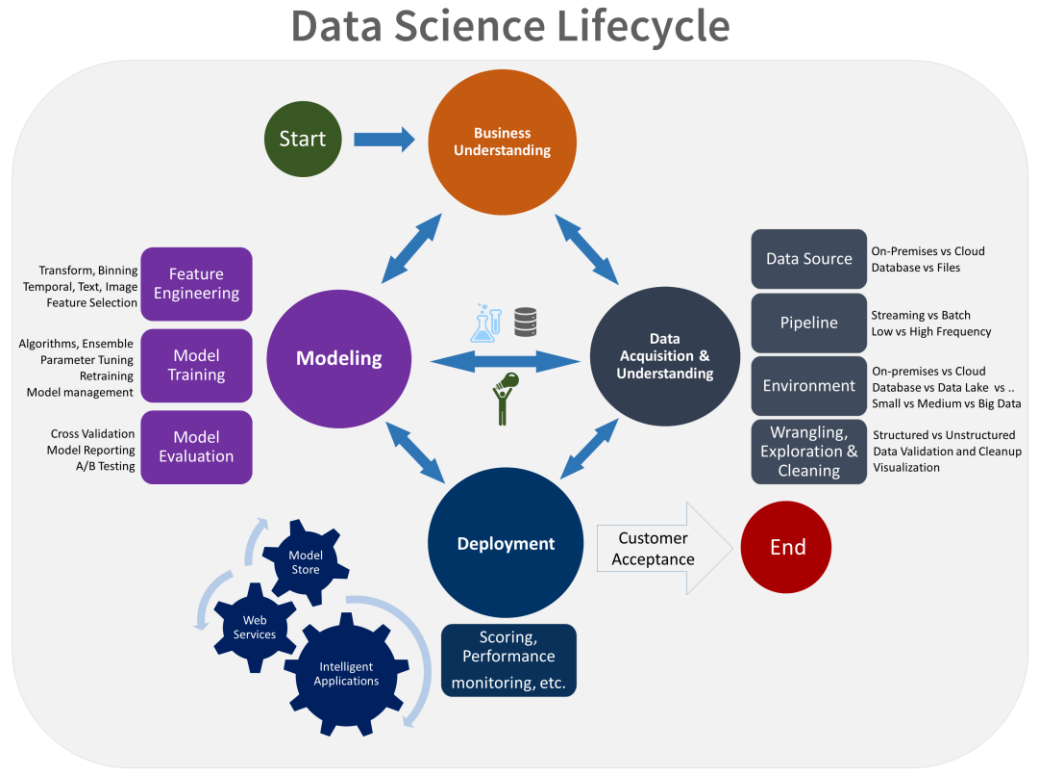
DATA SCIENCE FOR PREDICTIVE MODELING

Tessy Badriyah, PhD.
Politeknik Elektronika Negeri Surabaya (PENS), INDONESIA

Data Mining and Data Science
webinar series #2
25 Agustus 2020

OUTLINE



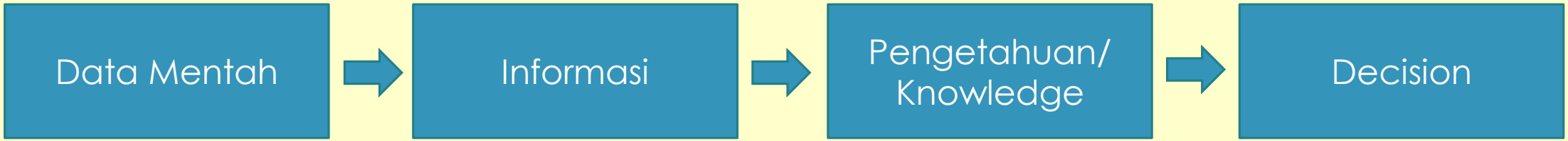
- Pendahuluan
- Data Science
- Predictive Modeling
- Workshop Predictive Modeling dengan Python

Courtesy: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle

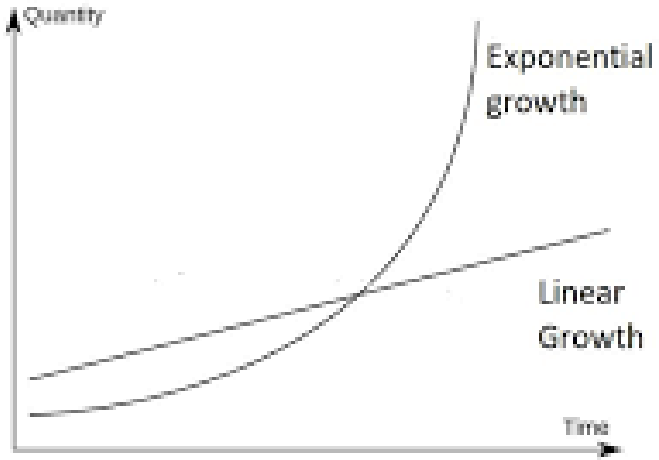
PENDAHULUAN

- Berbagai aktivitas kegiatan dalam kehidupan secara langsung maupun tidak memerlukan pengelolaan data
- Contoh:
- Bank : menabung, transfer, deposit.
- Reservasi : hotel, pesawat, kereta api.
- Belanja : toko, mall, supermarket.
- Dan lain-lain.

DATA EVOLUTION

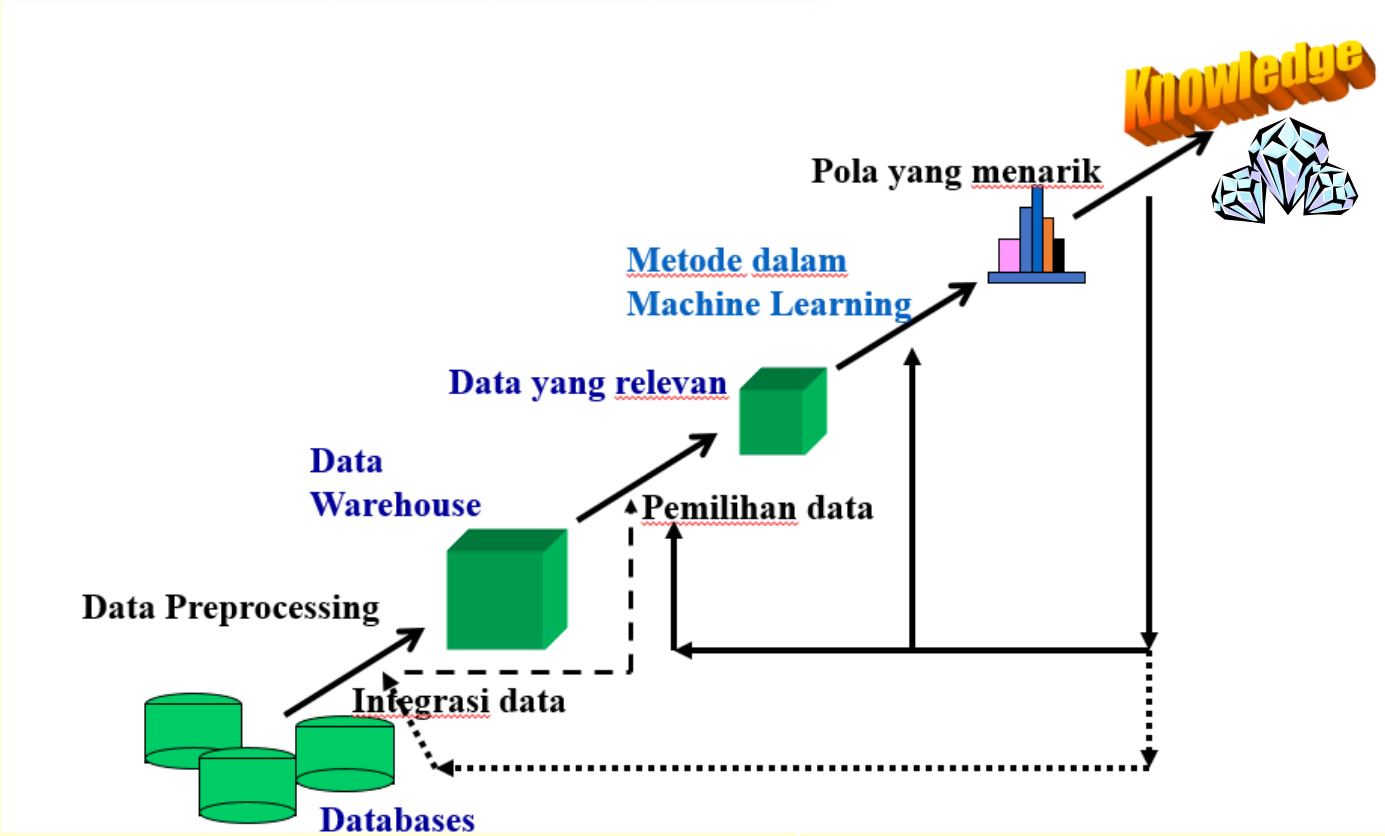


DARI INFORMASI MENJADI PENGETAHUAN



- Adanya ledakan data dengan pertumbuhan data secara eksponensial

Tahapan Data Mining



BIG DATA

- Istilah Big Data muncul pertama kali pada sekitar tahun 2000-an, ketika definisi Big Data dijelaskan dalam 3V oleh seorang analist bernama Doug Laney
  - Volume, data yang disimpan oleh suatu organisasi dalam jumlah yang besar
  - Velocity, ada kebutuhan untuk dapat mengakses data besar tersebut dengan cepat
  - Variety, data berasal dari berbagi macam variasi format data.

FORMAT DATA

- Berikut ini 3 jenis Format data:
  - Structured, relational database (RDBMS)
  - Semi-Structured, XML, JSON
  - Unstructured, document, jurnal, metadata, gambar, video, file teks, audio, ebooks, email message, social media, dll.

BIG DATA AND DATA SCIENCE

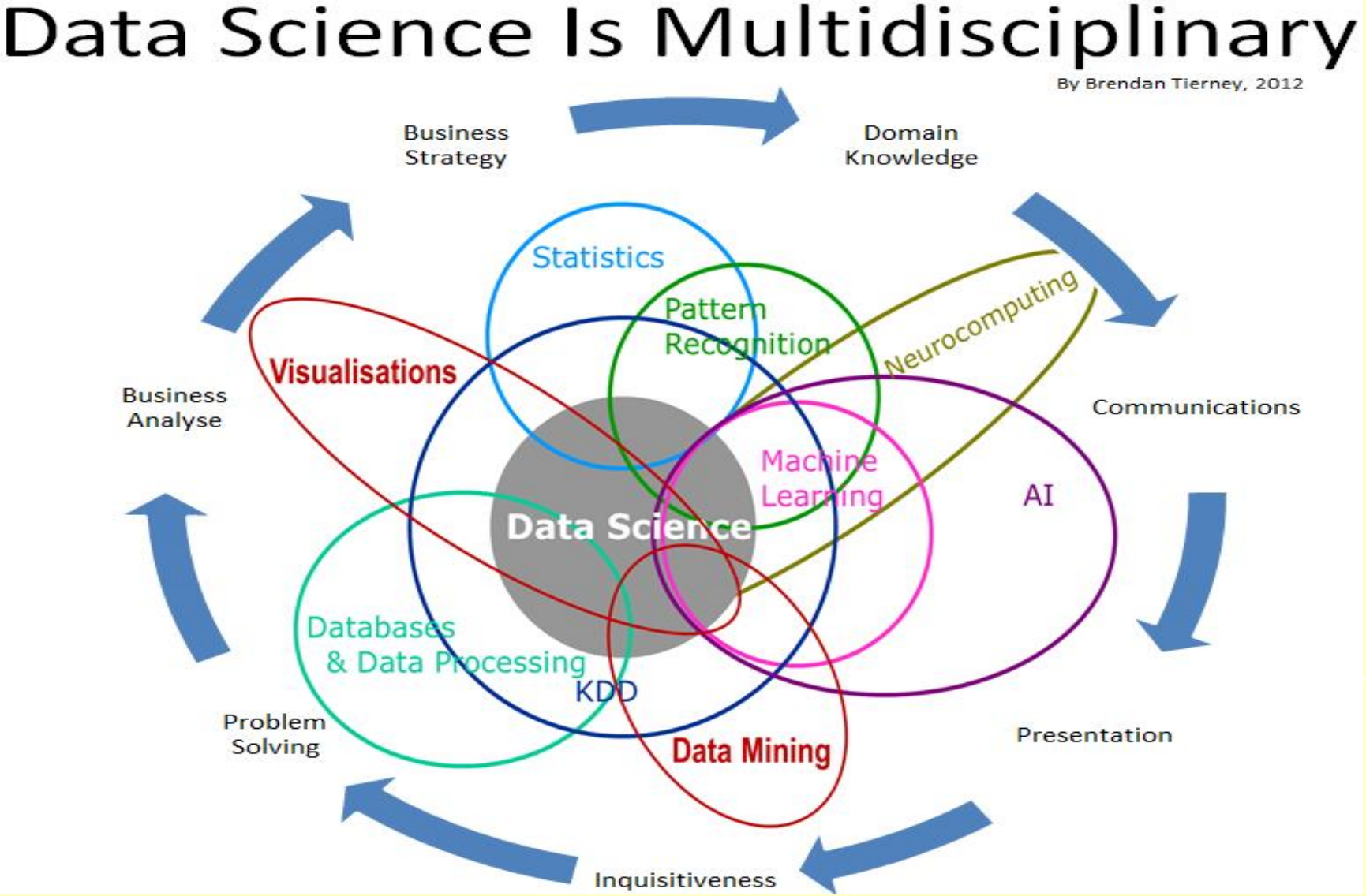
- Bidang ilmu Data Science berkaitan dengan penyelesaian permasalahan kompleks menggunakan data, tidak hanya data terstruktur seperti SQL tapi juga data yang tidak terstruktur dan semi-terstruktur dari era kemunculan Big Data.

- Data science adalah aplikasi dari proses data mining dan menggunakan metode machine learning dalam domain yang spesifik.

- Menurut survey, saat ini data science merupakan pekerjaan yang paling banyak dicari, bisa disebut sebagai the sexiest job of the twenty-first century.

DATA

Data Science Is Multidisciplinary



PREDICTIVE MODELING / PREDICTIVE ANALYTICS

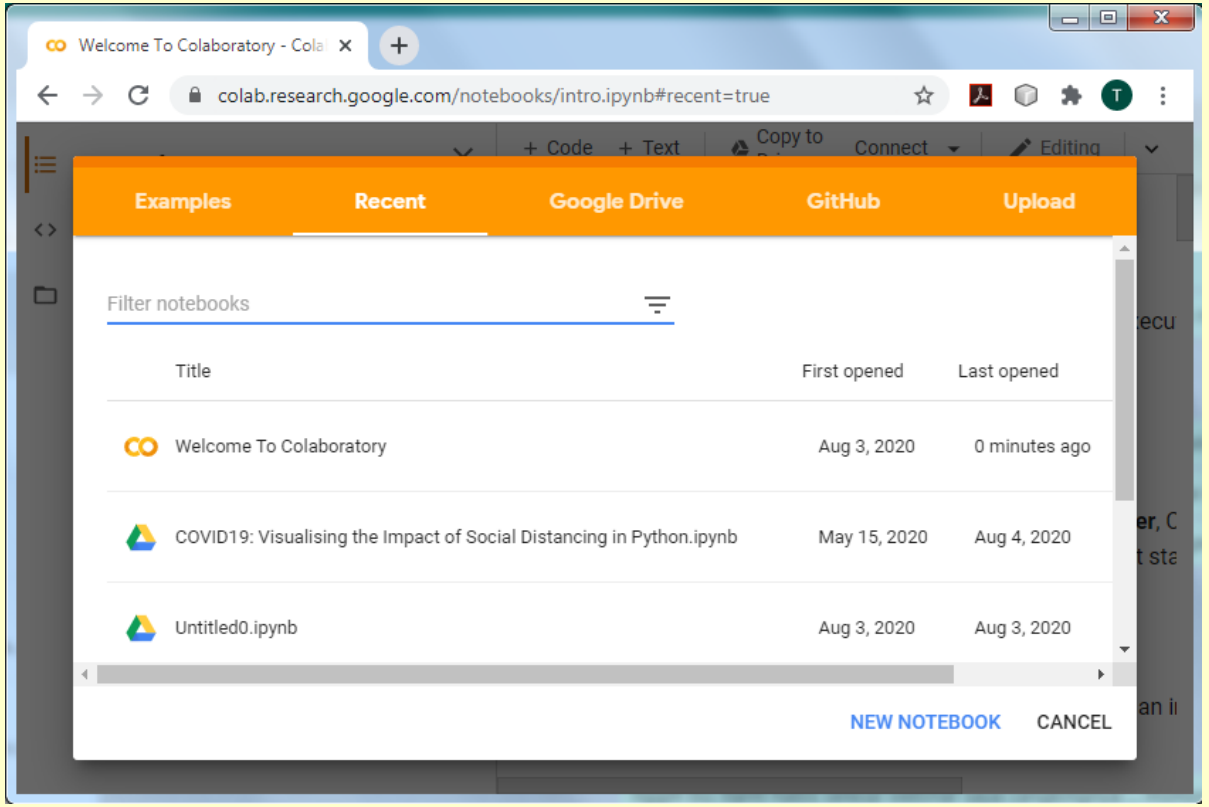
- Menggunakan data historical untuk memprediksi kejadian yang terjadi berikutnya.
- Data historical adalah data yang sudah diketahui atribut outputnya => disebut dengan data training => digunakan untuk men-generate model.
- Model yang dihasilkan kemudian diaplikasikan ke dalam data testing (data testing = data yang terjadi berikutnya yang tidak diketahui atribut output nya).

PYTHON PROGRAMMING

- Mengapa menggunakan Python?
- Karena kita perlu bekerja dengan pemrograman multipurpose, simple, dengan Bahasa yang efisien
  - ❑ Functional
  - ❑ Imperative
  - ❑ Object-oriented
  - ❑ Procedural

GOOGLE COLAB

- Dan untuk menjalankan pemrograman python kita gunakan google colab yang dapat diakses pada : colab.research.google.com
- Sebelumnya, anda harus login dulu dengan akun googlemail.

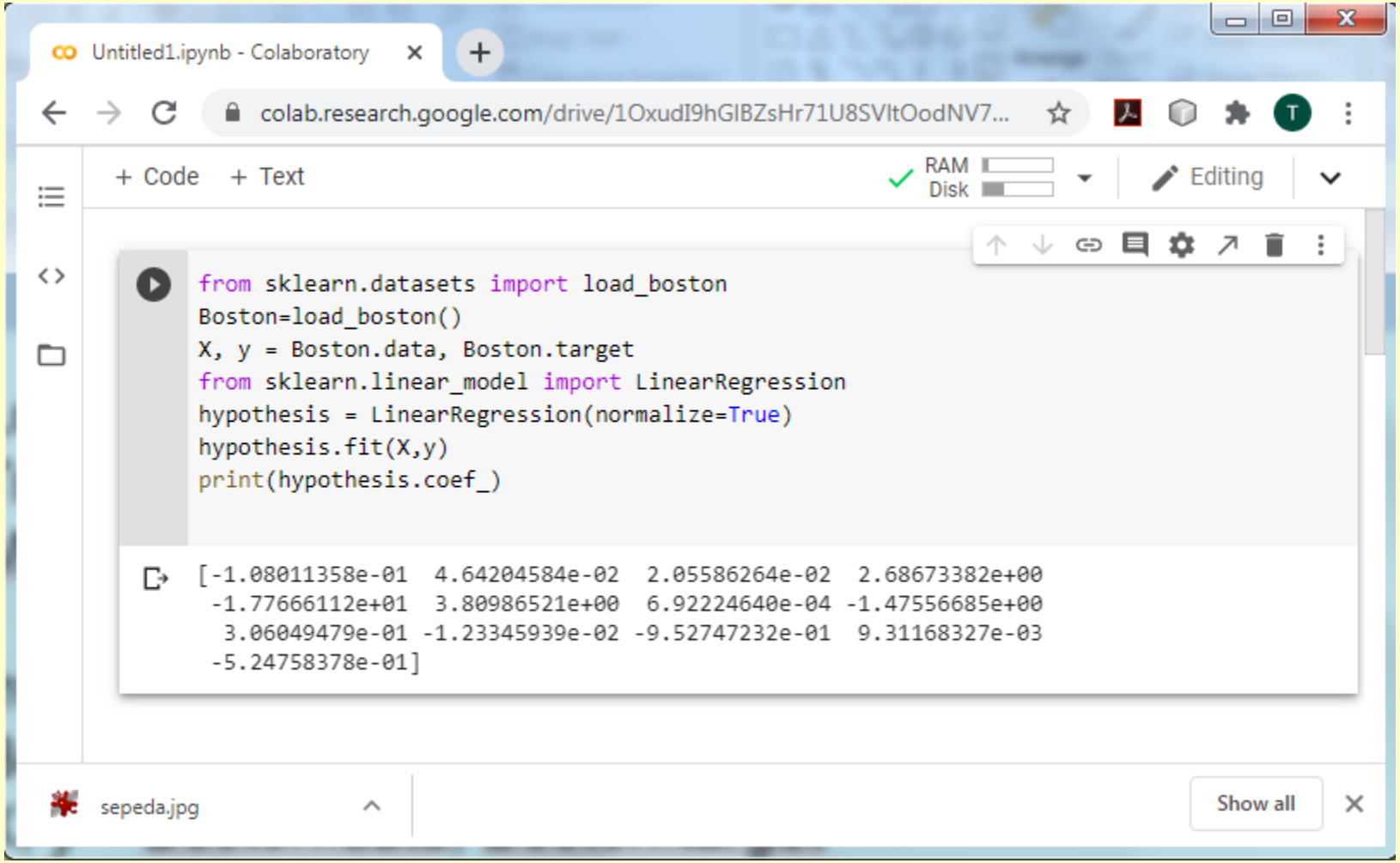


PEMBELAJARAN SEDERHANA

- Loading data
- Training sebuah data
- Menampilkan hasilnya

- from sklearn.datasets import load_boston
- Boston=load_boston()
- X, y = Boston.data, Boston.target
- from sklearn.linear_model import LinearRegression
- hypothesis = Linearregression(normalize=True_
- hypothesis.fit(X,y)
- print(hypothesis.coef_)

PEMBELAJARAN SEDERHANA



WORKSHOP PREDICTIVE MODELING

- Berfokus pada sub bidang yang spesifik yaitu predictive modeling.
- Bidang ini paling banyak digunakan dalam industry dan merupakan bidang yang menggunakan banyak menggunakan scikit-learn library yang ada di Python.
- Predictive modeling berfokus pada pengembangan model yang membuat prediksi yang akurat

PREDICTIVE MODELING WORKFLOW

- Mendefinisikan masalah: melakukan investigasi dan mengkarakterisasi persoalan dalam rangka memahami tujuan project.
- Menganalisa Data: Menggunakan descriptive statistics dan visualisasi untuk bisa lebih memahami data yang digunakan.
- Mempersiapkan Data: Menggunakan transformasi data untuk mendapatkan struktur data yang lebih baik dari persoalan prediksi dari algoritma pemodelan.
- Mengevaluasi Algoritma: Mendesain pengujian yang dapat mengevaluasi jumlah algoritma standart pada data dan memilih beberapa yang paling baik untuk bisa diinvestigasi lebih lanjut.
- Meningkatkan hasil: Menggunakan algorithm tuning dan metode ensemble untuk mendapatkan performansi yang lebih baik.
- Mempresentasikan hasil: Menyelesaikan model, membuat prediksi dan mempresentasikan hasilnya.

LOAD CSV FILES DENGAN PANDAS

- # Load CSV using Pandas
- from pandas import read_csv
- filename = 'diabetes.csv'
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- data = read_csv(filename, names=names)
- print(data.shape)

MELIHAT ISI DATA

- # Menampilkan 20 baris pertama
- from pandas import read_csv
- filename = "pima-indians-diabetes.data.csv"
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- data = read_csv(filename, names=names)
- peek = data.head(20)
- print(peek)

DESCRIPTIVE STATISTICS

- # Menampilkan 20 baris pertama
- from pandas import read_csv
- # Ringkasan Statistik
- from pandas import read_csv
- from pandas import set_option
- filename = "pima-indians-diabetes.data.csv"
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- data = read_csv(filename, names=names)
- set_option('display.width', 100)
- set_option('precision', 3)
- description = data.describe()
- print(description)

DISTRIBUSI CLASS (HANYA UNTUK KLASIFIKASI)

- # Distribusi Class
- from pandas import read_csv
- filename = "pima-indians-diabetes.data.csv"
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- data = read_csv(filename, names=names)
- class_counts = data.groupby('class').size()
- print(class_counts)

Output :
class
0 500
1 268

KORELASI DIANTARA ATRIBUT

- # Korelasi menggunakan Pairwise Pearson
- from pandas import read_csv
- from pandas import set_option
- filename = "pima-indians-diabetes.data.csv"
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- data = read_csv(filename, names=names)
- set_option('display.width', 100)
- set_option('precision', 3)
- correlations = data.corr(method='pearson')
- print(correlations)

	preg	plas	pres	skin	test	mass	pedi	age	class
preg	1.000	0.129	0.141	-0.082	-0.074	0.018	-0.034	0.544	0.222
plas	0.129	1.000	0.153	0.057	0.331	0.221	0.137	0.264	0.467
pres	0.141	0.153	1.000	0.207	0.089	0.282	0.041	0.240	0.065
skin	-0.082	0.057	0.207	1.000	0.437	0.393	0.184	-0.114	0.075
test	-0.074	0.331	0.089	0.437	1.000	0.198	0.185	-0.042	0.131
mass	0.018	0.221	0.282	0.393	0.198	1.000	0.141	0.036	0.293
pedi	-0.034	0.137	0.041	0.184	0.185	0.141	1.000	0.034	0.174
age	0.544	0.264	0.240	-0.114	-0.042	0.036	0.034	1.000	0.238
class	0.222	0.467	0.065	0.075	0.131	0.293	0.174	0.238	1.000

Pemodelan dengan Logistic Regression
Teknik sampling: train_test_split

- Evalusi menggunakan data training dan data testing
- from pandas import read_csv
- from sklearn.model_selection import train_test_split
- from sklearn.linear_model import LogisticRegression
- filename = 'pima-indians-diabetes.data.csv'
- names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
- dataframe = read_csv(filename, names=names)
- array = dataframe.values
- X = array[:,0:8]
- Y = array[:,8]
- test_size = 0.33
- seed = 7
- X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=test_size,
- random_state=seed)
- model = LogisticRegression()
- model.fit(X_train, Y_train)
- result = model.score(X_test, Y_test)
- print("Accuracy: %.3f%%") % (result*100.0)

Output :
Accuracy: 75.591%

Teknik sampling : k-fold Cross Validation

```
# Evaluasi menggunakan Cross Validation
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
filename = 'pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
num_folds = 10
seed = 7
kfold = KFold(n_splits=num_folds, random_state=seed)
model = LogisticRegression()
results = cross_val_score(model, X, Y, cv=kfold)
print("Accuracy: %.3f%% (%.3f%%)") % (results.mean()*100.0, results.std()*100.0)
```

Output :
Accuracy: 76.951% (4.841%)

Tuning Parameter pada Algoritma Support Vector Machine (SVM)

In [1]:
```
from pandas import read_csv
import numpy as np
```

In [2]:
```
from sklearn.datasets import make_blobs
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV
```

In [3]:
```
filename = 'pima-indians-diabetes.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
```

In [4]:
```
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
```

Tuning Parameter pada Algoritma Support Vector Machine (SVM)

In [20]:
```
from sklearn.svm import SVC
```

In [21]:
```
model = SVC()
```

In [22]:
```
kernel = ['poly', 'rbf', 'sigmoid']
C = [50, 10, 1.0, 0.1, 0.01]
gamma = ['scale']
```

In [23]:
```
param = dict(kernel=kernel,C=C,gamma=gamma)
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
search = GridSearchCV(estimator=model, param_grid=param, n_jobs=-1, cv=cv, scoring='accuracy',error_score=0)
result = search.fit(X, Y)
```

In [24]:
```
print("Best: %f using %s" % (result.best_score_, result.best_params_))
means = result.cv_results_['mean_test_score']
stds = result.cv_results_['std_test_score']
params = result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

```
Best: 0.759125 using {'C': 10, 'gamma': 'scale', 'kernel': 'poly'}
```

PENUTUP

Telah diselesaikan, pembelajaran webinar series #2 dengan topik Data Science for Predictive Modeling, dengan pembahasan topik sebagai berikut:

- Pendahuluan
- Data Science
- Predictive Modeling
- Workshop Predictive Modeling dengan Python

Semoga ilmu yang dipelajari dapat bermanfaat, aamiin …
Mohon maaf atas segala kekurangan dan terima kasih.


Thank
You