Direktorat Jenderal Pendidikan Tinggi, Riset, dan, Teknologi
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Republik Indonesia

DIKTI SIGAP MELAYANI

Kampus Merdeka
INDONESIA JAYA

IARC

MICROCREDENTIAL: ASSOCIATE DATA SCIENTIST

01 November – 10 Desember 2021

Pertemuan ke-10

Membangun Model 1 (Dasar Regresi dan Regresi Linier)

Profil Pengajar: Nama Lengkap dan Gelar Akademik

Poto Pengajar

Contak Pengajar:

Ponsel:

xxxxxx

Email:

xxxxxxx

Jabatan Akademik: .....

Latar Belakang Pendidikan:

- S1: .....
- S2: .....
- S3: .....

Riwayat/Pengalaman Pekerjaan:

- Dosen
- Xxxx
- Xxxx
- Xxxx
- xxxx

Deskripsi Topik

KODE UNIT : **J.62DMI00.012.1**

JUDUL UNIT : **Membangun Skenario Model**

DESKRIPSI UNIT : Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membangun skenario model.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Mengidentifikasi teknik pemodelan	1.1 **Asumsi-asumsi spesifik** mengenai data diidentifikasi sesuai karakteristik data. 1.2 **Teknik-teknik pemodelan** data diidentifikasi sesuai karakteristik data dan tujuan teknis *data science*.
2. Menentukan teknik pemodelan yang sesuai dengan karakteristik data dan tujuan teknis *data science*	2.1 Teknik pemodelan yang sesuai dengan karakteristik data ditentukan. 2.2 Deskripsi teknik pemodelan yang dipilih didokumentasikan sesuai SOP yang berlaku.
3. Menyiapkan skenario pengujian	3.1 **Skenario uji** yang mungkin diterapkan ditentukan sesuai tujuan teknis. 3.2 **Metrik evaluasi** pengujian diidentifikasi sesuai skenario uji. 3.3 Skenario uji yang dipilih didokumentasikan sesuai standar yang berlaku.

BATASAN VARIABEL

1. Konteks variabel
   - 1.1 Asumsi spesifik di antaranya, namun tidak terbatas pada nilai minimum dan maksimum data, periode minimum dan maksimum waktu untuk data *time series*.
   - 1.2 Teknik pemodelan di antaranya, namun tidak terbatas pada untuk klasifikasi: *decision tree, naive bayes, neural network, deep learning*; untuk klastering: *Self-Organizing Map* (SOM), *k-means*; untuk regresi: linier, regresi *Long Short Time Memory* (LSTM), *Recurrent Neural Network* (RNN); untuk rekomendasi: *apriori, asociacte rule, frequent item set*.
   - 1.3 Skenario uji antara lain *percentage splitting, cross validation*.
   - 1.4 Metrik evaluasi setidaknya terdiri dari paramater evaluasi, *interpretability*, waktu tanggap, dan nilai ambang batasnya (*threshold*). Yang dimaksud parameter evaluasi di antaranya: akurasi, presisi, *recall*, *f1-score*, kohesi, *Mean Absolute Error* (MAE).

Course Definition

- Modul ini adalah bagian pertama dari Membangun Model
- Membangun Model yang dibahas adalah:
  - Merancang Skenario Model
  - Membangun Model dengan regresi linier
- Terdapat beberapa algoritma yang akan dibahas
- Pembangunan model menggunakan library
- Modul ini akan dilanjutkan dengan pembahasan pembangunan model dengan menggunakan model supervised dan unsupervised lainnya.

Learning Objective

Dalam pelatihan ini diharapkan:

- Peserta mampu melakukan kegiatan persiapan pemodelan seperti pembagian data, penyusunan skenario pemodelan
- Peserta mampu melakukan proses pemodelan dengan regresi linier

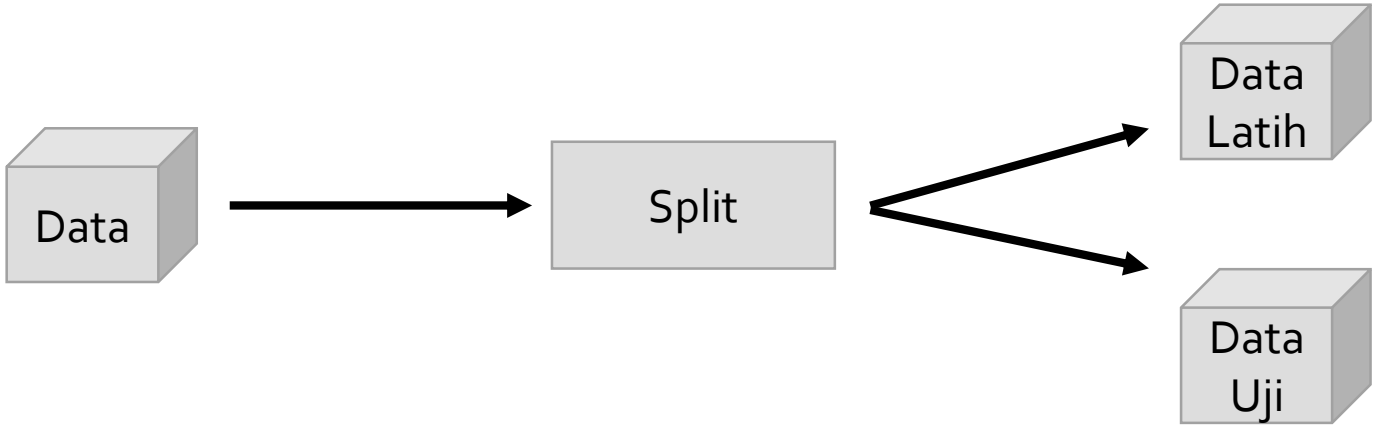
Outline

- Membangun Skenario Pemodelan :
  - Pembagian data: data latih, data uji, *k*-fold cross validation
  - Menentukan Langkah Eksperimen
  - Parameter Evaluasi
- Membangun Model regresi linier:
  - Algoritma yang diimplementasi menggunakan library
  - Matriks Performansi

Skenario Pemodelan

Pembagian Data

- Data dibagi menjadi 2 bagian:
  - Data Latih (*Training Data*): untuk mengembangkan model
  - Data Uji (*Testing Data*): untuk Mengukur performansi model



Pembagian Data

- Dataset Iris (https://archive.ics.uci.edu/ml/datasets/iris):
  - Data Latih (*Training Data*) : 70%
  - Data Uji (*Testing Data*) : 30%

X_train | y_train

Panjang Sepal	Lebar Sepal	Panjang Petal	Lebar Petal	Kelas
5.1	3.5	1.4	0.2	Iris Setosa
6.3	3.3	6	2.5	Iris Virginica
7	3	4.6	1.4	Iris Versicolour
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
5.8	3.3	6	2.4	Iris Virginica
6.8	3.1	4.5	1.5	Iris Versicolour
4.9	3	1.4	0.2	Iris Setosa
...	...	...	...	...
6.8	3.2	4.4	1.6	Iris Versicolour

Training Data 70%

Testing Data 30%

X_test | y_test

Hands On

Data Latih : 70%
Data Uji : 30%

```
[5]  from sklearn.model_selection import train_test_split

     X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.7)
```

```
[6]  print("Banyak data latih setelah dilakukan Train-Test Split: ", len(X_train))
     print("Banyak data uji setelah dilakukan Train-Test Split: ", len(X_test))

     Banyak data latih setelah dilakukan Train-Test Split:  105
     Banyak data uji setelah dilakukan Train-Test Split:  45
```

Output, jumlah data latih dan data uji

k-Fold Cross Validation

- *k*-Fold Cross Validation digunakan pada dataset dengan jumlah data yang relatif sedikit
- *k*-Fold Cross Validation dilakukan pada data latih
- Data latih dibagi menjadi *k* bagian kemudian secara iteratif, 1 bagian menjadi data validasi

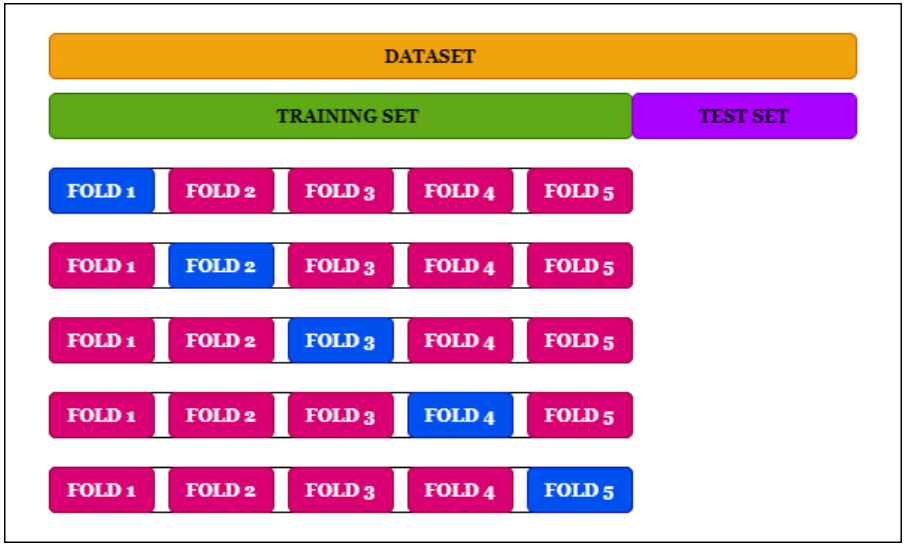


Image Source : https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538

Hands On

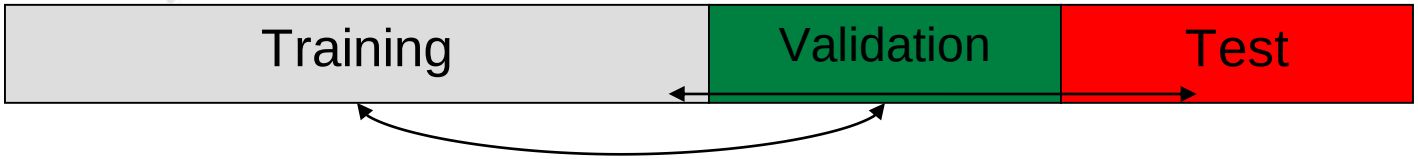
```
[ ]  from sklearn.model_selection import cross_val_score
     from sklearn.svm import SVC

     model = SVC(kernel = 'linear', C = 1)
     scores = cross_val_score(model, X, y, cv = 5)
     print("Akurasi model SVM untuk tiap fold: ", scores)
     print("Akurasi model SVM dengan 5-Fold Cross Validation: ", scores.mean())
```

5 Cross Validation

```
Akurasi model SVM untuk tiap fold:  [0.96666667 1.         0.96666667 0.96666667 1.
Akurasi model SVM dengan 5-Fold Cross Validation:  0.9800000000000001
```

Output akurasi dari setiap fold
Akurasi rata- rata dari seluruh fold

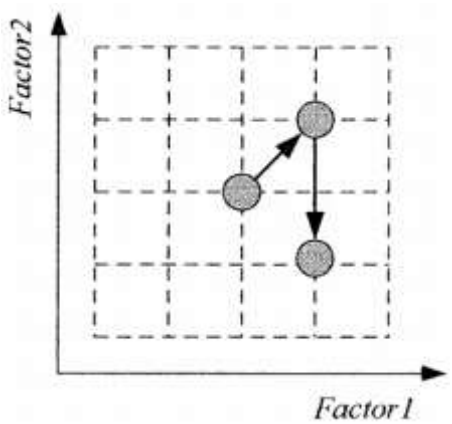
Training – Validation – Testing Data



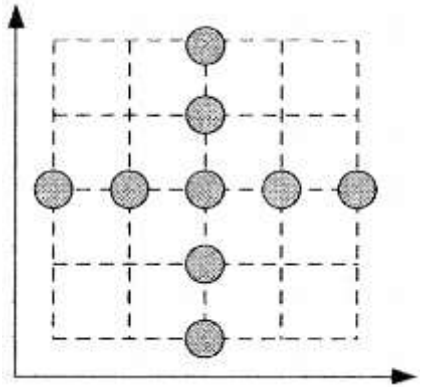
- Model Selection: Mengestimasi performa model - model yang berbeda untuk memilih model yang terbaik, yaitu model dengan minimum error
- Model Assessment: Dari model yang terpilih, mengestimasi error untuk data baru (data uji)

Menentukan Langkah Eksperimen

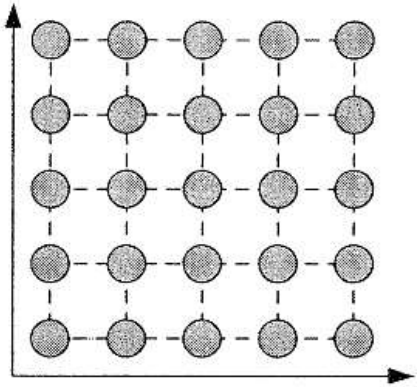
- Setiap metode memiliki parameter tertentu
- Dilakukan eksperimen dengan beberapa variasi parameter
- Parameter yang menghasilkan model performa terbaik akan digunakan selanjutnya
- Beberapa strategi pencarian parameter untuk menghasilkan model terbaik



Best Guess



One Factor at A Time



Grid Search

Parameter Evaluasi

- Klasifikasi
  - Akurasi
  - Presisi
  - Recall/Sensitivity
  - Specificity
  - F1-measure
  - ...
- Regresi
  - MSE (Mean Squared Error)
  - MAPE (Mean Absolute Percentage Error)
  - ...
- Klastering
  - Silhouette Score
  - Davies-Bouldin Index
  - ...
- Parameter Evaluasi akan dijelaskan secara detail pada materi pertemuan berikutnya

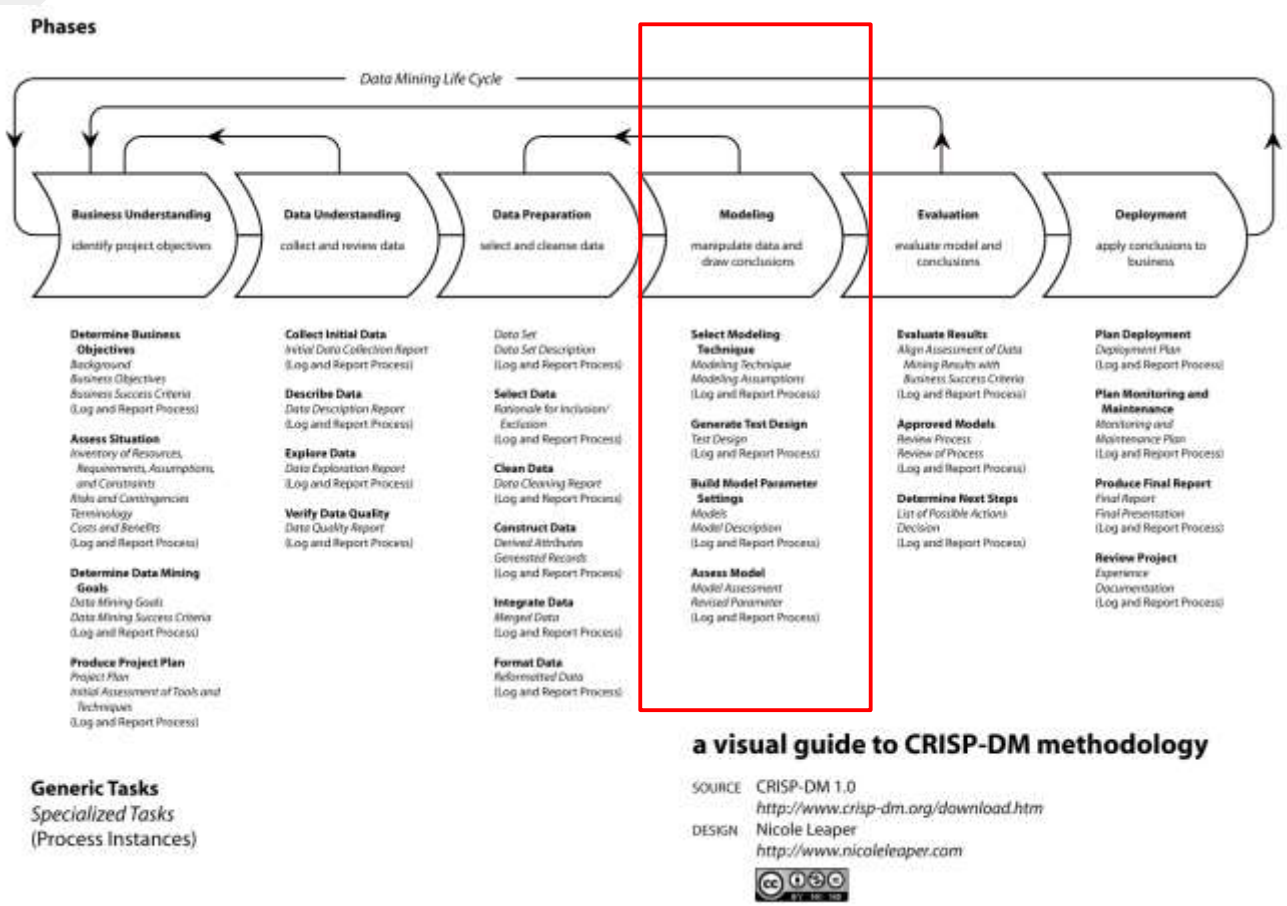
KODE UNIT : **J.62DMI00.013.1**

JUDUL UNIT : **Membangun Model**

DESKRIPSI UNIT : Unit kompetensi ini berhubungan dengan pengetahuan, keterampilan, dan sikap kerja yang dibutuhkan dalam membangun model.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Menyiapkan parameter model	1.1 Parameter-parameter yang sesuai dengan model diidentifikasi. 1.2 Nilai toleransi parameter evaluasi pengujian ditetapkan sesuai dengan tujuan teknis.
2. Menggunakan *tools* pemodelan	2.1 ***Tools* untuk membuat model** diidentifikasi sesuai dengan tujuan teknis *data science*. 2.2 Algoritma untuk teknik pemodelan yang ditentukan dibangun menggunakan *tools* yang dipilih. 2.3 Algoritma pemodelan dieksekusi sesuai dengan **skenario pengujian** dan *tools* untuk membuat model yang telah ditetapkan. 2.4 **Parameter model** algoritma dioptimasi untuk menghasilkan **nilai parameter evaluasi** yang sesuai dengan skenario pengujian.

1. Konteks variabel
   1.1 Termasuk di dalam skenario pengujian adalah komposisi *data training* dan *data testing*, cara pemilihan data *training* dan data testing seperti *percentage splitting, random selection,* atau *cross validation.*
   1.2 Yang dimaksud dengan parameter model di antaranya arsitektur model, banyaknya *layer* atau simpul, *learning rate* untuk *neural network,* nilai *k* untuk *k-means*, nilai *pruning* untuk *decision tree.*
   1.3 Nilai parameter evaluasi adalah nilai ambang batas *(threshold)* yang bisa diterima.
   1.4 Yang dimaksud dengan *tools* pemodelan di antaranya perangkat lunak *data science* di antaranya: *rapid miner, weka*, atau *development* untuk bahasa pemrograman tertentu seperti *python* atau R.


Phases
Data Mining Life Cycle
Business Understanding
identify project objectives
Data Understanding
collect and review data
Data Preparation
select and cleanse data
Modeling
manipulate data and draw conclusions
Evaluation
evaluate model and conclusions
Deployment
apply conclusions to business
Determine Business Objectives
Background
Business Objectives
Business Success Criteria
(Log and Report Process)
Assess Situation
Inventory of Resources, Requirements, Assumptions, and Constraints
Risks and Contingencies
Terminology
Costs and Benefits
(Log and Report Process)
Determine Data Mining Goals
Data Mining Goals
Data Mining Success Criteria
(Log and Report Process)
Produce Project Plan
Project Plan
Initial Assessment of Tools and Techniques
(Log and Report Process)
Collect Initial Data
Initial Data Collection Report
(Log and Report Process)
Describe Data
Data Description Report
(Log and Report Process)
Explore Data
Data Exploration Report
(Log and Report Process)
Verify Data Quality
Data Quality Report
(Log and Report Process)
Data Set
Data Set Description
(Log and Report Process)
Select Data
Rationale for Inclusion/ Exclusion
(Log and Report Process)
Clean Data
Data Cleaning Report
(Log and Report Process)
Construct Data
Derived Attributes
Generated Records
(Log and Report Process)
Integrate Data
Merged Data
(Log and Report Process)
Format Data
Reformatted Data
(Log and Report Process)
Select Modeling Technique
Modeling Technique
Modeling Assumptions
(Log and Report Process)
Generate Test Design
Test Design
(Log and Report Process)
Build Model Parameter Settings
Models
Model Description
(Log and Report Process)
Assess Model
Model Assessment
Revised Parameter
(Log and Report Process)
Evaluate Results
Align Assessment of Data Mining Results with Business Success Criteria
(Log and Report Process)
Approved Models
Review Process
Review of Process
(Log and Report Process)
Determine Next Steps
List of Possible Actions
Decision
(Log and Report Process)
Plan Deployment
Deployment Plan
(Log and Report Process)
Plan Monitoring and Maintenance
Monitoring and Maintenance Plan
(Log and Report Process)
Produce Final Report
Final Report
Final Presentation
(Log and Report Process)
Review Project
Experience Documentation
(Log and Report Process)
Generic Tasks
Specialized Tasks
(Process Instances)
a visual guide to CRISP-DM methodology
SOURCE CRISP-DM 1.0
http://www.crisp-dm.org/download.htm
DESIGN Nicole Leaper
http://www.nicoleleaper.com

Definisi Kursus

Pelatihan ini menjelaskan regresi dan bagaimana membangun model (regresi), yaitu:
a. menyiapkan parameter model
b. menggunakan tools pemodelan

selanjutnya menjelaskan algoritma dan menggunakan Regresi Linier, dan performansi regresi dengan Python dan Scikit-learn.

Capaian Pembelajaran

Peserta dapat menjelaskan, menyiapkan, dan mengimplementasikan model regresi dengan algoritma Regresi Linier sederhana dan variabel jamak

Beserta pengukuran performansinya menggunakan Python dan Scikit-learn.

Tujuan Pembelajaran

Peserta mempelajari pengertian, cara menyiapkan, dan cara implementasi model regresi dengan algoritma Regresi Linier sederhana dan variabel jamak.

Beserta pengukuran performansinya menggunakan Python dan Scikit-learn.

Regresi

Pengertian Regresi

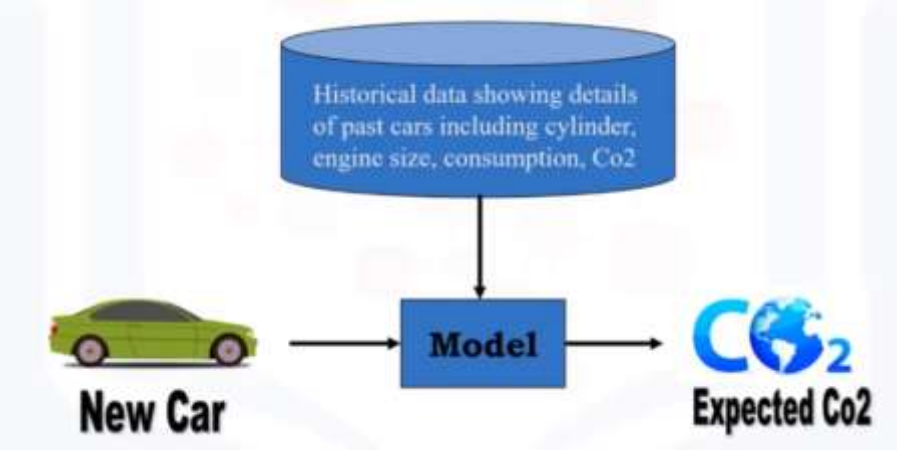
x : variabel bebas

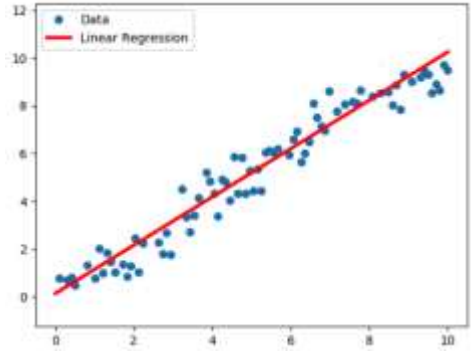
y : variabel tak bebas

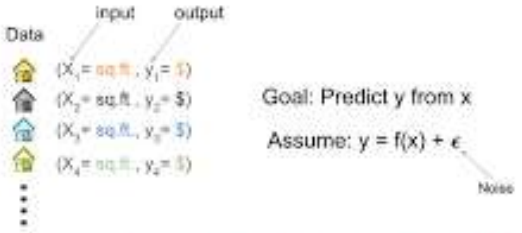
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Regresi adalah proses Memprediksi nilai kontinu

Model Regresi



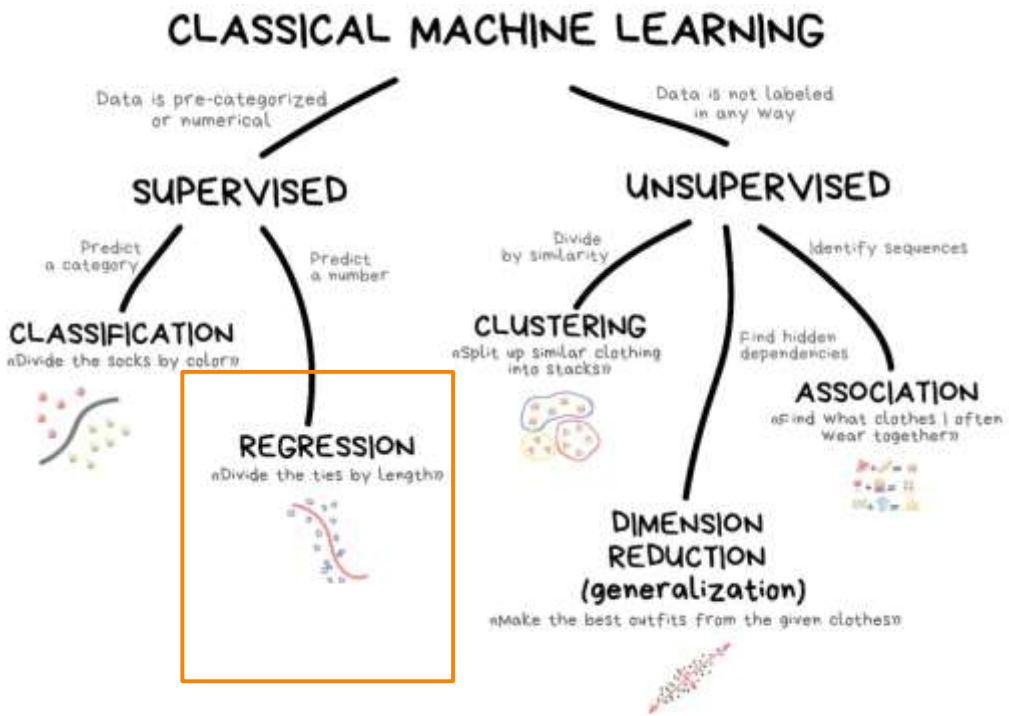


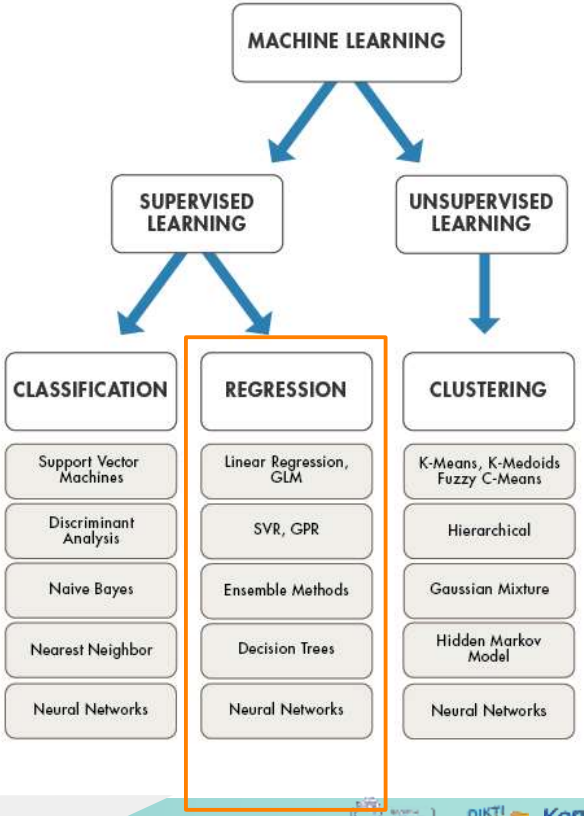


Task: Learn function f that "best" approximates the data.

f(sq.ft.) = $

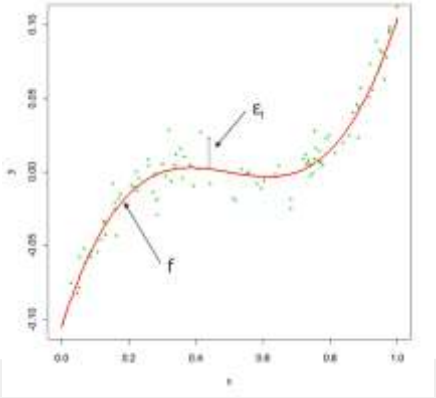
Regresi dalam Machine Learning

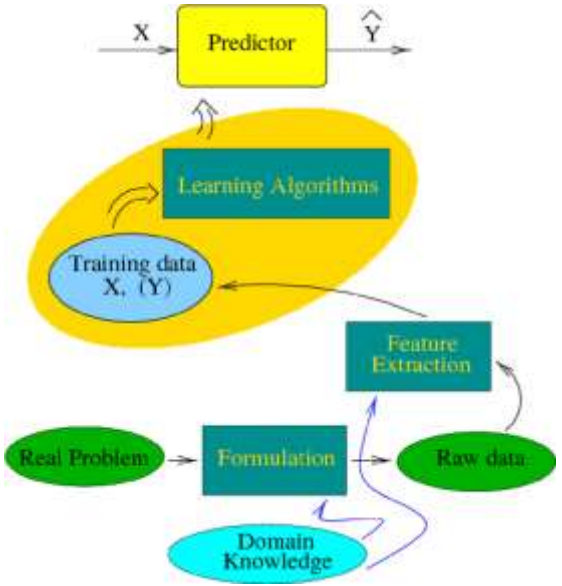




Machine Learning

- Pembelajaran dari data digunakan pada situasi, saat tidak ada solusi analitis, tetapi kita dapat menggunakan data untuk membuat suatu solusi empiris
- Premis dasar pada proses ini adalah menggunakan suatu himpunan pengamatan untuk menguak proses yang ada di dalamnya (underlying process)
- Anggap kita mengobservasi suatu ruang output dan ruang input.
- Ada suatu hubungan antara Y dan paling tidak satu X. Maka dapat dimodelkan hubungan tersebut sebagai *f*, dimana f adalah suatu fungsi yang tidak diketahui dan ε adalah random error (noise), tak bergantung dari *X* dengan mean nol.





Mengapa – Bagaimana Melakukan Estimasi *f*

- Mengapa melakukan estimasi *f*:
- Untuk keperluan:
  - Prediksi
  - Inferensi

- Pertama, kita asumsikan kita memiliki himpunan **training data**

$$\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \square\ , (\mathbf{X}_n, Y_n)\}$$

- Kedua, digunakan training data dan metode machine learning untuk melakukan estimasi *f*.
- Metode yang digunakan:
  - Parametric
  - Non-parametric methods

Prediksi

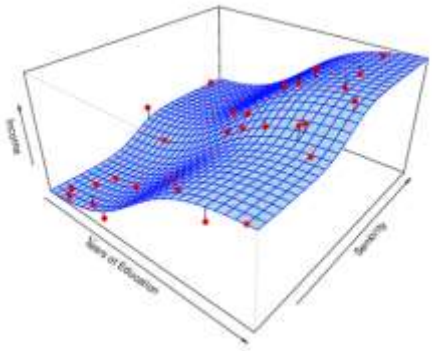
- By producing a good estimate for *f* where the variance of ε is not too large, then we can make accurate predictions for the response variable, *Y*, based on a new value of ***X***.
- We can predict *Y* using (***X***) where represents our estimate for *f*, and represents the resulting prediction for *Y*.
- The accuracy of as a prediction for *Y* depends on:
  - *Reducible error*
  - *Irreducible error*
- Note that will not be a perfect estimate for *f*; this inaccuracy introduces error.
- This error is *reducible* because we can potentially improve the accuracy of the estimated (i.e. hypothesis) function by using the most appropriate learning technique to estimate the target function *f*.
- Even if we could perfectly estimate *f*, there is still variability associated with ε that affects the accuracy of predictions = *irreducible* error.
- Average of the squared difference between the predicted and actual value of *Y*.
- Var(ε) represents the *variance* associated with ε.
- Our aim is to minimize the reducible error!!

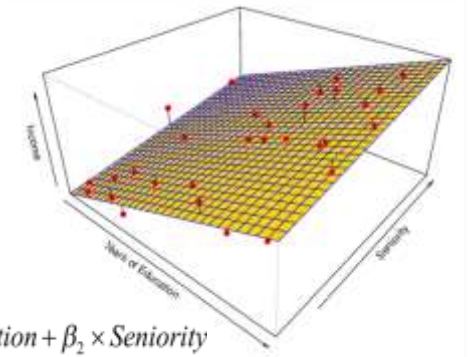
$$E[(Y - \hat{f}(X))^2 | X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{Reducible} + \underbrace{\mathrm{Var}(\epsilon)}_{Irreducible}$$

Inferensi

- Disamping melakukan prediksi, kita mungkin tertarik terhadap relasi antara Y dan X
- Pertanyaan kunci
  - Prediktor mana yang mempengaruhi response
  - Relasi negatif atau positif
  - Relasi sederhana, linier atau lebih kompleks

Metode Parametric





$$f = \beta_0 + \beta_1 \times Education + \beta_2 \times Seniority$$

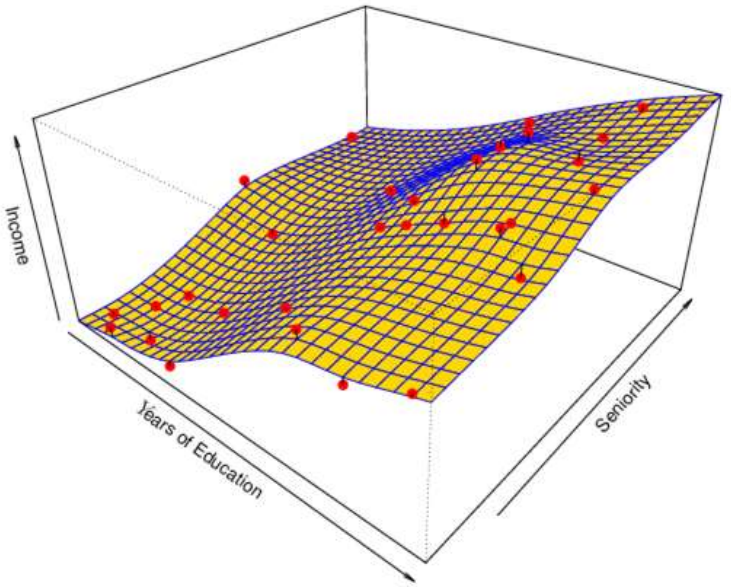
- This reduces the *learning problem* of estimating the target function *f* down to a problem of estimating a set of **parameters**.
- This involves a two-step approach…
- **Step 1:**
  - Make some assumptions about the functional form of *f*. The most common example is a linear model:

$$f(\mathbf{X}_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \square + \beta_p X_{ip}$$

- **Step 2:**
  - We use the *training data* to fit the model (i.e. estimate *f*….the unknown parameters).
  - The most common approach for estimating the parameters in a linear model is via ordinary least squares (OLS) linear regression

Pendekatan Non Parametric

- Pendekatan ini tidak memiliki asumsi yang eksplisit tentang bentuk fungsi *f*.
- **Keuntungan**: Akurat untuk beragam jenis bentuk fungsi *f*
- **Kerugian**: Membutuhkan jumlah observasi untuk memperoleh estimasi dari fungsi *f*
- Regresi non linier lebih flexible dan berpotensi memberikan estimasi lebih akurat
- Tetapi metode ini berisiko over-fitting data (mengikuti error, noise terlalu ditail). Terlalu fleksibel dapat menghasilkan estimasi f yang buruk



Pertimbangan

- **Notasi**
  - Input *X*: *feature*, *predictor*, or *independent variable*
  - Output *Y*: *response*, *dependent variable*
- **Kategorisasi**
  - Supervised learning vs. unsupervised learning. *Pertanyaan kunci*: Apakah Y tersedia pada data training
  - Regression vs. Classification. Pertanyaan kunci: Apakah Y kuantitatif atau kualitatif?
- **Kuantitatif:**
  - Pengukuran atau perhitungan yang tercatat sebagai nilai numeris (tinggi, suhu dll)
- **Qualitative:** kelompok atau kategori
  - Ordinal: kelompok kategori berurut (ukuran baju seperti S, M, L)
  - Nominal: nama kategori (e.g. status pernikahan, jenis kelamin)

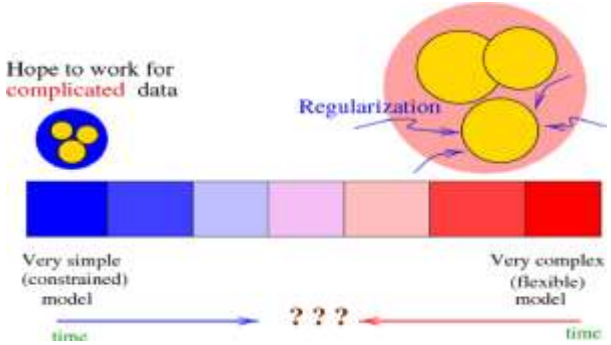
Akurasi Prediktif vs Interpretibilitas

- Conceptual Question:
- Why not just use a more flexible method if it is more realistic?

- Reason 1: A simple method (such as OLS regression) produces a model that is easier to interpret (especially for inference purposes).
- Reason 2: Even if the primary purpose of learning from the data is for prediction, it is often possible to get more accurate predictions with a simple rather than a complicated model.

Pertimbangan Machine Learning

- There are always two aspects to consider when designing a learning algorithm:
  - Try to fit the data well
  - Be as robust as possible
- The predictor that you have generated using your training data must also work well on new data.
- When we create predictors, usually the simpler the predictor is, the more robust it tends to be in the sense of begin able to be estimated reliably. On the other hand, the simple models do not fit the training data aggressively.



- Training Error vs. Testing Error:
  - Training error – reflects whether the data fits well
  - Testing error – reflects whether the predictor actually works on new data
- Bias vs. Variance:
  - Bias – how good the predictor is, on average; tends to be smaller with more complicated models
  - Variance – tends to be higher for more complex models
- Fitting vs. Over-fitting:
  - If you try to fit the data too aggressively, then you may over-fit the training data. This means that the predictors works very well on the training data, but is substantially worse on the unseen test data.
- Empirical Risk vs. Model Complexity:
  - Empirical risk – error rate based on the training data
  - Increase model complexity = decrease empirical risk but less robust (higher variance)

Tipe Model Regresi

Regresi Sederhana:

- Regresi sederhana linier
- Regresi sederhana non-linier
- Contoh: memprediksi `co2emission` vs `EngineSize` dari semua mobil.

Regresi Variabel Jamak:

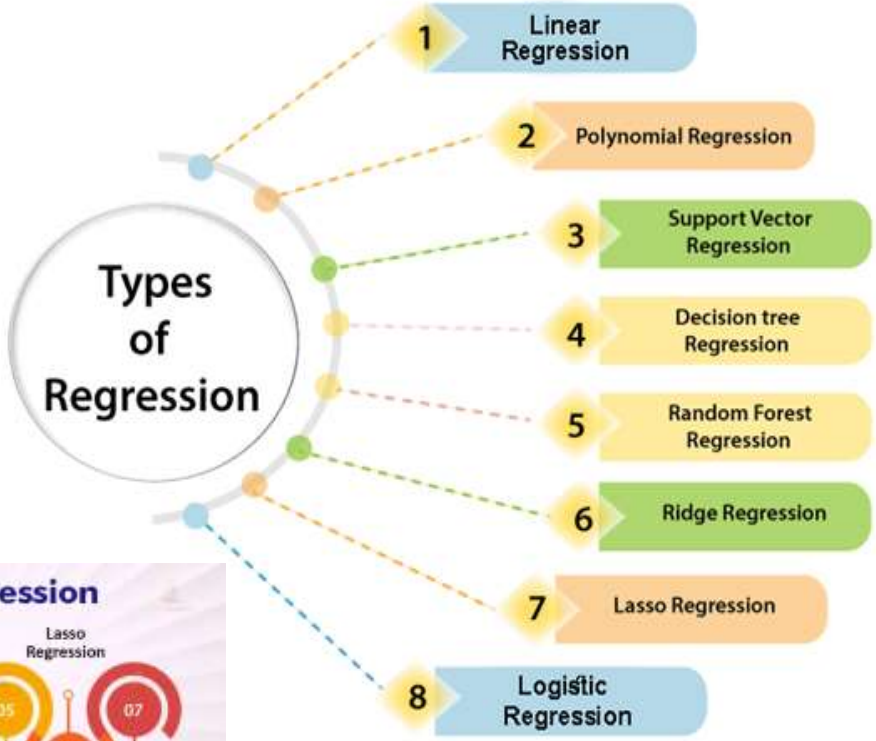
- Regresi variabel jamak linier
- Regresi variabel jamak non-linier
- Contoh: memprediksi `co2emission` vs `EngineSize` dan `Cylinders` dari semua mobil.

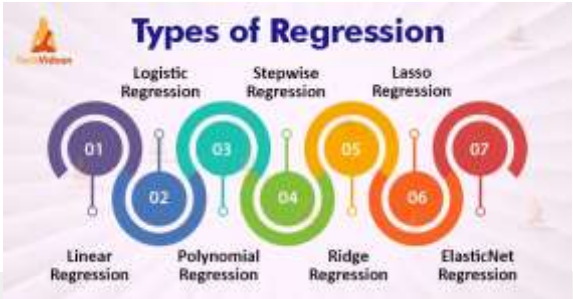
Aplikasi Regresi

- Prakiraan penjualan produk
- Analisis kepuasan
- Estimasi harga
- Pendapatan pekerjaan
- dst.

Algoritma Regresi

- Linier Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- LASSO Regression
- ANN Regression
- K-NN Regression
- dst.





Regresi Linier Sederhana

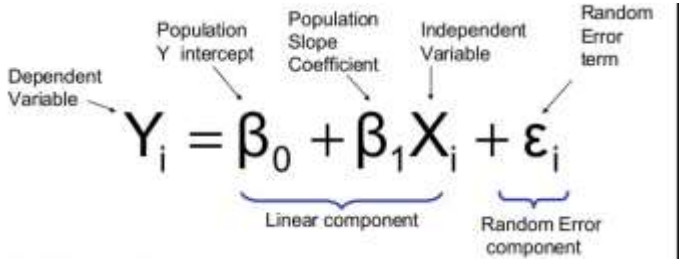
Regresi Linier Untuk Memprediksi Nilai Kontinu

x : variabel bebas

y : variabel tak bebas

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Nilai kontinu / numerik



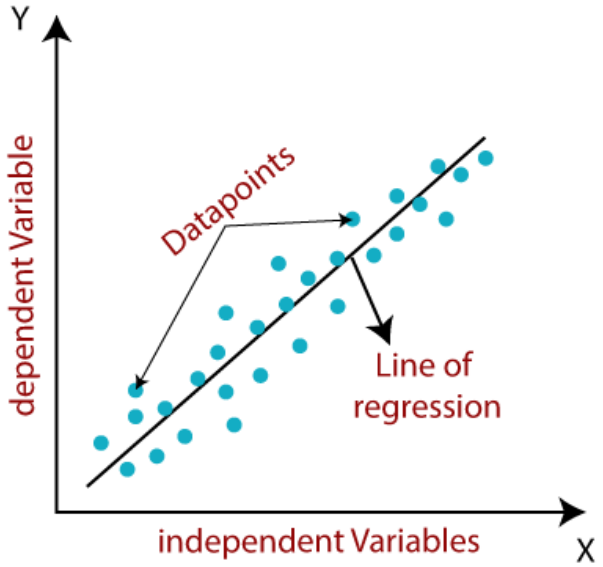
Topologi Regresi Linier

Regresi Linier Sederhana:

- Memprediksi `co2emission` vs `EngineSize` dari semua mobil
  - variabel bebas (x): `EngineSize`
  - variabel tak bebas (y): `co2emission`

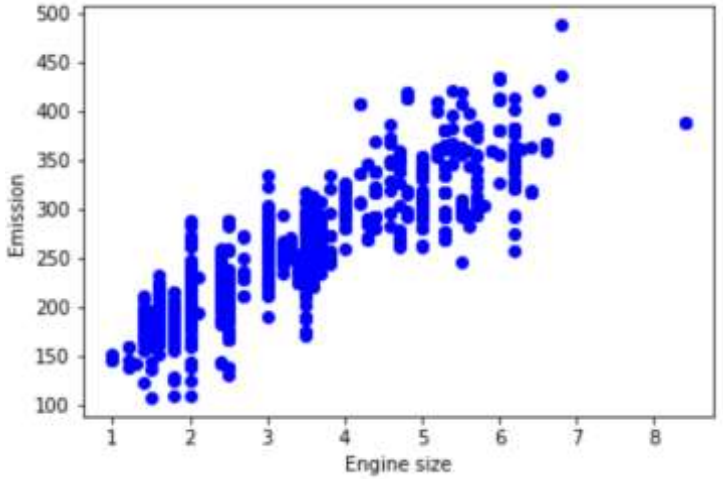
Regresi Linier Variabel Jamak:

- Memprediksi `co2emission` vs `EngineSize` dan `Cylinders` dari semua mobil
  - variabel bebas (x): `EngineSize`, `Cylinders`, dst.
  - variabel tak bebas (y): `co2emission`



Cara Kerja Regresi Linier

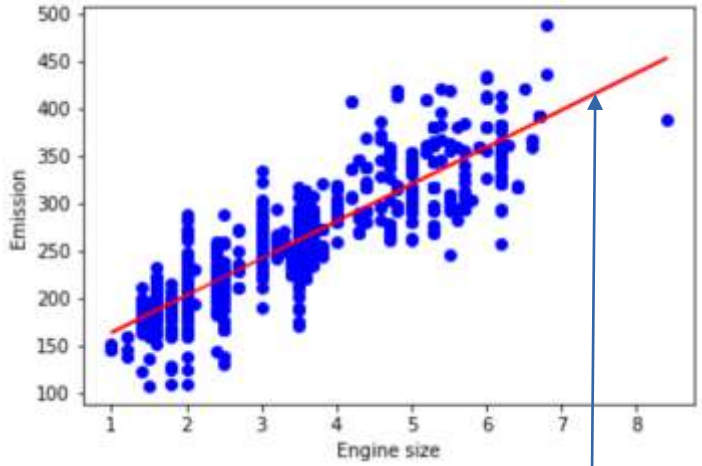
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



- Model regresi akan ditentukan parameternya dengan data satu variabel bebas untuk memprediksi variabel tak bebas, sebagai contoh memprediksi nilai kontinu CO2EMISSIONS dengan variabel ENGINESIZE berdasar data pembelajaran (No 0 sd No 8).
- Hasil pemodelan dapat digunakan memprediksi nilai numerik CO2EMISSIONS kasus baru yang belum pernah dihadapi yakni kasus No 9 dengan dasar ENGINESIZE.

Cara Kerja Regresi Linier

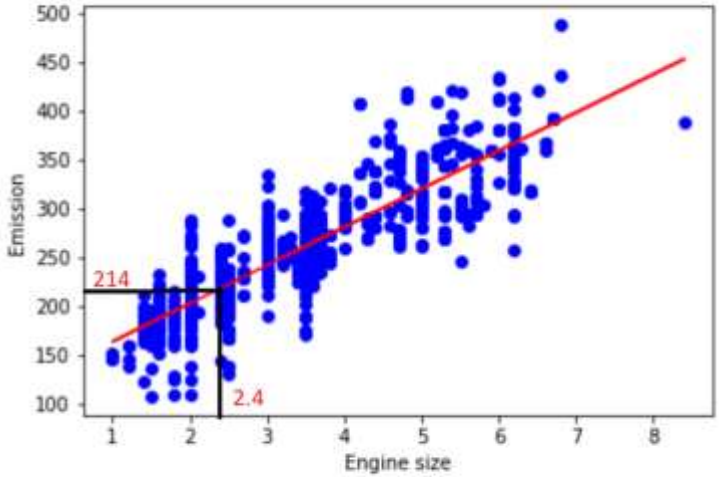
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



Persamaan linear

Cara Kerja Regresi Linier

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



- Garis merah pada Gambar adalah model linier yang dihasilkan oleh algoritma regresi linier. Berdasarkan garis merah itu dapat diketahui nilai CO2EMISSIONS sebagai variabel tak bebas dengan melihat nilai pada ENGINESIZE, dalam hal ini bernilai = 2,4 yang menghasilkan nilai CO2EMISSIONS = 214.
- Parameter regresi dari model persamaan linier yaitu θ0 dan θ1 sebagaimana dalam Gambar berikut ini.

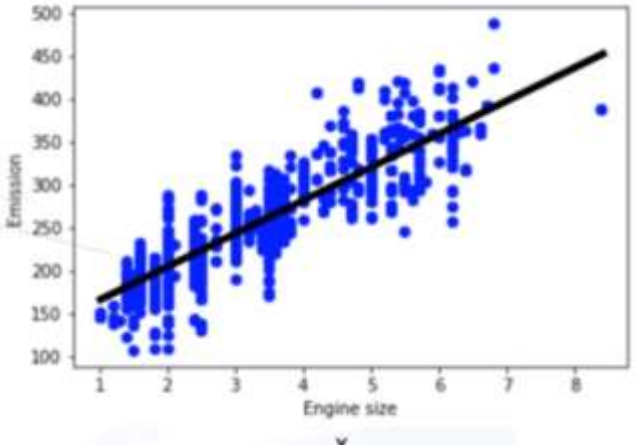
Cara Kerja Regresi Linier

$$\hat{y} = \theta_0 + \theta_1 x_1$$

response variable

a single predictor

y



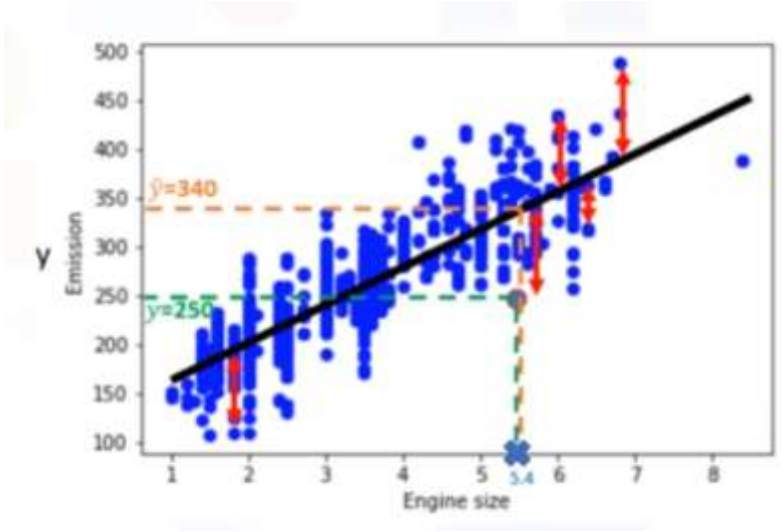
Cara Mencari Parameter Model Terbaik

$x_1 = 5.4$ independent variable
$y = 250$ actual Co2 emission of x1

$\hat{y} = \theta_0 + \theta_1 x_1$
$\hat{y} = 340$ the predicted emission of x1

Error = $y - \hat{y}$
$= 250 - 340$
$= -90$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



- Untuk mendapatkan parameter model regresi terbaik maka dicari parameter yang membuat selisih terkecil antara prediksi dengan nilai aktual yang disebut error.
- Dalam hal ini metrik yang paling sering digunakan adalah Mean Squared Error (MSE) sebagaimana ditunjukkan dalam Gambar untuk menghindari saling menegasikan antara error positif dan error negatif.

Estimasi Parameter

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

X_1 = ENGINESIZE, y = CO2EMISSIONS

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$\theta_1 = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$$\bar{x} = (2.0 + 2.4 + 1.5 + \dots)/9 = 3.34$$

$$\bar{y} = (196 + 221 + 136 + \dots)/9 = 256$$

$$\theta_1 = \frac{(2.0 - 3.34)(196 - 256) + (2.4 - 3.34)(221 - 256) + \dots}{(2.0 - 3.34)^2 + (2.4 - 3.34)^2 + \dots}$$

$$\theta_1 = 39$$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\theta_0 = 256 - 39 * 3.34$$

$$\theta_0 = 125.74$$

Parameter yang terbaik dicari dengan menggunakan metode Least Square seperti pada Gambar sehingga menghasilkan parameter terbaik dilihat dari MSE.

$$\hat{y} = 125.74 + 39x_1$$

Prediksi dengan Model Regresi Linier

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$Co2Emission = \theta_0 + \theta_1\ EngineSize$$

$$Co2Emission = 125 + 39\ EngineSize$$

$$Co2Emission = 125 + 39 \times 2.4$$

$$Co2Emission = 218.6$$

- Prediksi nilai kontinu variabel tak bebas dilakukan dengan memasukkan nilai variabel bebas ke dalam model yang sudah ditemukan.
- Dalam contoh Gambar, variabel EngineSize = 2,4 memberikan hasil Co2Emission = 218,6.

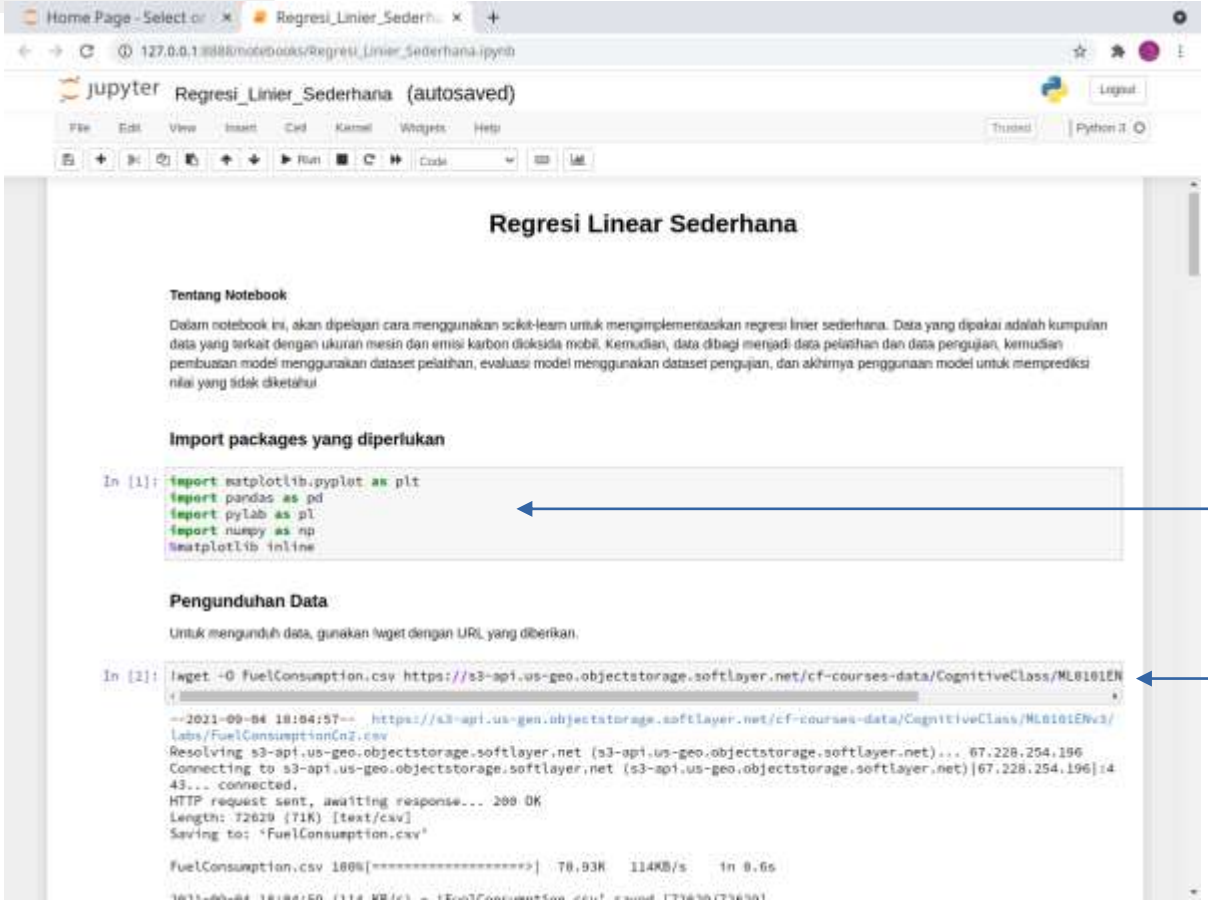
Kelebihan Regresi Linier

- Ringan
- Tidak perlu tuning parameter
- Mudah dipahami dan diinterpretasikan

Lab

- Jalankan file Jupyter Notebook untuk Regresi Linier Sederhana

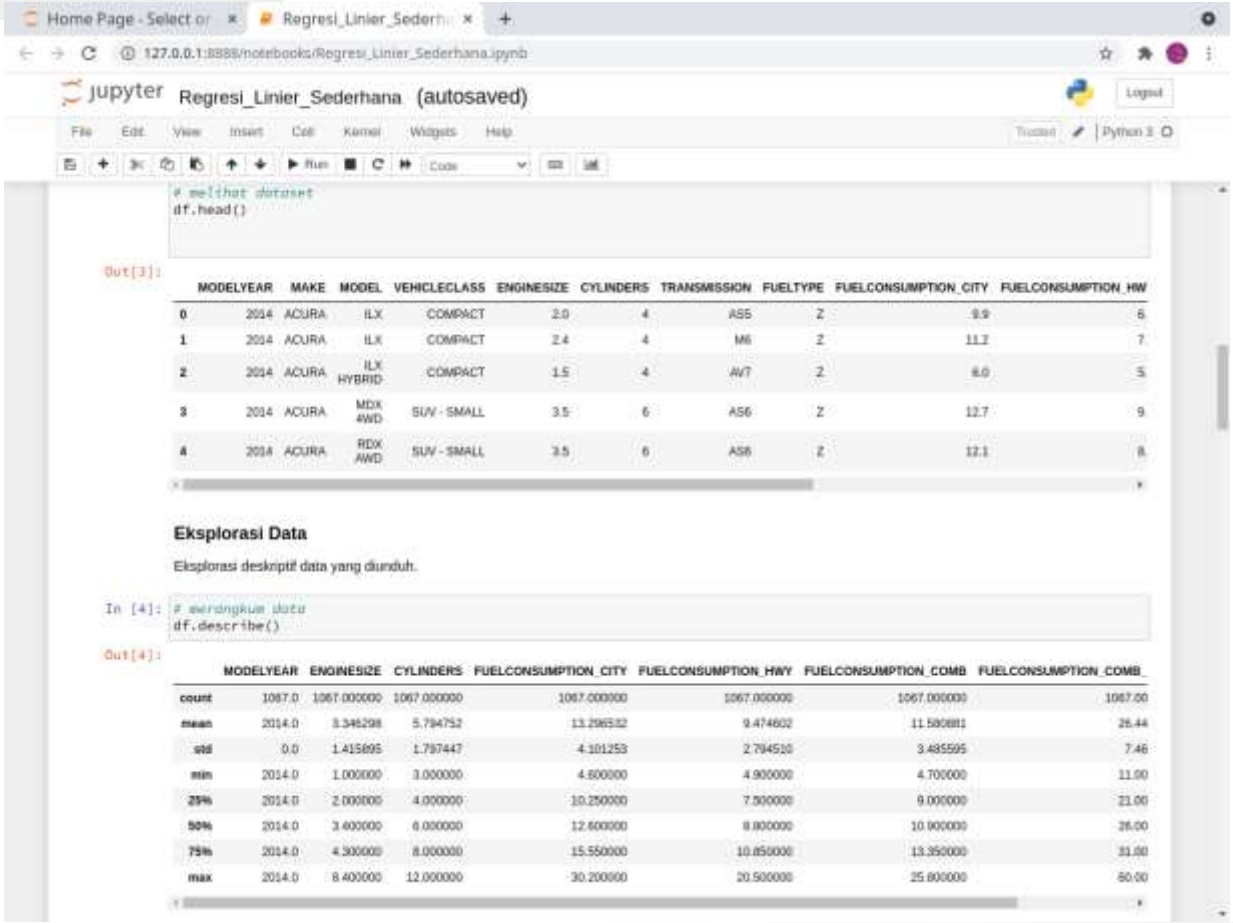
Regresi Linear Sederhana (1)



Library yang harus terinstal

Pastikan "wget" terinstal pada sistem anda

Regresi Linear Sederhana (2)



Regresi Linear Sederhana (3)

Home Page - Select or | Regresi_Linier_Sederh... | 127.0.0.1:8888/notebooks/Regresi_Linier_Sederhana.ipynb

jupyter Regresi_Linier_Sederhana (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3 | Logout

Run | Markdown

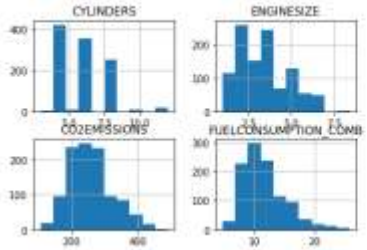
In [5]:
```
cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
cdf.head(10)
```

Out[5]:

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	212

Fitur-fitur tersebut dapat diplot sebagai berikut:

In [6]:
```
viz = cdf[['CYLINDERS','ENGINESIZE','CO2EMISSIONS','FUELCONSUMPTION_COMB']]
viz.hist()
plt.show()
```



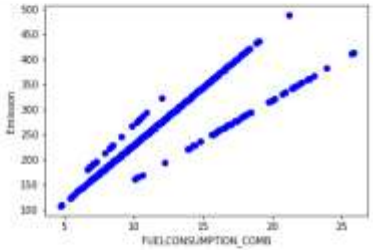
Regresi Linear Sederhana (4)

Home Page - Select or | Regresi_Linier_Sederh | 127.0.0.1:8888/notebooks/Regresi_Linier_Sederhana.ipynb

jupyter Regresi_Linier_Sederhana (autosaved)

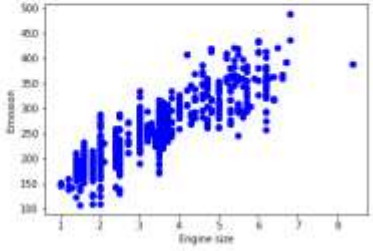
File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

In [7]:
```
plt.scatter(cdf.FUELCONSUMPTION_COMB, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("FUELCONSUMPTION_COMB")
plt.ylabel("Emission")
plt.show()
```



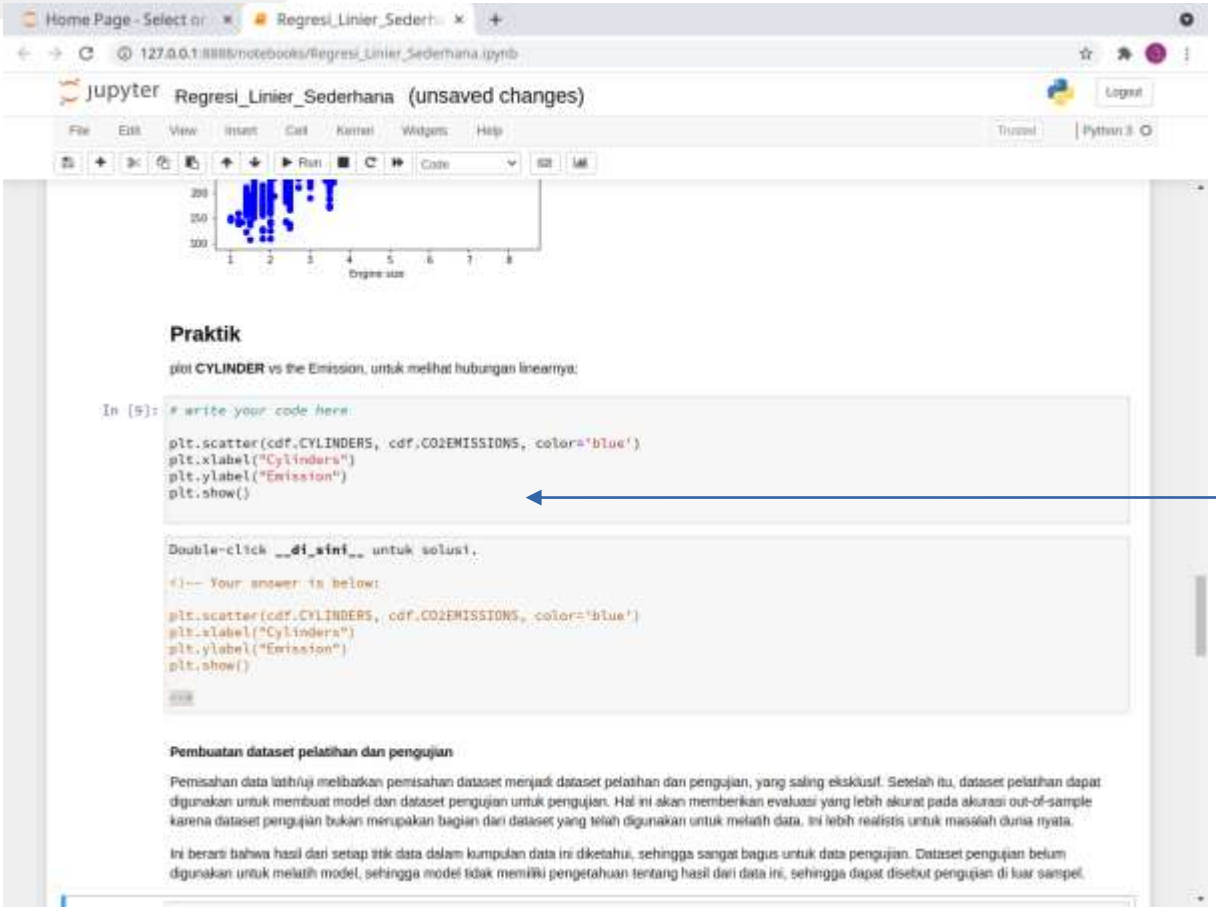
Plot scatter

In [8]:
```
plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



Plot scatter pasangan variabel lainnya

Regresi Linear Sederhana (5)



Plot scatter pasangan variabel lainnya

Regresi Linear Sederhana (6)

Home Page - Select or | Regresi_Linier_Sederh | 127.0.0.1:8888/notebooks/Regresi_Linier_Sederhana.ipynb

jupyter Regresi_Linier_Sederhana (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

digunakan untuk melatih model, sehingga model tidak memiliki pengetahuan tentang hasil dari data ini, sehingga dapat disebut pengujian di luar sampel.

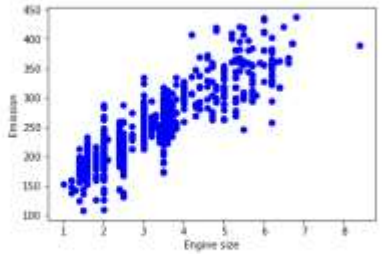
In [10]:
```
msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]
```

Model Regresi Sederhana

Regresi Linier cocok dengan model linier dengan koefisien B = (B1, ..., Bn) untuk meminimalkan 'jumlah kuadrat sisa (residu)' antara x bebas dalam kumpulan data, dan y tak bebas dengan pendekatan linier.

Distribusi Data Pelatihan

In [11]:
```
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



Pemodelan

Menggunakan sklearn package untuk memodelkan data.

Plot scatter pasangan variabel lainnya

Regresi Linear Sederhana (7)

Menggunakan sklearn package untuk memodelkan data.

```
In [12]: from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINESIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (train_x, train_y)
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[38.90563765]]
Intercept:  [126.04461402]
```

Regresi linear yang digunakan

Lakukan regresi linear

Koefisien persamaan garis regresi

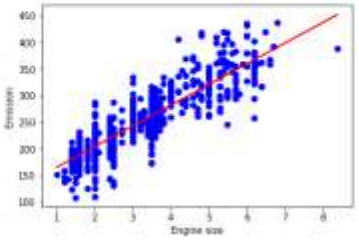
Seperti disebutkan sebelumnya, **koefisien** dan **intercept** dalam regresi linier sederhana, adalah parameter dari garis yang cocok dengan data. Mengingat bahwa ini adalah regresi linier sederhana, dengan hanya 2 parameter, dan mengetahui bahwa parameternya adalah intercept dan kemiringan atau gradien garis (koefisien), sklearn dapat memperkirakannya langsung dari data. Perhatikan bahwa semua data harus tersedia untuk menghitung parameter.

Plot output model

Plotting garis yang mencocoki terhadap data:

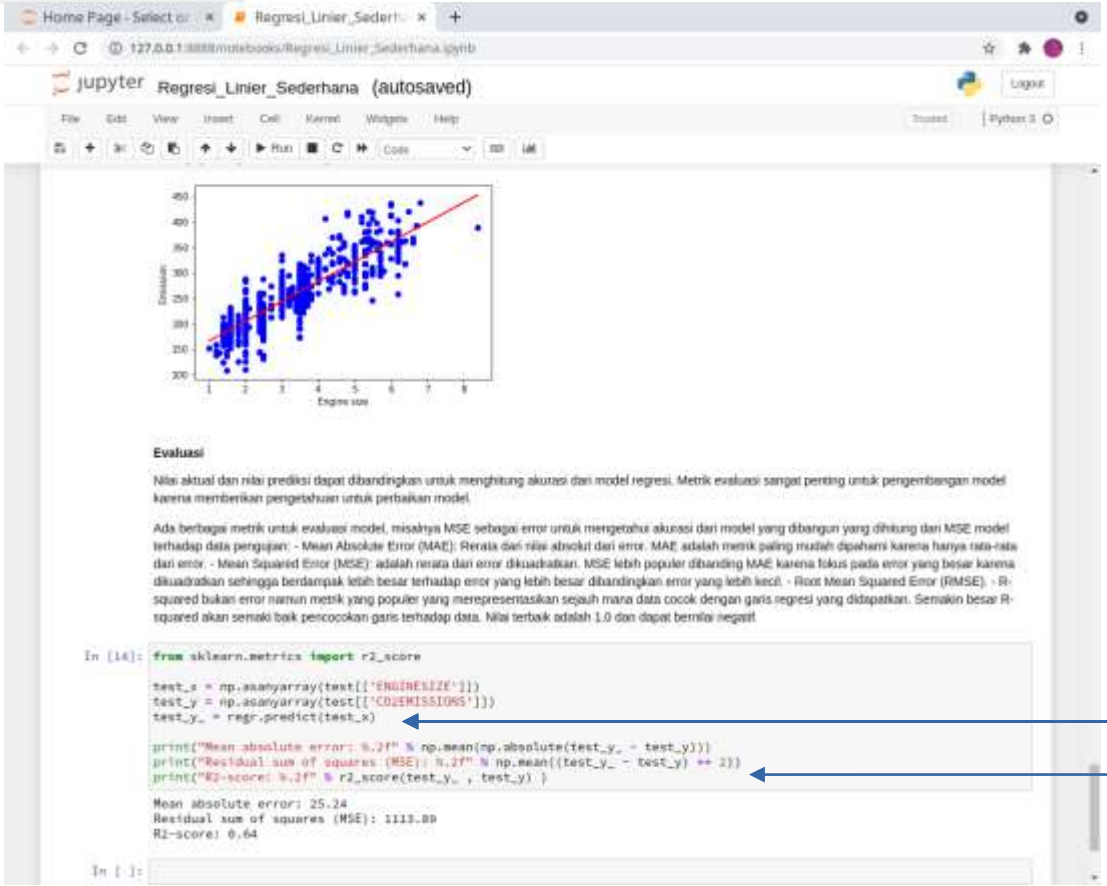
```
In [13]: plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

```
Out[13]: Text(0, 0.5, 'Emission')
```



Gambar garis persamaan tsb

Regresi Linear Sederhana (8)



Melakukan prediksi

Melakukan evaluasi

Regresi Linier Variabel Jamak

Contoh Regresi Linier Variabel Jamak

Efektivitas variabel-variabel bebas terhadap prediksi

- Apakah kegelisahan, kehadiran dosen, dan jenis kelamin mempunyai efek pada kinerja ujian mahasiswa?

Prediksi dampak perubahan

- Seberapa besar kenaikan/penurunan tekanan darah terhadap kenaikan/penurunan BMI dari pasien?

Prediksi Nilai Kontinu pada Regresi Linier Variabel Jamak

X: Independent variable Y: Dependent variable

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$Co2\ Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \ ...$$

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ ... + \theta_n x_n$$

$$\hat{y} = \theta^T X$$

$$\theta^T = [\theta_0, \theta_1, \theta_2, ...] \qquad X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ ... \end{bmatrix}$$

- Regresi linier variabel jamak menggunakan lebih dari satu variabel bebas antara lain ENGINESIZE, CYLINDERS, FUELCONSUMPTION_COMB
- Untuk memprediksi nilai kontinu variabel tak bebas dalam hal ini CO2EMISSION sebagaimana ditunjukkan pada Gambar.

MSE Untuk Menunjukkan Error Pada Model

$\hat{y} = \theta^T X$

$\hat{y}_i = 140$ the predicted emission of x_i

$y_i = 196$ actual value of x_i

$y_i - \hat{y}_i = 196 - 140 = 56$ residual error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267

- Regresi linier variabel jamak menggunakan MSE sebagai metrik kesalahan
- MSE adalah selisih antara hasil prediksi dengan nilai aktual variabel tak bebas sebagaimana ditunjukkan pada Gambar 10.

Estimasi Parameter Regresi Linier Variabel Jamak

Cara-cara mengestimasi parameter θ

Least Squares

- Operasi aljabar linier
- Perlu waktu yang lama untuk dataset yang besar (lebih dari 10000 baris)

Algoritma optimisasi

- Gradient Descent
- Metode yang sesuai apabila dataset sangat besar

Prediksi Menggunakan Regresi Linier Variabel

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$\hat{y} = \theta^T X$$

$$\theta^T = [125, 6.2, 14, \dots]$$

$$\hat{y} = 125 + 6.2x_1 + 14x_2 +$$

$$Co2Em = 125 + 6.2EngSize + 14\ \text{Cylinders} + \dots$$

$$Co2Em = 125 + 6.2 \times 2.4 + 14 \times 4 + \dots$$

$$Co2Em = 214.1$$

Gambar menunjukkan bagaimana prediksi nilai numerik CO2EMISSIONS berdasarkan variabel bebas jamak yaitu ENGINESIZE = 2,4 serta CYLINDERS = 4 dan FUELCONSUMPTION = 9,2 dengan hasil prediksi = 214,1 menggunakan parameter terbaik yang sudah didapatkan dari data latih.

Hands-On : Regresi Linear Variabel Jamak

```
import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
%matplotlib inline
```

→ Import library

```
df = pd.read_csv("FuelConsumptionCo2.csv")

# melihat dataset
df.head()
```

→ Load data

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUELTYPE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_H
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	Z	9.9	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	Z	11.2	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	Z	6.0	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	Z	12.7	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	Z	12.1	

→ Output

Hands-On : Regresi Linear Variabel Jamak

```
df.describe()
```

cek statistika dasar

	MODELYEAR	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_COMB	FUELCONSUMPTION_COM
count	1067.0	1067.000000	1067.000000	1067.000000	1067.000000	1067.000000	1067
mean	2014.0	3.346298	5.794752	13.296532	9.474602	11.580881	26
std	0.0	1.415895	1.797447	4.101253	2.794510	3.485595	7
min	2014.0	1.000000	3.000000	4.600000	4.900000	4.700000	11
25%	2014.0	2.000000	4.000000	10.250000	7.500000	9.000000	21
50%	2014.0	3.400000	6.000000	12.600000	8.800000	10.900000	26
75%	2014.0	4.300000	8.000000	15.550000	10.850000	13.350000	31
max	2014.0	8.400000	12.000000	30.200000	20.500000	25.800000	60

output

Hands-On : Regresi Linear Variabel Jamak

```
cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
cdf.head(10)
```

seleksi data dan tampilkan

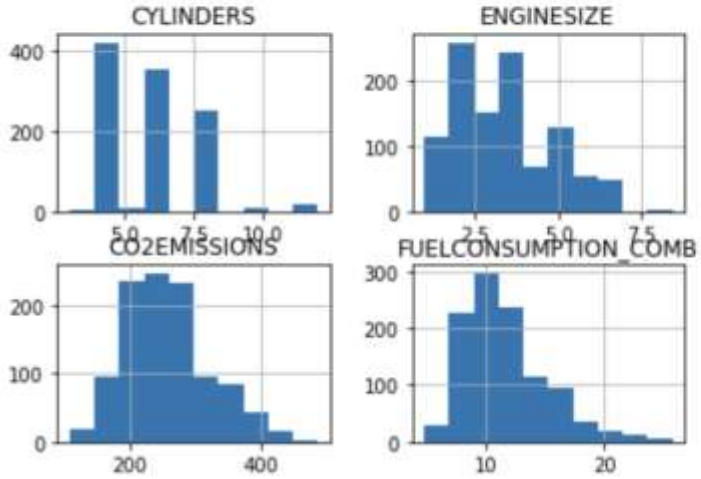
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	212

output

Hands-On : Regresi Linear Variabel Jamak

```
viz = cdf[['CYLINDERS','ENGINESIZE','CO2EMISSIONS','FUELCONSUMPTION_COMB']]
viz.hist()
plt.show()
```

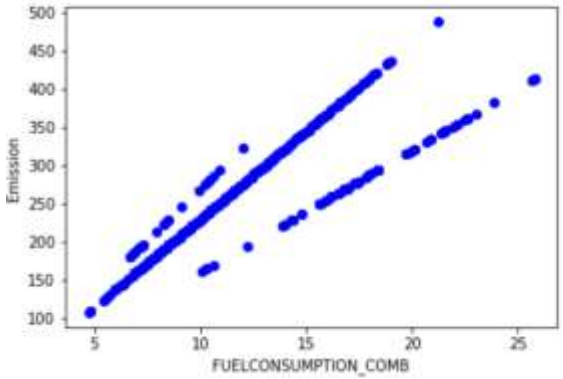
visualisasi data yang telah di seleksi



output

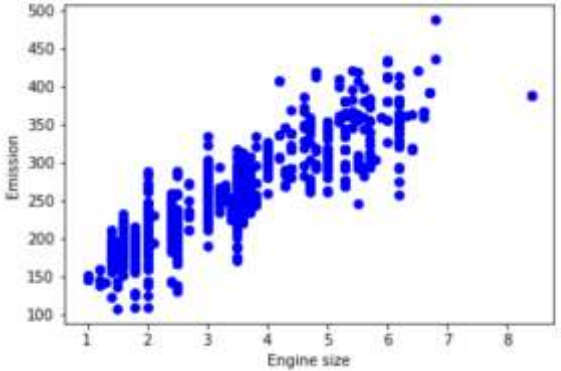
Hands-On : Regresi Linear Variabel Jamak

```
lt.scatter(cdf.FUELCONSUMPTION_COMB, cdf.CO2EMISSIONS,  color='blue')
lt.xlabel("FUELCONSUMPTION_COMB")
lt.ylabel("Emission")
lt.show()
```



Fuel Consumption vs Emssion

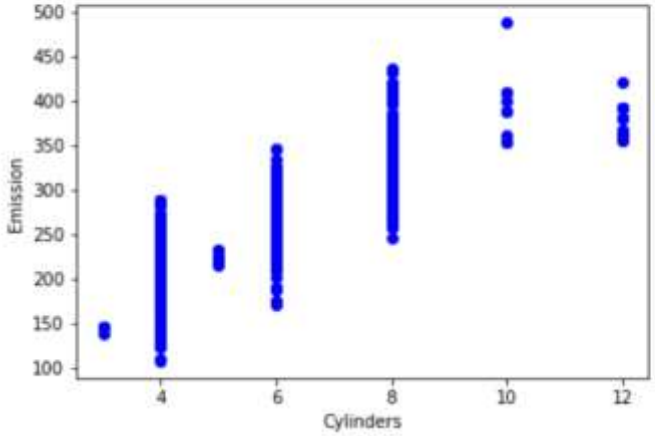
```
plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



Engine size vs Emssion

Hands-On : Regresi Linear Variabel Jamak

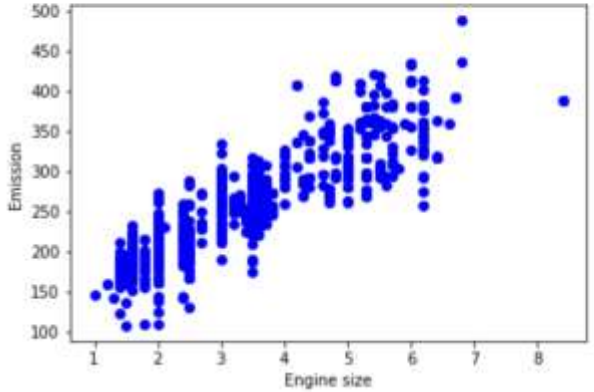
```
plt.scatter(cdf.CYLINDERS, cdf.CO2EMISSIONS,  color='blue')
plt.xlabel("Cylinders")
plt.ylabel("Emission")
plt.show()
```



Cylinders vs Emssion

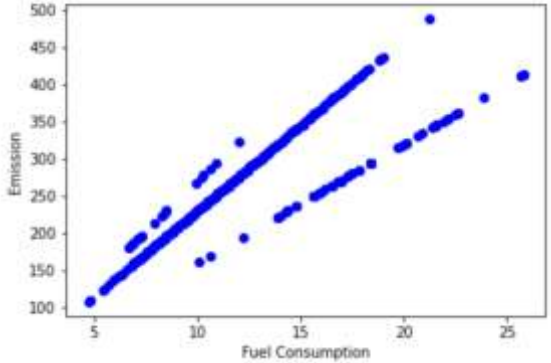
Hands-On : Regresi Linear Variabel Jamak

```
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='blue')
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



Train Engine size vs Emission

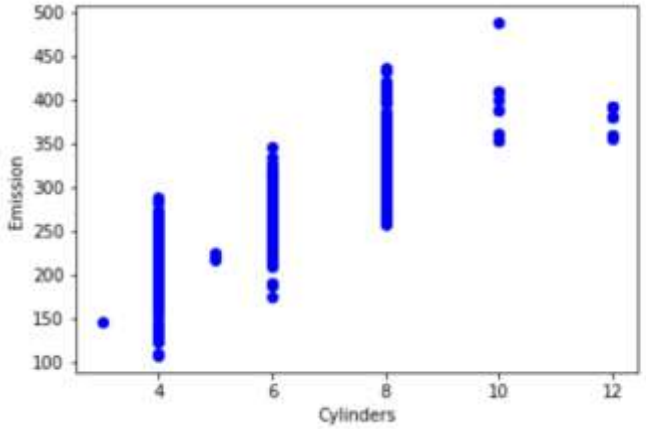
```
plt.scatter(train.FUELCONSUMPTION_COMB, train.CO2EMISSIONS,  color='blue')
plt.xlabel("Fuel Consumption")
plt.ylabel("Emission")
plt.show()
```



Train Fuel Consumpion vs Emission

Hands-On : Regresi Linear Variabel Jamak

```
plt.scatter(train.CYLINDERS, train.CO2EMISSIONS,  color='blue')
plt.xlabel("Cylinders")
plt.ylabel("Emission")
plt.show()
```



Train Cylinders vs Emission

Hands-On : Regresi Linear Variabel Jamak

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINESIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (train_x, train_y)
```

```
LinearRegression()
```

→ Melakukan pemodelan

```
# The coefficients
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```

```
Coefficients:  [[38.91748284]]
Intercept:  [126.2611611]
```

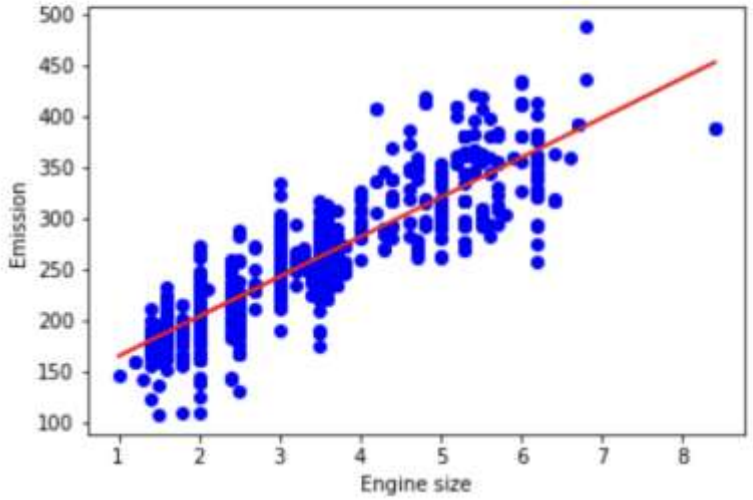
→ Menampilkan coeficients & intercept

Hands-On : Regresi Linear Variabel Jamak

```
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='blue')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-r')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

plotting garis engine vs emission

```
Text(0, 0.5, 'Emission')
```



output

Hands-On : Regresi Linear Variabel Jamak

```
from sklearn.metrics import r2_score

test_x = np.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_ = regr.predict(test_x)

print("Mean absolute error: %.2f" % np.mean(np.absolute(test_y_ - test_y)))
print("Residual sum of squares (MSE): %.2f" % np.mean((test_y_ - test_y) ** 2))
print("R2-score: %.2f" % r2_score(test_y_ , test_y) )
```

menampilkan RAE, MSE, dan R2-Score

```
Mean absolute error: 22.26
Residual sum of squares (MSE): 910.64
R2-score: 0.69
```

output

Lab

- Jalankan file Jupyter Notebook untuk Regresi Linier Variabel Jamak

Tools / Lab Online

- Scikit-Learn
- Jupyter Notebook
- Conda / Anaconda
- Google Colaboratory
- PyPI (pip)

Referensi

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.
- Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F. and Mueller, A., 2015. Scikit-learn: Machine learning without learning the machinery. *GetMobile: Mobile Computing and Communications*, *19*(1), pp.29-33.
- https://scikit-learn.org/stable/index.html

Terima kasih