

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
วิชา Machine Learning Laboratory

**การทดลองที่ 1 : พื้นฐานการอ่านไฟล์ข้อมูล การแก้ปัญหามูลค่าหาย การปรับช่วงค่าของข้อมูล การ  
ปรับลดมิติข้อมูล และ แสดงผลข้อมูลในเชิงกราฟ**

**วัตถุประสงค์**

1. เพื่อศึกษาและทดลองการอ่านไฟล์ข้อมูล (.csv)
2. เพื่อศึกษาและทดลองการแก้ปัญหามูลค่าหาย
3. เพื่อศึกษาและทดลองการปรับช่วงค่าของข้อมูล
4. เพื่อศึกษาและทดลองการปรับลดมิติข้อมูล
5. เพื่อศึกษาและทดลองการแสดงผลข้อมูลเชิงกราฟ
6. เพื่อศึกษาและทดลองการเตรียมรูปแบบข้อมูลเพื่อเข้าโมเดล

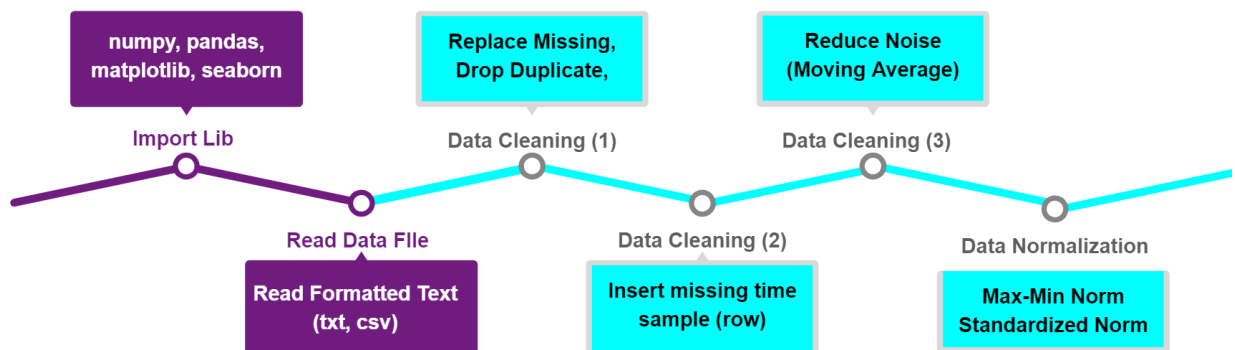
**อุปกรณ์ และเครื่องมือที่ใช้ในการทดลอง**

1. โปรแกรม python

**ข้อกำหนดในการตรวจการทดลอง**

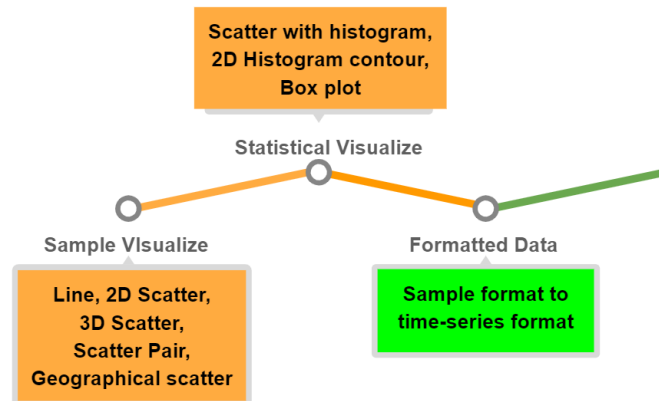
1. โฟสภาพผลการทดลองที่ต้องการตรวจใน Facebook group ในหัวข้อแลปที่ต้องการส่ง
2. อาจารย์จะตรวจ source code และภาพผลการทดลองที่ทำพร้อมอธิบาย เช็กส่งงานในระบบให้
3. นศ.ทุกคน ส่ง source code และ ตอบคำถามท้ายการทดลอง (สำหรับนศ.ที่ส่งทันในคาบเรียน) และ  
เอกสารอธิบายการทดลองและผลการทดลอง (เพิ่มเติมสำหรับนศ.ที่ส่งไม่ทันในคาบเรียน) ใน Google  
form ตามลิงค์ที่ให้ใน Facebook group ภายในวันที่ 28 มค. 2563 เวลา 18.00 น.

ตอนที่ 1: การทดลองอ่านไฟล์ข้อมูล การแก้ปัญหา ข้อมูลหาย และการปรับช่วงค่าของข้อมูล แสดงข้อมูลเชิงกราฟ และ การจัดเตรียมรูปแบบข้อมูลเพื่อนำเข้าโมเดล



1.1 ขั้นตอนการทดลองในการนำเข้าข้อมูล แก้ปัญหาข้อผิดพลาดของข้อมูล และการปรับช่วงค่าของข้อมูล

- Import Library  
Lib พื้นฐานที่ใช้งานหลักได้แก่ Numpy, Pandas, matplotlib.pyplot, seaborn
- Read Data File  
อ่านข้อมูล Dataset ที่ใช้ โดยไฟล์จะอยู่ในรูปของ text file (.txt) หรือ formatted text (.csv) ให้อ่านข้อมูลเข้ามาอยู่ในรูปของ dataframe [ `pd.read_csv()` ]
- Data Cleaning  
ทำการลดปัญหาข้อผิดพลาดของข้อมูลโดย
  - (1) จัดการข้อมูลหาย หรือ ขาดข้อมูล โดยสามารถเลือกตัดทิ้ง หรือ แทนที่ด้วย 0, mean, median [ `pd.fillna()` , `pd.replace()` , `pd.drop_duplicates()` ]
  - (2) แทรกข้อมูลในช่วงเวลาที่ขาดหายไป [ `pd.interpolate()` ]
  - (3) จัดการลดสัญญาณรบกวนในข้อมูลด้วยการทำ Moving Average [ `pd.rolling()` ]
- Data Normalization  
ทำการ Normalize ข้อมูลด้วย 1 เทคนิค (Max-Min Norm , Standardized Norm)



## 1.2 ขั้นตอนการแสดงผลเชิงกราฟ

- แสดงกราฟข้อมูลแต่ละ feature (Column) ด้วย Line Plot เพื่อดูค่าที่แท้จริง
- แสดงกราฟข้อมูลความสัมพันธ์ระหว่างคู่ features ด้วย 2D Scatter Pair Plot หรือ 2D sns.jointplot หรือ 3D Scatter Plot เพื่อดูความสัมพันธ์ของข้อมูลเชิง 3 มิติ (accelerateX, accelerateY, accelerateZ) หรือ (gyro.x, gyro.y, gyro.z)

Ex. 3D Scatter plot

```

fig = plt.figure(figsize=(w,h))
#=====
# First subplot
#=====
# set up the axes for the first plot
ax = fig.add_subplot(1, 2, 1, projection='3d')

ax.scatter(x,y,z, c='cyan',s=20,edgecolor='k')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# optional view configurations
ax.view_init(30, -60)

#=====
# Second subplot
#=====
# set up the axes for the second plot
ax = fig.add_subplot(1, 2, 2, projection='3d')

ax.scatter((x,y,z, c='violet',s=20,edgecolor='k')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

# optional view configurations
ax.view_init(30, -10)

plt.show()

```

- แสดงข้อมูลเชิงพิกัด Geolocation ของ ข้อมูล (gps.x, gps.y)

- Export ภาพแผนที่จาก

# lats longs max min from

<https://www.openstreetmap.org/export#map=6/12.780/102.986>

- อ่านไฟล์ภาพแผนที่

- แสดงตำแหน่งพิกัด GPS

```
#Map image from --> https://www.openstreetmap.org/export#map=6/12.780/102.986
map_im = plt.imread("map.png")
```

```
fig, ax = plt.subplots(figsize = (7,14))
ax.scatter(df['latitude'], df['longitude'], zorder=1, alpha= 0.5, c='r', s=20)
ax.set_title('Plotting Spatial Data on Map')
ax.set_xlim(BBox[0],BBox[1])
ax.set_ylim(BBox[2],BBox[3])
ax.imshow(map_im, zorder=0, extent = BBox, aspect= 'auto')
```

### 1.3 ขั้นตอนการจัดเตรียมข้อมูลเพื่อนำเข้าโมเดล

- ทำการจัดข้อมูล 5 Features [accelerateX, accelerateY, accelerateZ, compass, heartrate]

ในรูปของอะเรย์ 2 มิติ 2 ชุด

- อะเรย์ชุดที่ 1: เป็นการจัดเรียงข้อมูล โดยต้องการ row: single time sample / column: 5 features เพื่อให้ได้ผลลัพธ์เป็นอะเรย์ขนาด

(shape: ( #sample, #features ))

- อะเรย์ชุดที่ 2: เป็นการจัดเรียงข้อมูล time series ในรูปของ อะเรย์ 3 มิติ โดยต้องการตัดข้อมูลตาม time series เงื่อนไข time\_step และ time stride ที่นศ.กำหนด เพื่อให้ได้ผลลัพธ์เป็นอะเรย์ขนาด

(shape: ( #ชุด time\_series, #time\_step, #features ))

จากนั้นปรับอะเรย์ 3 มิติที่ได้ ให้อยู่ในรูปของ 2 มิติขนาด

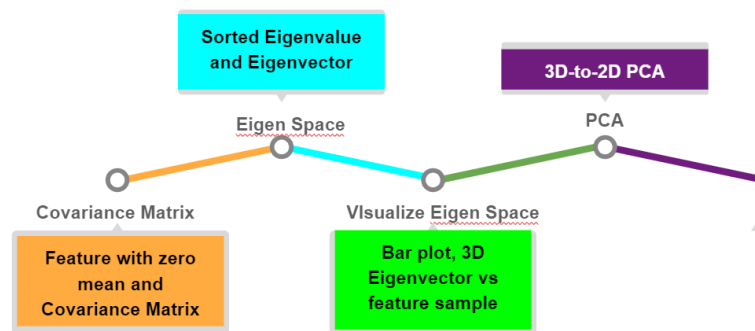
(shape: ( #ชุด\*#time\_step, #features ))

- แสดงภาพของอะเรย์ชุดที่ 1 และ ชุดที่ 2 [plt.imshow() หรือ plt.annotate\_heatmap() หรือ sns.heatmap() ]

---

(อาจารย์ตรวจผลการทดลอง)

## ตอนที่ 2: การทดลองการลดมิติของข้อมูลด้วยค่า Principle Component Analysis



- 2.1 เตรียมชุดข้อมูล feature 3 ค่า  $x$  (accelerateX, accelerateY, accelerateZ)
- ปรับให้เป็น zero mean

$$X_{zero\_mean} = x - \bar{x}$$

- คำนวณค่า covariance matrix ของชุดข้อมูล  $X_{norm}$  [ np.T, np.dot() ]

$$cov = X_{zero\_mean}^T \cdot X_{zero\_mean} / (N - 1)$$

- 2.2 คำนวณค่า eigenvalue / eigenvector จาก covariance matrix ที่คำนวณได้จากข้อ 2.1
- [ np.linalg.eig(covM) ]

- 2.3 แสดงกราฟ Eigen Space (Eigenvalue, Eigenvector)

- แสดงกราฟแท่ง (Bar graph) ของค่า Eigenvalue ที่จัดเรียงค่าจากมากไปน้อย
- แสดงปรับขนาดของ Eigenvector ด้วยค่า Eigenvalue

$$\text{Eigenvector} * \sqrt{\text{Eigenvalue}}$$

```
# Split each eigenvector and scale with its sqrt(eigenvalue)
ev1 = eig_vecs[:,0]*np.sqrt(eig_vals[0])
ev2 = eig_vecs[:,1]*np.sqrt(eig_vals[1])
ev3 = eig_vecs[:,2]*np.sqrt(eig_vals[2])
```

- แสดงกราฟความสัมพันธ์ของ feature และ eigen vector
  - แสดงกราฟ 3D Scatter Plot features (accelerateX, accelerateY, accelerateZ)
  - แสดงเส้น Eigen vector ลงบนกราฟ 3D Scatter Plot ที่สร้างไว้

```
#####
#plotting eigenvectors
#####

fig = plt.figure(figsize=(20,8))

# 3D eigenvector view
ax = fig.add_subplot(141, projection='3d')

ax.plot(Xnorm['accelerate.x'], Xnorm['accelerate.y'], Xnorm['accelerate.z'], 'o', markersize=10, color='green', alpha=0.2)
ax.plot([mean_x], [mean_y], [mean_z], 'o', markersize=10, color='red', alpha=0.5)

ax.plot([0, ev1[0]], [0, ev1[1]], [0, ev1[2]], color='red', alpha=0.8, lw=2)
ax.plot([0, ev2[0]], [0, ev2[1]], [0, ev2[2]], color='violet', alpha=0.8, lw=2)
ax.plot([0, ev3[0]], [0, ev3[1]], [0, ev3[2]], color='cyan', alpha=0.8, lw=2)

ax.set_xlabel('x_values')
ax.set_ylabel('y_values')
ax.set_zlabel('z_values')

plt.title('Eigenvectors')

# optional view configurations (x,y,z)=(10,60); (x,z)=(0,90); (x,y)=(90,90); (y,z)=(180,180)
ax.view_init(10,60)

plt.show()
```

## 2.4 ทำ PCA เพื่อลดข้อมูลจาก 3D -> 2D

- ลดมิติของข้อมูลจาก 3D features  $x$  (accelerateX, accelerateY, accelerateZ) ลงเหลือ 2D โดยเลือก eigenvector 2 vector แรก ที่สัมพันธ์กับ eigenvalue ที่มีค่าสูงสุด 2 อันดับแรก [np.dot()]

$$X_{PCA}(i) = x \cdot Eigen\_vector(i)$$

- แสดงภาพ  $X_{PCA}$  ด้วย [plt.imshow() หรือ plt.annotate\_heatmap() หรือ sns.heatmap() ]

---

(อาจารย์ตรวจผลการทดลอง)

## Tutorial

[Panda \(Reading .csv / Drop Duplicate\)](#)

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

[2] <http://www.datasciencemadesimple.com/delete-drop-duplicate-row-dataframe-python-pandas/>

[Panda \(Fill missing value\)](#)

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.fillna.html>

[Panda \(Replace with specific value\)](#)

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.replace.html>

Panda (drop row or column)

[https://chrisalbon.com/python/data\\_wrangling/pandas\\_dropping\\_column\\_and\\_rows/](https://chrisalbon.com/python/data_wrangling/pandas_dropping_column_and_rows/)

PCA

Inside calculation example

<https://plot.ly/ipython-notebooks/principal-component-analysis/>

Using built in function

<https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html>

Visualization in Graph Plot

[1] [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/visualization.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html)

[2] [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/visualization.html#visualization-hist](https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html#visualization-hist)

[3] 3D Subplot: <https://matplotlib.org/gallery/mplot3d/subplot3d.html>

[4] Jointdistribution: <https://seaborn.pydata.org/tutorial/distributions.html>

<https://python-graph-gallery.com/matplotlib/>

Color code

[https://matplotlib.org/examples/color/named\\_colors.html](https://matplotlib.org/examples/color/named_colors.html)

[https://seaborn.pydata.org/tutorial/color\\_palettes.html#setting-the-default-color-palette](https://seaborn.pydata.org/tutorial/color_palettes.html#setting-the-default-color-palette)

Geographical Plot

<https://www.tutorialspoint.com/plotting-google-map-using-gmplot-package-in-python>

<https://plot.ly/python/scatter-plots-on-maps/>