DigitalSkola

LEARNING PROGRESS REVIEW

Week 10

Diaz Jubairy - Hermulia Hadie
Desi Sulistyowati - Farahul Jannah

DigitalSkola

Table of Content

DigitalSkola

1.

Advanced Visualization

Seaborn, Folium, WordCloud

Seaborn

Seaborn adalah salah satu library Python yang berguna menciptakan visualisasi data statistik dengan tampilan yang berkualitas tinggi. Library ini di bangun berdasarkan library matplotlib serta terintegrasi dengan struktur data pada Pandas. Secara sederhana, Seaborn adalah ekstensi dari Matplotlib


```
import seaborn as sns
```

Seaborn vs Matplotlib

Seaborn

- Mempunyai berbagai macam plot dan tema untuk visualisasi data
- Menggunakan syntax yang sederhana
- Bekerja dengan Pandas dataframe
- Menghindari tumpang tindih plot dengan bantuan tema defaultnya.
- Memakai banyak memori.
- Tidak ada pie chart

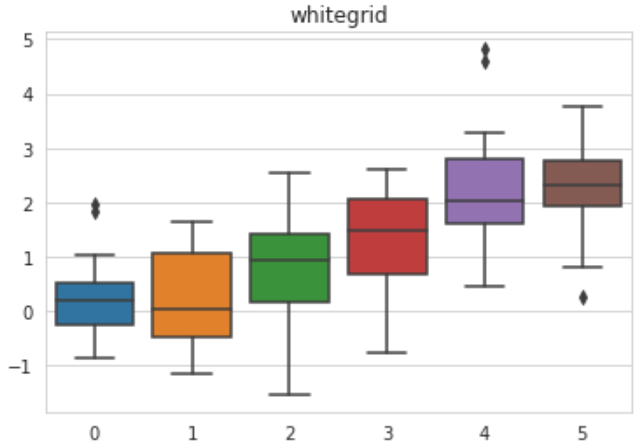
Matplotlib

- Digunakan untuk membuat grafik dasar.
- Menggunakan syntax yang kompleks dan panjang
- Bekerja dengan Numpy dan Pandas.
- Dapat disesuaikan secara personal.

Mengatur Background Grafik

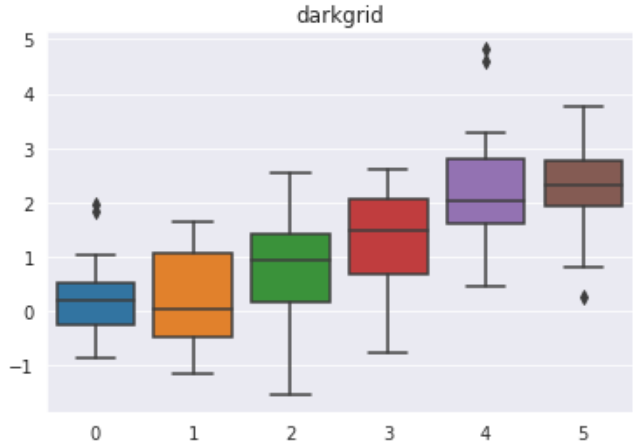
- White grid : background warna putih dengan garis(grid)

```
sns.set_style("whitegrid")
```



- Dark grid : background warna abu-abu dengan garis(grid)

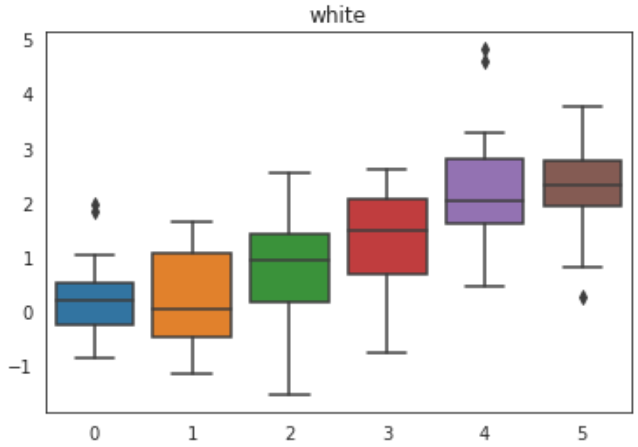
```
sns.set_style("darkgrid")
```



Mengatur Background Grafik

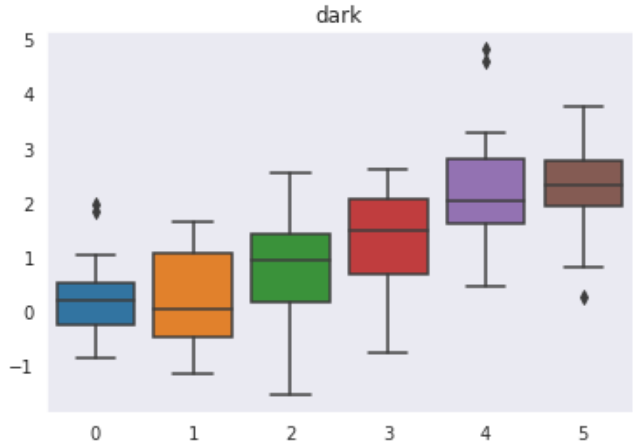
- White : warna background putih tanpa garis (grid)

```
sns.set_style("white")
```



- Dark : warna background abu-abu tanpa garis (grid)

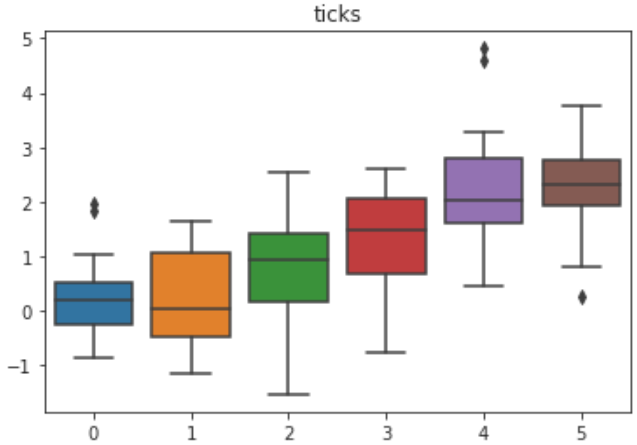
```
sns.set_style("dark")
```



Mengatur Background Grafik

- Ticks : warna background putih dan di setiap axis ada tanda

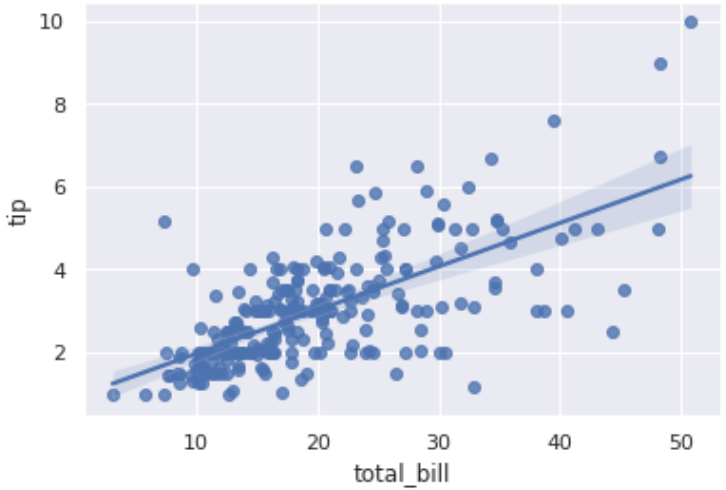
```
sns.set_style("ticks")
```



Regplot

sns.regplot() untuk menampilkan grafik yang menggambarkan informasi tentang sebaran dan hubungan regresi data

```
ax = sns.regplot(x="total_bill", y="tip", data=tips)
```

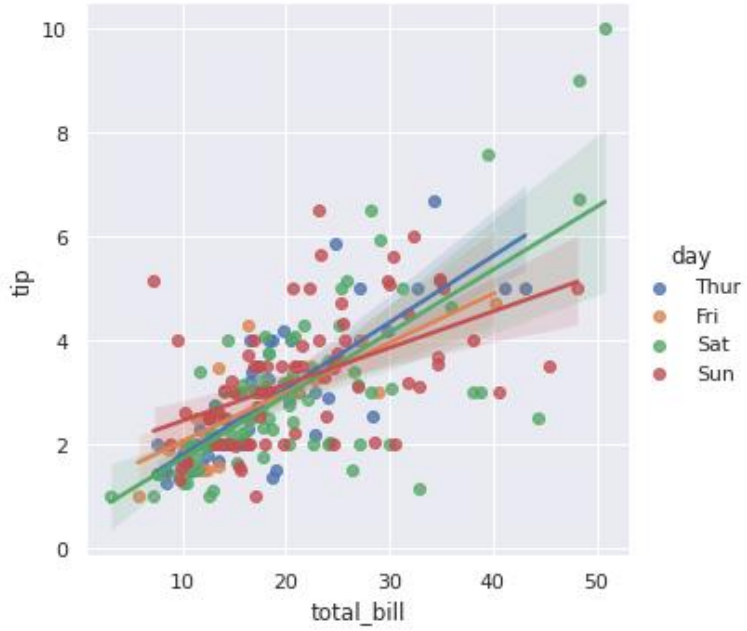


DigitalSkola

Lmplot

Apabila regplot hanya bisa menampilkan sebaran dan hubungan regresi data secara sederhana. Lmplot bisa menampilkan sebaran dan hubungan regresi data dengan mengacu pada variabel pembeda.

```
g = sns.lmplot(
    x="total_bill",
    y="tip",
    hue="day",
    data=tips)
```

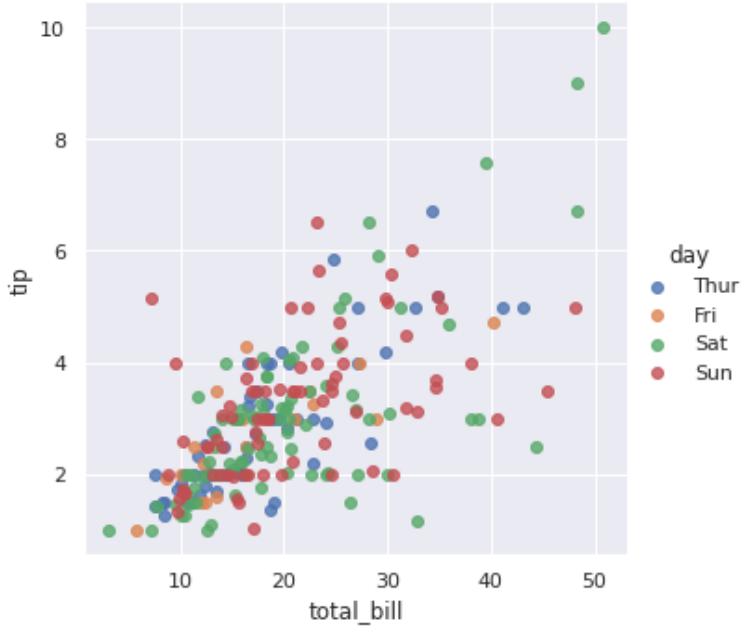


DigitalSkola

Lmplot

Apabila hanya ingin menampilkan scatter plotnya saja, maka bisa ditambah dengan fungsi "fit_reg=False"

```
ax = sns.lmplot(x="total_bill",
                y="tip", data=tips,
                fit_reg=False,
                hue='day')
```

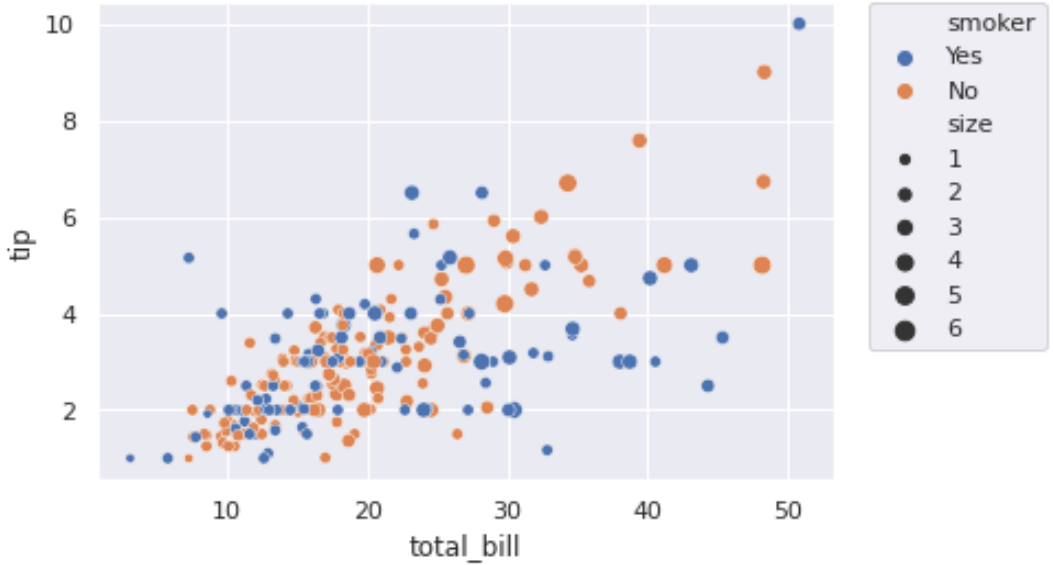


DigitalSkola

Scatterplot

Scatterplot tidak hanya bisa ditampilkan dengan syntax lmplot, tapi juga bisa menggunakan syntax "sns.scatterplot()"

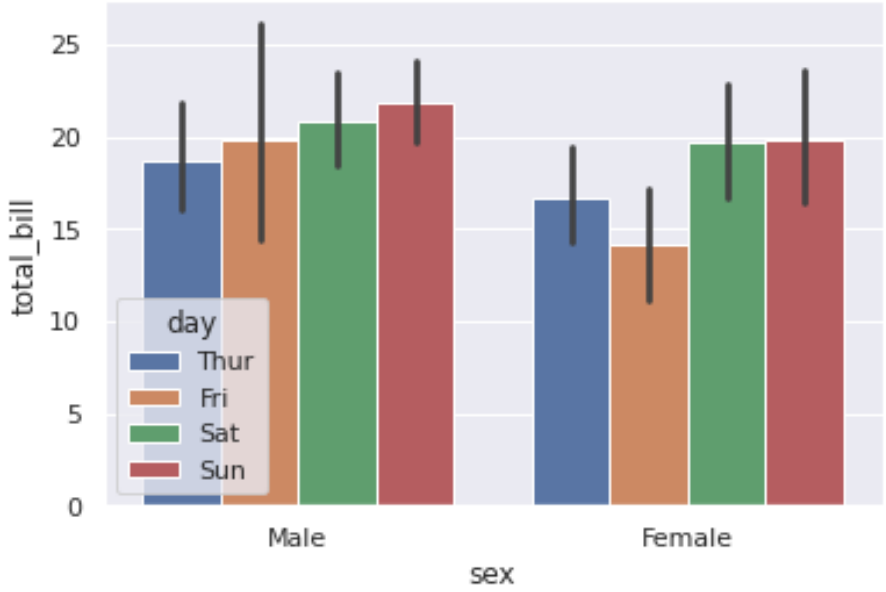
```
ax = sns.scatterplot(x="total_bill",
                     y="tip",
                     data=tips,
                     size="size",
                     hue='smoker')

# combine matplotlib with seaborn
import matplotlib.pyplot as plt

# mengatur legend di luar scatter plot-nya
plt.legend(bbox_to_anchor=(1.05, 1),
           loc=2,
           borderaxespad=0.)
```



DigitalSkola

Barchart

```
ax = sns.barplot(x="sex", y="total_bill", hue="day", data=tips)
```



DigitalSkola

Catplot

Catplot dapat digunakan untuk menunjukkan hubungan antara variabel numerik dan satu atau lebih variabel kategorik menggunakan salah satu dari beberapa representasi visual. Beberapa jenis plot yang bisa digunakan dalam catplot adalah:

Categorical scatterplots

- stripplot() (with kind = "strip"; ini adalah default dari catplot)
- swarmplot() (with kind ="swarm")

Categorical distribution plots

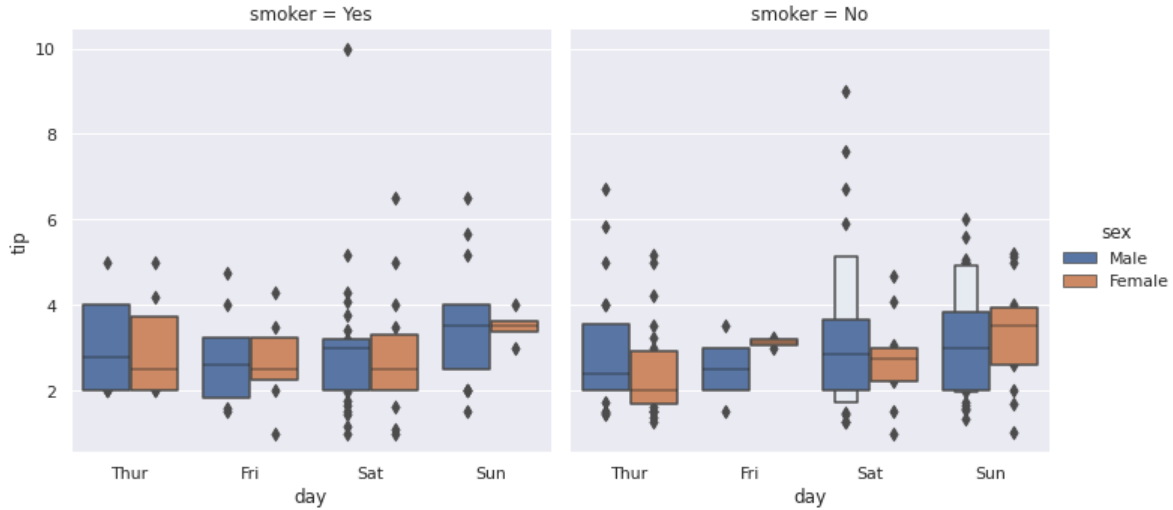
- boxplot() (with kind = "box")
- violinplot() (with kind = "violin"
- boxenplot() (with kind = "boxen")

Categorical estimate plot

- poinplot() (with kind = "point"
- barplot() (with kind = "bar"
- countplot() (with kind = "count)

Catplot

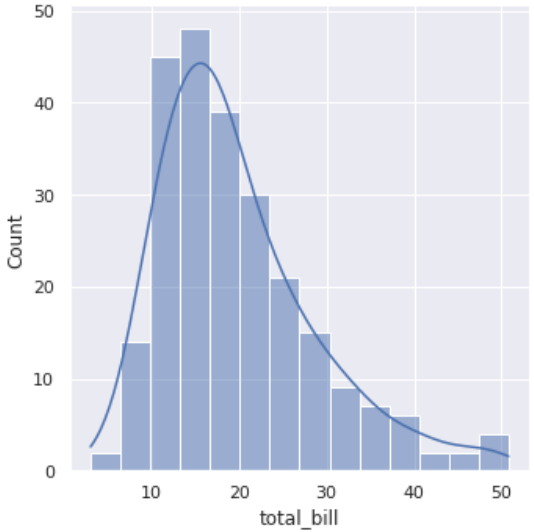
```
g = sns.catplot(x="day", y="tip", hue="sex", col='smoker', kind='boxen', data=tips)
```



DigitalSkola

Histogram

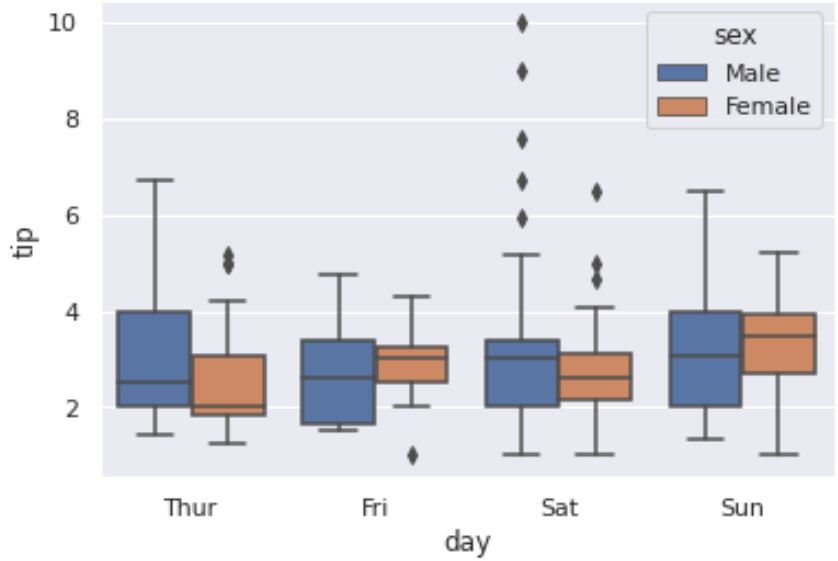
```
g = sns.displot(tips.total_bill, kde=True)
```



DigitalSkola

Boxplot

```
g = sns.boxplot(x="day", y="tip", hue='sex', data=tips)
```

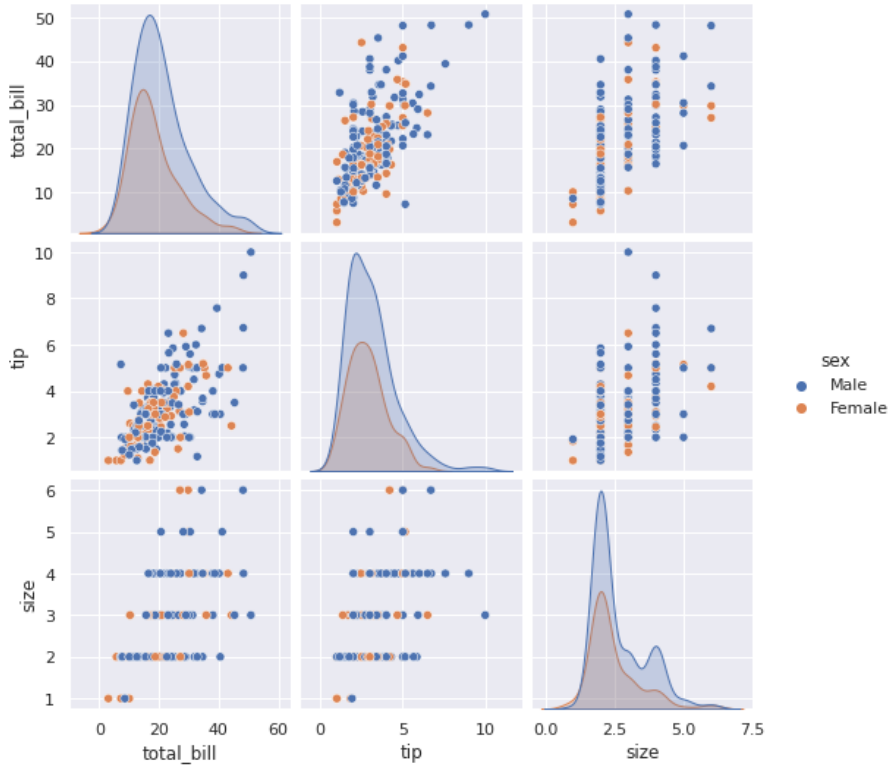


DigitalSkola

Scatterplot Matrix

Menampilkan seluruh hubungan antar variabel yang mempunyai data numerik

```
g = sns.pairplot(tips, hue="sex")
```

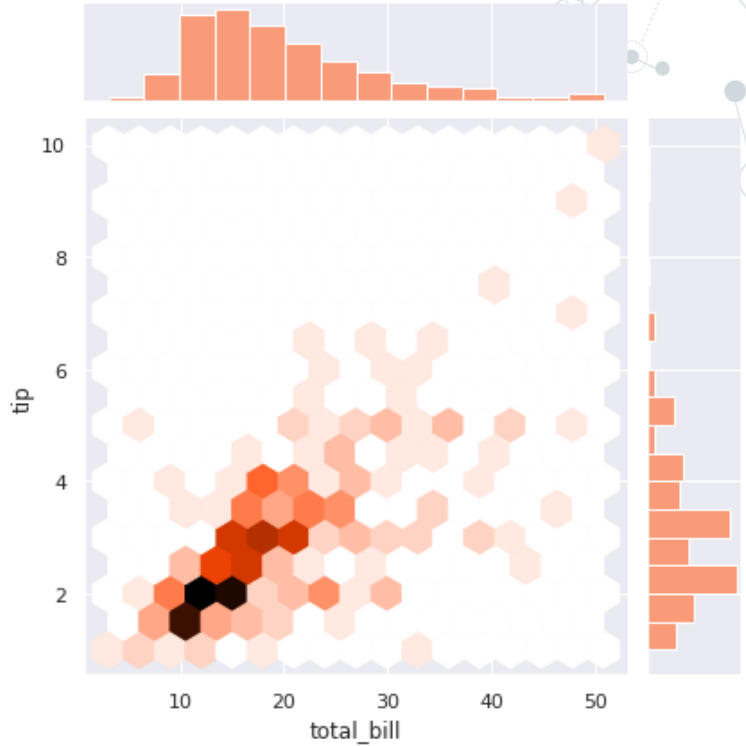


Joinplot

Menampilkan plot dari dua variabel dengan grafik bivariat dan univariat.
Joinplot juga mempunyai beberapa kind, yaitu:

- scatter
- kde
- hist
- hex
- reg
- resid

```
g = sns.jointplot(x="total_bill",
                  y="tip", data=tips,
                  kind="hex",
                  color="coral")
```



Folium

Folium memudahkan untuk memvisualisasikan data yang telah dimanipulasi dengan Python pada peta interaktif. Folium memungkinkan untuk mengaplikasikan data ke peta untuk visualisasi choropleth serta meneruskan visualisasi vektor/raster/HTML yang kaya sebagai penanda pada peta.


```
import folium
```

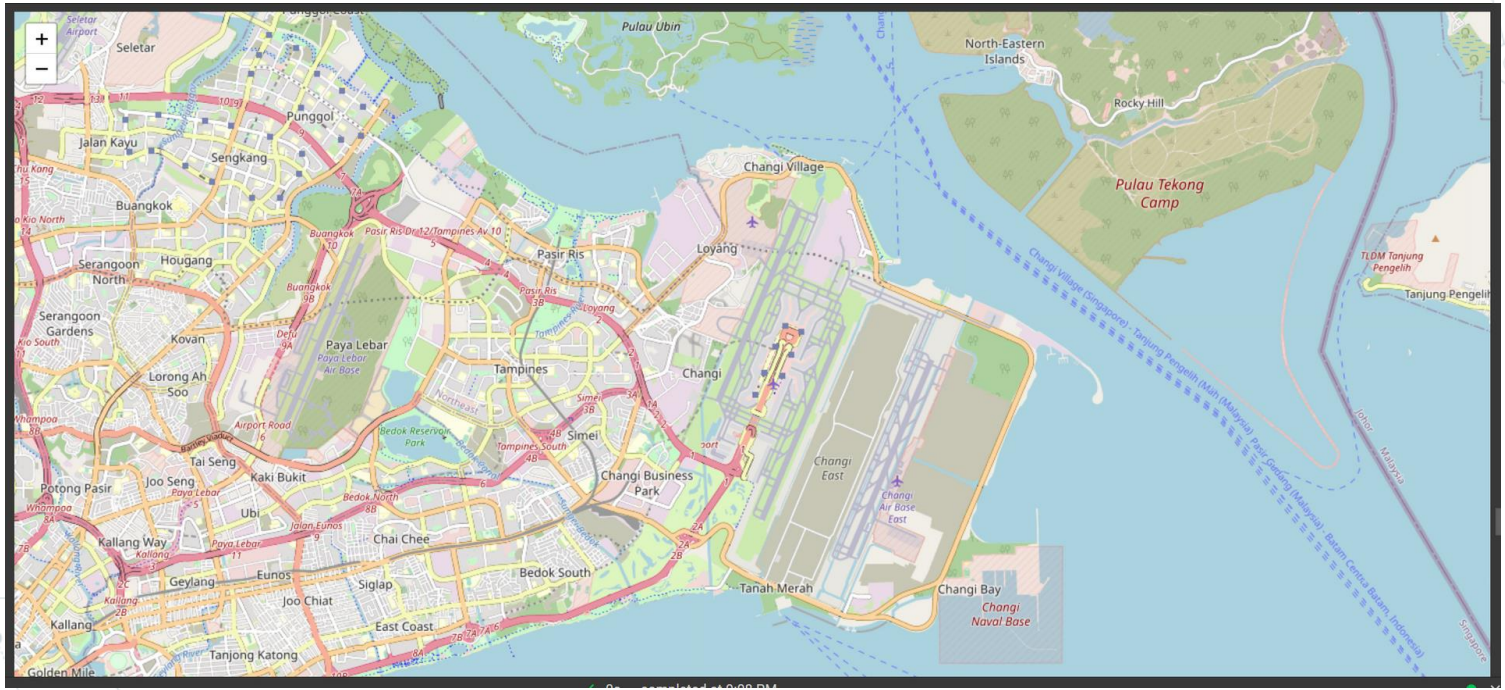
Folium

Syntax

```
#!pip install folium

import folium
m=folium.Map(location=[1.3396704621219861, 103.98355242289945],
             # diisi dengan titik koordinat tempat yang ingin ditampilkan
             zoom_start=13)
m
```

Folium

Result



WordCloud

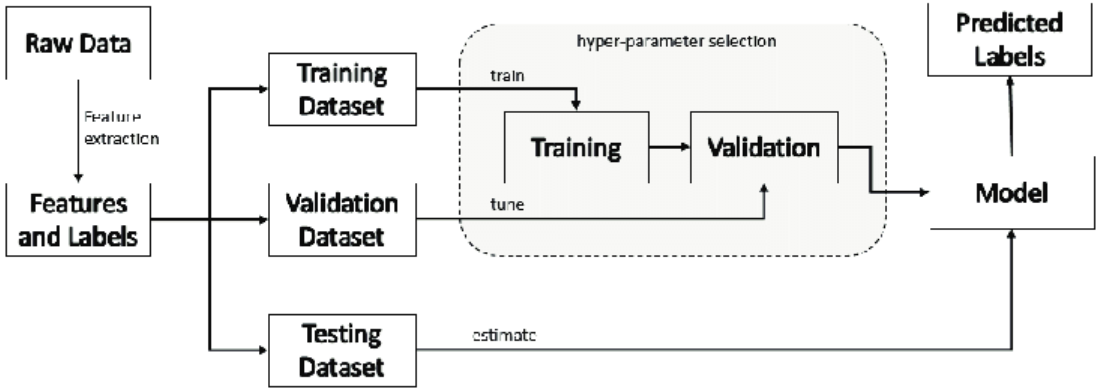
Word Cloud adalah teknik visualisasi data yang digunakan untuk merepresentasikan
data teks di mana ukuran setiap kata menunjukkan frekuensi atau kepentingannya. Word cloud banyak digunakan untuk enganalisis data dari situs web jejaring sosial.

```
import wordcloud
```

2. Introduction to Machine Learning

What is Machine Learning

- Cabang kecerdasan buatan (Artificial Intelligence/ AI), yang berkaitan dengan desain dan pengembangan algoritma yang memungkinkan komputer mengembangkan perilaku berdasarkan data empiris.
- Karena kecerdasan membutuhkan pengetahuan, maka komputer perlu memperoleh pengetahuan.



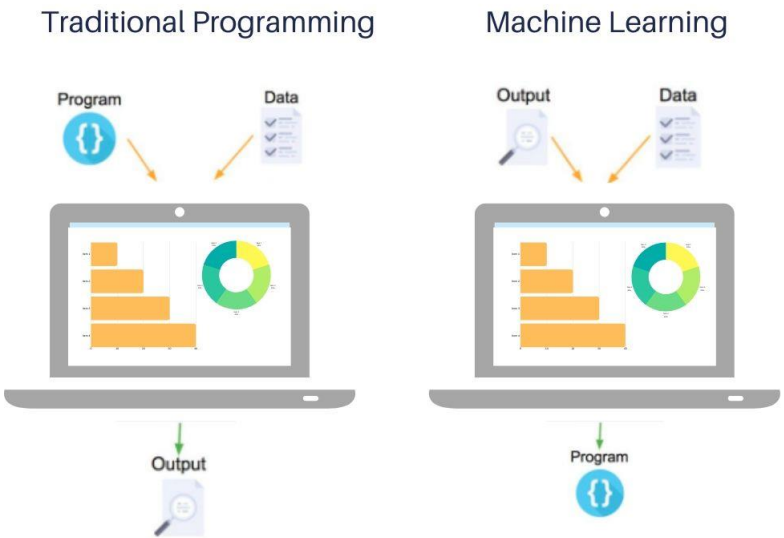
DigitalSkola

Why “Learn”?

- Machine learning adalah pemrograman komputer untuk mengoptimalkan kinerja menggunakan contoh data atau pengalaman masa lalu.
- Machine Learning digunakan ketika:
  - Keahlian manusia tidak ada (navigating on Mars)
  - Manusia tidak mampu menjelaskan keahliannya (speech recognition)
  - Solusi yg perlu disesuaikan dengan kasus tertentu (user biometrics)

ML vs Traditional Programming

- Traditional programming adalah proses manual—artinya seseorang (programmer) membuat program yang didalamnya terdapat aturan-aturan.
- Sedangkan di machine learning, algoritma secara otomatis merumuskan aturan (rules) dari data.



DigitalSkola

Types of Learning

1. Supervised learning

- Training data mempunyai target class

- Classification, regression/ prediction

2. Unsupervised learning

- Training data tidak mempunyai target class

- Clustering, association rules

3. Semi-supervised learning

Sebagian training data memiliki output

4. Reinforcement learning

Rewards (hadiah) diberikan ketika agent sukses dalam tugas tertentu, dan dihukum (punishment) ketika salah.

Supervised learning

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Unsupervised learning

Tid	Attrib1	Attrib2	Attrib3
1	Yes	Large	125K
2	No	Medium	100K
3	No	Small	70K
4	Yes	Medium	120K
5	No	Large	95K
6	No	Medium	60K
7	Yes	Large	220K
8	No	Small	85K
9	No	Medium	75K
10	No	Small	90K

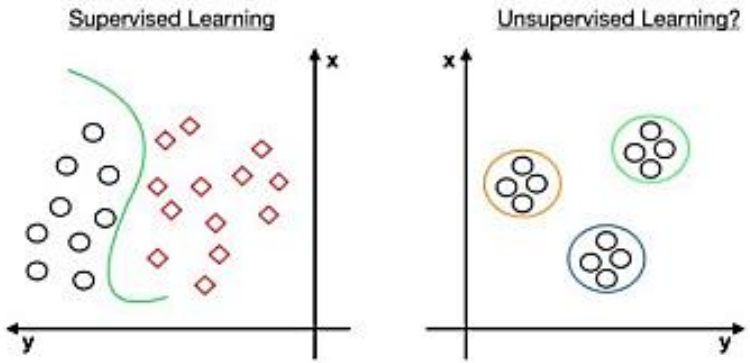
Semi-supervised learning

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	
3	No	Small	70K	
4	Yes	Medium	120K	
5	No	Large	95K	
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	
10	No	Small	90K	Yes

Supervised vs Unsupervised Learning

Supervised = Mempelajari data untuk memprediksi output. Kita tahu target label, sehingga kita membuat model untuk memprediksi label.

Unsupervised = Menemukan pattern/ characteristic dari data. Kita tidak mengetahui target label, sehingga kita membuat model yang mencoba mengelompokkan data.



DigitalSkola

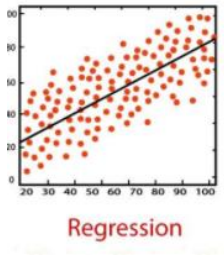
Supervised Learning

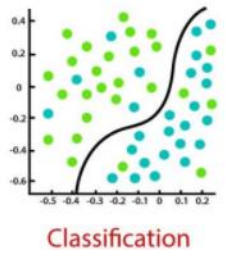
- Classification (klasifikasi) = metode yang menarik beberapa kesimpulan dari nilai input yangdiberikan pada saat training dan kemudian akan memprediksi label/kelas untuk data baru.
- Regression (regresi) = metode yang mencoba untuk menentukan kekuatan dan karakter hubungan antara satu variabel dependen dan serangkaian variabel lainnya (variabel independen).
- Algoritma regresi = nilai kontinu (seperti harga, gaji, usia, dll). § Algoritma klasifikasi = nilai diskrit (seperti stroke atau normal, spam atau bukan spam, dll)
- Both are supervised learning

Classification, regression, clustering

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built
221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955
538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951
180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933
604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965
510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987

Regression (house price dataset)





	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

Classification (stroke dataset)

	ID	Sex	Marital status	Age	Education	Income	Occupation
0	100000001	0	0	67	2	124670	1
1	100000002	1	1	22	1	150773	1
2	100000003	0	0	49	1	89210	0
3	100000004	0	0	45	1	171565	1
4	100000005	0	0	53	1	149031	1

Clustering (customer dataset)

DigitalSkola

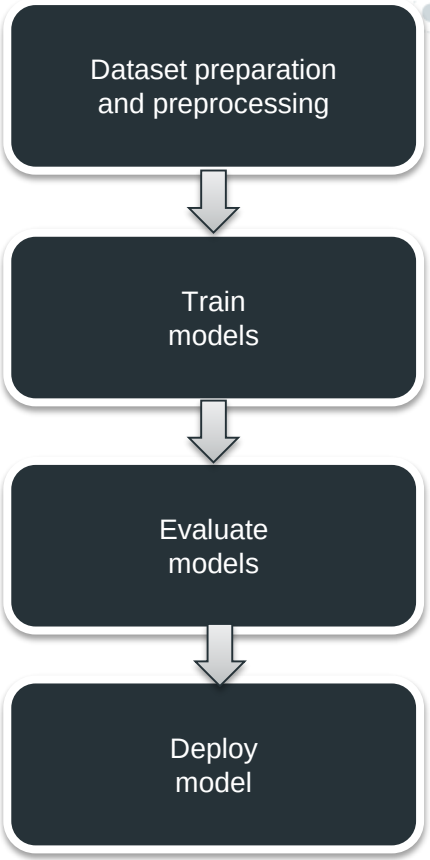
Stage in Machine Learning

Data preprocessing
Data cleaning, filling missing value, remove outlier

Train models
Select the algorithm
Feature selection and extraction

Evaluate model
Assess performance
Model comparison

Deploy model
Apply model to new data
Real-time demonstration



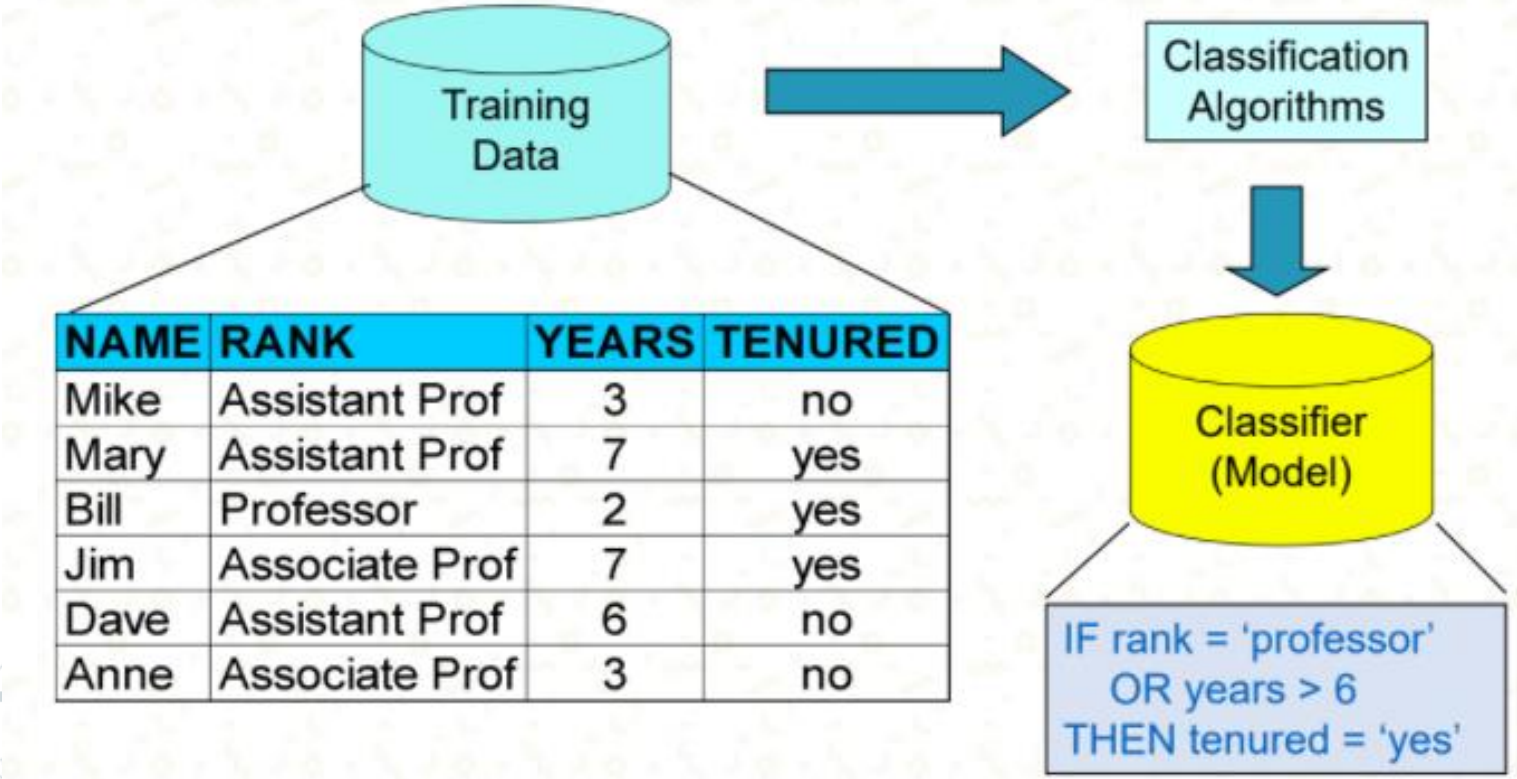
Why Data Preprocessing?

- Data di dunia nyata itu umumunya tidak bersih, banyak hal-hal yang perlu ditanggulangi seperti :
  - Data hilang:
    e.g., occupation=“ ”
  - Noisy: mengandung pencilan
    e.g., Salary=“-10” codes or names
  - Inkonsisten: ketodakasamaan format
    e.g., sex=“Girl” vs. sex=“Female”

Sex	Age	BMI	DM type	DM duration	FBS	Sys BP	Dias BP	Retinopathy
Male	65	25	II	20	129	130	80	Yes
Male	42	27	II	300	210	140	90	No
Female	31	21	I	11	164	145	80	Yes
Male	70	32	II	29	208	160	100	Yes
Female	54	34	II	6	183	155	95	No
	46	29	II	7	198	160	100	No
Female	16	24	I	-1	250	135	80	No
Male	67	30	II	12	243	165	90	Yes
Female	51	28	II	7	163	130	85	No
Girl	70	36	II	20	250	150	90	Yes
Female	63	35	II	14	203	160	110	No
Male	44	39	II	3	149	140	90	No
Boy	51	24	II	9	160	155	80	No
Male	27	19	I	5	170	140	90	No

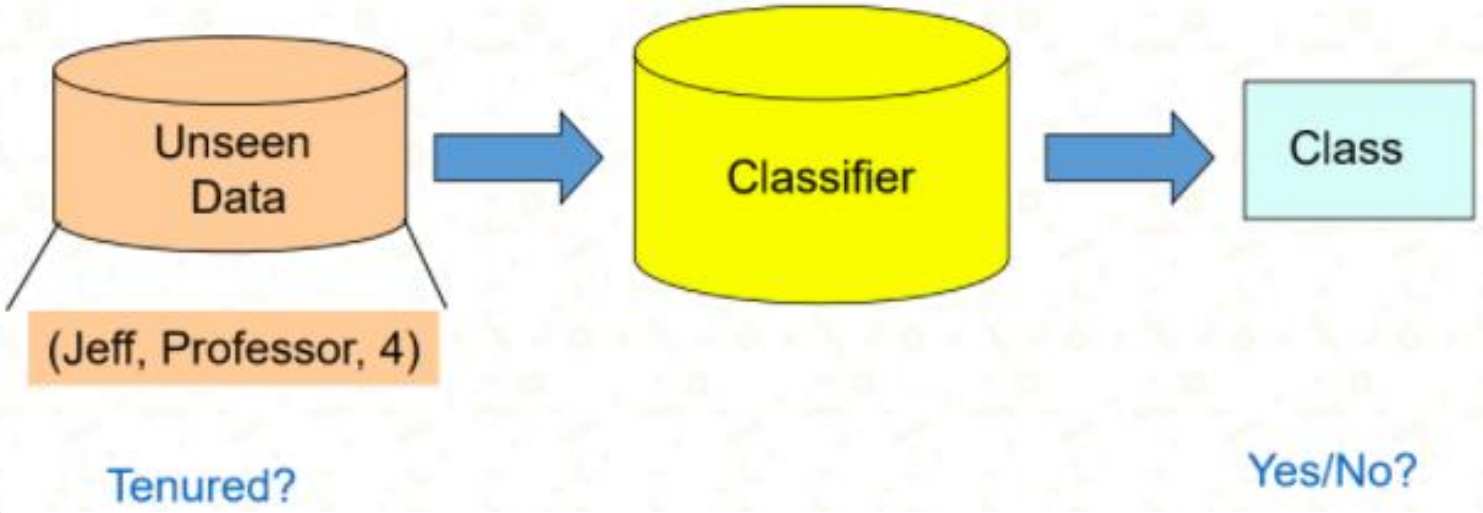
- No quality data, no quality mining results!
  - Keputusan yang berkualitas harus didasarkan pada data yang berkualitas

Model construction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Use the Model in Prediction



Bias and variance

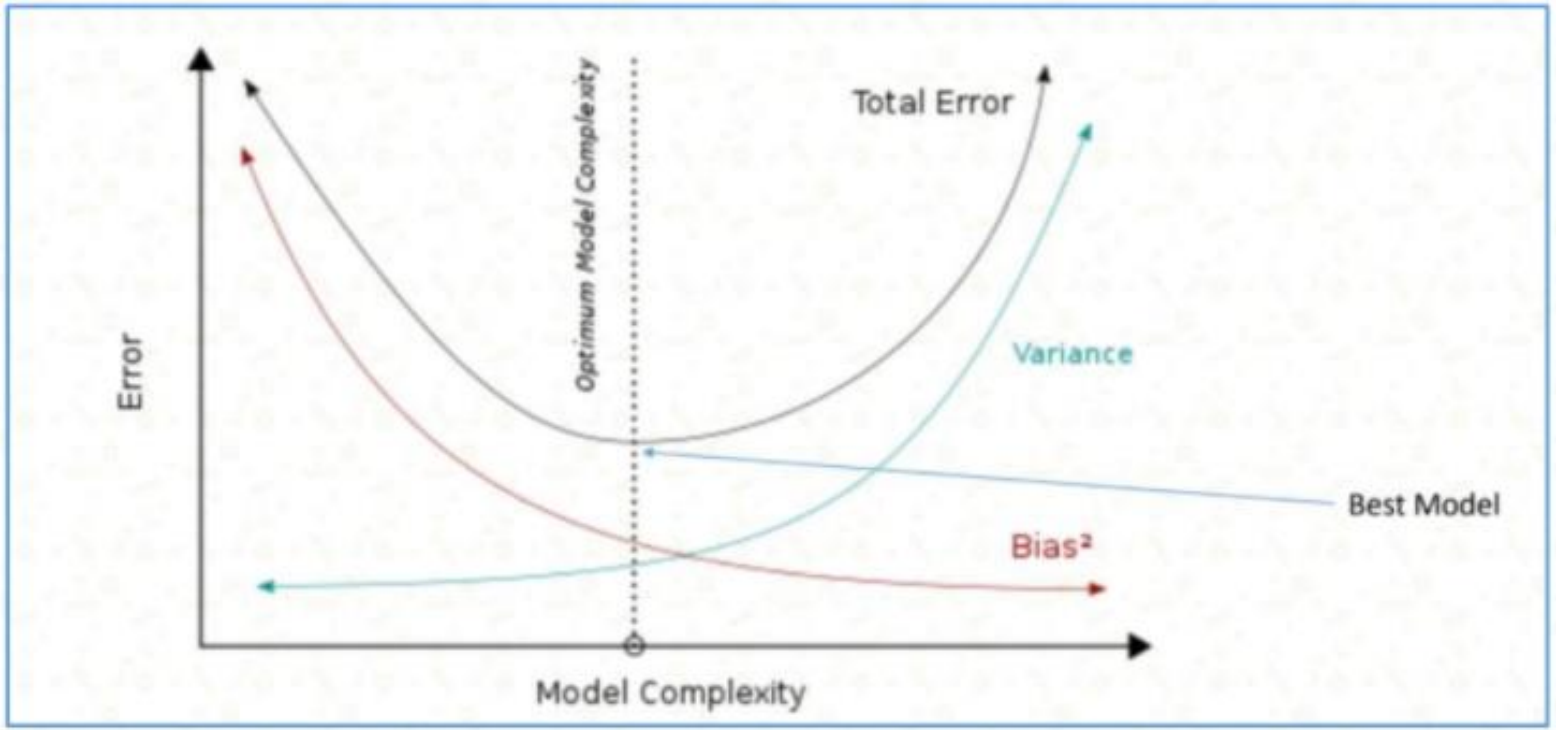
Bias

- Bias adalah perbedaan antara rata rata hasil prediksi dari model ML yang kita develop dengan data nilai yang sebenarnya.
- Bias yang tinggi dikarenakan dalam pembangunan model ML, dilakukan terlalu sederhana (oversimplified).

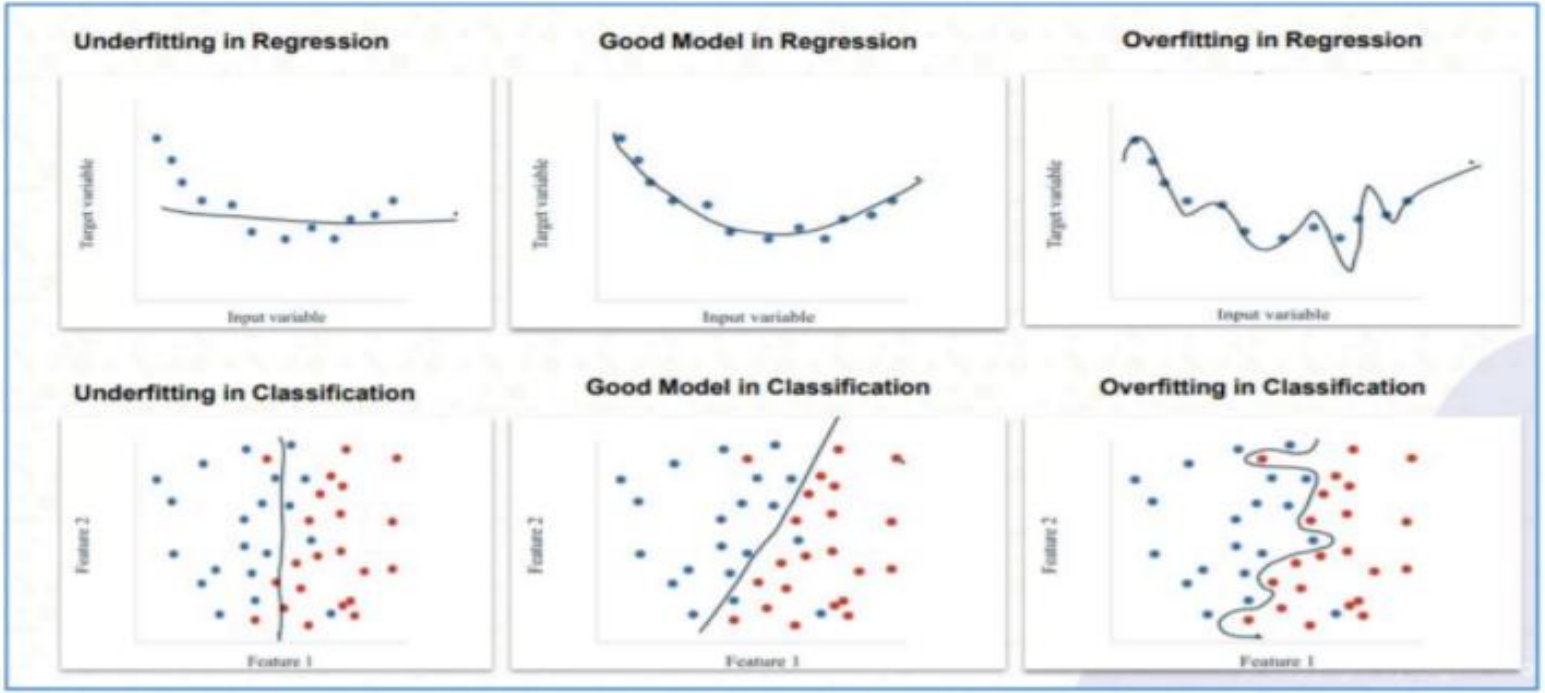
Variance

- Variance adalah variabel dari prediksi yang memberikan kita informasi perserbaran data hasil prediksi.
- Model yang memiliki variance tinggi memiliki korelasi kuat hanya pada training set, sehingga akan berkinerja baik pada training data saja.

Bias variance tradeoff



Underfitting and overfitting



Linear Regression

Membentuk hubungan antara dua variabel menggunakan garis lurus.

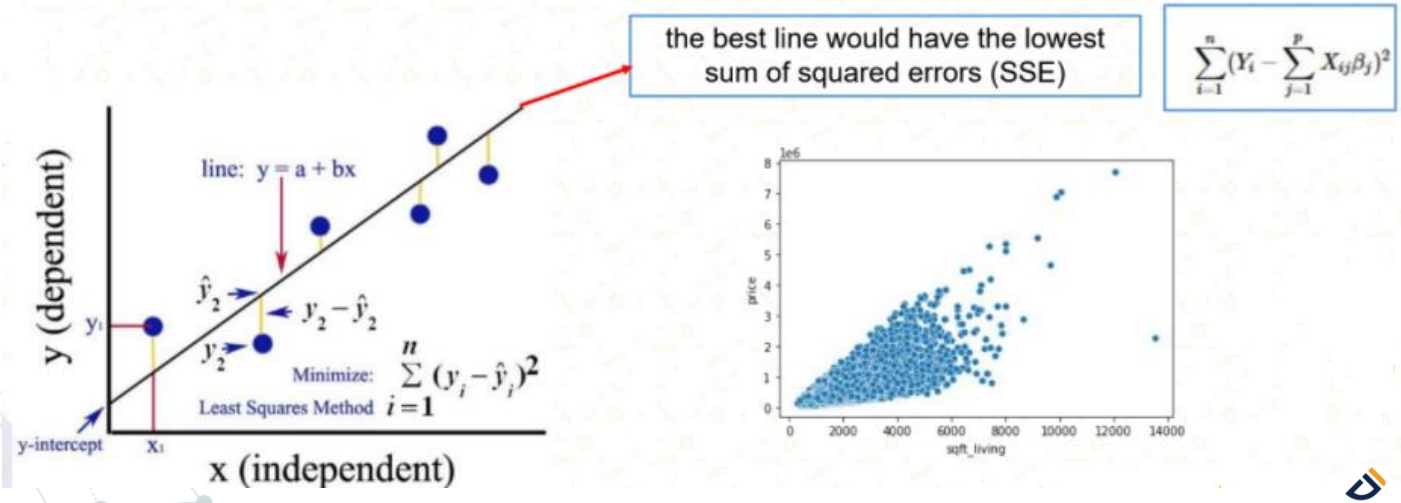
- **Simple linear regression:** $Y = a + bX + u$
- **Multiple linear regression:** $Y = a + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_tX_t + u$

Where:

- Y = the variable that you are trying to predict (dependent variable).
- X = the variable that you are using to predict Y (independent variable).
- a = the intercept.
- b = the slope.
- u = the regression residual.

Linear Regression

- Regresi linier mencoba menggambar garis yang paling dekat dengan data dengan menemukan slope dan intercept dan meminimalkan regression errors.
- Ordinary Least Squares (OLS) adalah metode estimasi yang paling umum untuk model linier

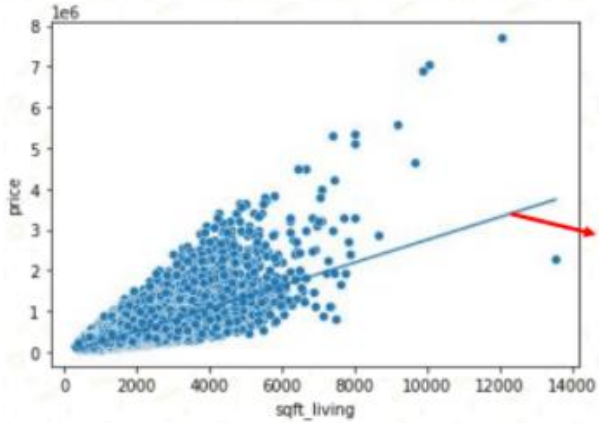


DigitalSkola

Linear Regression

Example

- y (dependent variable) = price (house price)
- x (independent variable) = sqft_living (square feet)

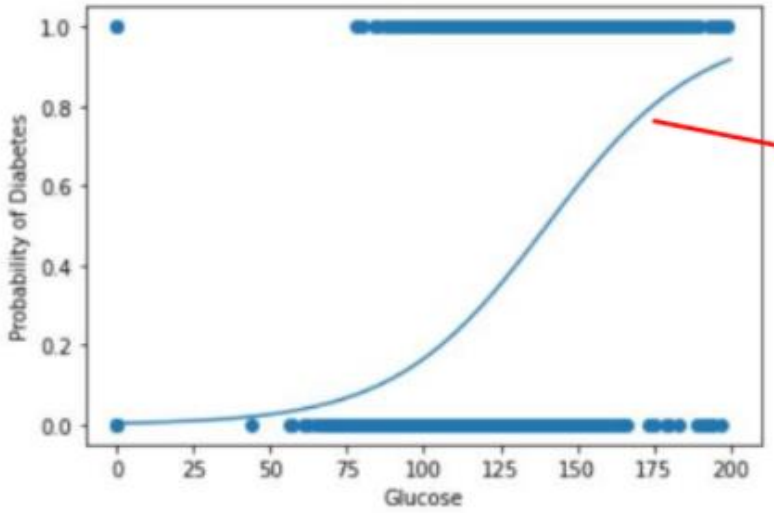


price = 279.51011741*sqft_living + -41947.45401876257

Q = House with 1000 square feet, approximate price?
A = USD 237562.663

DigitalSkola

Logistic Regression




```
p = 1/(1 + np.exp
        (-(0.04033676*x -5.6523997)))
```

Q = Patient with BG 190 mg/dL, is it diagnosed as diabetes?
A = Probability diabetes is 0.882

DigitalSkola

Logistic Regression

```
#hold out, dibagi menjadi training dan testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

#scaling
scaler = StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

# data preprocessing selesai

#mulai melakukan modelling. model ML learning dari training set
model=LogisticRegression()
model.fit(X_train, y_train)

# membuat prediksi
y_pred = model.predict(X_test)

#menghitung performa model, dengan accuracy dll
print('Accuracy ',accuracy_score(y_test, y_pred))
print('Precision ',precision_score(y_test, y_pred, average='macro'))
print('Recall ',recall_score(y_test, y_pred, average='macro'))
print('Confusion matrix ', confusion_matrix(y_test, y_pred))
plot_confusion_matrix(model, X_test, y_test, cmap=plt.cm.Blues)
plt.show()
```

3.

Data Preprocessing for Machine Learning

Features/Variable

- Fitur adalah properti terukur dari objek yang kita coba analisis.
- Fitur muncul sebagai kolom dalam table.

Sex	Age	BMI	DM type	DM duration	FBS	Sys BP	Dias BP	Retinopathy
Male	65	25	II	20	129	130	80	Yes
Male	42	27	II	300	210	140	90	No
Female	31	21	I	11	164	145	80	Yes
Male	70	32	II	29	208	160	100	Yes
Female	54	34	II	6	183	155	95	No
	46	29	II	7	198	160	100	No
Female	16	24	I	-1	250	135	80	No
Male	67	30	II	12	243	165	90	Yes
Female	51	28	II	7	163	130	85	No
Girl	70	36	II	20	250	150	90	Yes
Female	63	35	II	14	203	160	110	No
Male	44	39	II	3	149	140	90	No
Boy	51	24	II	9	160	155	80	No
Male	27	19	I	5	170	140	90	No

- Kualitas fitur berdampak besar pada kualiatas wawasan (insight) yang diperoleh saat pemodelan ML

Jenis-jenis Fitur

Jenis Fitur Kategoris

- Fitur Nominal: skala data yang berfungsi membedakan dan tidak ada tingkatan diantaranya. Contoh: gender, warna rambut, warna mata.
- Fitur Ordinal: data dikelompokan menjadi orde atau tingkatan. Contoh: jenjang pendidikan, kepuasan pelanggan.

Jenis Fitur Numerik

- Fitur Discrete: data diskrit mewakili item yang dapat dihitung. Contoh: jumlah siswa, jumlah kendaraan, dll
- Fitur Continuous: data kontinu mewakili item yang dapat diukur. Contoh: tinggi, suhu, kecepatan.

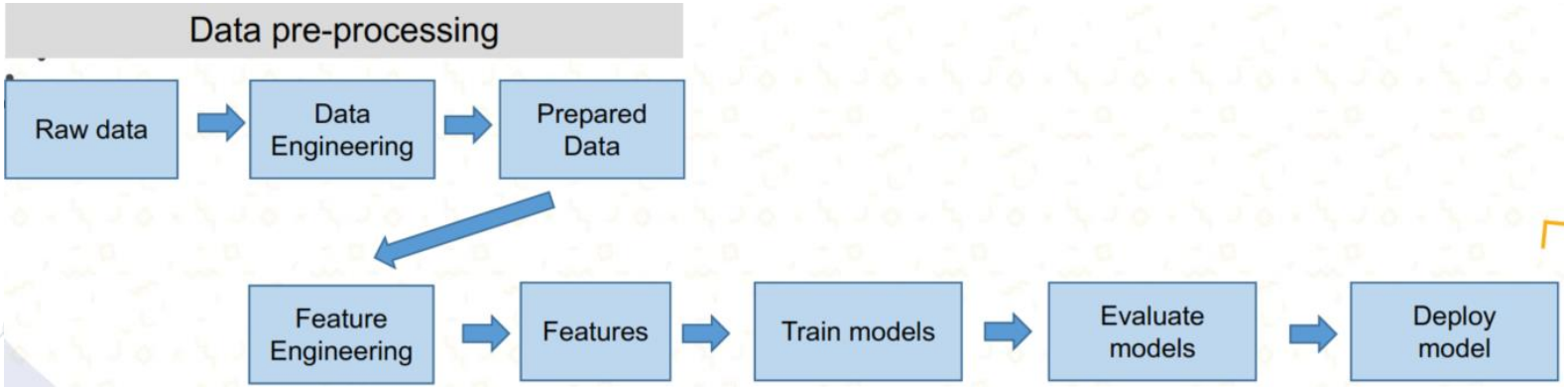
Data Preprocessing

- *Data preprocessing* merupakan sekumpulan teknik yang diterapkan pada dataset untuk menghapus noise, meng-handle missing value, dan data yang tidak konsisten.
- *Data preprocessing* diperlukan karena data mentah seringkali tidak lengkap dan memiliki format yang tidak konsisten.

- Data cleaning
- Data integration
- Data reduction
- Data transformation

Data Preprocessing

- *Feature engineering* adalah proses mengubah data mentah menjadi fitur yang siap dipakai oleh model ML.
- *Feature engineering* terdiri dari pembuatan fitur, sedangkan data preprocessing melibatkan pembersihan data.



Tugas Utama Data Preprocessing

Data cleaning

- Fill in missing values,
- mooth noisy data,
- Identify or remove outliers, and
- Resolve inconsistencies

Data integration

Integration of multiple databases, or files
Data reduction
Dimensionality reduction

Data transformation

- Normalization
- Standardization
- Encoding

Data Cleaning

Data di dunia nyata itu ‘kotor’

- Kosong atau tidak lengkap:
  pekerjaan=“ ”
- Noisy (nilai yg salah atau outliers):
  gaji=“-10”
- Nilai tidak konsisten:
  jenis kelamin=“perempuan” vs.
  jenis kelamin=“wanita”
- Data yang sama/ duplicate

No quality data, no quality mining results!

Sex	Age	BMI	DM type	DM duration	FBS	Sys BP	Dias BP	Retinopathy
Male	65	25	II	20	129	130	80	Yes
Male	42	27	II	300	210	140	90	No
Female	31	21	I	11	164	145	80	Yes
Male	70	32	II	29	208	160	100	Yes
Female	54	34	II	6	183	155	95	No
	46	29	II	7	198	160	100	No
Female	16	24	I	-1	250	135	80	No
Male	67	30	II	12	243	165	90	Yes
Female	51	28	II	7	163	130	85	No
Girl	70	36	II	20	250	150	90	Yes
Female	63	35	II	14	203	160	110	No
Male	44	39	II	3	149	140	90	No
Boy	51	24	II	9	160	155	80	No
Male	27	19	I	5	170	140	90	No

Incomplete (Missing) Data

Data tidak selalu tersedia

Misalnya, banyak baris tidak memiliki nilai untuk beberapa atribut, seperti pendapatan pelanggan dalam data penjualan

Data yang hilang

mungkin karena :

- Kerusakan peralatan
- Data tidak masuk karena ada kesalah pahaman
- Data tertentu mungkin tidak dianggap penting pada waktu proses entri

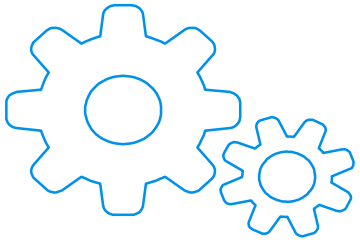
Handling Missing Data:

- Abaikan baris:
- Isi nilai yang hilang secara manual: butuh waktu lama?
- Isi secara otomatis dengan
  - konstanta global: misalnya, "unknown",
  - atribut mean, median (untuk numerik)
  - rata-rata atribut untuk semua sampel yang termasuk dalam kelas yang sama
  - nilai yang paling sering muncul (untuk kategoris)

Noisy Data

- Noise adalah data yang berisi nilai-nilai yang salah atau anomali, yang biasanya disebut juga outlier.

- Nilai atribut yang salah mungkin karena:
  - instrumen pengumpulan data yang salah
  - terjadi masalah pada saat entri data
  - terjadi masalah pada transmisi data



Handling Noisy Data:

- ✓ Binning:
  - urutkan data dan partisi terlebih dahulu ke dalam bin (frekuensi yang sama)
  - kemudian dapat mengganti nilai outlier dengan nilai rata rata atau median dalam bin tersebut.
- ✓ Regression: smooth training data dengan fungsi regresi / mengganti outlier berdasarkan fungsi regresi
- ✓ Clustering: mendeteksi dan menghapus outlier
- ✓ Combined computer and human inspection: mendeteksi nilai yang mencurigakan dan diperiksa oleh manusia (misalnya, menangani kemungkinan outlier)

Feature Encoding

- One-Hot Encoding: mengubah setiap kategori sehingga memiliki nilai angka 1 atau angka 0

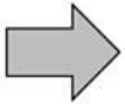
id	color
1	red
2	blue
3	green
4	blue

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

- Label Encoding: mengubah setiap kategori menjadi angka 1,2,3, … dst

petallength	petalwidth	iris_class
1.4	0.2	Iris-setosa
1.4	0.2	Iris-versicolor
1.3	0.2	Iris-virginica



petallength	petalwidth	iris_class
1.4	0.2	1
1.4	0.2	2
1.3	0.2	3

Normalizalization dan Standardization

- Normalization: mengubah nilai-nilai suatu feature menjadi skala tertentu [0,1].
- Standardization: mengubah nilai-nilai feature sehingga mean = 0 dan standard deviation = 1

- **Min-Max Scaling**

  Uses MinMaxScaler

  Transform to defined range

$$y = \frac{x - \min x_i}{\max x_i - \min x_i}$$

- **Standardization**

  Uses StandardScaler

  Transform to mean=0, sd=0

$$y = \frac{x - \bar{x}}{s}$$

Where

$\bar{x}$ = mean

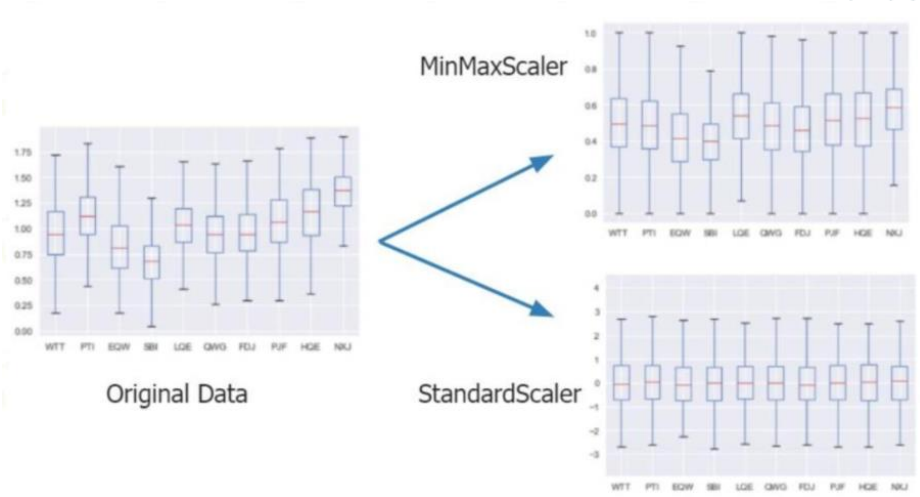
s = Standard deviation

DigitalSkola

Normalizalization dan Standardization

Tujuan:

- Data dengan skala yang sama akan menjamin algoritma pembelajaran memperlakukan semua fiture dengan adil
- Data dengan skala yang sama dan centered akan mempercepat algoritma pembelajaran
- Data dengan skala yang sama akan mempermudah interpretasi beberapa model ML



Kapan penggunaan:

- Gunakan standardization bila kita tahu data punya sebaran normal/gaussian

DigitalSkola

Train test split

- Training adalah proses ketika model mempelajari data
- Hasil dari training disebut model machine learning (trained model)
- Untuk membuktikan keakuratan model, diperlukan data uji (test data)
- Training set : subset untuk melatih model.
- Test set : subset untuk menguji model yang dilatih.
- Karena kurangnya data, kita bisa memisahkan dataset menjadi dua bagian yaitu training dan testing

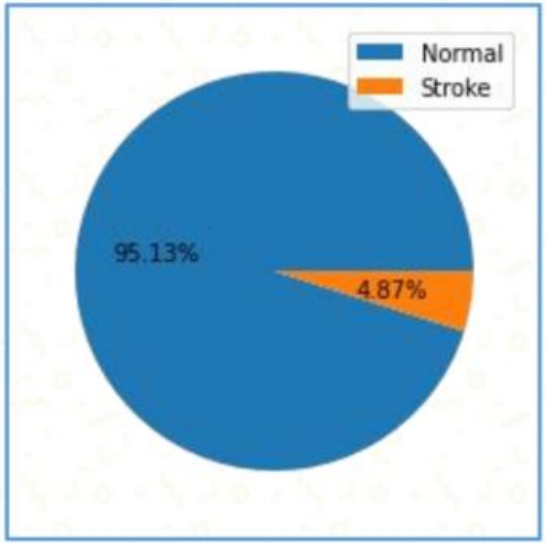
Training

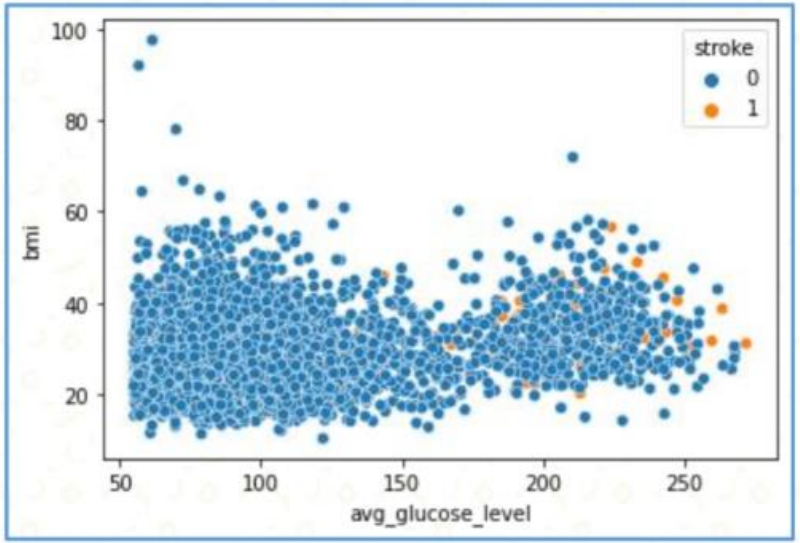
Testing / Proving

Imbalanced dataset

Imbalanced data mengacu pada masalah klasifikasi di mana jumlah pengamatan per kelas tidak merata.

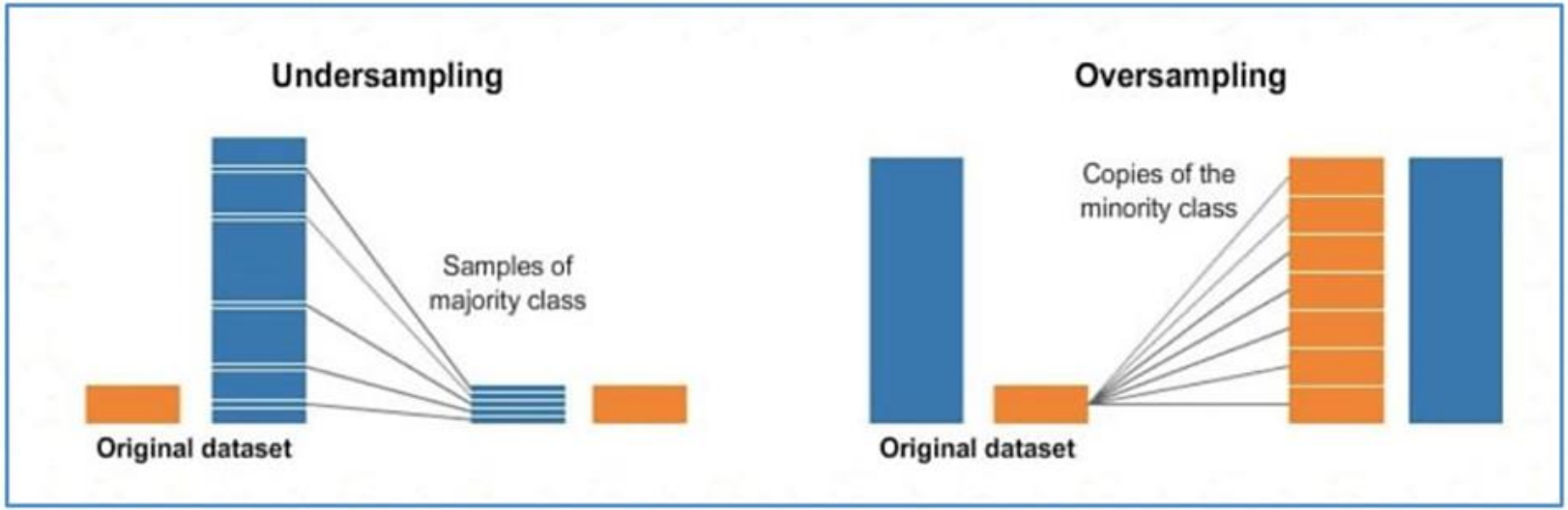
(https://www.kaggle.com/fedesoriano/stroke-prediction-dataset?select=healthcare-dataset-strokedata.csv)





How to handle Imbalanced dataset

- **Under sampling** = Menyeimbangkan distribusi kelas dengan menghilangkan contoh kelas mayoritas secara acak.
- **Oversampling** = Meningkatkan jumlah instance di kelas minoritas dengan mereplikasinya secara acak



Thank you