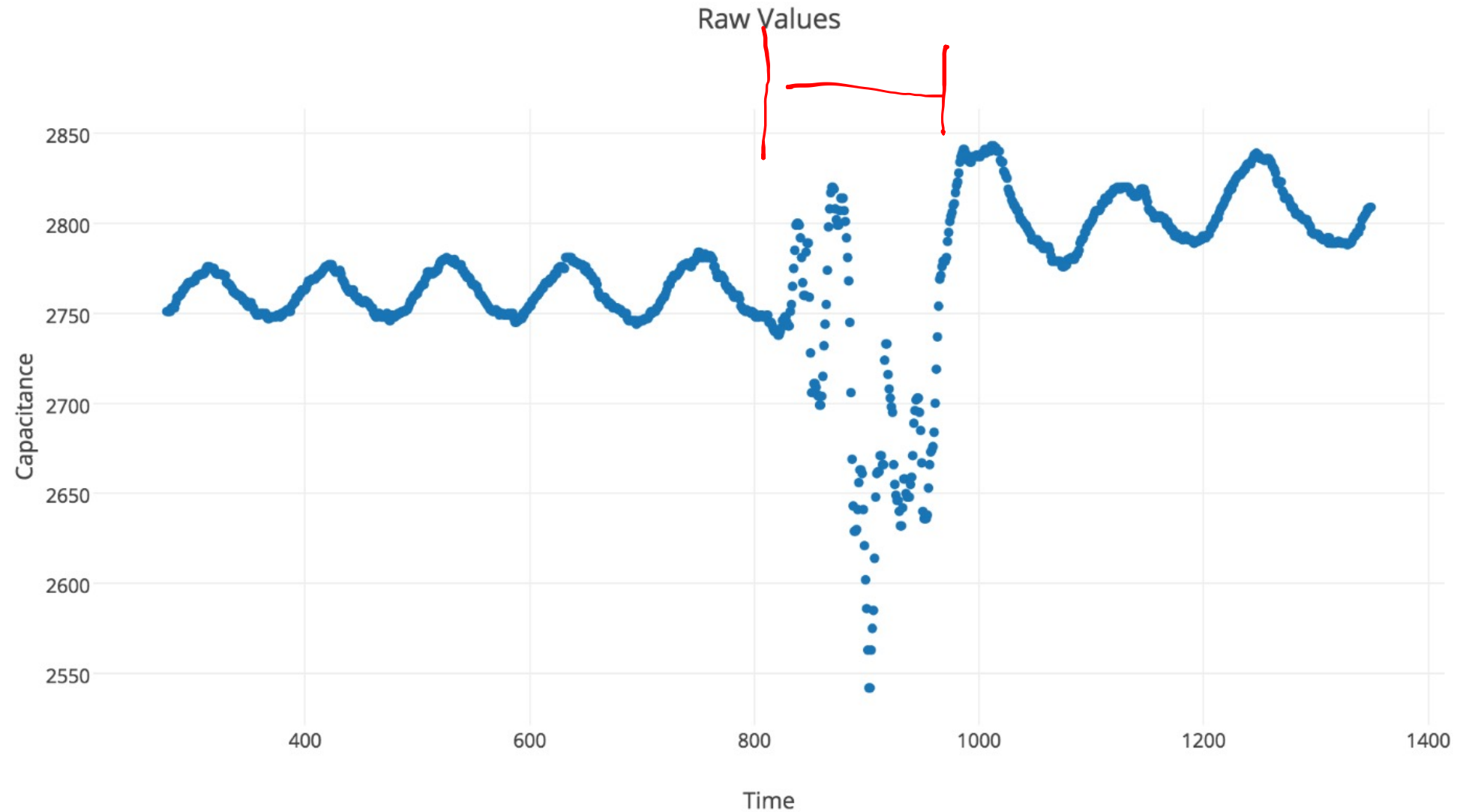
Data tidak representatif

- Data yang tidak representatif menghasilkan bias pada model



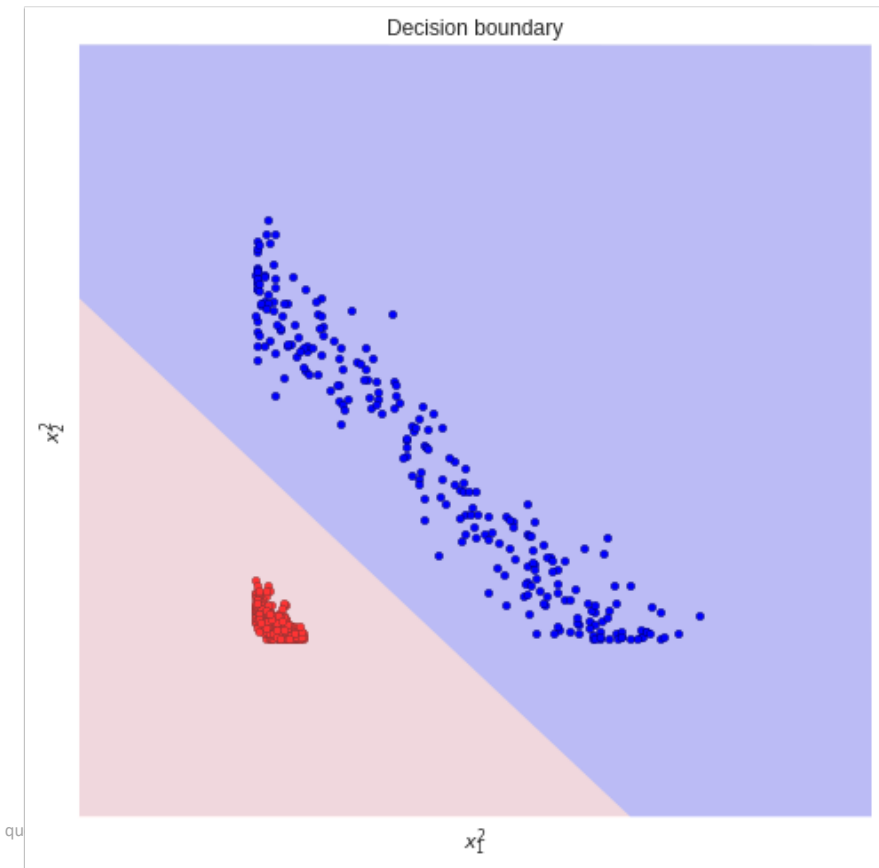
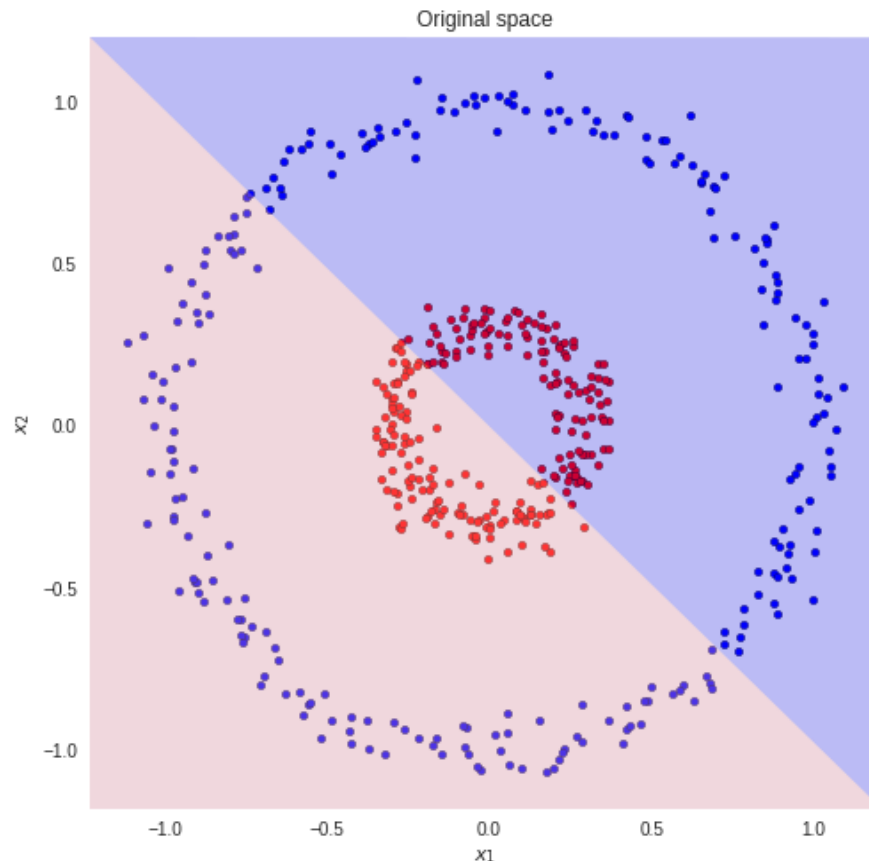
Kualitas data jelek

- Missing
- Error
- Outlier
- Noise



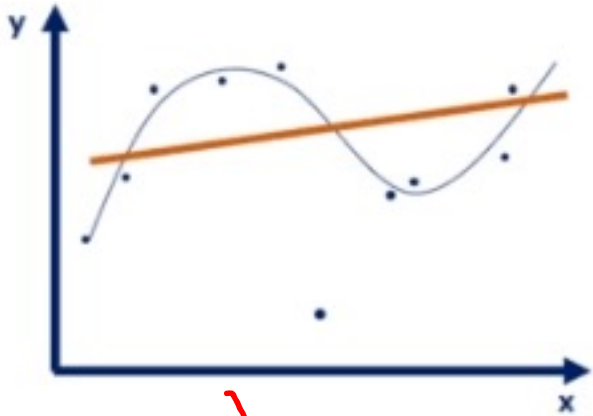
Fitur yang irelevan

- Banyak data tidak relevan terhadap model sehingga model mempelajari hal yang tidak penting (Garbage in garbage out)
- Maka diperlukan feature selection (memilih fitur yang penting) dan feature extraction (menggabungkan beberapa fitur yang ada)

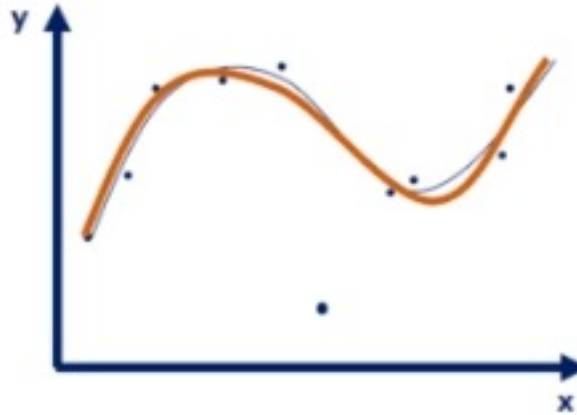


Overfitting dan Underfitting

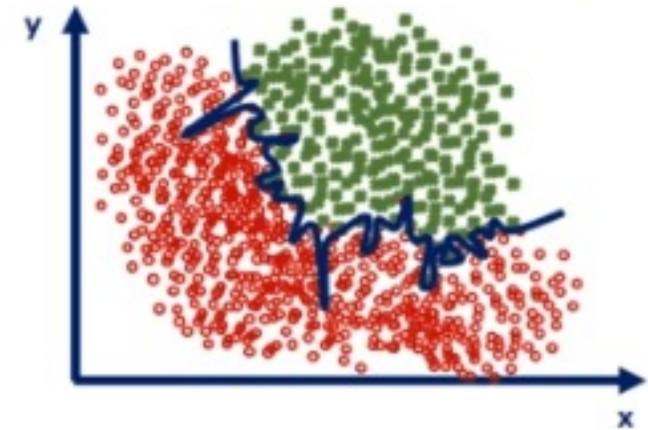
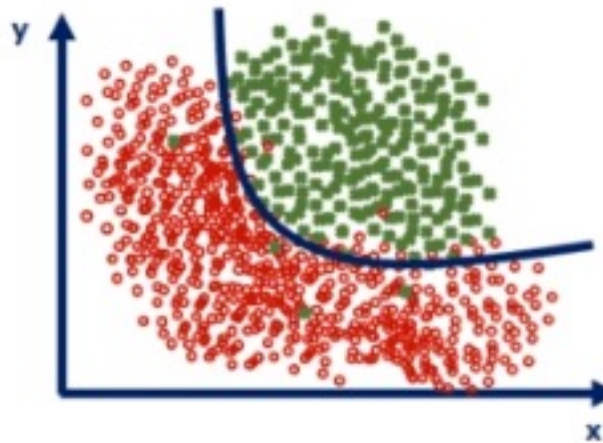
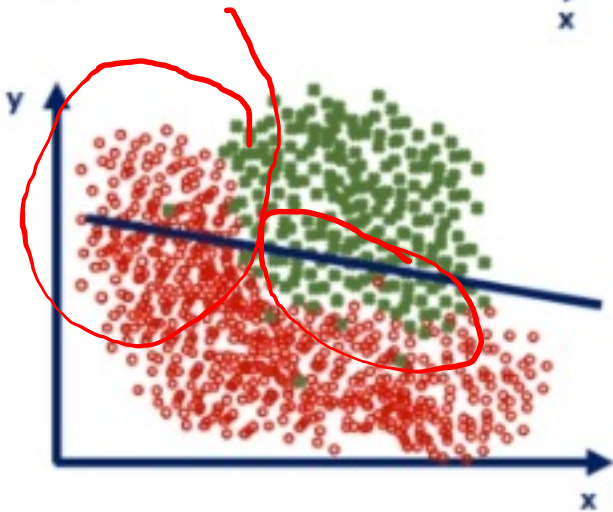
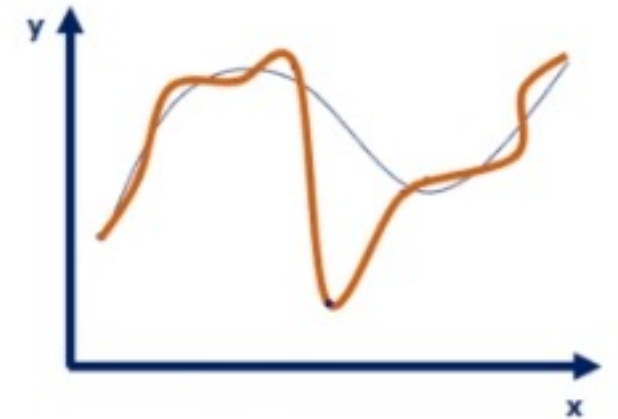
An **underfitted** model



A **good** model



An **overfitted** model



No Free Lunch Theorem

- https://en.wikipedia.org/wiki/No_free_lunch_theorem
- <https://direct.mit.edu/neco/article-abstract/8/7/1341/6016/The-Lack-of-A-Priori-Distinctions-Between-Learning?redirectedFrom=fulltext>
- Tidak ada satu model yang superior dari yang lain, suatu model cocok untuk suatu data dan tidak cocok untuk data yang lain

End-to-end ML

Alur Kerja ML

- Pada bagian ini, kita akan mengikuti contoh proyek ML di bab 2 pustaka utama
- Dimana terdapat beberapa tahapan:
 - Mendapatkan gambaran besar
 - Memperoleh data
 - Melakukan penelusuran dan visualisasi data
 - Mempersiapkan data untuk algoritma machine learning
 - Memilih dan melatih model
 - Melakukan fine-tune pada model
 - Launch, monitor, maintain sistem

Working with Real Data

- Belajar machine learning paling baik jika langsung bereksperimen dengan menggunakan real-world data
- Walau tidak mudah untuk memperoleh data asli, namun sudah ada beberapa sumber data online yang dapat digunakan.
- Popular open data:
 - UC Irvine Machine Learning Repository (<http://archive.ics.uci.edu/ml/>)
 - Kaggle datasets (<https://kaggle.com/datasets>)
 - Amazon's AWS datasets (<https://registry.opendata.aws/>)
- Meta portals:
 - <http://dataportals.org/>
 - <http://opendatamonitor.eu/>
 - <http://quandl.com/>

Mendapatkan gambaran besar

- Anda diminta membuat model untuk memprediksi data dengan menggunakan data sensus di California. Di dalam data ini terdapat informasi berupa populasi, median pendapatan, dan median harga rumah untuk blok lokasi di California.
- Hal pertama yang harus Anda lakukan adalah bertanya mengenai tujuan bisnis. Membuat model bukanlah tujuan akhir. Kita harus berpikir, kira-kira apa yang ingin model ini lakukan?
- Memahami tujuan sangatlah penting karena akan mempengaruhi bagaimana kita akan memetakan masalah, algoritma yang akan digunakan, alat ukur performansi yang akan dipilih, dan waktu yang harus kita luangkan.

Gambaran Besar

- Setelah mencari tahu, kita menemukan kalau keluaran model machine learning kita (berupa prediksi median harga rumah per blok) akan digunakan untuk analisis lain.
- Sampai dengan saat ini harga rumah dihitung oleh tenaga ahli secara manual: tim mengumpulkan data harga rumah yang paling update, ketika ada harga rumah yang tidak ditemukan, mereka melakukan estimasi dengan menggunakan suatu rumus yang kompleks. Hal ini memiliki biaya tinggi dan banyak memakan waktu, ditambah dengan hasil prediksi yang dibuat seringkali kurang 20% dari harga yang seharusnya.
- Data sensus sepertinya menjadi sumber data yang baik untuk keperluan ini, karena dataset ini sudah tercantum harga rumah dari berbagai kecamatan, dan data tambahan lainnya

Gambaran Besar

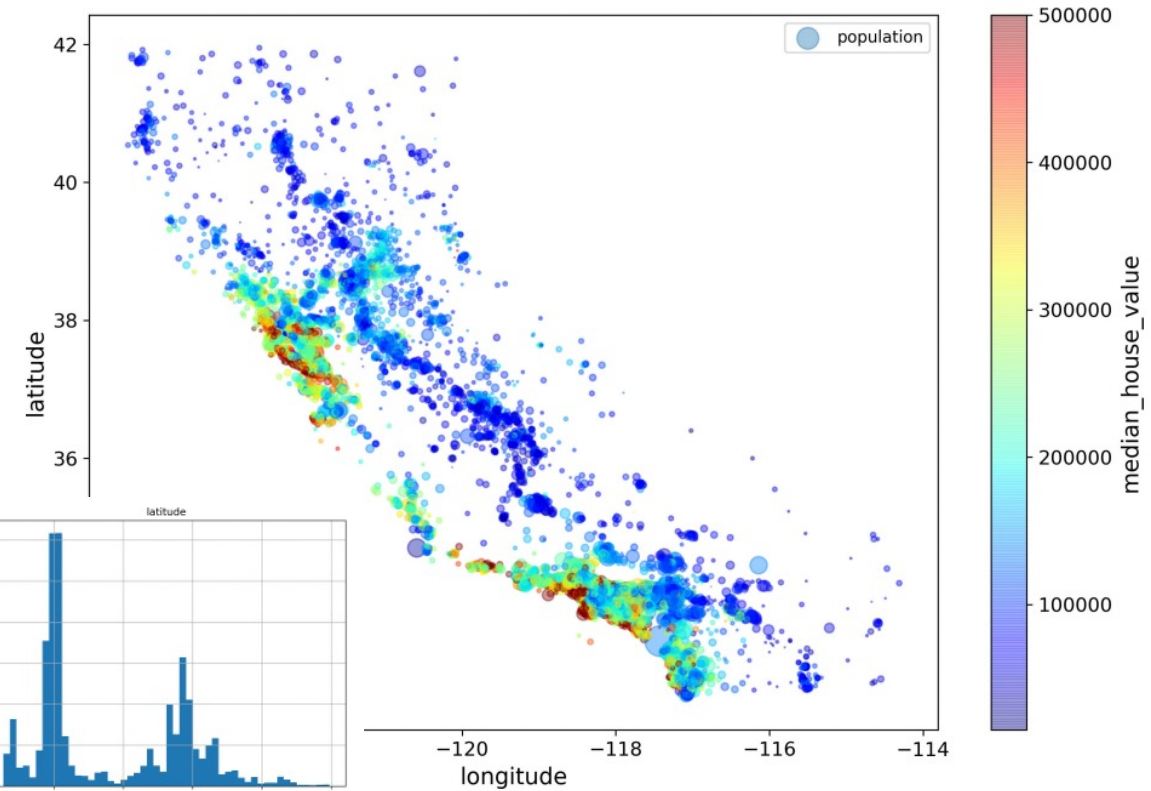
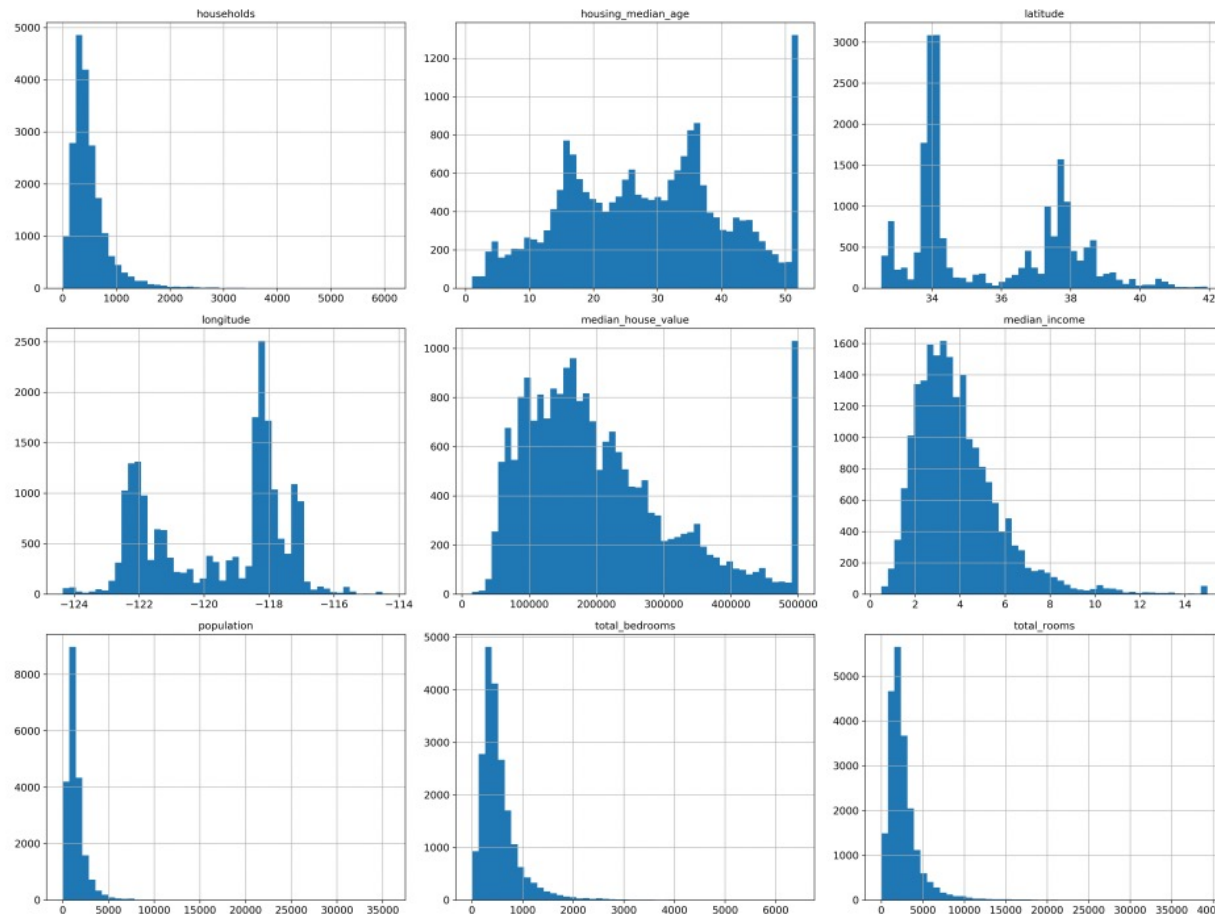
- Masalah ini adalah permasalahan supervised learning
 - Alasan: data yang kita miliki adalah data yang sudah dilabeli, karena data tersebut sudah memiliki informasi harga rumah
- Masalah ini adalah masalah regresi, secara spesifik adalah multiple regression
 - Alasan: data yang kita miliki memiliki beberapa fitur dan akan digunakan untuk melakukan prediksi
- Masalah ini adalah masalah univariate regresion
 - Alasan: karena kita hanya ingin melakukan prediksi satu nilai saja per blok
- Alat ukur performa adalah RMSE atau MAE
 - Alasan: karena kita ingin hasil prediksi sedekat mungkin dengan nilai aslinya

Memperoleh data

- Download data
 - Biasanya data tersimpan di dalam folder komputer di mana kita bekerja. Kadang data tersebut berada di dalam database atau url. Di dalam contoh berikut ini data tersebut berada di dalam url github dan filenya berupa csv.
<https://github.com/ageron/handson-ml2/tree/master/datasets/housing>
- Panggil Data
 - Jika data yang digunakan sudah berada di dalam komputer kita dalam format csv, maka dapat data tersebut dapat dipanggil ke dalam lingkungan Jupyter Notebook dalam format dataframe dengan menggunakan library pandas.

Penelusuran dan visualisasi data

- Cek tipe data
- Deskripsi data secara statistik
- Visualisasi data

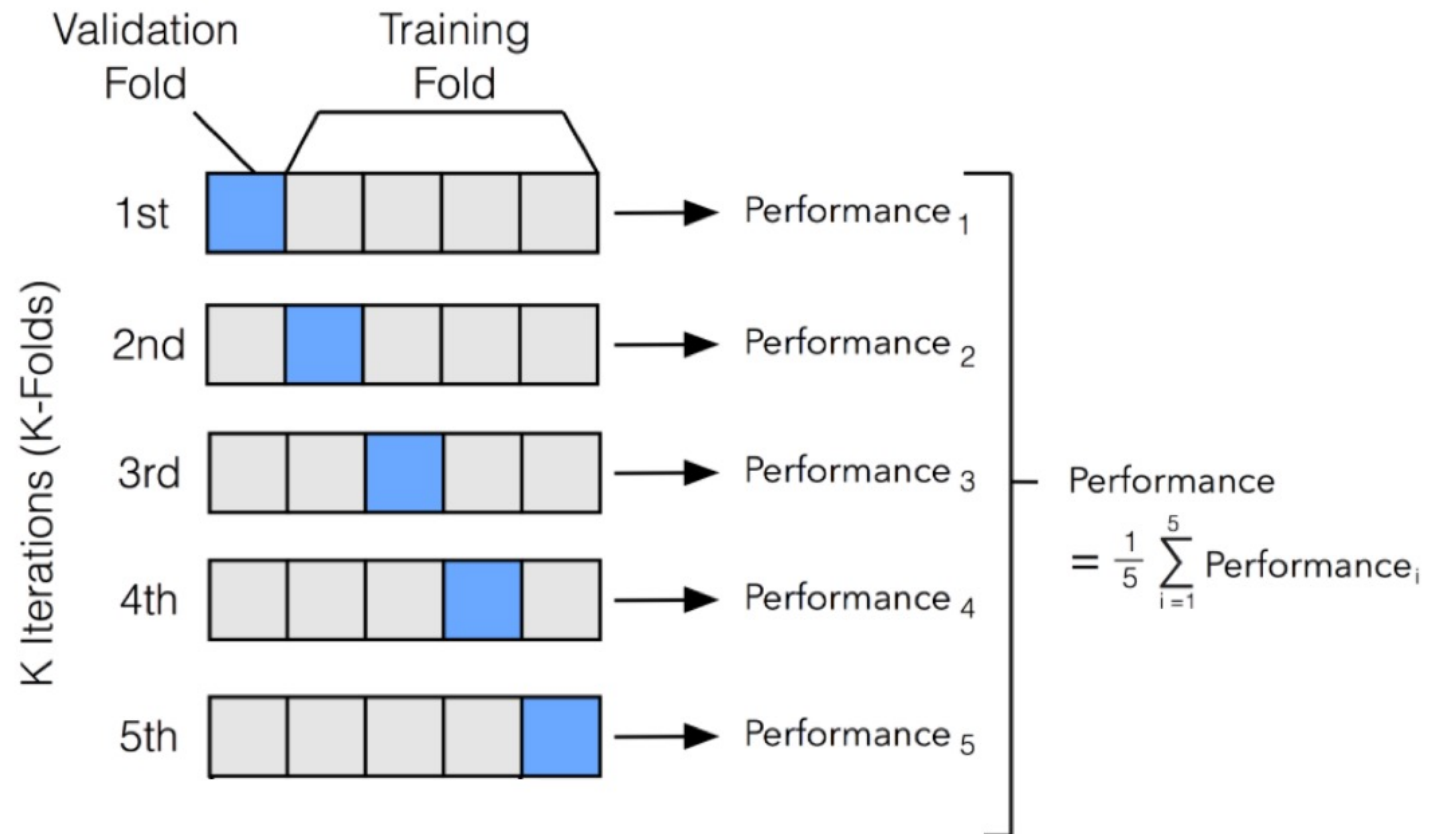


Mempersiapkan data untuk algoritma ML

- Data latih dan data uji
- Pembersihan data: Cek dan buang/isi nilai null/outlier
- Menangani data teks dan kategorial
- Pengamatan dan rekayasa fitur
- Feature scaling

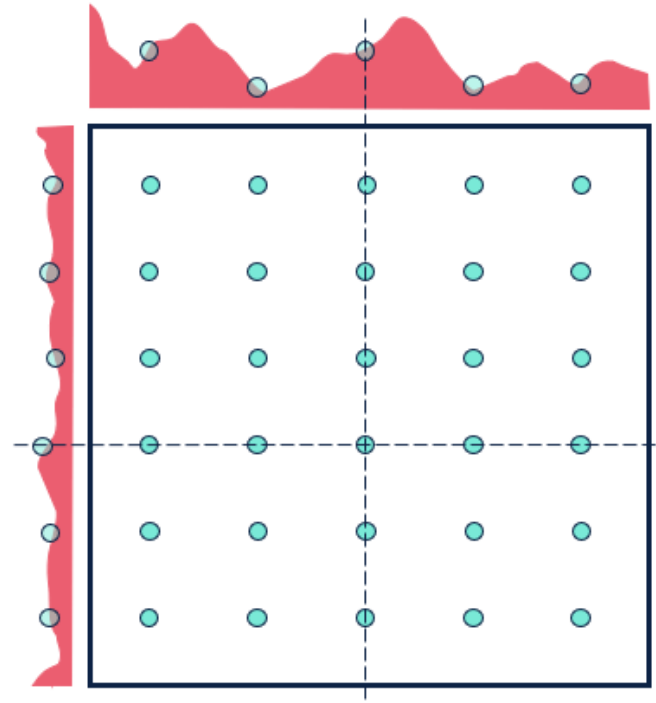
Memilih dan melatih model

- Memilih model: regresi linear
- Melatih model: algoritma gradient descent pada data latih (optimisasi parameter)
- Evaluasi model: RMSE pada data uji dengan cross-validation (5-fold)
- Menyimpan model

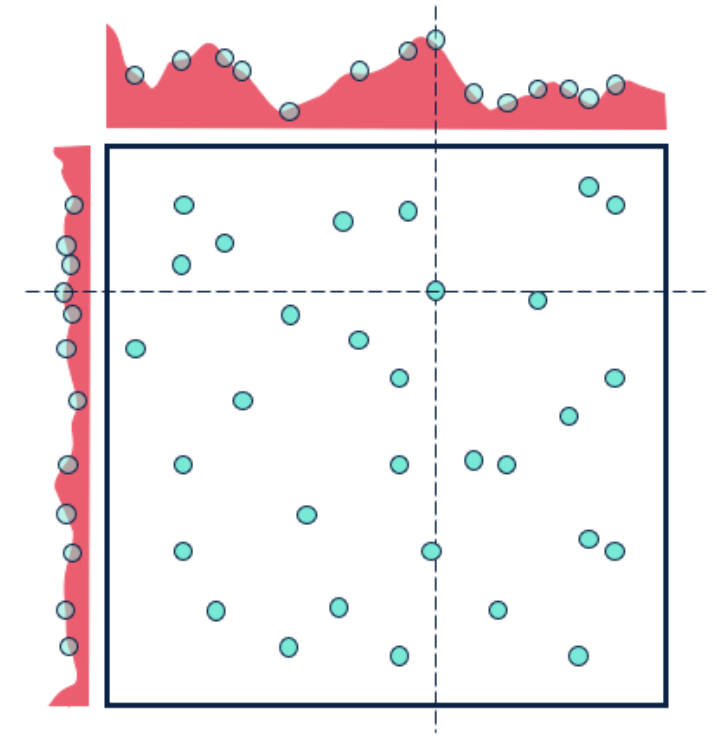


Melakukan finetune

- GridSearch
- RandomizedSearch








Grid Search



Random Search

Launch, monitor, maintain sistem

- Deploy model  Aplikasi
- Monitor distribusi data  balance
- Monitor performa model  $RMSE < \epsilon$
- Monitor dependensi pipeline 
- Perbaiki jika ada problem 

Tuhan Memberkati