

# รายงาน

เรื่อง CS:GO Round Winner Classification

## เสนอ

ดร. นิวัตรณ วัฒนกิจรุ่งโรจน์

## จัดทำโดย

นาย ปิยชาติ เอี่ยมสำอางค์ รหัส 63130500081

นาย พีรพัฒน์ คำเกิด รหัส 63130500088

นาย ภัทรพล มรรคหิรัญ รหัส 63130500091

นาย วริศชัย สุรชัยธนวัฒน์ รหัส 63130500103

นาย สหทัศน์ ยิ่งสกุลเกียรติ 63130500113

นาย อภิวัชรณ์ อาทิตย์เที่ยง รหัส 63130500130

นาย อริย์รัช กรุดมินบุรี รหัส 63130500133

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา INT491 Introduction to Data Analytics and Big  
Data & Big Data Analytics  
ภาคเรียนที่ 1 ปีการศึกษา 2565

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Introduction to Data Analytics and Big Data & Big Data Analytics รหัสวิชา INT491 โดยมีจุดประสงค์เพื่อศึกษาความรู้ที่ได้จากเรื่อง ความแตกต่างด้านประสิทธิภาพจากการใช้ PySpark และ ไม่ได้ใช้ PySpark ในการประมวลผลเพื่อวิเคราะห์การทำนายผล ซึ่งรายงานนี้มีเนื้อหาเกี่ยวกับความรู้จาก การที่กลุ่มของเรานั้นได้ทำการทดลองและสรุปผลจากการสร้างเพื่อการวิเคราะห์การทำนายผล และ นำมาเปรียบเทียบเพื่อสรุปผล ถึง ความแตกต่างของ PySpark และ ไม่ได้ใช้ PySpark

ทางกลุ่มผู้จัดทำหวังว่ารายงานฉบับนี้จะให้ความรู้ และเป็นประโยชน์แก่ผู้อ่านทุกๆ ท่านหากมีข้อผิดพลาดประการใดทางกลุ่มผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย

ลงชื่อ  
กลุ่มผู้จัดทำ

# สารบัญ

เรื่อง	หน้า
<b>1.Dataset</b>	<b>4</b>
<b>2.แนวคิดในการวิเคราะห์</b>	<b>4</b>
<b>3.ผลการทดลอง (confusion matrix, accuracy, precision, recall, f1 และเวลาที่ใช้)</b>	<b>5</b>
วิธี A แบบ non-PySpark : MLPClassifier	5
วิธี A แบบ PySpark : MultilayerPerceptronClassifier	7
วิธี B แบบ non-PySpark : DecisionTreeClassifier	9
วิธี B แบบ PySpark: DecisionTreeClassifier	11
สรุปจากผล AVG ทั้ง 2 model ที่ใช้ทั้ง PySpark และ non-PySpark	12
<b>4.ตารางเปรียบเทียบประสิทธิภาพในแง่เวลาที่ใช้ในการประมวลผล</b>	<b>15</b>
<b>5.กราฟแสดงผลการทดลอง</b>	<b>16</b>
<b>6.สรุปผล</b>	<b>18</b>
<b>บรรณานุกรม</b>	<b>21</b>

# 1.Dataset

Dataset ที่ใช้ มีชื่อว่า CS:GO Round Winner โดยทำการ Classification เป็น dataset เกี่ยวกับผลการแพ้ชนะของเกม CS:GO โดยใน dataset มีข้อมูลได้แก่ timeleft , ct score , t score , map , bomb planted , ct health , t health , ct armor , t armor , ct money , ct helmets , t helmets , ct defuse kits , ct players alive , t players alive และ round winner ซึ่งใช้ dataset ดังกล่าวในการวิเคราะห์ผลแพ้ชนะของแต่ละรอบ และประเมินประสิทธิภาพในการทำนายผลแพ้ชนะของแต่ละรอบ

## 2.แนวคิดในการวิเคราะห์

### 2.1) วิธี A : Neural network แบบ Multi layer Perceptron

เป็นวิธีในการเรียนรู้เชิงลึก (deep learning) ของตัว model โดยอาศัย neural cell หรือเซลล์ประสาทของมนุษย์ในหลาย layer ในการประมวลผล เพื่อให้ได้ผลลัพธ์ที่มีประสิทธิภาพมากขึ้น จึงมีการทำซ้ำไปเรื่อยๆ จากข้อผิดพลาดที่เกิดขึ้นทำให้เกิดการพัฒนาและเพิ่มประสิทธิภาพในการทำนายผล

- model: non-PySpark
  - classification method: MLPClassifier
  - import จาก sklearn.neural\_network
- วิเคราะห์ข้อมูลจาก dataset โดยวิธีการแบบ MLP(Multilayer Perceptron) ด้วย MLPClassifier สำหรับการวิเคราะห์ข้อมูล ในวิธี A แบบ non-PySpark model
  - model: PySpark
    - classification method: MultilayerPerceptronClassifier
    - import จาก pyspark.ml.classification
- วิเคราะห์ข้อมูลจาก dataset โดยวิธีการแบบ MultilayerPerceptronClassifier สำหรับการวิเคราะห์ข้อมูล ในวิธี A แบบ PySpark model

### 2.2) วิธี B : DecisionTree

- model: non-PySpark
  - classification method : DecisionTreeClassifier
  - import จาก sklearn.tree
- วิเคราะห์ข้อมูลจาก dataset โดยวิธีการแบบ DecisionTreeClassifier สำหรับการวิเคราะห์ข้อมูล ในวิธี B แบบ non-PySpark model
  - model: PySpark
    - classification method: DecisionTreeClassifier
    - import จาก pyspark.ml.classification
- วิเคราะห์ข้อมูลจาก dataset โดยวิธีการแบบ DecisionTreeClassifier สำหรับการวิเคราะห์ข้อมูล ในวิธี B แบบ PySpark model

### 3.ผลการทดลอง (confusion matrix, accuracy, precision, recall, f1 และเวลาที่ใช้)

layer ที่เราใช้มี 3 layer ทดลองคือ (16,8,2) โดยที่ layer ชั้นแรก เป็น จำนวน column + bias unit ที่ใช้ในการสร้าง model และเวลา learn จะใช้ layer ในการ learn แต่ละครั้ง จนได้ผลลัพธ์ เป็น 2 ค่า คือผู้ชนะคือ counter terrorist หรือ terrorist ในการทำนายในแต่ละครั้ง ทำให้ใช้เวลาที่นานขึ้น ถ้า layer ของ neural network มีขนาดที่เพิ่มขึ้น

#### วิธี A แบบ non-PySpark : MLPClassifier

##### รอบที่ 1

Summary	precision	recall	f1-score	support
CT	0.73	0.78	0.75	12001
T	0.77	0.72	0.74	12481
accuracy	-	-	0.75	24482
macro avg	0.75	0.75	0.75	24482
weighted avg	0.75	0.75	0.75	24482

##### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	9307	2694
Actual Class T	3515	8966

##### รอบที่ 2

Summary	precision	recall	f1-score	support
CT	0.73	0.77	0.75	12001
T	0.77	0.72	0.75	12481
accuracy	-	-	0.75	24482
macro avg	0.75	0.75	0.75	24482
weighted avg	0.75	0.75	0.75	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	9290	2711
Actual Class T	3461	9020

### รอบที่ 3

Summary	precision	recall	f1-score	support
CT	0.73	0.78	0.76	12001
T	0.78	0.73	0.75	12481
accuracy	-	-	0.75	24482
macro avg	0.76	0.75	0.75	24482
weighted avg	0.76	0.75	0.75	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	9411	2590
Actual Class T	3432	9049

### รอบที่ 4

Summary	precision	recall	f1-score	support
CT	0.73	0.78	0.76	12001
T	0.78	0.72	0.75	12481
accuracy	-	-	0.75	24482
macro avg	0.75	0.75	0.75	24482
weighted avg	0.75	0.75	0.75	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	9394	2607
Actual Class T	3460	9021

### รอบที่ 5

Summary	precision	recall	f1-score	support
CT	0.73	0.75	0.74	12000
T	0.75	0.73	0.74	12482
accuracy	-	-	0.74	24482
macro avg	0.74	0.74	0.74	24482
weighted avg	0.74	0.74	0.74	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	9009	2991
Actual Class T	3405	9077

python --- 150.62591242790222 วินาที ---

### วิธี A แบบ PySpark : MultilayerPerceptronClassifier

Summary	รอบที่ 1	รอบที่ 2	รอบที่ 3	รอบที่ 4	รอบที่ 5
accuracy	0.72	0.73	0.74	0.72	0.7
precision	0.68	0.75	0.7	0.88	0.8
recall	0.85	0.71	0.85	0.52	0.55
f1	0.71	0.73	0.73	0.71	0.7

## Confusion Matrix

### รอบที่ 1

	Predict Class CT	Predict Class T
Actual Class CT	6908	5089
Actual Class T	1834	10624

### รอบที่ 2

	Predict Class CT	Predict Class T
Actual Class CT	8964	2926
Actual Class T	3591	8808

### รอบที่ 3

	Predict Class CT	Predict Class T
Actual Class CT	7389	4503
Actual Class T	1922	10751

### รอบที่ 4

	Predict Class CT	Predict Class T
Actual Class CT	11054	842
Actual Class T	5930	6421

### รอบที่ 5

	Predict Class CT	Predict Class T
Actual Class CT	10381	1666
Actual Class T	5596	6772

PySpark --- 480.99161863327026 วินาที---



## วิธี B แบบ non-PySpark : DecisionTreeClassifier

### รอบที่ 1

Summary	precision	recall	f1-score	support
CT	0.65	0.66	0.66	12001
T	0.67	0.67	0.67	12481
accuracy	-	-	0.66	24482
macro avg	0.66	0.66	0.66	24482
weighted avg	0.66	0.66	0.66	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	7916	4085
Actual Class T	4175	8306

### รอบที่ 2

Summary	precision	recall	f1-score	support
CT	0.66	0.65	0.66	12001
T	0.67	0.68	0.68	12481
accuracy	-	-	0.67	24482
macro avg	0.67	0.67	0.67	24482
weighted avg	0.67	0.67	0.67	24482

### Confusion Matrix

	Predict Class CT	Predict Class T
Actual Class CT	7804	4197
Actual Class T	3947	8534

**รอบที่ 3**

Summary	precision	recall	f1-score	support
CT	0.66	0.65	0.66	12001
T	0.67	0.68	0.68	12481
accuracy	-	-	0.67	24482
macro avg	0.67	0.67	0.67	24482
weighted avg	0.67	0.67	0.67	24482

**Confusion Matrix**

	Predict Class CT	Predict Class T
Actual Class CT	7783	4218
Actual Class T	3924	8557

**รอบที่ 4**

Summary	precision	recall	f1-score	support
CT	0.65	0.60	0.62	12001
T	0.64	0.70	0.67	12481
accuracy	-	-	0.65	24482
macro avg	0.65	0.65	0.65	24482
weighted avg	0.65	0.65	0.65	24482

**Confusion Matrix**

	Predict Class CT	Predict Class T
Actual Class CT	7156	4845
Actual Class T	3806	8675

**รอบที่ 5**

Summary	precision	recall	f1-score	support
CT	0.66	0.64	0.65	12000
T	0.66	0.68	0.67	12482
accuracy	-	-	0.66	24482
macro avg	0.66	0.66	0.66	24482
weighted avg	0.66	0.66	0.66	24482

**Confusion Matrix**

	Predict Class CT	Predict Class T
Actual Class CT	7684	4316
Actual Class T	4014	8468

**วิธี B แบบ PySpark: DecisionTreeClassifier**

Summary	รอบที่ 1	รอบที่ 2	รอบที่ 3	รอบที่ 4	รอบที่ 5
accuracy	0.74	0.74	0.74	0.74	0.74
precision	0.79	0.8	0.8	0.83	0.79
recall	0.66	0.66	0.66	0.61	0.66
f1	0.74	0.74	0.74	0.73	0.74

**Confusion Matrix**

**รอบที่ 1**

	Predict Class CT	Predict Class T
Actual Class CT	10024	2133
Actual Class T	4186	7961

## รอบที่ 2

	Predict Class CT	Predict Class T
Actual Class CT	9877	2086
Actual Class T	4222	8256

## รอบที่ 3

	Predict Class CT	Predict Class T
Actual Class CT	9946	2105
Actual Class T	4208	8234

## รอบที่ 4

	Predict Class CT	Predict Class T
Actual Class CT	10511	1545
Actual Class T	4895	7559

## รอบที่ 5

	Predict Class CT	Predict Class T
Actual Class CT	9789	2147
Actual Class T	4263	8139

## สรุปจากผล AVG ทั้ง 2 model ที่ใช้ทั้ง PySpark และ non-PySpark

### 1. model A : MLPClassifier

#### 1.1 non-PySpark

Summary	precision	recall	f1-score	support
---------	-----------	--------	----------	---------

CT	0.73	0.772	0.752	12000.8
T	0.77	0.724	0.746	12481.2
accuracy	-	-	0.748	24482
macro avg	0.75	0.748	0.748	24482
weighted avg	0.75	0.748	0.748	24482

AVG: confusion matrix

	Predict Class CT	Predict Class T
Actual Class CT	9282.2	2718.6
Actual Class T	3454.6	9026.6

1.2 PySpark

AVG : สรุปค่าเฉลี่ยจากทุก 5 รอบ

Summary	
accuracy	0.722
precision	0.762
recall	0.696
f1	0.716

AVG : สรุปจากการ predict class ทั้ง CT , T

	Predict Class CT	Predict Class T
Actual Class CT	8939.2	3005.2
Actual Class T	3774.6	8675.2

จากการสังเกต AVG ของ model-A โดยใช้ method-MLPClassifier แบบ non-PySpark vs. PySpark **สรุปได้ว่า ::**

model แบบ non-PySpark สามารถทำนายผลข้อมูลได้ดีกว่า model แบบ PySpark สังเกตได้จากตาราง confusion matrix ของทั้ง 2 แบบ จะพบว่า model แบบ non-PySpark มีจำนวนข้อมูลที่ทำนายถูกเยอะกว่าแบบ PySpark อธิบายคือ มีค่าของ True positive และ True negative มากกว่า

## 2. model B : DecisionTree

### 2.1 non-PySpark

Summary	precision	recall	f1-score	support
CT	0.656	0.64	0.65	12000.8
T	0.662	0.682	0.674	12481.2
accuracy	-	-	0.662	24482
macro avg	0.662	0.662	0.662	24482
weighted avg	0.662	0.662	0.662	24482

#### AVG : confusion matrix

	Predict Class CT	Predict Class T
Actual Class CT	7668.6	4332.2
Actual Class T	3973.2	8508

### 2.2 PySpark

Summary	Average
accuracy	0.74
precision	0.802
recall	0.65
f1	0.738

#### AVG : confusion matrix

	Predict Class CT	Predict Class T
Actual Class CT	10029.4	2003.2
Actual Class T	4354.8	8029.8

จากการสังเกต AVG ของ model-B โดยใช้ method-DecisionTreeClassifier แบบ non-PySpark vs. PySpark สรุปได้ว่า ::

จากmodel B เปรียบเทียบระหว่างวิธี PySpark และ non-PySpark สามารถสรุปได้ว่า วิธี model PySpark นั้นมีการทำนายที่ถูกต้องมากกว่า model non-PySpark โดยสามารถดูได้จากตาราง Average ของ confusion matrix ของทั้ง2 model จะพบว่า model ของ PySpark นั้นมีข้อมูลการทำนายถูกมากกว่าของ model non-PySpark

## 4.ตารางเปรียบเทียบประสิทธิภาพในแง่เวลาที่ใช้ในการประมวลผล

### MLPClassifier

วิธี A แบบ non-PySpark

รอบที่ 1 - 35.73 วินาที  
รอบที่ 2 - 30.7 วินาที  
รอบที่ 3 - 29.33 วินาที  
รอบที่ 4 - 25.23 วินาที  
รอบที่ 5 - 29.12 วินาที  
เวลาโดยเฉลี่ย - 30.02 วินาที

วิธี A แบบ PySpark

รอบที่ 1 - 52.66 วินาที  
รอบที่ 2 - 153.06 วินาที  
รอบที่ 3 - 143.75 วินาที  
รอบที่ 4 - 66.91 วินาที  
รอบที่ 5 - 56.87 วินาที  
เวลาโดยเฉลี่ย - 94.65 วินาที

### DecisionTreeClassifier

วิธี B แบบ non-PySpark

รอบที่ 1 - 7.34 วินาที  
รอบที่ 2 - 8.96 วินาที  
รอบที่ 3 - 9.01 วินาที  
รอบที่ 4 - 9.03 วินาที  
รอบที่ 5 - 8.68 วินาที  
เวลาโดยเฉลี่ย - 8.604 วินาที

วิธี B แบบ PySpark

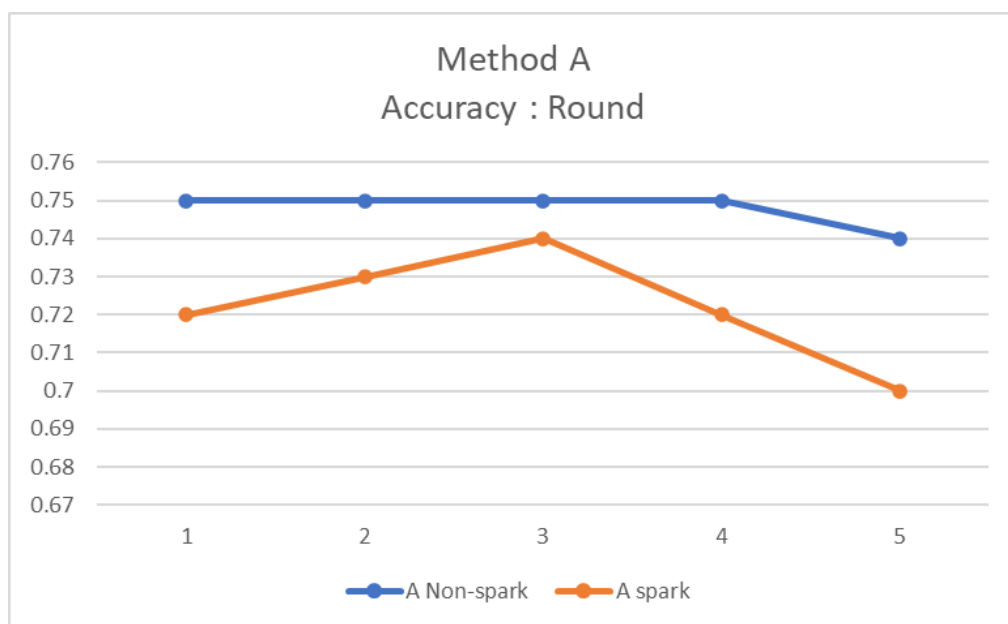
รอบที่ 1 - 50.14 วินาที  
รอบที่ 2 - 31.69 วินาที  
รอบที่ 3 - 31.17 วินาที  
รอบที่ 4 - 32.06 วินาที

รอบที่ 5 - 30.70 วินาที  
เวลาโดยเฉลี่ย - 35.152 วินาที

จะเห็นว่าทำไม non-PySpark ทำการประมวลผลได้เร็วกว่า PySpark เพราะขนาด dataset ของเราใช้ในการทำนายผลแพ้ชนะของเกม CSGO มีจำนวน column และ record ที่อาจจะยังไม่มากพอสำหรับการการประมวลผลแบบ PySpark รวมถึงการใช้ Session ของ PySpark ทำให้ใช้เวลาการทำงานของ PySpark ทำได้ช้ากว่า non PySpark

## 5.กราฟแสดงผลการทดลอง

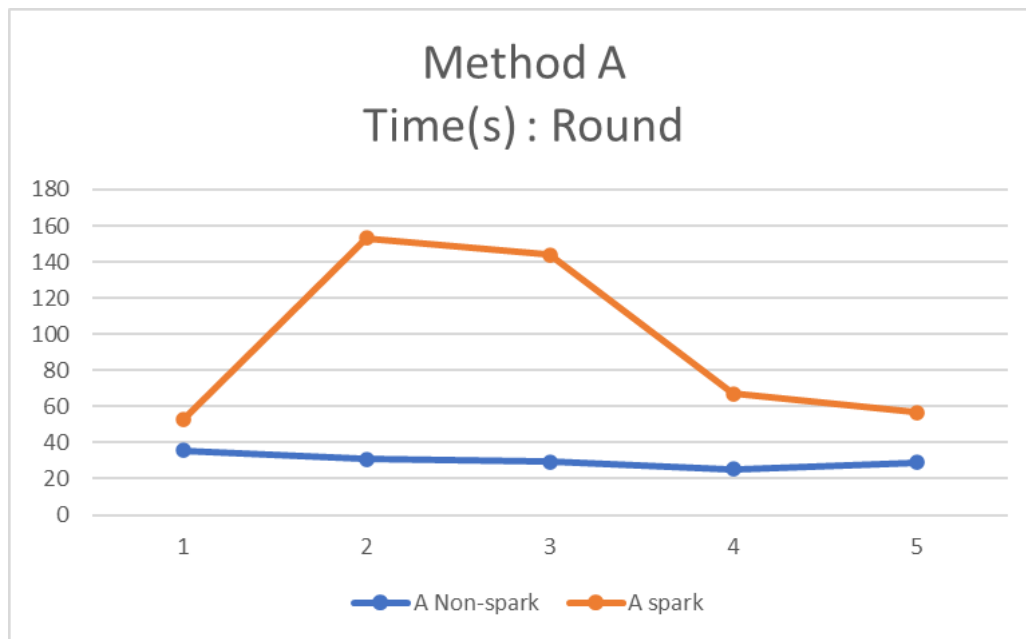
### Method A



Accuracy

- จะเห็นได้ว่าจากกราฟนั้น method ที่เราใช้นั้นถ้าเป็น Non-PySpark นั้นจะมี Accuracy ที่คงที่กว่า PySpark และมี Accuracy ที่สูงกว่าตัวของ PySpark

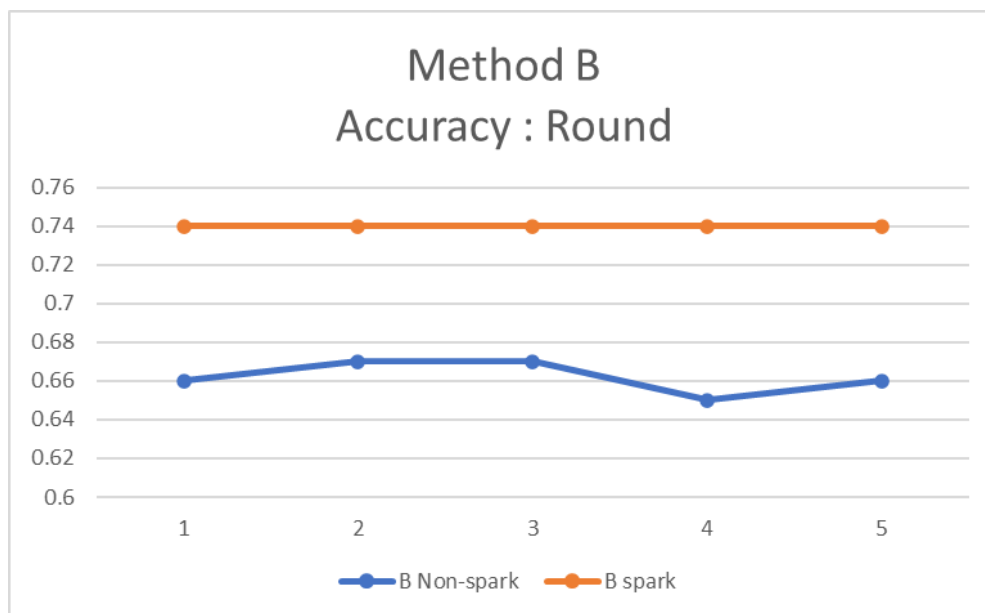




Time

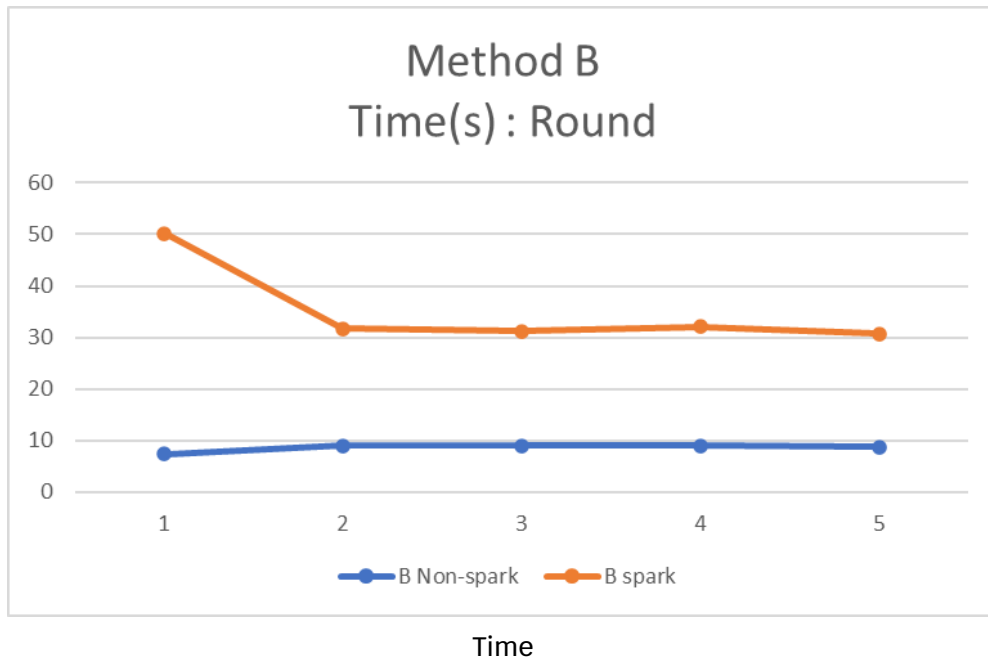
- จะเห็นได้ว่าจากกราฟนั้น method ที่ใช้ PySpark นั้นมีการใช้เวลาที่เยอะหรือสูงกว่า Non-PySpark และ เวลานั้น Non-PySpark นั้นมีความนิ่งกว่าแบบ PySpark

## Method B



Accuracy

- จะเห็นได้ว่าจากกราฟนั้น method ที่เราใช้นั้นถ้าเป็น PySpark นั้นจะมี Accuracy ที่คงที่กว่า Non-PySpark และ มี Accuracy ที่สูงกว่าตัวของ Non-PySpark



- จะเห็นได้ว่าจากกราฟนั้น method ที่ใช้ PySpark นั้นมีการใช้เวลาที่เยอะหรือสูงกว่า Non-PySpark และ เวลานั้น Non-PySpark นั้นมีความนิ่งกว่าแบบ PySpark

## 6.สรุปผล

จากผลการทดสอบ เป็นจำนวน 5 ครั้ง ของ วิธี A และ B สรุปได้ว่า

วิธี A แบบ non-PySpark (MLPClassifier :: sklearn.neural\_network)

- ใช้เวลาโดยเฉลี่ยที่ 30.02 วินาที มีความถูกต้อง(accuracy) ในการทำนายผลแพ้ชนะของเกมส์เฉลี่ยที่ 74.8%

วิธี A แบบ PySpark(MLPClassifier :: pyspark.ml.classification)

- ใช้เวลาโดยเฉลี่ยที่ 94.65 วินาที มีความถูกต้อง(accuracy) ในการทำนายผลแพ้ชนะของเกมส์เฉลี่ยที่ 72.2%

วิธี A แบบ non-PySpark จะใช้เวลาที่น้อยกว่า ในการประเมินผล/วิเคราะห์ จึงสรุปได้ว่า method: MLPClassifier(sklearn.neural\_network) มีความเหมาะสมกว่า ในเรื่องของการใช้เวลาประเมินผลที่น้อยกว่า แบบ PySpark

วิธี B แบบ non-PySpark (DecisionTreeClassifier :: sklearn.tree)

- ใช้เวลาโดยเฉลี่ยที่ 8.604 วินาที มีความถูกต้อง(accuracy) ในการทำนายผลแพ้ชนะของเกมส์เฉลี่ยที่ 66.2%

วิธี B แบบ PySpark(DecisionTreeClassifier :: pyspark.ml.classification)

- ใช้เวลาโดยเฉลี่ยที่ 35.152 วินาที มีความถูกต้อง(accuracy) ในการทำนายผลแพ้ชนะของเกมส์เฉลี่ยที่ 74%

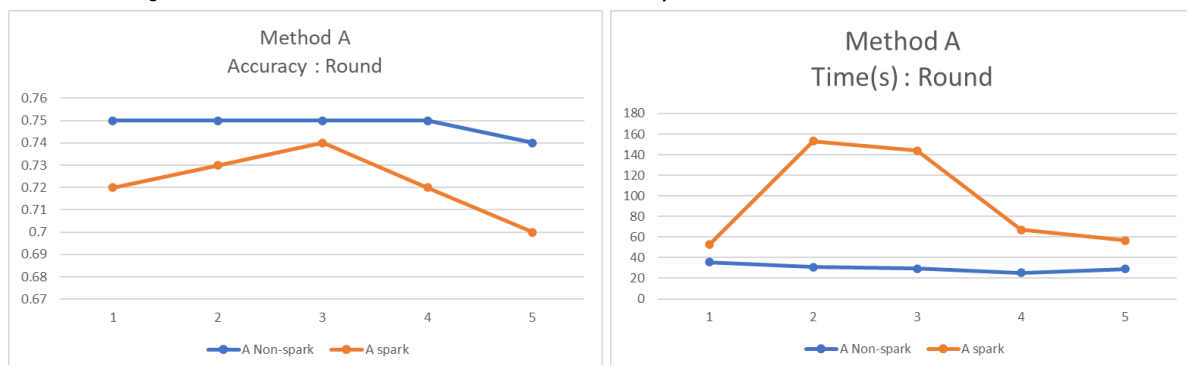
วิธี B แบบ non-PySpark จะใช้เวลาที่น้อยกว่า ในการประเมินผล/วิเคราะห์ จึงสรุปได้ว่า method: DecisionTreeClassifier(sklearn.tree) มีความเหมาะสมกว่า ในเรื่องของการใช้เวลาประเมินผลที่น้อยกว่า แบบ PySpark

อย่างไรก็ตาม data set นี้ มีปริมาณของข้อมูลที่น้อย และยังไม่เหมาะกับการดึงประสิทธิภาพของ method จาก PySparkเท่าไร จึงส่งผลให้เวลาโดยเฉลี่ยในการวิเคราะห์ข้อมูลมากกว่าแบบ non-PySpark

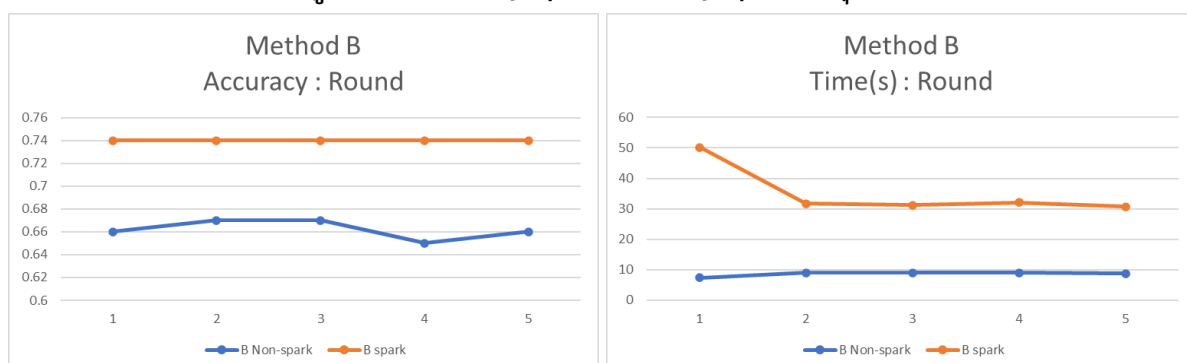
ดังนั้นการใช้ pure-python / non-PySpark มีความเหมาะสมกว่าสำหรับการวิเคราะห์ข้อมูลใน data set นี้ จากผลสรุปของเวลาโดยเฉลี่ยในการวิเคราะห์ข้อมูล

สรุปผลจากกราฟเปรียบเทียบ accuracy & time ของการวิเคราะห์ข้อมูลแบบ non-PySpark vs. PySpark(โดยใช้กราฟเส้นในการสรุปผล/เปรียบเทียบ)

- method:A (MLPClassifier) เมื่อเทียบผลของ accuracy & time จากการวิเคราะห์ข้อมูลแบบ non-PySpark vs. PySparkสรุปได้ว่า



- non-PySpark นั้น มีค่า “Accuracy” ที่มั่นคง และสูงกว่า PySpark อีกทั้งการเปรียบเทียบโดยใช้ “Time(s)” เป็นตัววัด พบว่า non-PySpark ใช้เวลาในการวิเคราะห์ข้อมูลที่น้อยกว่า และน้อยกว่าแบบ PySpark
- ดังนั้น การวิเคราะห์ข้อมูลสำหรับ data set นี้ โดยใช้ classification method: MLPClassifier แบบ non-PySpark มีความเหมาะสมในการวิเคราะห์มากกว่า PySpark
- method:B (DecisionTreeClassifier) เมื่อเทียบผลของ accuracy & time จากการวิเคราะห์ข้อมูลแบบ non-PySpark vs. PySpark สรุปได้ว่า



- PySpark มีค่า “Accuracy” ที่คงที่ และสูงกว่า non-PySpark แต่เมื่อเปรียบเทียบโดยใช้ “Time(s)” เป็นตัววัด พบว่า non-PySpark มีการใช้เวลาในการวิเคราะห์ที่น้อย และน้อยกว่า PySpark
- ดังนั้น การวิเคราะห์ข้อมูลสำหรับ data set นี้ โดยใช้ classification method: DecisionTreeClassifier แบบ PySpark อาจมีความเหมาะสมกว่า non-PySpark ในเรื่องของ Accuracy/ความแม่นยำ แต่อย่างไรก็ตาม PySpark ยังคงวิเคราะห์ข้อมูลโดยใช้ระยะเวลาที่มากกว่า non-PySpark อยู่ดี

## บรรณานุกรม

CHRISTIAN LILLELUND.(2563).CS:GO Round Winner Classification.[ออนไลน์].

เข้าถึงได้จาก

<https://www.kaggle.com/datasets/christianlillelund/csgo-round-winner-classification> (วันที่สืบค้น : 21 พฤศจิกายน 2565)

Pranav Thaenraj.(2564). Spark MultiLayer Perceptron Classifier for POI

Classification.[ออนไลน์].เข้าถึงได้จาก

<https://towardsdatascience.com/spark-multilayer-perceptron-classifier-for-poi-classification-99e5c68b4a77> (วันที่สืบค้น : 21 พฤศจิกายน 2565)

Vicky Zhang.(2559). Why does my Spark run slower than pure Python?

Performance comparison.[ออนไลน์].เข้าถึงได้จาก

<https://stackoverflow.com/questions/34625410/why-does-my-spark-run-slower-than-pure-python-performance-comparison> (วันที่สืบค้น : 21 พฤศจิกายน 2565)

scikit-learn developers.(2550).sklearn.preprocessing.StandardScaler.[ออนไลน์].

เข้าถึงได้จาก

<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> (วันที่สืบค้น : 28 พฤศจิกายน 2565)

scikit-learn developers.(2550).sklearn.tree.DecisionTreeClassifier.[ออนไลน์].

เข้าถึงได้จาก

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> (วันที่สืบค้น : 28 พฤศจิกายน 2565)

scikit-learn developers.(2550).sklearn.neural\_network.MLPClassifier.[ออนไลน์].

เข้าถึงได้จาก

[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html) (วันที่สืบค้น : 28 พฤศจิกายน 2565)

scikit-learn developers.(2550).sklearn.metrics.classification\_report.[ออนไลน์].

เข้าถึงได้จาก

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification\\_report.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html) (วันที่สืบค้น : 28 พฤศจิกายน 2565)

scikit-learn developers.(2550).sklearn.metrics.confusion\_matrix.[ออนไลน์].เข้าถึงได้จาก

[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) (วันที่สืบค้น : 28 พฤศจิกายน 2565)

Apache Spark developers.(2557).StringIndexer.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.feature.StringIndexer.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Apache Spark developers.(2557).VectorAssembler.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/3.1.3/api/python/reference/api/pyspark.ml.feature.VectorAssembler.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Apache Spark developers.(2557).MultilayerPerceptronClassifier.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.MultilayerPerceptronClassifier.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Apache Spark developers.(2557).DecisionTreeClassifier.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.classification.DecisionTreeClassifier.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Apache Spark developers.(2557).MulticlassMetrics.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.mllib.evaluation.MulticlassMetrics.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Apache Spark developers.(2557).MulticlassClassificationEvaluator.[ออนไลน์].เข้าถึงได้จาก

<https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.MulticlassClassificationEvaluator.html> (วันที่สืบค้น : 30 พฤศจิกายน 2565)

Vicky Zhang.(2559). Why does my Spark run slower than pure Python?

Performance comparison.[ออนไลน์].เข้าถึงได้จาก

<https://stackoverflow.com/questions/34625410/why-does-my-spark-run-slower-than-pure-python-performance-comparison> (วันที่สืบค้น : 30 พฤศจิกายน 2565)