

Homework Topic: Regression Analysis & Random Forest

Week : 03 (Class date : 5-Nov-22)

Student Name : Puriwat Sangrawee
Student ID : 65056071

1. จากการศึกษาเพื่อพยากรณ์จำนวนวันเฉลี่ยที่นักท่องเที่ยวใช้ในการพักผ่อน ซึ่งมีความสัมพันธ์กับจำนวนปีที่ทำงาน (Years of working) และรายได้ต่อปี (Yearly income) จึงใช้ข้อมูลในไฟล์ HW04.xlsx เพื่อสร้างโมเดล Linear Regression ของข้อมูลดังกล่าวโดยใช้ Azure Machine Learning

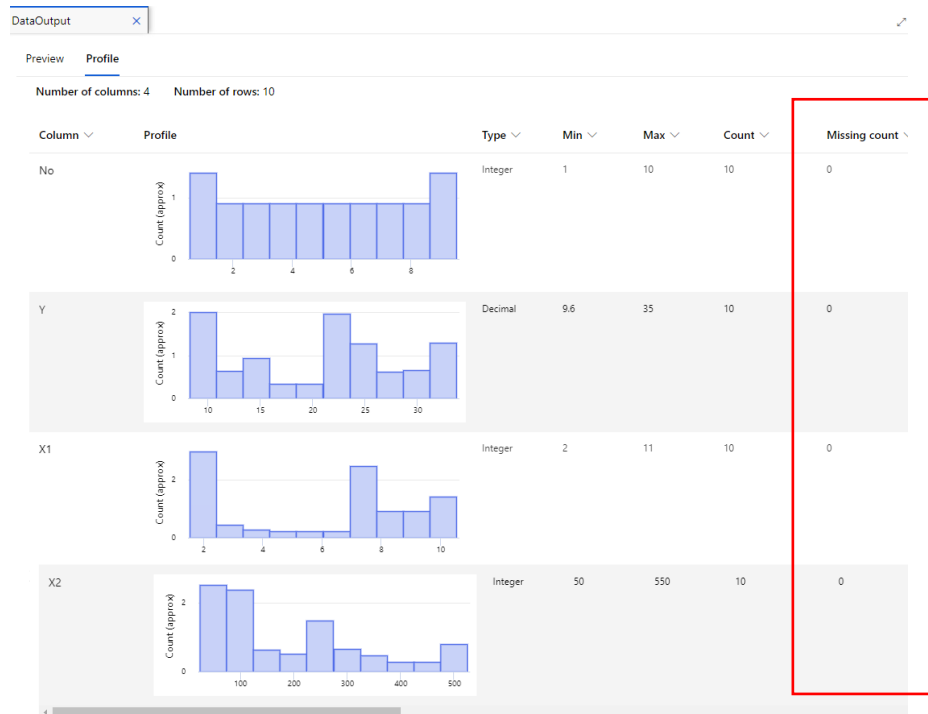
Y : จำนวนวันเฉลี่ยที่นักท่องเที่ยวใช้ในการท่องเที่ยว

X1 : จำนวนปีที่ทำงาน

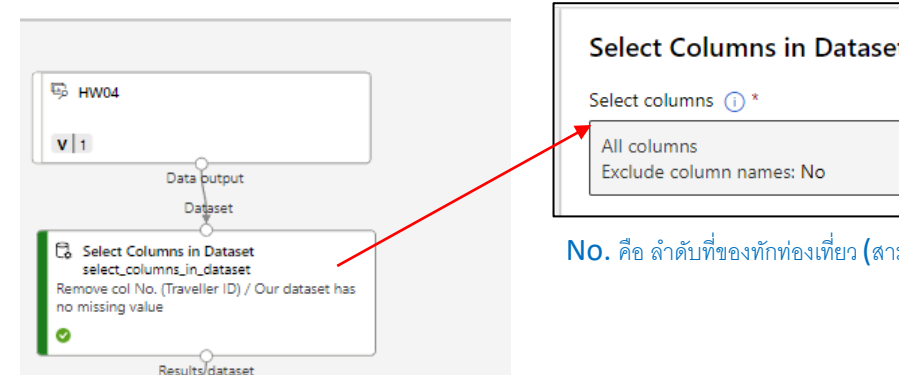
X2 : รายได้ต่อปี

1) แสดงหน้าจอการเตรียมข้อมูล

1.1) เริ่มต้นจากการตรวจสอบค่า Missing Value พบว่าข้อมูลชุดนี้ไม่มีค่า Missing value จึงข้ามขั้นตอนการ clean data ได้เลย

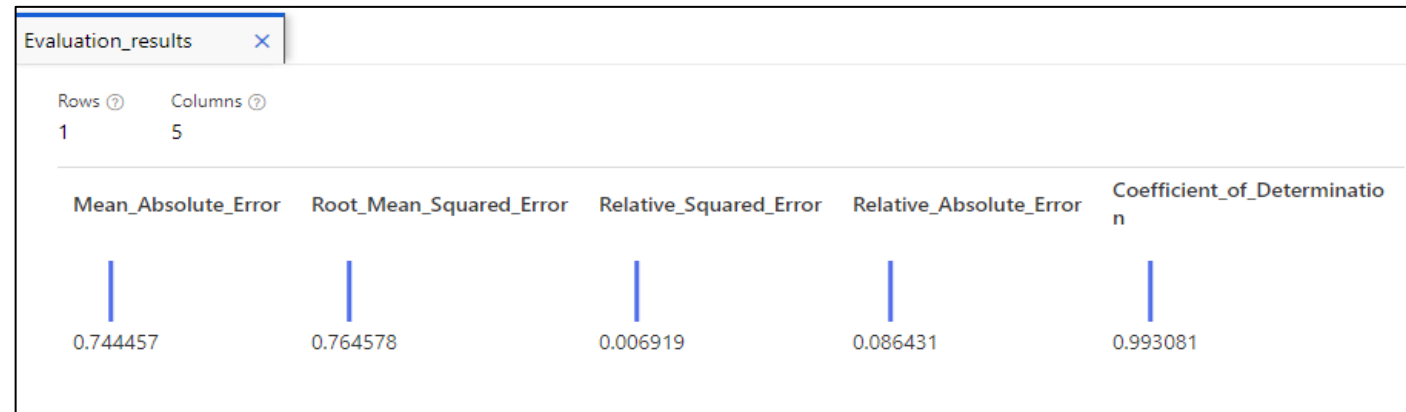


1.2) จากนั้นทำการเลือกคอลัมน์เฉพาะที่จำเป็นเพื่อนำไปเทรนโมเดลในขั้นตอนถัดไป



No. คือ ลำดับที่ของนักท่องเที่ยว (สามารถตัดออกได้)

2) แสดงภาพผลการวัดประสิทธิภาพโมเดล



สำหรับข้อมูลชุดนี้ เมื่อกำหนดค่าพารามิเตอร์ต่างๆ เท่ากัน และทำการเปรียบเทียบ ระหว่างข้อมูลที่ผ่านมาและไม่ผ่านการ Normalize พบว่า ประสิทธิภาพเท่ากัน จึงแสดงผลเฉพาะวิธีนี้

และ Overall Process Flowchart เป็นไปตามหน้าที่ 2



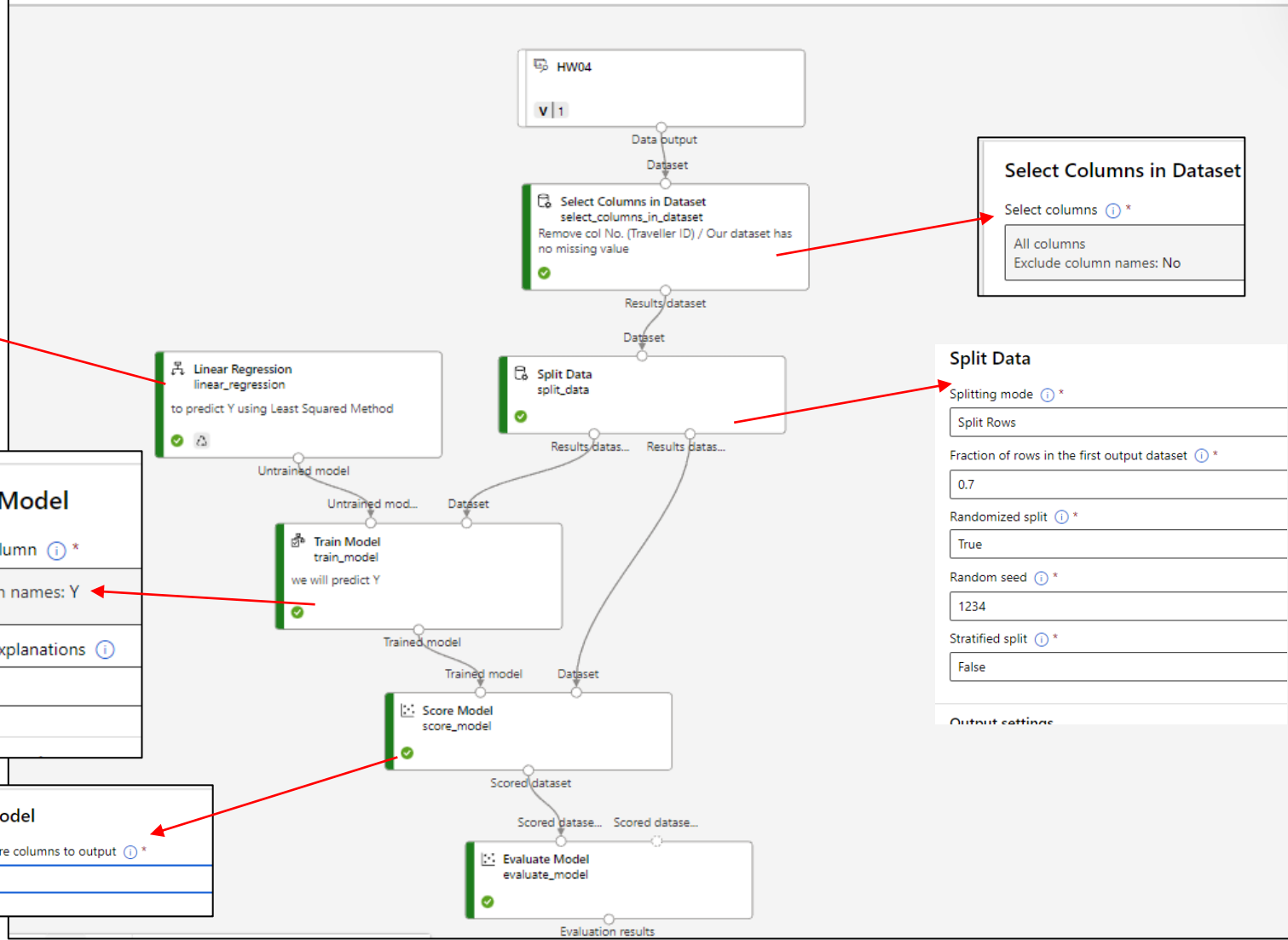
[Sign out](#)

DE_HW_wk03_article01

python 3.8. This may impact your component outputs and/or endpoint deployments from inference pipelines. [Learn more](#)

[Clone](#) [Resubmit](#) [Publish](#) [Show lineage](#) [Create inference pipeline](#) [Delete](#)

DE_HW_wk03_article01 [Completed](#)



Linear Regression

Solution method ⓘ *

Ordinary Least Squares

L2 regularization weight ⓘ *

0.001

Include intercept term ⓘ *

True

Random number seed ⓘ

1234

Train Model

Label column ⓘ *

Column names: Y

Model explanations ⓘ

False

Score Model

Append score columns to output ⓘ *

True

Select Columns in Dataset

Select columns ⓘ *

All columns

Exclude column names: No

Split Data

Splitting mode ⓘ *

Split Rows

Fraction of rows in the first output dataset ⓘ *

0.7

Randomized split ⓘ *

True

Random seed ⓘ *

1234

Stratified split ⓘ *

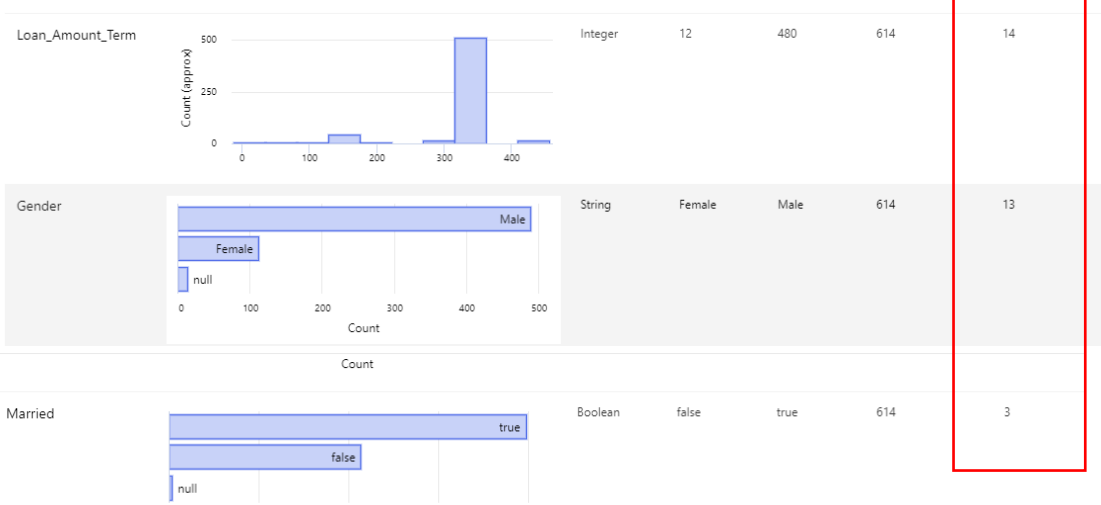
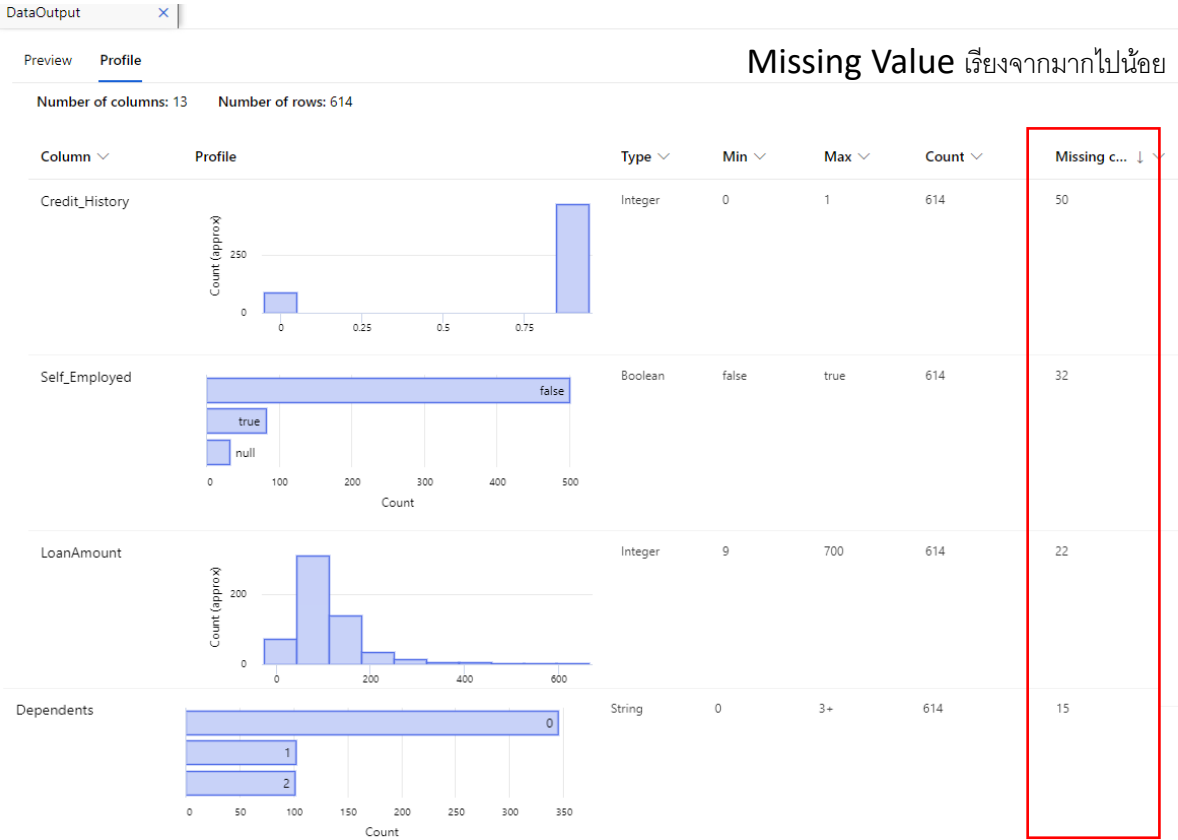
False

2. ในการพิจารณาวงเงินกู้ของธนาคารแห่งหนึ่งจะพิจารณาจากปัจจัยหลายๆ อย่างเช่น การศึกษา สถานะภาพ เพศ และอื่น ๆ จากข้อมูลการกู้เงินของลูกค้าจำนวน 614 รายที่ธนาคารเก็บไว้ (Train_loan.csv) ให้ทำการสร้างโมเดลโดยใช้ Azure Machine Learning เพื่อพิจารณาอนุมัติการกู้เงิน ของธนาคารดังกล่าวโดยใช้หลักการต่าง ๆ ของวิทยาการข้อมูลอย่างครบถ้วน และตอบคำถามต่อไปนี้ โดยใช้ Logistic Regression (แบ่งข้อมูล Train:Test 70:30, Random seed = 1234)

1) Accuracy = **74.1%** 2) Precision = **66.7%**
3) Recall = **90.6%** 4) F1 Score = **76.8%**

>>>ข้อมูลชุดดังกล่าวประกอบด้วยข้อมูลเริ่มต้น 13 คอลัมน์ 614 รายการ

1. ในขั้นตอนแรก เราจะทำการ Explore/Replace Missing Value กันก่อน



และเพื่อให้เราสามารถนำประโยชน์จากข้อมูลได้อย่างเต็มที่ เราจะไปเปลี่ยนข้อมูลต้นทางจาก categorical เป็น Numerical เพื่อให้เข้ามาเข้าสมการได้ โดยการเปลี่ยนแปลงเป็นไปตามด้านล่าง จากนั้นอัปโหลด dataset เข้ามาใหม่

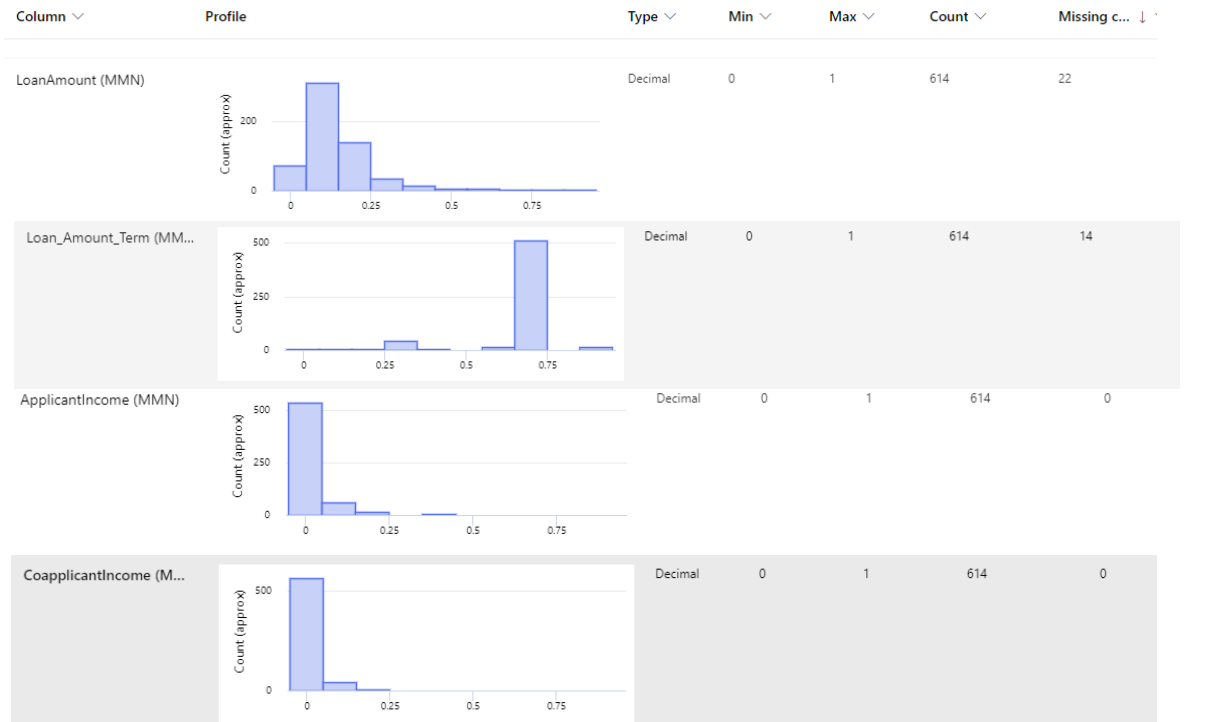
Column	Old value	New Value
Married	Yes (True)	1
	No (False)	0
Dependency	3+	3
Education	Graduated	1
	Not Graduated	0
Self_Employed	Yes (True)	1
	No (False)	0
Gender	Male	1
	Female	2
Property	Rural	1
	Urban	2
	Semiurban	3

แต่เนื่องจากเกิด **Error** บางอย่างใน **Program** ทำให้ผมไม่สามารถใช้คำสั่ง **Normalize** ได้ (ผมลองใช้แล้ว หากเลือกแค่กับ **column ApplicantIncome** คอลัมน์เดียวจะใช้ได้ครับ รูปแบบอื่นคือใช้ไม่ได้เลย บางทีก็ **Error SMOTE** หรือ **Clean Missing data** แบบไม่ทราบสาเหตุ ทั้ง ๆ ที่ทำเหมือนตัวอย่างในห้องเรียน) ผมเลยขออนุญาตทำ **Min-Max Normalize** ก่อน **Import dataset** ครับ

ภาพรวมของข้อมูล

Number of columns: 17 Number of rows: 50 (of 614)																
loan_ID	Gender	Married	Depen...	Educati...	Self_E...	Applica...	Applica...	Coappli...	Coappli...	LoanA...	LoanA...	Loan_A...	Loan_A...	Credit_...	Proper...	Loan_S...
P001002	1	0	0	1	0	5849	0.07	0	0	null	null	360	0.744	1	2	true
P001003	1	1	1	1	0	4583	0.055	1508	0.036	128	0.172	360	0.744	1	1	false
P001005	1	1	0	1	1	3000	0.035	0	0	66	0.082	360	0.744	1	2	true
P001006	1	1	0	0	0	2583	0.03	2358	0.057	120	0.161	360	0.744	1	2	true
P001008	1	0	0	1	0	6000	0.072	0	0	141	0.191	360	0.744	1	2	true
P001011	1	1	2	1	1	5417	0.065	4196	0.101	267	0.373	360	0.744	1	2	true

ทำการเช็ค **Missing Value** สำหรับคอลัมน์ที่เพิ่มมาใหม่ (4 คอลัมน์)



จากนั้นจะทำการลบคอลัมน์ที่ไม่จำเป็นทิ้งไป (ประกอบด้วย4 คอลัมน์บนก่อน **MMN** และ **Loan_ID**)

และจากนั้นจะทำการเตรียมข้อมูลต่อการ **Replace Missing value** ตามตารางด้านล่าง

NO.	Col Name with Missing Value	How to Solve
1	Credit_History	Replace with Mode
2	Self_Employed	Replace with Mode (FALSE)
3	LoanAmout (MMN)	Replace with Mean
4	Dependents	Replace with Mode
5	Loan_Amount_Term (MMN)	Replace with Mode
6	Gender	Remove Entire Row
7	Married	Remove Entire Row

สำหรับขั้นตอนต่อจากนี้สามารถดูรายละเอียดเพิ่มเติมได้ใน **Process flowchart** (หน้า 5-6)

Puriwat SangraweeX
65056071@KMITL.AC.TH
Switch Directory

Sign out

Select Columns in Dataset

Overview Parameters Outputs + logs

Refresh Register model Connect to compute

Select columns 1 *

Edit column

All columns
Exclude column names:
Loan_ID, ApplicantIncome, CoapplicantIncome, LoanAmount,
Loan_Amount_Term

Clean Missing Data

Parameters

Refresh Register model

Columns to be cleaned 1 *

Column names: Gender, Married

Minimum missing value ratio 1 *

0.0

Maximum missing value ratio 1 *

1.0

Cleaning mode 1 *

Remove entire row

Clean Missing Data

Parameters

Refresh Register model

Columns to be cleaned 1 *

Column names:
Credit_History, Self_Employed, Dependents,
Loan_Amount_Term (MMN)

Minimum missing value ratio 1 *

0.0

Maximum missing value ratio 1 *

1.0

Cleaning mode 1 *

Replace with mode

Generate missing value indicator column 1 *

False

Cols with all missing values 1 *

Remove

Clean Missing Data

Overview Parameters

Refresh Register model

Columns to be cleaned 1 *

Column names: LoanAmount (MMN)

Minimum missing value ratio 1 *

0.0

Maximum missing value ratio 1 *

1.0

Cleaning mode 1 *

Replace with mean

Generate missing value indicator column 1 *

False

Cols with all missing values 1 *

Remove

SMOTE

Overview Parameters

Refresh Register model

Label column 1 *

Column names: Loan_Status

SMOTE percentage 1 *

120

Number of nearest neighbors 1 *

2

Random seed 1 *

1234

5



Two-Class Logistic Regression

Overview Parameters Outputs + logs

Refresh + Register model Connect to

Create trainer mode ⓘ *

SingleParameter

Optimization tolerance ⓘ *

1e-07

L2 regularization weight ⓘ *

1.0

Random number seed ⓘ

1234

Filter Based Feature Selection

Overview Parameters ...

Refresh + Register model ...

Operate on feature columns only ⓘ

True

Number of desired features ⓘ *

6

Feature scoring method ⓘ *

PearsonCorrelation

Target column ⓘ *

Column names: Loan_Status

Split Data

Overview Parameters Outputs + logs

Refresh + Register model Connect to

Splitting mode ⓘ *

Split Rows

Fraction of rows in the first output dataset ⓘ *

0.7

Randomized split ⓘ *

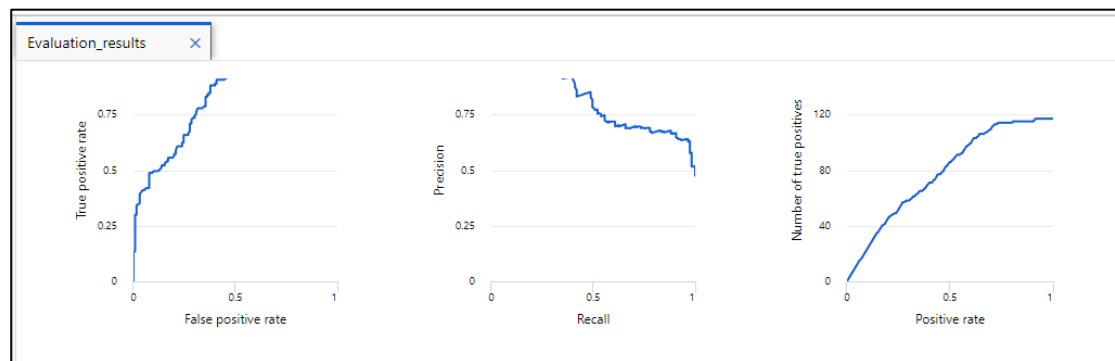
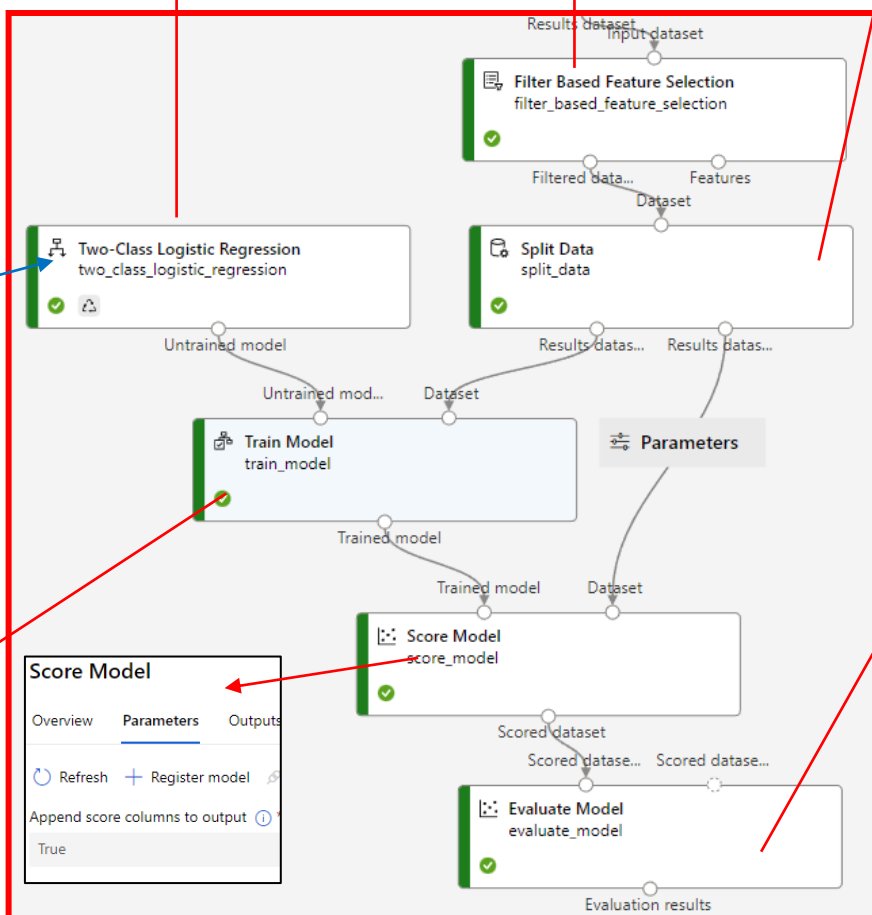
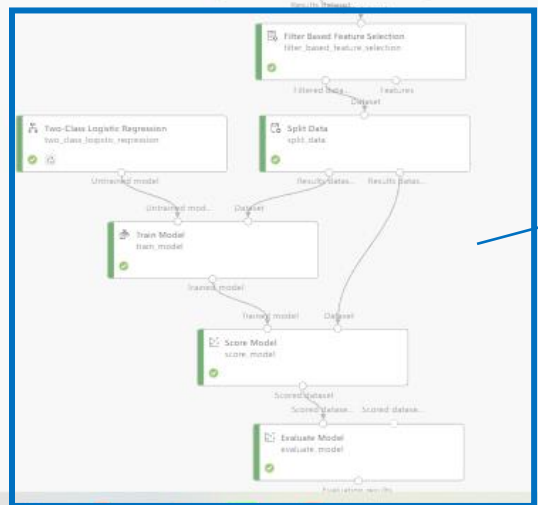
True

Random seed ⓘ *

1234

Stratified split ⓘ *

False



Threshold 0.5

Accuracy 0.741

Precision 0.667

Recall 0.906

F1 Score 0.768

AUC 0.822

		Actual	
		True	False
Predicted	True	106	53
	False	11	77

Train Model

Overview Parameters Outputs + logs

Refresh + Register model Connect to

Label column ⓘ *

Column names: Loan_Status

Model explanations ⓘ

False

Score Model

Overview Parameters Outputs

Refresh + Register model

Append score columns to output ⓘ

True

3. จากข้อมูลของผู้โดยสารเรือไททานิค จงใช้สร้างโมเดลทำนายการรอดชีวิตของผู้โดยสาร โดยใช้ Decision Forest และกำหนดค่าพารามิเตอร์ต่าง ๆ ดังรูป (แบ่งข้อมูล Train:Test 70:30, Random seed = 1234)

สังเกตค่า Model1

Accuracy = **82.8%**
F1 Score = **79.1%**

ทดลองทำการปรับพารามิเตอร์ของแบบจำลองดังนี้

- 1) เพิ่มจำนวน Minimum number of samples per leaf node =5 ค่า Accuracy เปลี่ยนแปลงหรือไม่อย่างไร

สังเกตค่า Model2

Accuracy = **83.1%**
F1 Score = **78.0%**

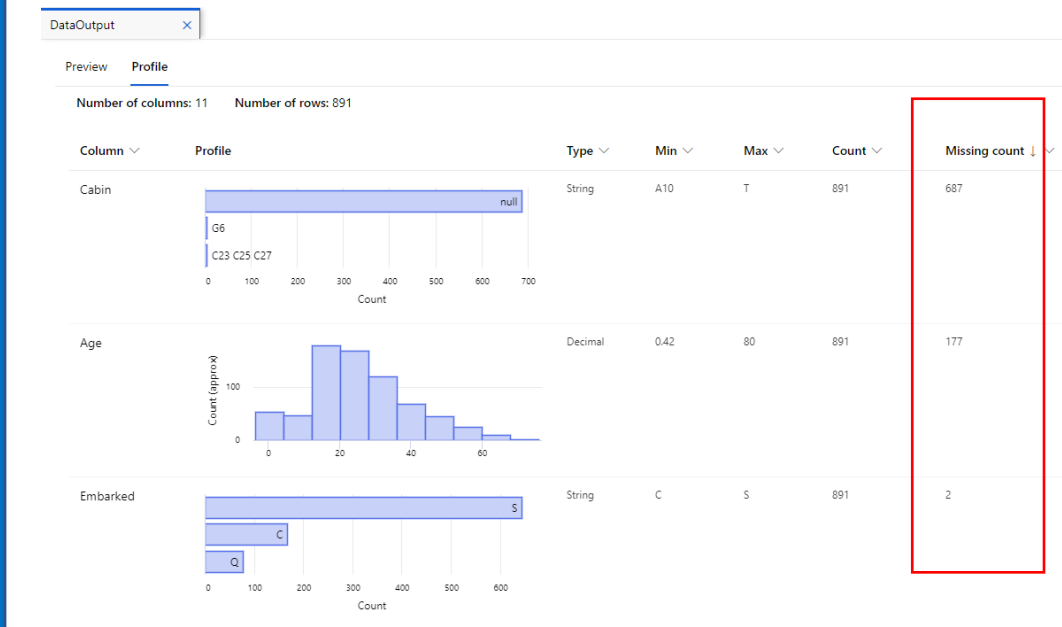
- 2) จากข้อ 1) ทำการลด Maximum depth of the decision trees = 6 ค่า Accuracy เปลี่ยนแปลงหรือไม่อย่างไร

สังเกตค่า Model2

Accuracy = **82.8%**
F1 Score = **77.7%**

- 3) ควรเลือก Model ใดมาใช้งาน เพราะเหตุใด

ตามปกติแล้วการเลือกโมเดลใดขึ้นอยู่กับประเภทของงาน แต่ในกรณีนี้เป็นโมเดลที่เกี่ยวกับชีวิตคน อาจนำไปออกแบบระบบเครื่องจักร หรือการวางแผนบริหารจัดการบนเรือสำราญเพื่อป้องกันการสูญเสียชีวิต ในลักษณะนี้ ค่า **recall** จึงมีความสำคัญเพิ่มขึ้น และค่าที่สามารถ **cover** ได้ทั้งค่า **Precision** และ **recall** ก็คือ **F1 score** ดังนั้นที่ **Threshold 50%** เราจะเลือกตัวที่ **F1 Score** สูงสุด นั่นคือ **Model1**



ข้อมูลส่วนใหญ่ set parameter เหมือน KDAI15 : Decision Tree Model แต่มีการเปลี่ยนแปลง Random Seed ใน split data เป็น 1234 เปลี่ยน Model Algorithm จาก Decision Tree เป็น Decision Forest

→ ดู Model Process Flowchart เพิ่มเติมที่หน้า 8-10



[Sign out](#)

Default value

Two-Class Decision Forest

Create trainer mode ⓘ *

SingleParameter

Number of decision trees ⓘ *

8

Maximum depth of the decision trees ⓘ *

32

Minimum number of samples per leaf node ⓘ *

1

Resampling method ⓘ *

Bagging Resampling

Train Model

Label column ⓘ *

Column names: Survived

Model explanations ⓘ

False

Score Model

Append score columns to output ⓘ *

True

Filter Based Feature Selection

Operate on feature columns only ⓘ

True

Number of desired features ⓘ *

6

Feature scoring method ⓘ *

ChiSquared

Target column ⓘ *

Column names: Survived

Split Data

Splitting mode ⓘ *

Split Rows

Fraction of rows in the first output dataset ⓘ *

0.7

Randomized split ⓘ *

True

Random seed ⓘ *

1234

Stratified split ⓘ *

False

Clean Missing Data

Columns to be cleaned ⓘ *

Column names: Age

Minimum missing value ratio ⓘ *

0.0

Maximum missing value ratio ⓘ *

1.0

Cleaning mode ⓘ *

Replace with median

Generate missing value indicator column ⓘ *

False

Cols with all missing values ⓘ *

Remove

Clean Missing Data

Columns to be cleaned ⓘ *

Column names: Embarked

Minimum missing value ratio ⓘ *

0.0

Maximum missing value ratio ⓘ *

1.0

Cleaning mode ⓘ *

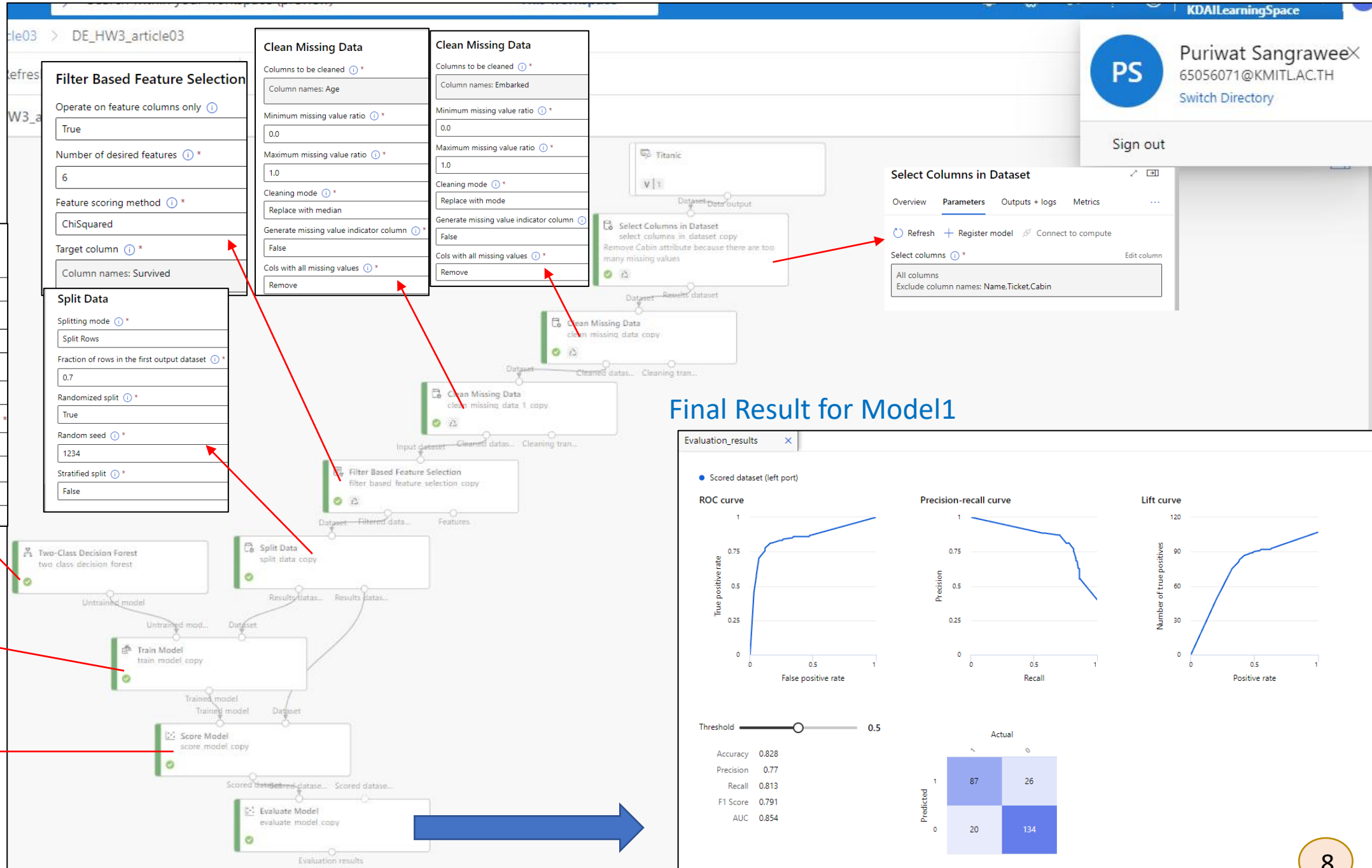
Replace with mode

Generate missing value indicator column ⓘ

False

Cols with all missing values ⓘ *

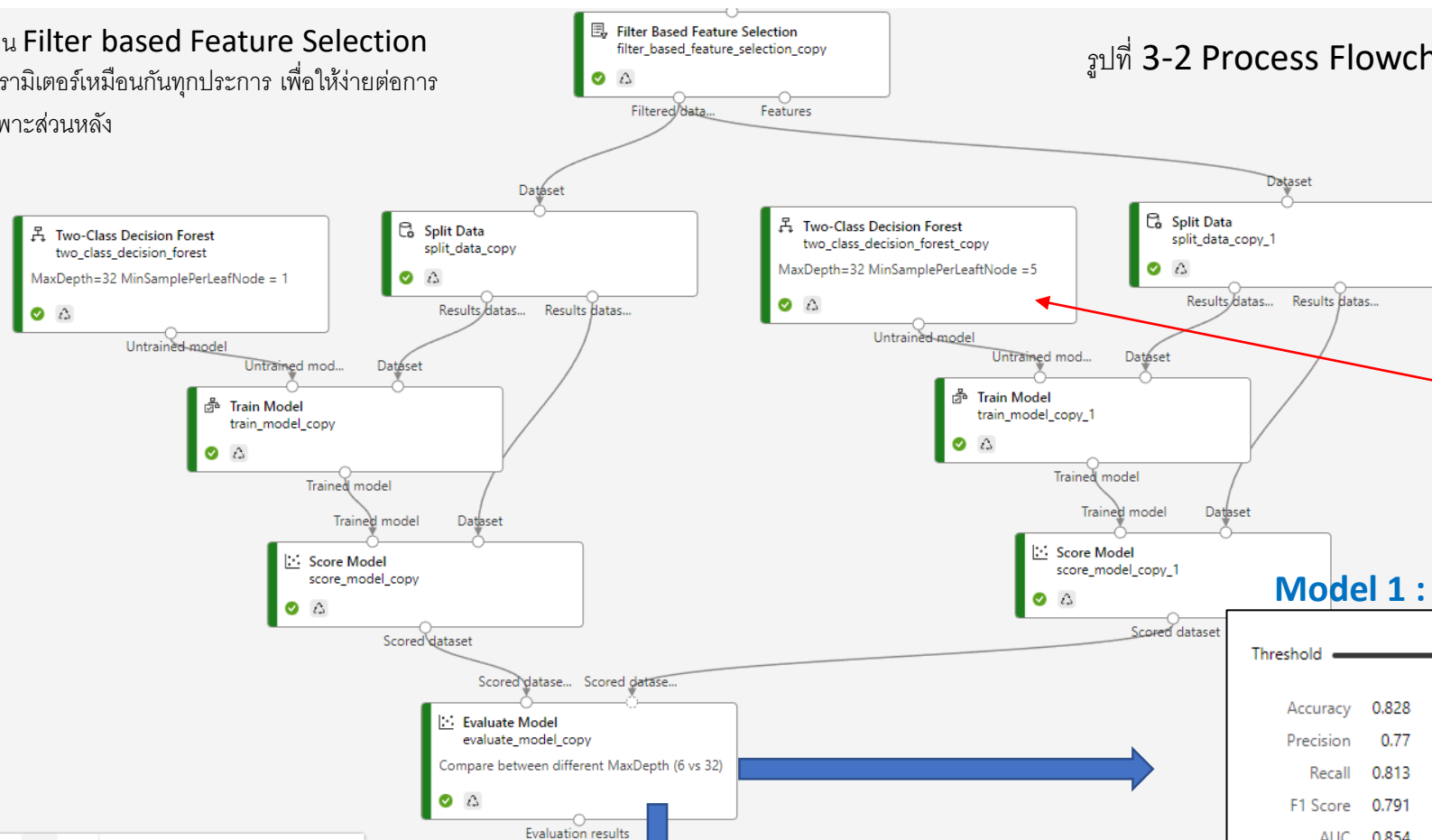
Remove



รูปที่ 3-1 Process Flowchart และการ set ค่าใน Model 1

หมายเหตุ : ในขั้นตอนก่อน Filter based Feature Selection
ขั้นตอนการทำงานและพารามิเตอร์เหมือนกันทุกประการ เพื่อให้ง่ายต่อการ
อ่าน จึงทำการตัดลงมาเฉพาะส่วนหลัง

รูปที่ 3-2 Process Flowchart after Configuring Parameter (Model1 vs Model 2-1)



Two-Class Decision Forest

Create trainer mode ⓘ *

SingleParameter

Number of decision trees ⓘ *

8

Maximum depth of the decision trees ⓘ *

32

Minimum number of samples per leaf node ⓘ *

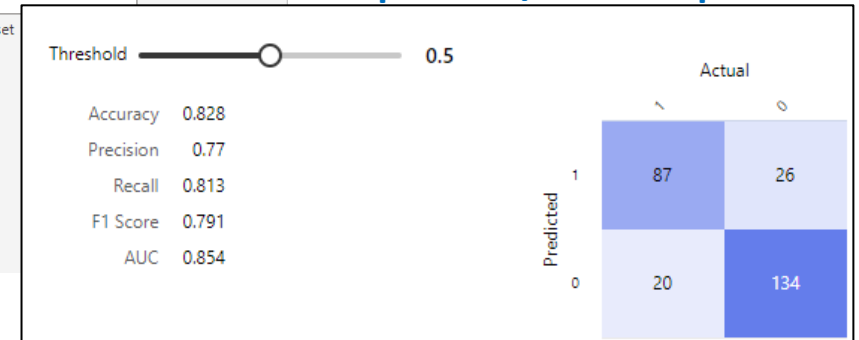
5

Resampling method ⓘ *

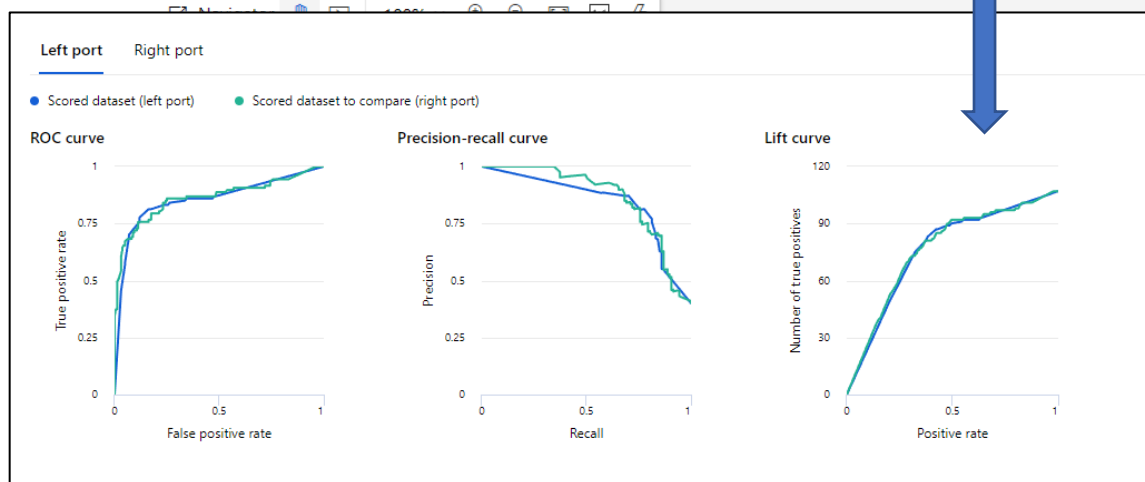
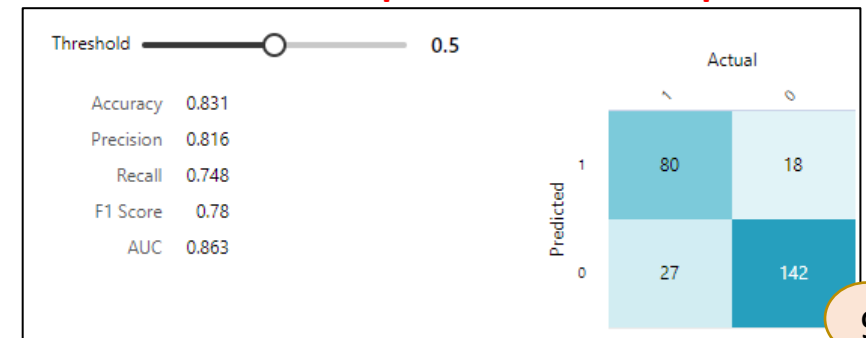
Bagging Resampling

Output settings

Model 1 : MaxDepth= 32 / MinSample=1



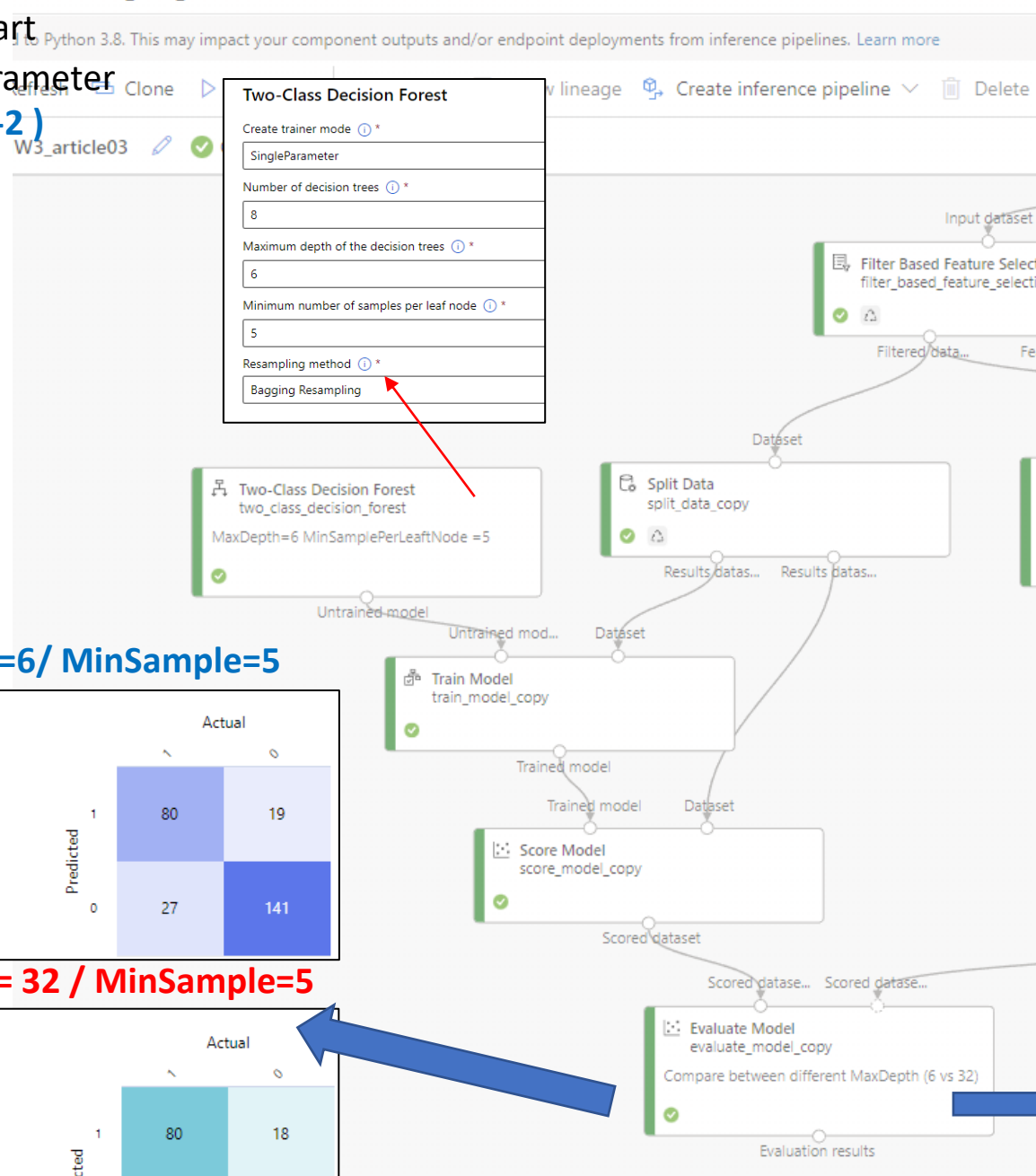
Model 2-1 : MaxDepth= 32 / MinSample=5



Sign out

หมายเหตุ : ในขั้นตอนก่อน Filter based Feature Selection ขั้นตอนการทำงานและพารามิเตอร์เหมือนกันทุกประการ เพื่อให้ง่ายต่อการอ่าน จึงทำการตัดลงมาเฉพาะส่วนหลัง

รูปที่ 3-3 Process Flowchart after Configuring Parameter (Model 2-1 vs Model 2-2)



Two-Class Decision Forest

Create trainer mode ⓘ *

SingleParameter

Number of decision trees ⓘ *

8

Maximum depth of the decision trees ⓘ *

6

Minimum number of samples per leaf node ⓘ *

5

Resampling method ⓘ *

Bagging Resampling

Two-Class Decision Forest

Create trainer mode ⓘ *

SingleParameter

Number of decision trees ⓘ *

8

Maximum depth of the decision trees ⓘ *

32

Minimum number of samples per leaf node ⓘ *

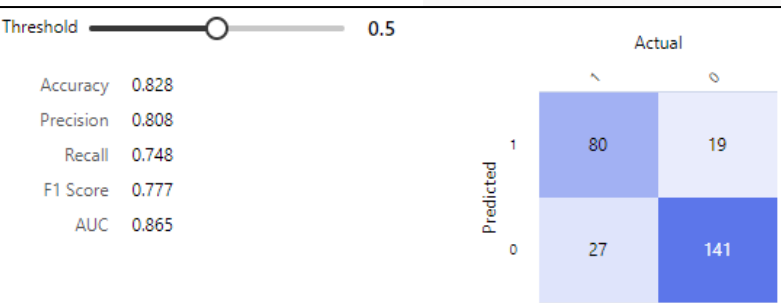
5

Resampling method ⓘ *

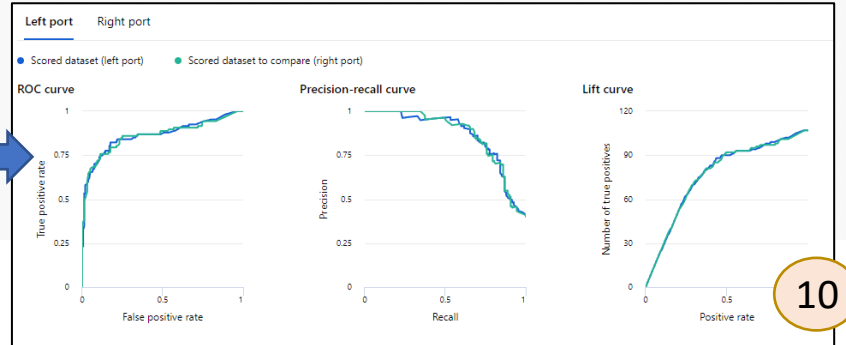
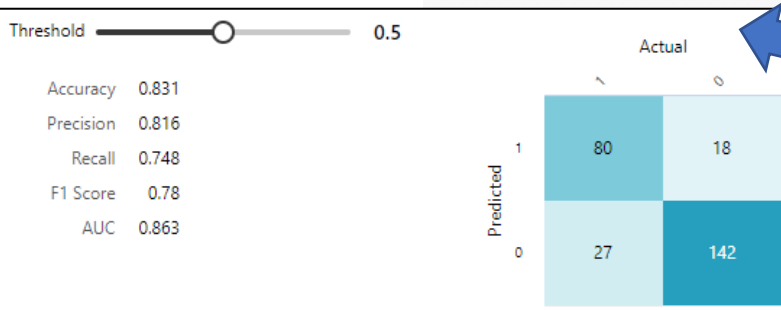
Bagging Resampling

Output settings

Model 2-2 : MaxDepth=6/ MinSample=5



Model 2-1 : MaxDepth= 32 / MinSample=5





4. จากข้อมูลบ่งชี้การเกิดอาการปวดหลังส่วนล่างของผู้ป่วยจำนวน 310 คน ซึ่งประกอบไปด้วยแอตทริบิวต์จำนวน 13 แอตทริบิวต์ (ไฟล์ Lower_back_pain.csv) ได้แก่


- Class_att คือ ผลการวินิจฉัย โดย Abnormal หมายถึง มีอาการปวดหลังส่วนล่าง และ Normal หมายถึง ไม่มีอาการปวดหลังส่วนล่าง
- แอตทริบิวต์อื่น ๆ ที่เป็นตัวแปรต้นจำนวน 12 แอตทริบิวต์ ซึ่งเป็นข้อมูลชนิดตัวเลข

ให้นักศึกษาใช้เครื่องมือใน Azure Machine Learning ทำการจัดเตรียมข้อมูลและสร้างโมเดลในการจำแนกว่าเป็นผู้ป่วยที่มีอาการปวดหลังส่วนล่าง (Abnormal) หรือ ไม่มีอาการปวดหลังส่วนล่าง (Normal) โดยใช้อัลกอริทึม Support Vector Machine (SVM) โดยกำหนดค่าพารามิเตอร์ที่เกี่ยวข้องดังรูป พร้อมทั้งตอบคำถามต่อไปนี้

1) Accuracy = 86.0 %
Precision = 77.4%
Recall = 80.0%
F1 Score = 78.7%

Two-Class Support Vecto...  

Create trainer mode ⓘ *

SingleParameter 


Number of iterations ⓘ * ...

10

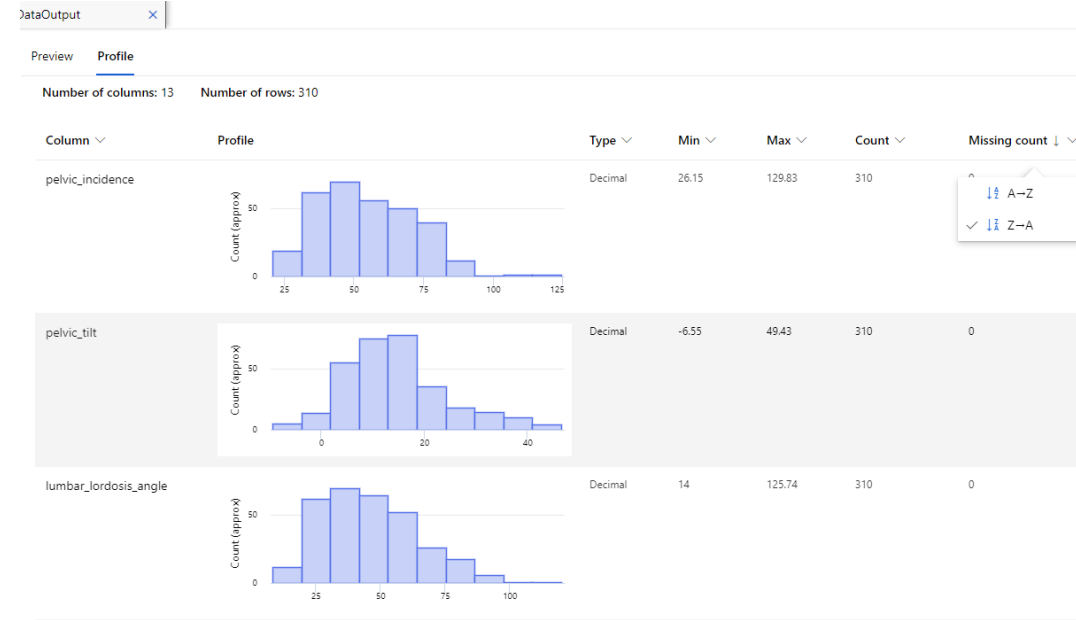
Lambda ⓘ * ...

0.001

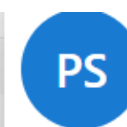
Normalize features ⓘ *

True 

Random number seed ⓘ ...



- เนื่องจาก data ของเราไม่มี Missing Value เราจึงสามารถข้ามขั้นตอนการ Clean ไปได้
- เนื่องจาก data ของเรามีค่าในแต่ละ col ค่อนข้างใกล้เคียงกัน เราจึงไม่จำเป็นต้องทำการ Normalize
- ดู Process Flowchart จากหน้า 11



[Sign out](#)

title04 [Completed](#)

Two-Class Support Vector Machine

Create trainer mode ⓘ *

SingleParameter

Number of iterations ⓘ *

10

Lambda ⓘ *

0.001

Normalize features ⓘ *

True

Random number seed ⓘ

1234

Two-Class Support Vector Machine
two_class_support_vector_machine

Train Model

Label column ⓘ *

Column names: Class_att

Model explanations ⓘ

False

Train Model
train_model

Score Model

Append score columns to output ⓘ *

True

Lower_back_pain

v | 1

Data output

Input dataset

Filter Based Feature Selection
filter_based_feature_selection

Select 6 features from 12 total features

Filtered data...

Features

Dataset

Untrained model

Untrained mod...

Results datas...

Results datas...

Dataset

Train Model
train_model

Trained model

Trained model

Dataset

Score Model
score_model

Scored dataset

Scored data...

Scored data...

Evaluate Model
evaluate_model

Evaluation results

Filter Based Feature Selection

Operate on feature columns only ⓘ

True

Number of desired features ⓘ *

8

Feature scoring method ⓘ *

PearsonCorrelation

Target column ⓘ *

Column names: Class_att

เลือก 6-8 Features

Split Data

Splitting mode ⓘ *

Split Rows

Fraction of rows in the first output dataset ⓘ *

0.7

Randomized split ⓘ *

True

Random seed ⓘ *

1234

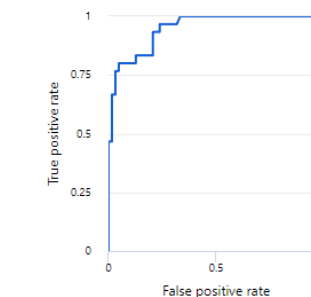
Stratified split ⓘ *

False

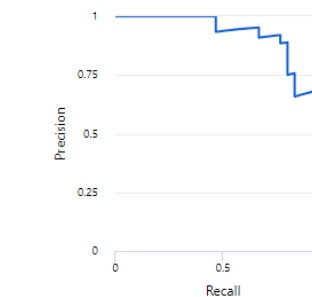
Evaluation_results

Scored dataset (left port)

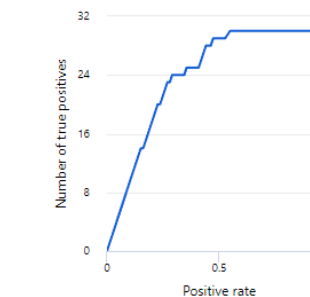
ROC curve



Precision-recall curve



Lift curve



Threshold 0.5

Accuracy 0.86
Precision 0.774
Recall 0.8
F1 Score 0.787
AUC 0.949

		Actual	
Predicted	Normal	24	7
	Abnormal	6	56