ชื่อ-สกุล_____ห้อง_____รหัสนักศึกษา_____

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วิชา Machine Learning Laboratory

การทดลองที่ 2 : การทดลองปรับค่าพารามิเตอร์เพื่อสร้างโมเดลการทำนายค่าแบบต่างๆ และเปรียบเทียบประสิทธิภาพของโมเดลทดสอบ

วัตถุประสงค์

1. เพื่อศึกษาและทดลองการใช้งานโมเดลการทำนายค่าแบบต่างๆ
2. เพื่อศึกษาและทดลองการปรับค่าพารามิเตอร์ที่เหมาะสมกับโมเดลทำนายค่าสำหรับชุดข้อมูลทดสอบ
3. เพื่อศึกษาและทดลองเทคนิคการวัดประสิทธิภาพโมเดลการทำนายค่า
4. เพื่อศึกษาและทดลองการเปรียบเทียบประสิทธิภาพของโมเดลต่างๆเชิงกราฟ

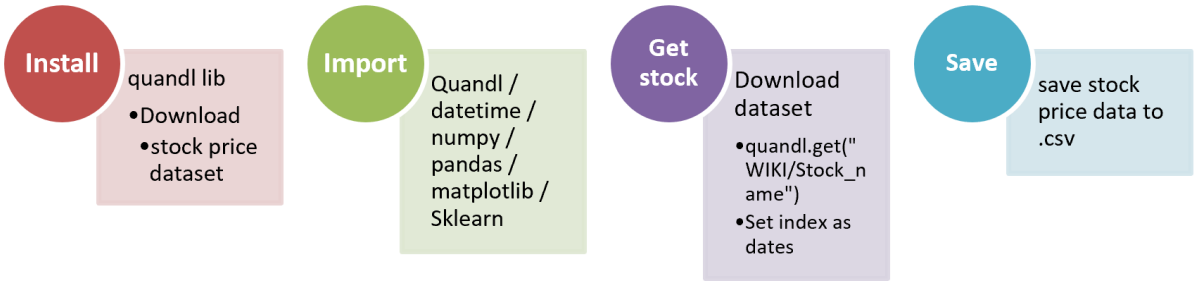
อุปกรณ์ และเครื่องมือที่ใช้ในการทดลอง

1. โปรแกรม python

ข้อกำหนดในการตรวจการทดลอง

1. แสดงโค๊ดและภาพผลการทดลองที่ทำพร้อมอธิบาย
2. นศ.ที่ได้รับการตรวจจากอาจารย์เรียบร้อย อาจารย์จะเช็คส่งงานในระบบ
3. ให้นศ. นำรูปภาพให้แสดง**ทุกภาพ** โพสลงใน facebook group ในหัวข้อ Lab#3.1, 3.2, 3.3 และส่ง source code พร้อม ตอบคำถามท้ายการทดลองใน google form ส่งภายในวันที่ 2 มีค. 2563 เวลา 18.00 น.

ตอนที่ 1: การทดลองเตรียมข้อมูล ปรับค่าข้อมูล และจัดแบ่งชุด Train, Test เพื่อสอนโมเดล



1.1 เตรียมข้อมูลทดลอง

- Install quandl ซึ่งเป็น library ในการดึงข้อมูลราคาหุ้น US

```
pip install -U quandl
```

ถ้าไม่ได้ให้ลอง

```
conda install quandl
```

หรือ

```
conda update quandl
```

- Import Lib (quandl, datetime, numpy, pandas, matplotlib, sklearn)
- ฟังก์ชันใช้งาน

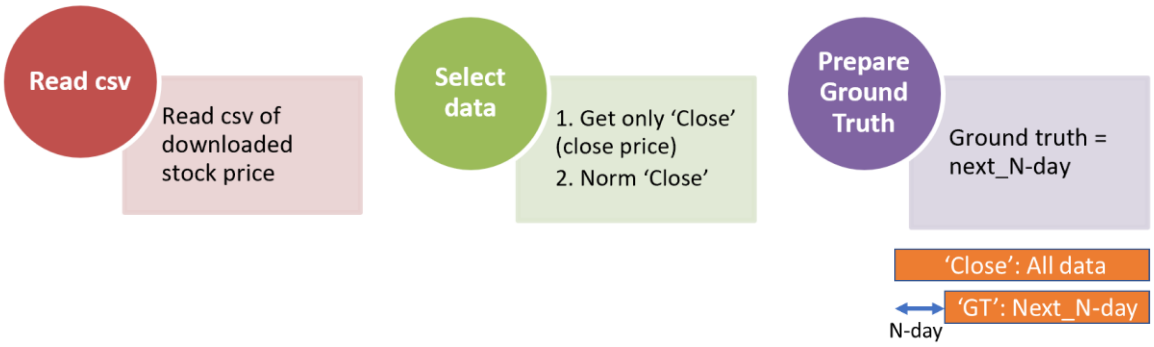
```
# Stock data
import quandl
import datetime
# Analyzing
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn import model_selection
from sklearn import preprocessing
from sklearn import metrics
```

- โหลดข้อมูลหุ้น

โดยใช้ฟังก์ชัน quandl.get("WIKI/Stock_name") และ กำหนด index ข้อมูลด้วย dates

หมายเหตุ ให้นศ.เลือก Stock_name 1 ชื่อ โดยสามารถเป็น (AMZN: amazon, MSFT: Microsoft, GOOG: Google, DELL: Dell, HPQ: HP)

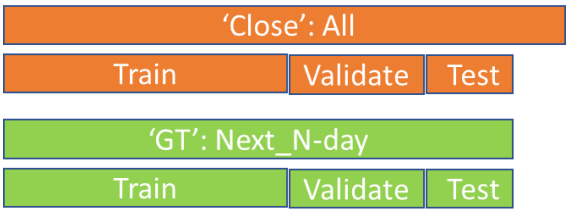
- เก็บข้อมูลที่อ่านเข้ามา ลงไฟล์ .csv ด้วย (to_csv())



1.2 ปรับรูปแบบของข้อมูล

- เลือกใช้ข้อมูล 'Close' ซึ่งเป็นราคาปิดหรือราคาสุดท้ายของแต่ละวันมาใช้
- สร้างข้อมูลทางเลือกด้วยการทำ Normalization ข้อมูล 'Close' ที่เลือก
- สร้างข้อมูลราคาวันถัดไป Next_N-day เพื่อใช้เป็นคำตอบ (Ground Truth) ในการคาดการณ์ (predict ข้อมูลในอีกหลายวันข้างหน้า กำหนดวันในการคาดการณ์จำนวน Next_N-day วันล่วงหน้า

Ex. Next_N-day = 30 # predict ข้อมูลอีก 30 วันข้างหน้า



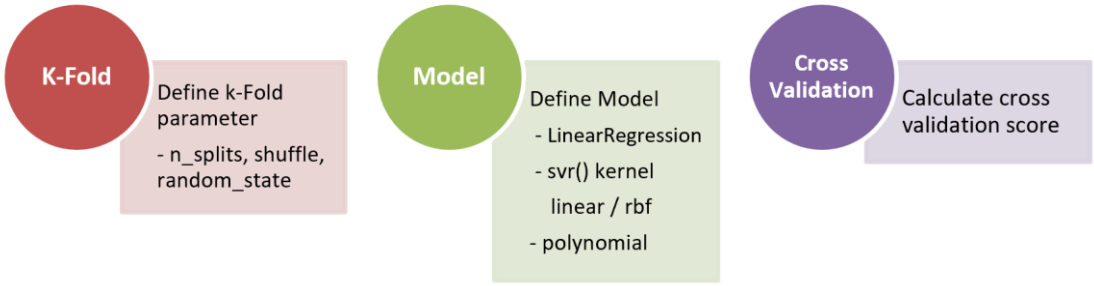
1.3 จัดเตรียมข้อมูลสำหรับ train validation และ test

- ข้อมูล test ให้ใช้ข้อมูล 'GT' ช่วง 60 วันท้าย ซึ่งตรงกับข้อมูล 'Close' ตัดจำนวนวันคาดการณ์ล่วงหน้า (Next_N-day) ไปแล้ว 60 วันท้าย
- เตรียมข้อมูล train, validate ด้วยการใช้ฟังก์ชัน model_selection.train_test_split() กำหนด random_seed เพื่อให้ผลการจัดข้อมูลคงที่เหมือนกันทุกครั้งที่สอนโมเดล

1.4 แสดงรูปกราฟการกระจายของ train validate ที่แบ่งจากข้อ 1.3

```
# ตั้งค่าขนาดพื้นที่ภาพ
plt.figure(figsize=(30,10))
# scatter plot ความสัมพันธ์ของค่า X_train, y_train และ X_test, y_test
plt.scatter(x, y, marker='o', color='blue');
```

ตอนที่ 2: การทดลองทำ Cross validation และ prediction เพื่อดูค่าความแม่นยำของแต่ละโมเดล



2.1 ทำการทดสอบพารามิเตอร์สำหรับโมเดลด้วยการทำ Cross Validation

- สร้างโมเดลรูปแบบ cross validation ที่ต้องการใช้ ในที่นี้ใช้ K-Fold โดยใช้ฟังก์ชัน

    model_selection.KFold()

กำหนดค่า random_state=seed, shuffle = True เพื่อให้ random ได้ชุดข้อมูลเดียวกันในทุกครั้ง

- สร้าง prediction model พร้อมระบุพารามิเตอร์ที่ต้องการ โดยกำหนดให้ใช้โมเดลต่อไปนี้

```
# Linear Regression Model
LRM = LinearRegression()
# Support Vector Regression (SVR) Model จำนวน 3 รูปแบบ kernel
c_val # ค่าปรับเข้มงวดกับ outlier bound [10^-6, 10^6] ค่าที่น่าสนใจ 1000
gramma_value # ค่าการควบคุมรูปร่างของโมเดล [10^-6, 10^6] ค่าน่าสนใจ 0.1

svr_lin = SVR(kernel='linear', C=c_val)
svr_rbf = SVR(kernel='rbf', C=c_val, gamma=gmm)
svr_poly = SVR(kernel='poly', C=c_val, degree=2)
```

- ทำ cross validation สำหรับแต่ละโมเดล model_name โดยใช้ฟังก์ชัน

    model_selection.cross_val_score()

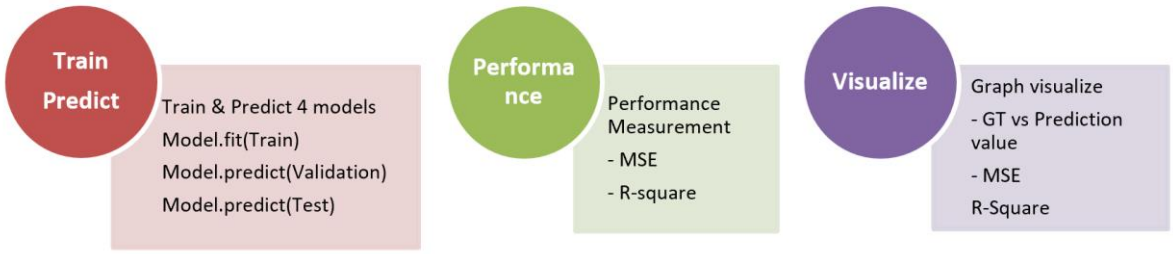
หมายเหตุ ให้ทำการคำนวณ score ซึ่งเป็นค่า (‘Accuracy’) ของโมเดลทั้ง 4 แบบที่สร้างไว้

- แสดงรูปกราฟเปรียบเทียบ score ที่ได้จากโมเดลทั้ง 4 แบบที่คำนวณข้างบน ในรูปแบบต่อไปนี้

    กราฟ #1 score แต่ละ k-fold

    กราฟ #2 score.mean

    กราฟ #3 score.std



2.2 ทดสอบโมเดลทั้ง 4 แบบ ที่กำหนดพารามิเตอร์ไว้ในข้อ 2.1

- ทำการ train โมเดลทั้ง 4 แบบ ด้วยข้อมูล Train ที่แบ่งไว้

model_name.fit() / model_name.predict()

- ทำการ predict ข้อมูลชุด Validation และ Test
- คำนวณค่าตัววัดประสิทธิภาพของการทำนายจากโมเดลทั้ง 4 แบบ โดยวัดค่า MSE และ R^2

metrics.mean_squared_error() / metrics.r2_score()

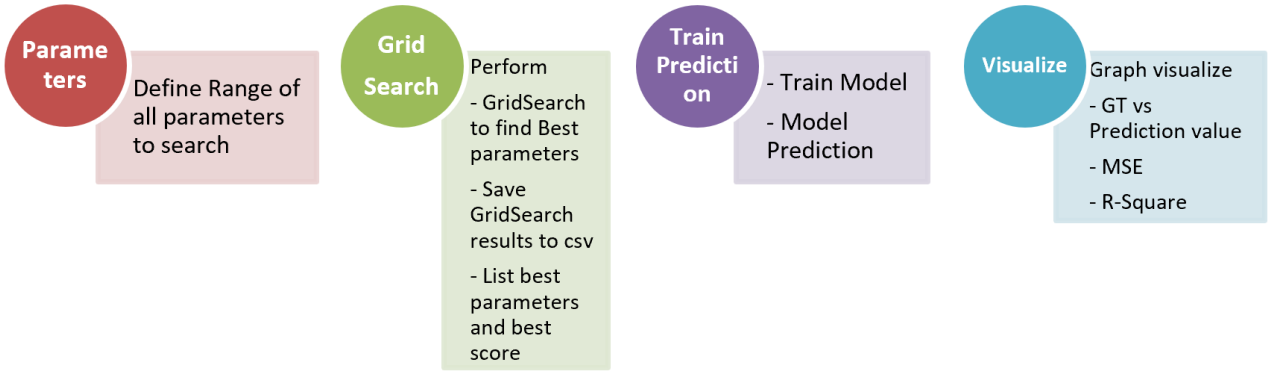
ของข้อมูล Predict validation และ predict test

- แสดงรูปกราฟเปรียบเทียบผลการ Predict validation และ Predict test ข้างต้นจากโมเดลทั้ง 4 แบบ โดยในรูปแบบกราฟที่แสดงเห็นความแตกต่างชัดเจน เช่น กราฟ plot, bar, scatter เป็นต้น

_____

(อาจารย์ตรวจผลการทดลอง)

ตอนที่ 3: การทดลองการค้นหาพารามิเตอร์ที่ดีที่สุดสำหรับโมเดล



3.1 กำหนดรายการพารามิเตอร์ทั้งหมดที่ต้องการทดสอบหาค่าที่ดีที่สุดของโมเดล SVC

svc_kernel='rbf'

k-Fold # เลือก 1 ค่า

c_param # เลือกค่าในช่วง [0.1, 1000] จำนวน 4 ค่า

gamma = # เลือกค่าในช่วง [0.1, 1.0] จำนวน 3 ค่า

tuned_parameters = [{'kernel': svc_kernel, 'C': c_param, 'gamma': gamma}]

3.2 เตรียมการค้นหาพารามิเตอร์ที่ดีที่สุดโดยใช้ฟังก์ชั่น GridSearchCV

- # กำหนดโมเดล

  model = SVR()

- ใช้ cross validation (cv) เป็น kfold ที่กำหนดไว้ในตอนข้อ 2.1

  model_selection.GridSearchCV()

- นำค่าพารามิเตอร์ที่ดีที่สุดที่ได้จาก GridSearchCV() ไปสอนโมเดล แสดงค่า score และพารามิเตอร์ที่ดีที่สุด

  fit() / best_params_/ best_score

- save ผลลัพธ์จากการทำ GridSearchCV cv_results ลงบนไฟล์ .csv

3.3 ทำการ predict ข้อมูลชุด Validation และ Test

3.5 คำนวณค่าตัววัดประสิทธิภาพของการทำนายที่ได้จากข้อ 3.5 โดยวัดค่า MSE และ R^2

3.6 แสดงรูปกราฟเปรียบเทียบผลการ Predict validation และ Predict test ข้างต้นจากโมเดลทั้ง 4 แบบ โดยในรูปแบบกราฟที่แสดงเห็นความแตกต่างชัดเจน เช่น กราฟ plot, bar, scatter เป็นต้น

_____

(อาจารย์ตรวจผลการทดลอง)

<u>Tutorial</u>

Panda (Reading .csv / Drop Duplicate / drop row column / concat append)

[1] https://www.ritchieng.com/pandas-removing-duplicate-rows/

[2] https://chrisalbon.com/python/data_wrangling/pandas_dropping_column_and_rows/

[3] https://jakevdp.github.io/PythonDataScienceHandbook/03.06-concat-and-append.html

Get US Stock Information

[4] https://github.com/WillKoehrsen/Data-Analysis/tree/master/stocker

pip install -U quandl

Stock Prediction

[5] https://enlight.nyc/projects/stock-market-prediction/?fbclid=IwAR1wnwE1abkH38otIr3ddlUPIiJr1KCbdl-tj8-zGoYPU33xEAJTqLU85VA

[6] https://programmingforfinance.com/2018/01/predicting-stock-prices-with-linear-regression/?fbclid=IwAR24L0873fd0GMK_r1svieMQi-YYUWhE8wxpDN9nA7UI37iDncCEbW0CHvE

[7] https://towardsdatascience.com/stock-prediction-in-python-b66555171a2?fbclid=IwAR2BqdV9oVHinmXATN_VyPWDBtcGmqaF-HEDhmk90y7fBn0HzOyajxG7pDQ

Visualization in Graph Plot

[8] https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

Color code

[9] https://matplotlib.org/examples/color/named_colors.html