

Project

กลุ่มเทคโนโลยี

DATA

1

องค์กร : สำนักงานเขตพื้นที่การศึกษามัธยมศึกษา เขต 4

ข้อมูลชุดที่ 1: จำนวนนักเรียนแยกตามศาสนา ปีการศึกษา 2564 เทอม 1

<https://data.go.th/dataset/reli2564-1>

ข้อมูลชุดที่ 2 : จำนวนนักเรียนแยกตามน้ำหนักและส่วนสูง ปีการศึกษา 2564 เทอม 1

<https://data.go.th/dataset/high2564-1>

ข้อมูลชุดที่ 3 : จำนวนนักเรียนแยกตามสัญชาติ ปีการศึกษา 2564 เทอม 1

<https://data.go.th/dataset/nation2564-1>

Preprocessing

2

รวม 2 ตาราง (.merge())



```
1 merged_table1 = data_clean1.merge(data2_new,how='left',left_on='Level_School',right_on='Level_School') #รวมตาราง ซึ่งเอาข้อมูลชุดที่ 1 ไว้ทางด้านซ้าย ข้อมูลชุดที่ 2 ไว้ทางขวา และข้อมูลชุดที่ 1 เป็นตารางหลัก เชื่อมกันด้วย Level_School
2 merged_table1
```



	No.	School_Name	School_ID	YearTerm	Religion	Level	Level_School	Total_Religion	WeightHigh	Total_WeightHigh
0	31	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M1	1.07	627	สูงตามเกณฑ์	495
1	32	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M2	1.08	634	สูงตามเกณฑ์	558
2	33	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M3	1.09	632	สูงตามเกณฑ์	528
3	34	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M4	1.10	468	สูงตามเกณฑ์	389
4	35	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M5	1.11	482	สูงตามเกณฑ์	426
...
133	884	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M2	22.08	93	สูงตามเกณฑ์	64
134	885	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M3	22.09	89	สูงตามเกณฑ์	63
135	886	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M4	22.10	123	สูงตามเกณฑ์	99
136	887	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M5	22.11	107	สูงตามเกณฑ์	84
137	888	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M6	22.12	83	สูงตามเกณฑ์	58

โดยทำการรวมตาราง ซึ่งเอาข้อมูลชุดที่ 1 ไว้ทางด้านซ้าย ข้อมูลชุดที่ 2 ไว้ทางขวา และข้อมูลชุดที่ 1 เป็นตารางหลัก เชื่อมกันด้วย Level_School

Preprocessing

3

เชื่อมโยงกัน 3 ตาราง

In []: `data3_new = data_clean3.iloc[:,[4,6,7]]` #ตัดข้อมูลด้วย *iloc* เอาแค่คอลัมน์ *Level_School*, *Nationality* และ *Total_Nationality*
`data3_new`

Out[]:

	Nationality	Level_School	Total_Nationality
12	ไทย	1.07	632
13	ไทย	1.08	640
14	ไทย	1.09	639
15	ไทย	1.10	474
16	ไทย	1.11	489
...
649	ไทย	22.08	90
650	ไทย	22.09	88
651	ไทย	22.10	123
652	ไทย	22.11	106
653	ไทย	22.12	85

138 rows × 3 columns

ตัดข้อมูลด้วย *iloc* เอาแค่คอลัมน์ *Level_School*, *Nationality* และ *Total_Nationality*

Preprocessing

4

```
In [ ]: merged_table_new = merged_table1.merge(data3_new,how='left',left_on='Level_School',right_on='Level_School') #ตารางเชื่อมโยงกัน 3 ตาราง ซึ่งเอาข้อมูลที่อยู่ทาง  
merged_table_new
```

Out[]:

	No.	School_Name	School_ID	YearTerm	Religion	Level	Level_School	Total_Religion	WeightHigh	Total_WeightHigh	Nationality	Total_Nationality
0	31	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M1	1.07	627	สูงตามเกณฑ์	495	ไทย	632
1	32	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M2	1.08	634	สูงตามเกณฑ์	558	ไทย	640
2	33	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M3	1.09	632	สูงตามเกณฑ์	528	ไทย	639
3	34	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M4	1.10	468	สูงตามเกณฑ์	389	ไทย	474
4	35	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M5	1.11	482	สูงตามเกณฑ์	426	ไทย	489
...
133	884	โรงเรียนมัธยม สังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M2	22.08	93	สูงตามเกณฑ์	64	ไทย	90
134	885	โรงเรียนมัธยม สังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M3	22.09	89	สูงตามเกณฑ์	63	ไทย	88
135	886	โรงเรียนมัธยม สังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M4	22.10	123	สูงตามเกณฑ์	99	ไทย	123
136	887	โรงเรียนมัธยม สังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M5	22.11	107	สูงตามเกณฑ์	84	ไทย	106
137	888	โรงเรียนมัธยม สังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M6	22.12	83	สูงตามเกณฑ์	58	ไทย	85

138 rows × 12 columns

ตารางเชื่อมโยงกัน 3 ตาราง ซึ่งข้อมูลที่อยู่ทางซ้ายเป็นข้อมูลที่เชื่อมกับ 2 ตาราง ส่วนตาราง data3_new ไว้ทางขวา ตารางหลักคือตารางด้านซ้ายเชื่อมกันด้วย Level_School

Preprocessing

5

✓
0
วินาที

▶

1 Total_table_max1_noindex = total_table_max1.reset_index()
2 Total_table_max1_noindex

👤

Level_School

Total_Religion

0	1.01	0
1	1.02	0
2	1.03	0
3	1.04	0
4	1.05	0
...
259	22.08	93
260	22.09	89
261	22.10	123
262	22.11	107
263	22.12	83

264 rows × 2 columns

หาค่ามากที่สุดของศาสนา

Preprocessing

6

1 Total_table_max2_noindex = total_table_max2.reset_index()
2 Total_table_max2_noindex



	Level_School	Total_WeightHigh
0	1.01	0
1	1.02	0
2	1.03	0
3	1.04	0
4	1.05	0
...
267	22.08	64
268	22.09	63
269	22.10	99
270	22.11	84
271	22.12	58

272 rows × 2 columns

หาค่ามากที่สุดน้ำหนักส่วนสูง



Classification

7

1. ต้องการทราบว่านักเรียนในแต่ละระดับชั้น/ชื่ออำเภอมีการนับถือศาสนาเป็นอย่างไร
2. ต้องการทราบว่านักเรียนในแต่ละระดับชั้น/ชื่ออำเภอ นำหนักส่วนสูงแต่ละโรงเรียนมีการกระจายตัวของข้อมูลเป็นอย่างไร

Classification8

1 merge_All = pd.merge(merged_table_new,District) # pd.merge เพื่อรวมตาราง merge_table_new และ ตาราง District เข้าด้วยกัน

2 merge_All

	No.	School_Name	School_ID	YearTerm	Religion	Level	Level_School	Total_Religion	WeightHigh	Total_WeightHigh	Nationality	Total_Nationality	District_Name	District_ID
0	31	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M1	1.07	627	สูงตามเกณฑ์	495	ไทย	632	เมืองปทุมธานี	0.01
1	32	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M2	1.08	634	สูงตามเกณฑ์	558	ไทย	640	เมืองปทุมธานี	0.01
2	33	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M3	1.09	632	สูงตามเกณฑ์	528	ไทย	639	เมืองปทุมธานี	0.01
3	34	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M4	1.10	468	สูงตามเกณฑ์	389	ไทย	474	เมืองปทุมธานี	0.01
4	35	โรงเรียนปทุมวิไล	1	2564/1	พุทธ	M5	1.11	482	สูงตามเกณฑ์	426	ไทย	489	เมืองปทุมธานี	0.01
...
133	884	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M2	22.08	93	สูงตามเกณฑ์	64	ไทย	90	สาธิต	0.07
134	885	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M3	22.09	89	สูงตามเกณฑ์	63	ไทย	88	สาธิต	0.07
135	886	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M4	22.10	123	สูงตามเกณฑ์	99	ไทย	123	สาธิต	0.07
136	887	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M5	22.11	107	สูงตามเกณฑ์	84	ไทย	106	สาธิต	0.07
137	888	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร	22	2564/1	พุทธ	M6	22.12	83	สูงตามเกณฑ์	58	ไทย	85	สาธิต	0.07

138 rows × 14 columns

ตารางนี้มาจากการลบค่า Missing ของข้อมูลชื่ออำเภอ แล้วรวมตารางข้อมูลเดิม กับข้อมูลชื่ออำเภอเข้าด้วยกัน

Classification

9

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ ศาสนา

```
1 ALL = merge_All.iloc[:,[5,4,8,12]]
2 ALL
```

	Level	Religion	WeightHigh	District_Name
0	M1	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
1	M2	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
2	M3	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
3	M4	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
4	M5	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
...
133	M2	พุทธ	สูงตามเกณฑ์	ลำลูกกา
134	M3	พุทธ	สูงตามเกณฑ์	ลำลูกกา
135	M4	พุทธ	สูงตามเกณฑ์	ลำลูกกา
136	M5	พุทธ	สูงตามเกณฑ์	ลำลูกกา
137	M6	พุทธ	สูงตามเกณฑ์	ลำลูกกา

138 rows × 4 columns

ตัดข้อมูลด้วย .iloc ดูเฉพาะข้อมูล Level , Religion, WeightHiht และ District_Name

ทำนายระดับชั้นและอำเภอ กับ ศาสนา

```
1 Religion = ALL.iloc[:,[0,1,3]]
2 Religion
```

	Level	Religion	District_Name
0	M1	พุทธ	เมืองปทุมธานี
1	M2	พุทธ	เมืองปทุมธานี
2	M3	พุทธ	เมืองปทุมธานี
3	M4	พุทธ	เมืองปทุมธานี
4	M5	พุทธ	เมืองปทุมธานี
...
133	M2	พุทธ	ลำลูกกา
134	M3	พุทธ	ลำลูกกา
135	M4	พุทธ	ลำลูกกา
136	M5	พุทธ	ลำลูกกา
137	M6	พุทธ	ลำลูกกา

138 rows × 3 columns

ตัดข้อมูลด้วย .iloc เพื่อดูข้อมูลระดับชั้น ศาสนาและชื่ออำเภอ

Classification

10

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ ศาสนา

```
1 Total_Level = Religion.groupby('Level').count()
2 Total_Level
```

	Religion	District_Name
Level		
M1	22	22
M2	22	22
M3	22	22
M4	22	22
M5	22	22
M6	22	22
P1	1	1
P2	1	1
P3	1	1
P4	1	1
P5	1	1
P6	1	1

ทำนายจากระดับชั้นกับศาสนา โดยใช้ .groupby เรียกดูระดับชั้น

```
1 Religion.groupby('Religion').count()
```

	Level	District_Name
Religion		
พุทธ	138	138

.groupby เพื่อดูข้อมูล Religion พบว่า คนที่นับถือ ศาสนาพุทธทุกคน มีจำนวนทั้งหมด 138 แถว

Classification

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ ศาสนา

11

Decision Tree

```
1 Dummy_Religion = pd.get_dummies(Religion)
2 Dummy_Religion
```



	Level_M1	Level_M2	Level_M3	Level_M4	Level_M5
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1
...
133	0	1	0	0	0
134	0	0	1	0	0
135	0	0	0	1	0
136	0	0	0	0	1
137	0	0	0	0	0

138 rows × 20 columns

ใช้คำสั่ง `pd.get_Dummies`
เพื่อเปลี่ยนศาสนาที่เป็นตัว
หนังสือให้เป็นตัวเลข

```
✓ [462] 1 X1 = merge_All[['District_ID','Level_School']]
        2 y1 = merge_All[['Total_Religion']]
```

```
✓ [463] 1 X1
```

	District_ID	Level_School
0	0.01	1.07
1	0.01	1.08
2	0.01	1.09
3	0.01	1.10
4	0.01	1.11
...
133	0.07	22.08
134	0.07	22.09
135	0.07	22.10
136	0.07	22.11
137	0.07	22.12

138 rows × 2 columns

```
1 y1
```



	Total_Religion
0	627
1	634
2	632
3	468
4	482
...	...
133	93
134	89
135	123
136	107
137	83

138 rows × 1 columns

X1 คือ ข้อมูลระดับชั้น และ ID ของอำเภอ และ y1 คือ ข้อมูลศาสนา

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ ศาสนา

▼ plot tree

✓ [468] 1 from sklearn.tree import plot_tree #import ข้อมูล

✓ 1 plot_tree(mytree); # plot แผนภาพต้นไม้



plot แผนภาพต้นไม้

▼ Evaluate

✓ [475] 1 from sklearn.metrics import accuracy_score #import ข้อมูล

✓ 1 mytree = DecisionTreeClassifier(max_depth=5,random_state=13)
2 mytree.fit(X1_train2,y1_train2)



DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
max_depth=5, max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=13, splitter='best')

✓ [477] 1 tree1_result = mytree.predict(X1_val)

✓ [478] 1 accuracy_score(y1_val,tree1_result) #แสดงค่า accuracy

0.047619047619047616

จากการวัดผล พบว่า ค่า accucy เท่ากับ 0.047

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ ศาสนา

```
✓ [250] 1 from sklearn.neighbors import KNeighborsClassifier #import ข้อมูล  
      2 from sklearn.metrics import accuracy_score
```

```
✓ ▶ 1 #knn1  
    2 # Define  
    3 neigh1 = KNeighborsClassifier(n_neighbors=3, weights='uniform')  
    4 # Train  
    5 neigh1.fit(X1_train2,y1_train2)  
    6 # Test  
    7 knn1_result = neigh1.predict(X1_val)  
    8 accuracy_score(y1_val,knn1_result)
```

```
👤 /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DataCor  
    0.38095238095238093
```

**ใช้วิธี knn โดยถามเพื่อนบ้านที่ใกล้ที่สุด 3 คน
พบว่า ค่า accuracy เท่ากับ 0.380**

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ น้ำหนักส่วนสูง

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ น้ำหนักส่วนสูง

```
1 ALL = merge_All.iloc[:,[5,4,8,12]]
2 ALL
```

	Level	Religion	WeightHigh	District_Name
0	M1	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
1	M2	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
2	M3	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
3	M4	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
4	M5	พุทธ	สูงตามเกณฑ์	เมืองปทุมธานี
...
133	M2	พุทธ	สูงตามเกณฑ์	ลำลูกกา
134	M3	พุทธ	สูงตามเกณฑ์	ลำลูกกา
135	M4	พุทธ	สูงตามเกณฑ์	ลำลูกกา
136	M5	พุทธ	สูงตามเกณฑ์	ลำลูกกา
137	M6	พุทธ	สูงตามเกณฑ์	ลำลูกกา

138 rows × 4 columns

ตัดข้อมูลด้วย .iloc ดูเฉพาะข้อมูล Level ,
Religion, WeightHiht และ District_Name

```
1 WeightHigh = ALL.iloc[:,[0,3,2]] #ทำการตัด .iloc เพื่อ
2 WeightHigh
```



	Level	District_Name	WeightHigh
0	M1	เมืองปทุมธานี	สูงตามเกณฑ์
1	M2	เมืองปทุมธานี	สูงตามเกณฑ์
2	M3	เมืองปทุมธานี	สูงตามเกณฑ์
3	M4	เมืองปทุมธานี	สูงตามเกณฑ์
4	M5	เมืองปทุมธานี	สูงตามเกณฑ์
...
133	M2	ลำลูกกา	สูงตามเกณฑ์
134	M3	ลำลูกกา	สูงตามเกณฑ์
135	M4	ลำลูกกา	สูงตามเกณฑ์
136	M5	ลำลูกกา	สูงตามเกณฑ์
137	M6	ลำลูกกา	สูงตามเกณฑ์

138 rows × 3 columns

ตัดข้อมูลด้วย .iloc เพื่อดูข้อมูลระดับชั้น
ศาสนาและข้อมูลน้ำหนักส่วนสูง

Classification

15

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ น้ำหนักส่วนสูง

```
[361] 1 X2 = merge_All[['District_ID','Level_School']]
      2 y2 = merge_All[['School_Name']]
```

```
1 X2
```

	District_ID	Level_School
0	0.01	1.07
1	0.01	1.08
2	0.01	1.09
3	0.01	1.10
4	0.01	1.11
...
133	0.07	22.08
134	0.07	22.09
135	0.07	22.10
136	0.07	22.11
137	0.07	22.12

138 rows × 2 columns

```
1 y2
```

	School_Name
0	โรงเรียนปทุมวิไล
1	โรงเรียนปทุมวิไล
2	โรงเรียนปทุมวิไล
3	โรงเรียนปทุมวิไล
4	โรงเรียนปทุมวิไล
...	...
133	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร
134	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร
135	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร
136	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร
137	โรงเรียนมัธยมสังคีตวิทยา กรุงเทพมหานคร

138 rows × 1 columns

X2 คือ ข้อมูลระดับชั้น และ ID ของอำเภอ และ y2 คือ ข้อมูลของชื่อโรงเรียนแต่ละโรงเรียน

เลือก Model และ การวัดผล

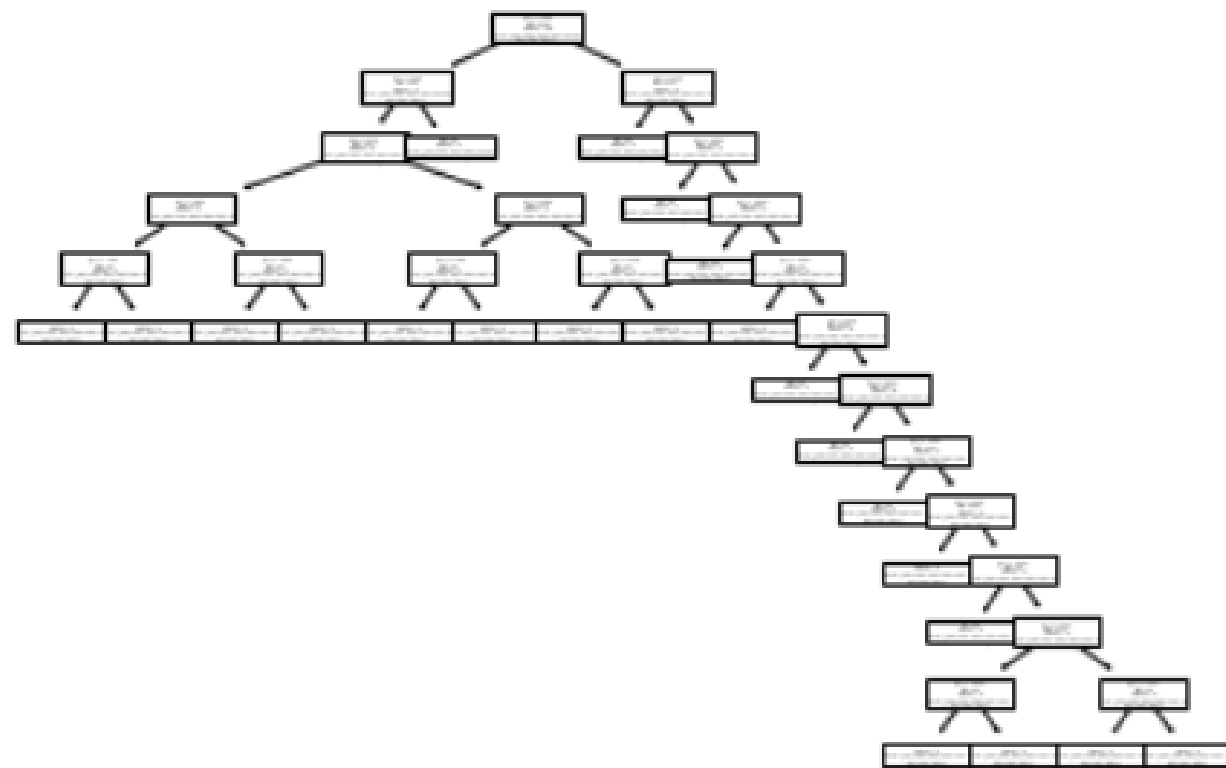
16

ทำนายระหว่าง ตัวแปร X คือ ระดับชั้นและอำเภอ กับ ตัวแปร y คือ น้ำหนักส่วนสูง

▼ plot tree

✓ [367] 1 from sklearn.tree import plot_tree #import ข้อมูล

✓ [368] 1 plot_tree(mytree2); # plot แผนภาพต้นไม้



plot แผนภาพต้นไม้

▼ Evaluate

✓ [374] 1 from sklearn.metrics import accuracy_score #import ข้อมูล

✓ [▶] 1 mytree2 = DecisionTreeClassifier()
2 mytree2.fit(X2_train2,y2_train2)

DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, presort='deprecated', random_state=None, splitter='best')


✓ [376] 1 tree2_result = mytree2.predict(X2_val) #predict ข้อมูลตัวแปร (X2_val)


✓ [377] 1 accuracy_score(y2_val,tree2_result) #รันค่า

0.9523809523809523



จากการวัดผล พบว่า ค่า accucy เท่ากับ 0.952


ค่า KNN

 1 #knn1
2 # Define
3 neigh1 = KNeighborsClassifier(n_neighbors=3, weights='uniform')
4 # Train
5 neigh1.fit(X1_train,y1_train)
6 # Test
7 knn1_result = neigh1.predict(X1_val)
8 accuracy_score(y1_val,knn1_result)

 /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DataCo
0.47619047619047616

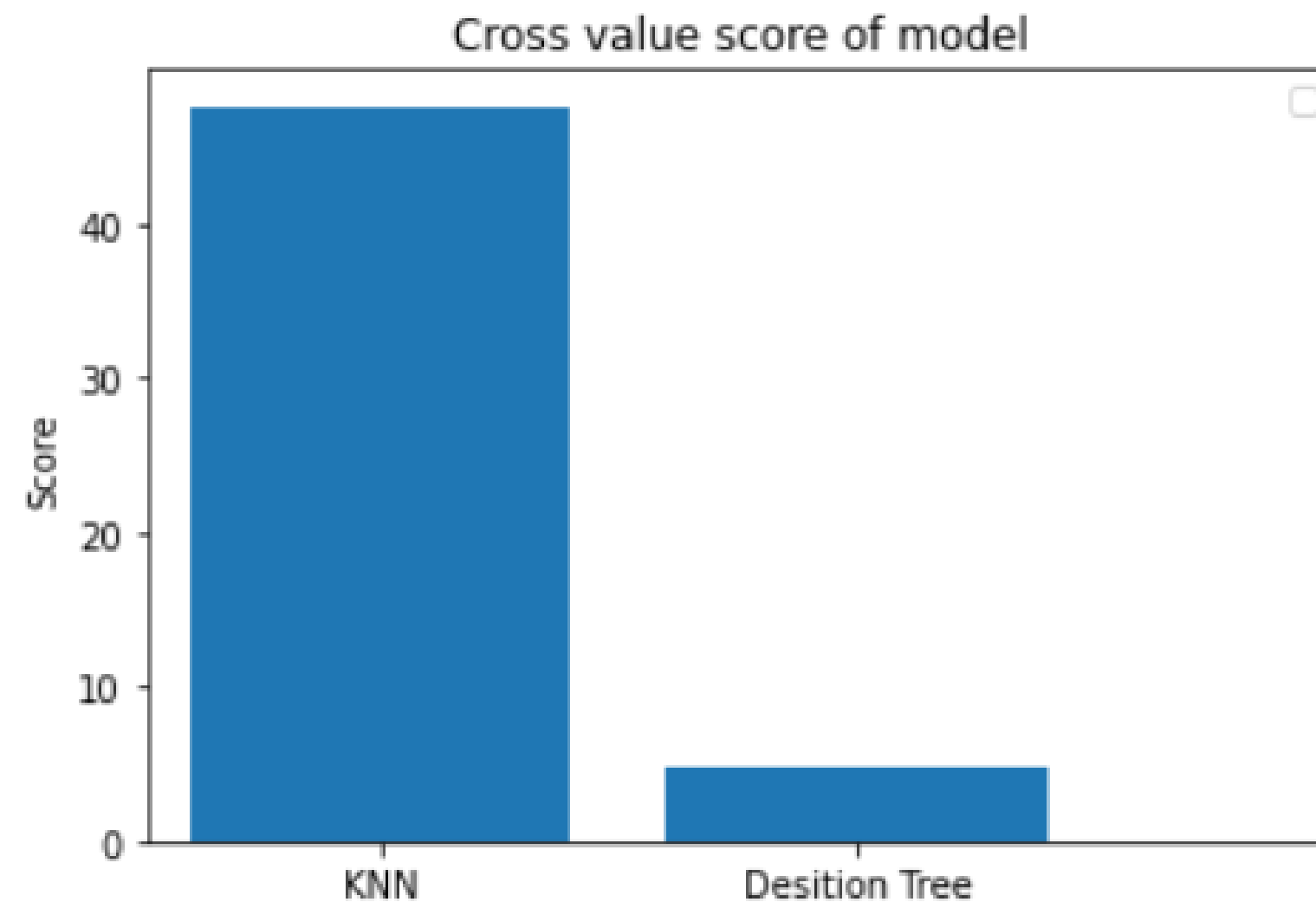
ชุดที่ 1 ใช้วิธี knn โดยถามเพื่อนบ้านที่ใกล้ที่สุด 3 คน
พบว่า ค่า accuracy เท่ากับ 0.476

 0 วินาที
 1 #knn2
2 # Define
3 neigh2 = KNeighborsClassifier(n_neighbors=3, weights='uniform')
4 # Train
5 neigh2.fit(X2_train2,y2_train2)
6 # Test
7 knn2_result = neigh2.predict(X2_val)
8 accuracy_score(y2_val,knn2_result)

 /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: DataCo
0.7142857142857143

ชุดที่ 2 ใช้วิธี knn โดยถามเพื่อนบ้านที่ใกล้ที่สุด 3 คน
พบว่า ค่า accuracy เท่ากับ 0.714

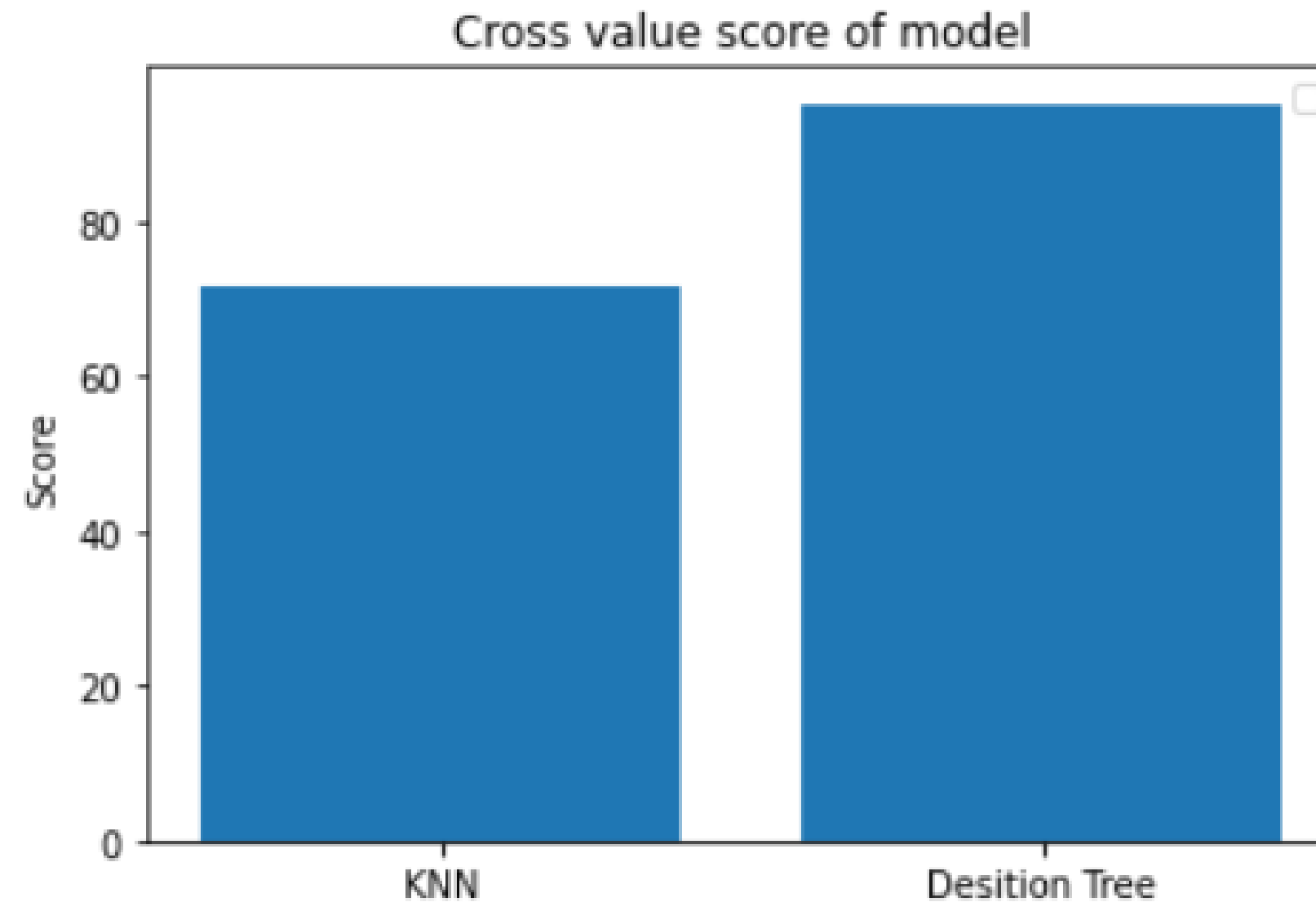
ชุดที่ 1



สรุป :

พบว่าข้อมูลชุดที่ 1 ที่ทำการทดสอบ
วิธี KNN มีความแม่นยำมากกว่า Decision Tree

ชุดที่ 2

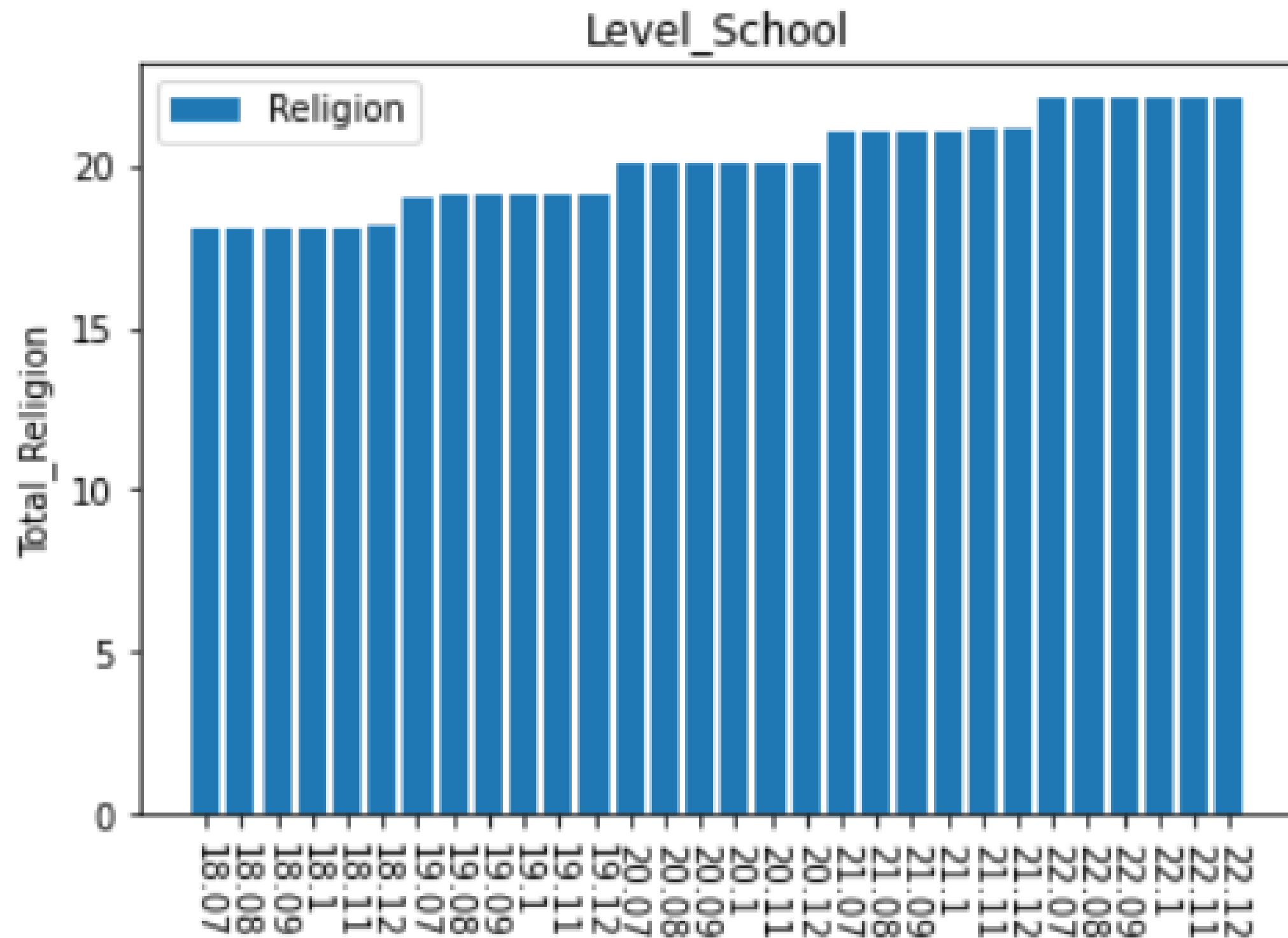


สรุป :

พบว่าข้อมูลชุดที่ 2 ที่ทำการทดสอบ
วิธี Decision Tree มีความแม่นยำมากกว่า KNN

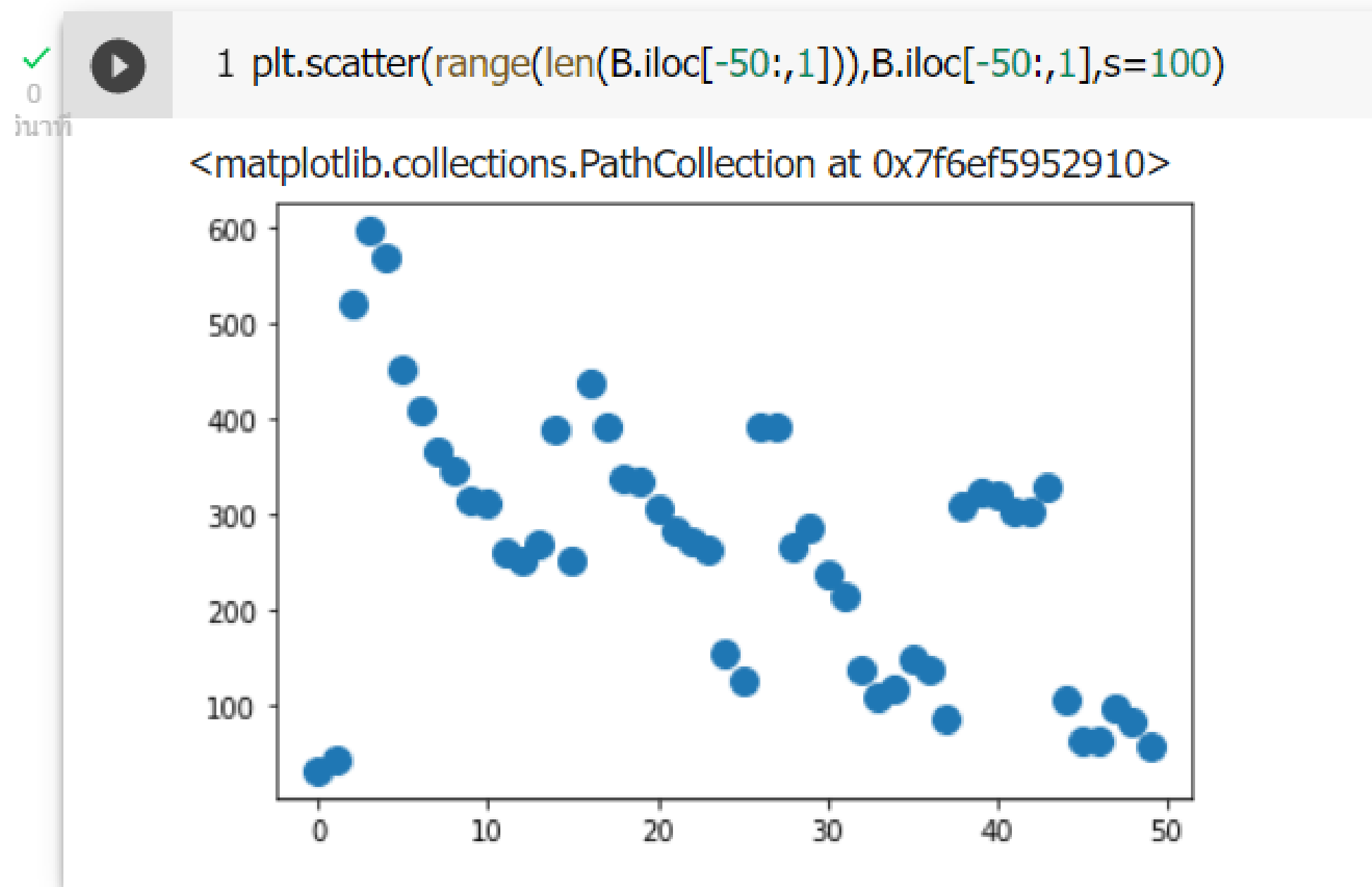
**Visualization ระหว่าง ตัวแปร X1 คือ ระดับชั้นและอำเภอ กับ
ตัวแปร y1 คือ ศาสนา**

```
merge_All.groupby('Level_School')[['Total_Religion']].sum().reset_index()  
plt.bar(np.arange(len(A.iloc[-30:,0])), A.iloc[-30:,0],label = 'Religion')  
  
plt.ylabel('Total_Religion')  
plt.title('Level_School')  
plt.xticks(np.arange(len(A.iloc[-30:,0])), A.iloc[-30:,0],rotation=-90)  
plt.legend();
```



สรุป จากกราฟจะเห็นได้ว่า
ศาสนาที่ทำมาพล็อตกราฟเป็น
ศาสนาพุทธ แกน X คือระดับ
ชั้นต่าง ๆ ในแต่ละโรงเรียน
และแกน y เป็นจำนวนคนที่
นับถือศาสนาพุทธ

Visualization ระหว่าง ตัวแปร X2 คือ ระดับชั้นและอำเภอ
กับ ตัวแปร y2 คือ น้ำหนักส่วนสูง



สรุป จากกราฟจะเห็นได้ว่าน้ำหนักส่วนสูง
เมื่อทำนายกับระดับชั้นและอำเภอจะเห็น
ได้ว่าการกระจายตัวกันมาก

BUONA SERA



ขอบคุณที่รับชมค่ะ

1. นางสาวภัทรสร	เทพบุตร	623020531-5
2. นางสาวสุพิชญา	ตั้งกิจวานิชย์	623020542-0
3. นางสาวสุภาวดี	คำทวย	623020544-6
4. นางสาวอุมาพร	คำภิชัย	623020547-0
5. นางสาวพลอยบงกช	แสงโทโพธิ์	623021051-4

