



GROUP PROJECT

กลุ่ม : เอกาไรสติ



PAGE 1

แหล่งที่มาของข้อมูล

01

รับเรื่องร้องเรียน/แจ้ง
เบาะแสยาเสพติด

องค์กร : สำนักงานคณะกรรมการ
ป้องกันและปราบปรามยาเสพติด

02

ปริมาณของกลางยา
เสพติด

องค์กร : สำนักงานคณะกรรมการ
ป้องกันและปราบปรามยาเสพติด

03

จำนวนคดีผู้ต้องหาคดี
ยาเสพติด

องค์กร : สำนักงานคณะกรรมการ
ป้องกันและปราบปรามยาเสพติด

04

ข้อมูลทั่วไปของจังหวัด



ปัญหา

คือ เพื่อทำนายชนิดของยาเสพติด โดยใช้จำนวนประชากร, พื้นที่, รายได้เฉลี่ยต่อเดือนของครัวเรือน, จำนวนเรื่องร้องเรียน/แจ้งเบาะแสยาเสพติด, ปริมาณของกลางยาเสพติด และจำนวนคดีผู้ต้องหาคดียาเสพติด

นำเข้าข้อมูลที่ต้องการนำมาใช้งาน

In [1]: `import pandas as pd` # pandas เป็น Package ที่ใช้เกี่ยวกับการจัดการข้อมูลที่เป็นตาราง , เรียกใช้โดยการ import แล้วตามด้วยชื่อ Package

In [2]: `from google.colab import drive` # เชื่อมต่อกับ google drive ที่เราเก็บข้อมูลไว้
`drive.mount('/content/drive')`

Mounted at /content/drive

In [3]: `import os` # เป็น package ที่เอาไว้จัดการไฟล์และเอาไว้ดูที่มาของไฟล์หรือ path

In [4]: `path = '/content/drive/My Drive/DWDM2021_DATAPROJECT'` # ระบุให้ชี้ไปที่โฟลเดอร์ใน google drive ที่เราเซฟไฟล์ไว้ คือ โฟลเดอร์ DWDM2021_DATAPROJECT

ชุดข้อมูลที่ 1 รับเรื่องร้องเรียน/แจ้งเบาะแส ยาเสพติด

In [5]: `data_01 = pd.read_csv(os.path.join(path, 'รับเรื่องร้องเรียนยาเสพติด.csv')) # โหลดไฟล์ข้อมูลการ
data_01 # print data ดูว่ามีค่าเป็น missing`

Out[5]:

	รหัส จังหวัด	ชื่อจังหวัด	เรื่องร้อง เรียนและ แจ้งเบาะแสร ยาเสพติด ทั้งหมด (เรื่อง)	การดำเนิน การต่อเรื่อง ร้องเรียนและ แจ้งเบาะแสร ยาเสพติด ทั้งหมด (เรื่อง)	เปอร์เซ็นต์การ ดำเนินการต่อ เรื่องร้องเรียน และแจ้งเบาะแสร ยาเสพติด ทั้งหมด (%)	เรื่องร้อง เรียนและ แจ้งเบาะแสร ยาเสพติด พื้นที่ (เรื่อง)	การดำเนิน การต่อเรื่อง ร้องเรียนและ แจ้งเบาะแสร ยาเสพติด พื้นที่ (เรื่อง)	เปอร์เซ็นต์การ ดำเนินการต่อ เรื่องร้องเรียน และแจ้งเบาะแสร ยาเสพติด พื้นที่ (%)	เรื่องร้อง เรียนและ แจ้งเบาะแสร ยาเสพติด บุคคล (เรื่อง)	การดำเนิน การต่อเรื่อง ร้องเรียนและ แจ้งเบาะแสร ยาเสพติด บุคคล (เรื่อง)	เปอร์เซ็นต์การ ดำเนินการต่อ เรื่องร้องเรียน และแจ้งเบาะแสร ยาเสพติด บุคคล (%)
0	11	สมุทรปราการ	519	506	97.50	108	104	96.30	411	402	97.81
1	12	นนทบุรี	526	526	100.00	110	110	100.00	416	416	100.00
2	13	ปทุมธานี	519	504	97.11	118	114	96.61	401	390	97.26
3	14	พระนครศรีอยุธยา	376	373	99.20	30	30	100.00	346	343	99.13
4	15	ฉะเชิงเทรา	106	106	100.00	7	7	100.00	99	99	100.00
...
72	93	พิจิตร	91	91	100.00	5	5	100.00	86	86	100.00
73	94	ปัตตานี	44	44	100.00	4	4	100.00	40	40	100.00
74	95	ยะลา	34	34	100.00	4	4	100.00	30	30	100.00
75	96	นราธิวาส	35	35	100.00	0	0	0.00	35	35	100.00
76	10	กรุงเทพมหานคร	2568	2402	93.54	545	510	93.58	2023	1892	93.52

77 rows x 11 columns

ตรวจสอบค่า Missing ของตาราง data_01

In [8]: `data_01.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missingบ้าง สรุปคือทุกคอลัมน์ไม่มีค่า missing`

Out[8]:

```

รหัสจังหวัด      False
ชื่อจังหวัด      False
เรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติดทั้งหมด (เรื่อง)  False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติดทั้งหมด (เรื่อง)  False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติดทั้งหมด (%)  False
เรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด พื้นที่ (เรื่อง)  False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด พื้นที่ (เรื่อง)  False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด พื้นที่ (%)  False
เรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด บุคคล (เรื่อง)  False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด บุคคล (เรื่อง)  False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสรยาเสพติด บุคคล (%)  False
dtype: bool

```

ชุดข้อมูลที่ 2 ปริมาณของกลางยาเสพติด

In [9]: data_02 = pd.read_csv(os.path.join(path,'ปริมาณของกลางยาเสพติด.csv')) # ไฟล์ที่เราต้องการ
data_02 # print data ดูว่าบันทึกค่าเป็นยังไง

Out[9]:

	รหัส จังหวัด	จังหวัด	ปริมาณของกลางยา เสพติดทั้งหมด (กิโลกรัม)	ชนิดของ กลางยาเสพติด	ยาบ้า (กิโลกรัม)	ไอซ์ (กิโลกรัม)	กัญชาแห้ง (กิโลกรัม)	เฮโรอีน (กิโลกรัม)	คัสตามีน (กิโลกรัม)	โคเคน (กิโลกรัม)	กระเทียม (กิโลกรัม)
0	32	สุรินทร์	87.14170	ยาบ้า (กิโลกรัม)	50.2577	1.43400	0.38000	0.000	0.0000	0.000	35.07000
1	41	อุดรธานี	164.12746	ยาบ้า (กิโลกรัม)	141.3443	11.26127	11.20389	0.000	0.0580	0.000	0.26000
2	49	มุกดาหาร	1830.13310	กัญชาแห้ง (กิโลกรัม)	604.1371	545.27100	679.70000	0.400	0.0400	0.000	0.58500
3	47	สกลนคร	812.34960	กัญชาแห้ง (กิโลกรัม)	42.1416	75.39300	670.28500	0.000	0.0000	0.000	24.53000
4	46	กาฬสินธุ์	384.13460	ยาบ้า (กิโลกรัม)	253.9826	127.45400	2.69800	0.000	0.0000	0.000	0.00000
...
72	92	ตรัง	2032.81350	กระเทียม (กิโลกรัม)	248.9255	10.50600	0.92700	0.000	0.0000	0.000	1772.45500
73	93	พัทลุง	1317.79318	กระเทียม (กิโลกรัม)	63.0449	4.14783	8.28795	0.000	0.0000	0.000	1242.31250
74	94	ปัตตานี	6683.02730	กระเทียม (กิโลกรัม)	352.6543	378.82900	0.44200	0.794	0.0000	0.000	5950.30800
75	95	ยะลา	4242.57840	กระเทียม (กิโลกรัม)	176.9054	21.12000	0.64900	0.007	0.0000	0.000	4043.89700
76	10	กรุงเทพมหานคร	3722.06515	กระเทียม (กิโลกรัม)	182.0511	336.05174	433.37700	9.389	5.3312	0.049	2755.81611

77 rows x 11 columns

ตรวจสอบค่า Missing ของตาราง data_02

```
In [12]: data_02.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง สรุปคือทุกคอลัมน์ไม่มีค่า missing
```

```
Out[12]: รหัสจังหวัด      False
จังหวัด      False
ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)  False
ชนิดของกลางยาเสพติด      False
ยาน้ำ (กิโลกรัม)      False
ไอซ์ (กิโลกรัม)      False
กัญชาแห้ง (กิโลกรัม)      False
เฮโรอีน (กิโลกรัม)      False
คีตาเมีน (กิโลกรัม)      False
โคเคน (กิโลกรัม)      False
กระพ้อม (กิโลกรัม)      False
dtype: bool
```

เลือกเฉพาะคอลัมน์ที่เป็น จำนวนของปริมาณยาเสพติด แต่ละชนิด

เพื่อดูว่ายาเสพติดชนิดใดที่มีเรื่อง
ร้องเรียนและถูกจับมากที่สุด
รวมทั้งมีปริมาณเท่าไร

```
In [13]: data_02_cut = data_02.iloc[:,4:] # เลือกเฉพาะคอลัมน์ที่เป็นจำนวนของปริมาณยาเสพติด (กิโลกรัม)
data_02_cut
```

Out[13]:

	ยาน้ำ (กิโลกรัม)	ไอซ์ (กิโลกรัม)	กัญชาแห้ง (กิโลกรัม)	เฮโรอีน (กิโลกรัม)	คีตาเมีน (กิโลกรัม)	โคเคน (กิโลกรัม)	กระพ้อม (กิโลกรัม)
76	182.0511	336.05174	433.37700	9.38900	5.3312	0.049	2755.81611
24	16.8905	35.66841	17.08120	0.00100	0.9550	0.000	332.85821
26	87.8015	13.21964	13.18900	0.51300	0.7890	0.030	626.33734
26	121.0734	40.26200	72.16300	0.17100	0.7310	0.000	981.02860
27	636.6864	26.73300	2.80200	0.00000	0.0100	0.000	932.65946
...
72	248.9255	10.50600	0.92700	0.00000	0.0000	0.000	1772.45500
73	63.0449	4.14783	8.28795	0.00000	0.0000	0.000	1242.31250
74	352.6543	378.82900	0.44200	0.79400	0.0000	0.000	5950.30800
76	176.9054	21.12000	0.64900	0.00700	0.0000	0.000	4043.89700
69	342.2397	2177.68196	4.63820	1.10272	0.0150	0.000	5569.70900

77 rows x 7 columns

แปลงให้กลายเป็นค่าของข้อมูล หลังจากนั้นทำการวนลูปเพื่อหาค่าที่มากที่สุดของแต่ละแถว เพื่อดูว่ายาสเสพติดที่มีเรื่องร้องเรียนและถูกจับมากที่สุดมีปริมาณเท่าไร

```
In [15]: import numpy as np
max_all = np.zeros(data_02_value.shape[0]) # หาค่าที่มากที่สุดของแต่ละแถว เพื่อดูว่ายาสเสพติดชนิดที่พบบ่อยที่สุดมีปริมาณเท่าไร
for row in range(data_02_value.shape[0]):
    max_all[row] = max(data_02_value[row,:])
print(max_all)
```

```
[2.75581611e+03 3.32858210e+02 6.26337340e+02 9.81028600e+02
 9.32659460e+02 3.85900000e+02 1.54666800e+02 5.14725000e+01
 1.08513000e+02 5.91140000e+02 5.21596000e+02 5.69710000e+01
 6.04940000e+01 4.16274000e+02 5.78895000e+01 2.89320000e+01
 5.42737000e+02 4.91995600e+01 3.01773000e+02 7.79296000e+01
 5.02577000e+01 1.30678400e+02 5.03481000e+02 4.23811000e+01
 5.87532000e+01 7.53441000e+01 1.30539320e+03 2.90345000e+01
 4.59846000e+02 1.41344300e+02 6.36172000e+01 1.21472780e+03
 6.91882000e+01 9.00012000e+01 2.53982600e+02 6.70285000e+02
 9.69362800e+02 6.79700000e+02 3.88423520e+03 1.77913600e+02
 1.24196880e+03 5.11268000e+01 6.60400500e+02 1.21619000e+01
 3.16730000e+01 3.54830150e+03 4.60160000e+00 3.79929000e+01
 3.22070000e+00 5.61840000e+01 1.50032800e+03 3.01403000e+01
 2.78173000e+01 1.40540000e+01 1.32001000e+02 4.85322590e+02
 1.39515700e+03 4.17618000e+01 4.62830000e+01 1.14475550e+02
 4.05020000e+01 6.94054280e+02 2.71459300e+02 2.30304260e+03
 1.80731700e+03 2.56040640e+03 2.84483950e+03 7.76736410e+03
 3.15614394e+03 6.69522000e+02 2.67276536e+04 3.57429400e+03
 1.77245500e+03 1.24231250e+03 5.95030800e+03 4.04389700e+03
 5.56970900e+03]
```


เพิ่มคอลัมน์ของ max_all เข้าไปยังตารางข้อมูล และเปลี่ยนชื่อคอลัมน์ของ max_all ให้เป็น 'ปริมาณของกลางยาเสพติดที่พบมากที่สุด (กิโลกรัม)'

In [18]: data_02.rename(columns={"max_all":'ปริมาณของกลางยาเสพติดที่พบมากที่สุด (กิโลกรัม)'}, inplace=True) # เปลี่ยนชื่อคอลัมน์ data_02

Out[18]:

	รหัส จังหวัด	จังหวัด	ปริมาณของ กลางยาเสพติด ทั้งหมด (กิโลกรัม)	ชนิดของ กลางยา เสพติด	ยาบ้า (กิโลกรัม)	ไอซ์ (กิโลกรัม)	กัญชาแห้ง (กิโลกรัม)	เฮโรอีน (กิโลกรัม)	คีตามีน (กิโลกรัม)	โคเคน (กิโลกรัม)	กระท่อม (กิโลกรัม)	ปริมาณของ กลางยาเสพติดที่ พบมากที่สุด (กิโลกรัม)
76	10	กรุงเทพมหานคร	3722.06515	กระท่อม (กิโลกรัม)	182.0511	336.05174	433.37700	9.38900	5.3312	0.049	2755.81611	2755.81611
24	11	สมุทรปราการ	403.45432	กระท่อม (กิโลกรัม)	16.8905	35.66841	17.08120	0.00100	0.9550	0.000	332.85821	332.85821
25	12	นนทบุรี	741.87948	กระท่อม (กิโลกรัม)	87.8015	13.21964	13.18900	0.51300	0.7890	0.030	626.33734	626.33734
26	13	ปทุมธานี	1215.42900	กระท่อม (กิโลกรัม)	121.0734	40.26200	72.16300	0.17100	0.7310	0.000	981.02860	981.02860
27	14	พระนครศรีอยุธยา	1598.89086	กระท่อม (กิโลกรัม)	636.6864	26.73300	2.80200	0.00000	0.0100	0.000	932.65946	932.65946
...
72	92	ตรัง	2032.81350	กระท่อม (กิโลกรัม)	248.9255	10.50600	0.92700	0.00000	0.0000	0.000	1772.45500	1772.45500
73	93	พัทลุง	1317.79318	กระท่อม (กิโลกรัม)	63.0449	4.14783	8.28795	0.00000	0.0000	0.000	1242.31250	1242.31250
74	94	ปัตตานี	6683.02730	กระท่อม (กิโลกรัม)	352.6543	378.82900	0.44200	0.79400	0.0000	0.000	5950.30800	5950.30800
75	95	ยะลา	4242.57840	กระท่อม (กิโลกรัม)	176.9054	21.12000	0.64900	0.00700	0.0000	0.000	4043.89700	4043.89700
69	96	นราธิวาส	8095.38658	กระท่อม (กิโลกรัม)	342.2397	2177.68196	4.63820	1.10272	0.0150	0.000	5569.70900	5569.70900

77 rows x 12 columns

ตรงคอลัมน์ ชนิดของกลางยาเสพติด ลบหน่วยคำว่า (กิโลกรัม) ออกไป
ให้เหลือแค่ชื่อของชนิดยาเสพติดเพียงอย่างเดียว

In [21]: data_02['ชนิดของกลางยาเสพติด'].replace({
 'กัญชาแห้ง (กิโลกรัม)': 'กัญชาแห้ง',
 'ไอซ์ (กิโลกรัม)': 'ไอซ์',
 'ยาบ้า (กิโลกรัม)': 'ยาบ้า',
 'กระเทียม (กิโลกรัม)': 'กระเทียม'
}, inplace=True)
data_02

Out[21]:

	รหัส จังหวัด	จังหวัด	ปริมาณของ กลางยาเสพติด ทั้งหมด (กิโลกรัม)	ชนิดของ กลางยา เสพติด	ยาบ้า (กิโลกรัม)	ไอซ์ (กิโลกรัม)	กัญชาแห้ง (กิโลกรัม)	เฮโรอีน (กิโลกรัม)	คีดามีน (กิโลกรัม)	โคเคน (กิโลกรัม)	กระเทียม (กิโลกรัม)	ปริมาณของกลาง ยาเสพติดที่พบ มากที่สุด (กิโลกรัม)
76	10	กรุงเทพมหานคร	3722.06515	กระเทียม	182.0511	336.05174	433.37700	9.38900	5.3312	0.049	2755.81611	2755.81611
24	11	สมุทรปราการ	403.45432	กระเทียม	16.8905	35.66841	17.08120	0.00100	0.9550	0.000	332.85821	332.85821
25	12	นนทบุรี	741.87948	กระเทียม	87.8015	13.21964	13.18900	0.51300	0.7890	0.030	626.33734	626.33734
26	13	ปทุมธานี	1215.42900	กระเทียม	121.0734	40.26200	72.16300	0.17100	0.7310	0.000	981.02860	981.02860
27	14	พระนครศรีอยุธยา	1598.89086	กระเทียม	636.6864	26.73300	2.80200	0.00000	0.0100	0.000	932.65946	932.65946
...
72	92	ตรัง	2032.81350	กระเทียม	248.9255	10.50600	0.92700	0.00000	0.0000	0.000	1772.45500	1772.45500
73	93	พัทลุง	1317.79318	กระเทียม	63.0449	4.14783	8.28795	0.00000	0.0000	0.000	1242.31250	1242.31250
74	94	ปัตตานี	6683.02730	กระเทียม	352.6543	378.82900	0.44200	0.79400	0.0000	0.000	5950.30800	5950.30800
75	95	ยะลา	4242.57840	กระเทียม	176.9054	21.12000	0.64900	0.00700	0.0000	0.000	4043.89700	4043.89700
69	96	นราธิวาส	8095.38658	กระเทียม	342.2397	2177.68196	4.63820	1.10272	0.0150	0.000	5569.70900	5569.70900

77 rows x 12 columns

ดูว่ามียาเสพติดชนิดใดบ้างที่มีเรื่องร้องเรียนและถูกจับมากที่สุด โดยเรียงข้อมูลตามปริมาณของกลางยาเสพติดที่พบมากที่สุด (กิโลกรัม)

In [22]: # ดูว่ามียาเสพติดชนิดใดบ้างที่พบปริมาณมากที่สุด
data_02.groupby(['ชนิดของกลางยาเสพติด']).sum().sort_values(by='ปริมาณของกลางยาเสพติดที่พบมากที่สุด (กิโลกรัม)')

Out[22]:

	รหัส จังหวัด	ปริมาณของกลางยา เสพติดทั้งหมด (กิโลกรัม)	ยาบ้า (กิโลกรัม)	ไอซ์ (กิโลกรัม)	กัญชาแห้ง (กิโลกรัม)	เฮโรอีน (กิโลกรัม)	คีตาเมีน (กิโลกรัม)	โคเคน (กิโลกรัม)	กระท่อม (กิโลกรัม)	ปริมาณของกลางยา เสพติดที่พบมากที่สุด (กิโลกรัม)
ชนิดของ กลางยาเสพติด										
กัญชาแห้ง	203	3595.90562	951.6886	674.11602	1941.83200	0.40000	0.1130	0.000	27.75600	1941.83200
ไอซ์	238	5924.24406	318.7240	4386.03556	27.88200	94.97000	509.3370	0.000	587.39550	4386.03556
ยาบ้า	1546	22482.09478	15267.5754	2999.66530	3297.10958	97.91100	148.7800	3.008	668.04550	15438.41810
กระท่อม	1950	89967.62917	4145.5856	5058.26213	2113.22031	70.50925	12.0913	0.279	78567.68158	78567.68158

จากตารางข้างต้นพบว่า ชนิดของกลางยาเสพติดที่พบมากที่สุด คือ กัญชาแห้ง, ไอซ์, ยาบ้า และกระท่อม

ชุดข้อมูลที่ 3 จำนวนคดี ผู้ต้องหาคดียาเสพติด

In [23]: data_03 = pd.read_csv(os.path.join(path,'คดีผู้ต้องหาคดียาเสพติด.csv')) # ไฟล์ที่เราต้องการ
data_03 # print data ดูว่ามีหน้าตาเป็นยังไง

Out[23]:

	ปีงบประมาณ	รหัสจังหวัด	จำนวนคดีทั้งหมด (คดี)	จำนวนผู้ต้องหาทั้งหมด (คน)	จำนวนคดีเสพ (คดี)	จำนวนผู้ต้องหาคดีเสพ (คน)	จำนวนคดีครอบครอง (คดี)	จำนวนผู้ต้องหาคดีครอบครอง (คน)	จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	จำนวนคดีจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีจำหน่าย (คน)	จำนวนคดีผลิต (คดี)	จำนวนผู้ต้องหาคดีผลิต (คน)	จำนวนคดีนำเข้า (คดี)	จำนวนผู้ต้องหาคดีนำเข้า (คน)	จำนวนคดีส่งออก (คดี)	จำนวนผู้ต้องหาคดีส่งออก (คน)	จำนวนคดีส่งออก (คดี)
0	2563	44	3209	3290	897	896	1176	1195	1018	1074	49	51	67	72	1.0	1.0	0	0	0
1	2563	49	3694	3745	2487	2486	644	656	477	515	41	50	37	29	3.0	2.0	0	0	2
2	2563	45	4558	4658	2141	2147	1065	1072	1242	1321	71	78	32	32	1.0	1.0	1	1	3
3	2563	42	2601	2741	874	879	906	942	713	803	44	51	53	56	5.0	3.0	0	0	6
4	2563	47	3336	3500	930	932	679	705	1048	1167	430	475	245	214	1.0	1.0	0	0	2
...
72	2563	34	7745	7892	3832	3840	988	1005	1394	1472	1190	1244	332	324	4.0	2.0	1	1	3
73	2563	46	4407	4524	1692	1697	1183	1195	1409	1502	55	61	55	56	NaN	NaN	0	0	9
74	2563	40	7642	7846	3595	3611	2010	2060	1814	1930	131	149	80	82	3.0	3.0	0	0	9
75	2563	48	2492	2586	1326	1330	401	414	610	667	120	139	25	25	4.0	3.0	0	0	1
76	2563	38	2976	3054	1405	1404	759	769	767	835	20	22	18	17	7.0	7.0	0	0	0

77 rows x 22 columns

ตรวจสอบค่า Missing ของตาราง data_03

In [32]: data_03.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง จะเห็นว่ามีคอลัมน์ที่มีค่า missing อยู่ทั้งหมด 4 คอลัมน์ คือ จำนวนคดีปาเข้า (คดี), จำนวนผู้ต้องหาคดีปาเข้า (คน), จำนวนคดีสมคบ (คดี) และ จำนวนผู้ต้องหาคดีสมคบ (คน) ดังนั้นจะทำการลบค่า missing ทิ้ง 4 คอลัมน์นี้

Out[32]:

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสนอ (คดี)	False
จำนวนผู้ต้องหาคดีเสนอ (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาคดีครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีจำหน่าย (คน)	False
จำนวนคดียัก (คดี)	False
จำนวนผู้ต้องหาคดียัก (คน)	False
จำนวนคดีปาเข้า (คดี)	True
จำนวนผู้ต้องหาคดีปาเข้า (คน)	True
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมคบ (คดี)	True
จำนวนผู้ต้องหาคดีสมคบ (คน)	True
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False
dtype: bool	

แก้ค่า missing โดยใช้ .dropna()

In [28]: data_03_dropped.shape # ตรวจสอบขนาดของข้อมูลหลังจากที่ .dropna()

Out[28]: (34, 22)

In [29]: data_03.shape # ตรวจสอบขนาดของข้อมูลก่อนที่จะ .dropna()

Out[29]: (77, 22)

In [30]: `## function หา % ของข้อมูลที่จะหายไปถ้า .dropna()`

```
def loss_data(data):  
    before = data.shape[0]  
    after = data.dropna().shape[0]  
    c = before - after  
    print(f'size before drop = {before}')  
    print(f'size after drop = {after}')  
    print(f'we loss {c*100/before} % of data')
```

In [31]: loss_data(data_03) # จะเห็นว่าหลังจากที่ .dropna() ข้อมูลหายไป 55.84 % ซึ่งเป็นจำนวนมาก ดังนั้นจึงเลือกที่จะลบค่า missing ด้วยวิธีอื่น

```
size before drop = 77  
size after drop = 34  
we loss 55.84415584415584 % of data
```

แก้ค่า missing โดยการเติมข้อมูลค่าที่เหมาะสมลงไป

```
In [33]: data_03[['จำนวนคดีนำเข้า (คดี)']] = data_03[['จำนวนคดีนำเข้า (คดี)']].fillna("0") # เติมค่า missing เป็น 0 เนื่องจากในตารางเป็นค่าว่างซึ่งอาจจะแปลว่าไม่มีจำนวนคดีเข้าเสียคดี  
นำเข้ายาเสพติด
```

```
In [34]: data_03.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง จะเห็นว่าคอลัมน์ของ จำนวนคดีนำเข้า (คดี) เป็น False แสดงว่าไม่มีค่า missing
```

```
Out[34]:
```

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสพ (คดี)	False
จำนวนผู้ต้องหาคดีเสพ (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาคดีครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีจำหน่าย (คน)	False
จำนวนคดีผลิต (คดี)	False
จำนวนผู้ต้องหาคดีผลิต (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหาคดีนำเข้า (คน)	True
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมคบ (คดี)	True
จำนวนผู้ต้องหาคดีสมคบ (คน)	True
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False
dtype:	bool

แก้ค่า missing โดยการเติมข้อมูลค่าที่เหมาะสมลงไป

```
In [35]: data_03[['จำนวนผู้ต้องหาคดีนำเข้า (คน)']] = data_03[['จำนวนผู้ต้องหาคดีนำเข้า (คน)']].fillna("0") # เดิมค่า missing เป็น 0 เนื่องจากในตารางเป็นค่าว่างซึ่งอาจจะแปลว่าไม่มีจำนวนผู้  
ต้องหาคดีในสเปซติดข้อหาป่าเข้ายาเสพติด
```

```
In [36]: data_03.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง จะเห็นว่า คอลัมน์ของ จำนวนผู้ต้องหาคดีนำเข้า (คน) เป็น False แสดงว่าไม่มีค่า missing
```

```
Out[36]:
```

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสพ (คดี)	False
จำนวนผู้ต้องหาคดีเสพ (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาคดีครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีจำหน่าย (คน)	False
จำนวนคดีผลิต (คดี)	False
จำนวนผู้ต้องหาคดีผลิต (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหาคดีนำเข้า (คน)	False
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมคบ (คดี)	True
จำนวนผู้ต้องหาคดีสมคบ (คน)	True
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False

dtype: bool

แก้ค่า missing โดยการเติมข้อมูลค่าที่เหมาะสมลงไป

```
In [37]: data_03[['จำนวนคดีสมคบ (คดี)']] = data_03[['จำนวนคดีสมคบ (คดี)']].fillna('0') # เติมค่า missing เป็น 0 เนื่องจากในตารางเป็นค่าว่างซึ่งอาจจะแปลว่าไม่มีจำนวนคดียาเสพติดข้อหาสมคบคิด
```

```
In [38]: data_03.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง จะเห็นว่า คอลัมน์ของ จำนวนคดีสมคบ (คดี) เป็น False แสดงว่าไม่มีค่า missing
```

```
Out[38]: ปึงบประมาณ      False
รหัสจังหวัด      False
จำนวนคดีทั้งหมด (คดี)      False
จำนวนผู้ต้องหาทั้งหมด (คน)      False
จำนวนคดีเสพ (คดี)      False
จำนวนผู้ต้องหาดีเสพ (คน)      False
จำนวนคดีครอบครอง (คดี)      False
จำนวนผู้ต้องหาดีครอบครอง (คน)      False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)      False
จำนวนผู้ต้องหาดีครอบครองเพื่อจำหน่าย (คน)      False
จำนวนคดีจำหน่าย (คดี)      False
จำนวนผู้ต้องหาดีจำหน่าย (คน)      False
จำนวนคดีผลิต (คดี)      False
จำนวนผู้ต้องหาดีผลิต (คน)      False
จำนวนคดีนำเข้า (คดี)      False
จำนวนผู้ต้องหาดีนำเข้า (คน)      False
จำนวนคดีส่งออก (คดี)      False
จำนวนผู้ต้องหาดีส่งออก (คน)      False
จำนวนคดีสมคบ (คดี)      False
จำนวนผู้ต้องหาดีสมคบ (คน)      True
จำนวนคดี 5 ข้อหาหลัก (คดี)      False
จำนวนผู้ต้องหาดี 5 ข้อหาหลัก (คน)      False
dtype: bool
```


แก้ค่า missing โดยการเติมข้อมูลค่าที่เหมาะสมลงไป

```
In [39]: data_03[["จำนวนผู้ต้องหาคดีสมคบ (คน)"]] = data_03[["จำนวนผู้ต้องหาคดีสมคบ (คน)"]].fillna('0') # เติมค่า missing เป็น 0 เนื่องจากในตารางเป็นค่าว่างซึ่งอาจจะแปลว่าไม่มีจำนวนผู้ต้องหาคดียาเสพติดข้อหาสมคบคิด
```

```
In [40]: data_03.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง จะเห็นว่า คอลัมน์ของ จำนวนผู้ต้องหาคดีสมคบ (คน) เป็น False แสดงว่าไม่มีค่า missing
```

```
Out[40]:
```

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสพ (คดี)	False
จำนวนผู้ต้องหาคดีเสพ (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาคดีครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีจำหน่าย (คน)	False
จำนวนคดีผลิต (คดี)	False
จำนวนผู้ต้องหาคดีผลิต (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหาคดีนำเข้า (คน)	False
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมคบ (คดี)	False
จำนวนผู้ต้องหาคดีสมคบ (คน)	False
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False

dtype: bool

ข้อมูลชุดที่ 4 ข้อมูลทั่วไปของจังหวัด

```
In [41]: data_04 = pd.read_csv(os.path.join(path, 'ข้อมูลทั่วไปของจังหวัด.csv')) # ไฟล์ที่เราต้องการ  
data_04 # print data ดูว่าหน้าตาเป็นอย่างไร
```

Out[41]:

	ชื่อจังหวัด	จำนวนประชากร (คน)	พื้นที่ (ตร.กม.)	รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)
0	เชียงใหม่	1,287,615	11,678.40	15056.38
1	เชียงใหม่	1,640,479	20,107.00	20443.21
2	น่าน	476,363	11,472.10	18041.55
3	พะเยา	486,304	6,335.10	14353.79
4	แพร่	447,564	6,538.60	21398.36
...
72	หนองคาย	509,395	3,027.00	20579.85
73	หนองบัวลำภู	502,868	3,859.00	18745.43
74	อุดรธานี	1,544,786	11,730.30	25766.53
75	อุบลราชธานี	1,813,088	15,744.80	18397.86
76	อำนาจเจริญ	372,137	3,161.20	20320.06

77 rows x 4 columns

ตรวจสอบค่า Missing ของตาราง data_04

```
In [43]: data_04.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้างเป็น False แสดงว่าไม่มีค่า missing
```

Out[43]:

ชื่อจังหวัด	False
จำนวนประชากร (คน)	False
พื้นที่ (ตร.กม.)	False
รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)	False
dtype:	bool

เชื่อมต่อตาราง

In [47]: # เชื่อมต่อตาราง data_03 (จำนวนคดี ผู้ต้องหาคดีอาญาเสพติด) กับ data_01 (รับแจ้งร้องเรียน/แจ้งเบาะแสยาเสพติด) โดยใช้ รหัสจังหวัด
merge_table = data_03.merge(data_01, how='left', left_on='รหัสจังหวัด', right_on='รหัสจังหวัด')
merge_table

Out[47]:

	ปีงบประมาณ	รหัสจังหวัด	จำนวนคดีทั้งหมด (คดี)	จำนวนผู้ต้องหาทั้งหมด (คน)	จำนวนคดีสาร (คดี)	จำนวนผู้ต้องหาสาร (คน)	จำนวนคดีครอบครอง (คดี)	จำนวนผู้ต้องหาครอบครอง (คน)	จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	จำนวนผู้ต้องหาครอบครองเพื่อจำหน่าย (คน)	จำนวนคดีจำหน่าย (คดี)	จำนวนผู้ต้องหาจำหน่าย (คน)	จำนวนคดีผลิต (คดี)	จำนวนผู้ต้องหาผลิต (คน)	จำนวนคดีนำเข้า (คดี)	จำนวนผู้ต้องหานำเข้า (คน)	จำนวนคดีส่งออก (คดี)	จำนวนผู้ต้องหาส่งออก (คน)	จ ค ส (
0	2563	10	21550	22641	9724	9754	6760	7142	4136	4395	352	373	489	899	34	18	0	0	4
1	2563	11	6990	7627	1816	1817	2646	2793	1906	2026	136	142	469	828	2	2	2	1	1
2	2563	12	3432	3632	937	937	1188	1231	1096	1184	80	87	120	177	0	0	0	0	1
3	2563	13	4235	4509	1192	1193	1378	1396	1369	1426	69	83	213	364	0	0	0	0	1
4	2563	14	4199	4355	1745	1747	940	945	1217	1323	167	198	119	129	0	0	0	0	9
...
72	2563	92	3688	3939	1496	1505	1015	1080	773	889	21	23	377	434	0	0	0	0	5
73	2563	93	2430	2661	489	488	896	961	632	754	35	46	374	408	0	0	0	0	2
74	2563	94	4563	4777	1009	1012	2061	2135	834	904	56	59	598	661	0	0	0	0	1
75	2563	95	3326	3679	330	335	2029	2182	616	755	65	74	282	328	3	4	0	0	0
76	2563	96	4231	4535	911	918	2258	2384	857	1006	15	17	185	205	2	2	0	0	1

77 rows x 32 columns

ตรวจสอบค่า Missing ของตาราง 2 ตารางที่เชื่อมต่อกันแล้ว

```
In [49]: merge_2table.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง สรุปคือไม่มีคอลัมน์ไหนที่มีค่า missing
```

```
Out[49]:
```

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสีย (คดี)	False
จำนวนผู้ต้องหาคดีเสีย (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาคดีครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำนำ (คดี)	False
จำนวนผู้ต้องหาคดีจำนำ (คน)	False
จำนวนคดียึด (คดี)	False
จำนวนผู้ต้องหาคดียึด (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหาคดีนำเข้า (คน)	False
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมคบ (คดี)	False
จำนวนผู้ต้องหาคดีสมคบ (คน)	False
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False
ชื่อจังหวัด	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (%)	False

dtype: bool

In [50]: # เชื่อมต่อตาราง data_03 (จำนวนคดี ผู้ต้องหาคดีอาสพคดี) กับ data_01 (รับเรื่องราวเรียน/แจ้งเบาะแสยาเสพติด) กับ data_02 (ปริมาณของกลางยาเสพติด) โดยใช้ รหัสจังหวัด
merge_3table = merge_2table.merge(data_02,how='left', left_on='รหัสจังหวัด',right_on='รหัสจังหวัด')
merge_3table

Out[50]:

	ปีงบประมาณ	รหัสจังหวัด	จำนวนคดีทั้งหมด (คดี)	จำนวนผู้ต้องหาทั้งหมด (คน)	จำนวนคดีเสพ (คดี)	จำนวนผู้ต้องหาคดีเสพ (คน)	จำนวนคดีครอบครอง (คดี)	จำนวนผู้ต้องหาคดีครอบครอง (คน)	จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	จำนวนคดีจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีจำหน่าย (คน)	จำนวนคดีผลิต (คดี)	จำนวนผู้ต้องหาคดีผลิต (คน)	จำนวนคดีนำเข้า (คดี)	จำนวนผู้ต้องหาคดีนำเข้า (คน)	จำนวนคดีส่งออก (คดี)	จำนวนผู้ต้องหาคดีส่งออก (คน)	จ ค ส (เ
0	2563	10	21550	22641	9724	9754	6760	7142	4136	4395	352	373	489	899	34	18	0	0	4
1	2563	11	6990	7627	1816	1817	2646	2793	1906	2026	136	142	469	828	2	2	2	1	1
2	2563	12	3432	3632	937	937	1188	1231	1096	1184	80	87	120	177	0	0	0	0	1
3	2563	13	4235	4509	1192	1193	1378	1396	1369	1426	69	83	213	364	0	0	0	0	1
4	2563	14	4199	4355	1745	1747	940	945	1217	1323	167	198	119	129	0	0	0	0	9
...
72	2563	92	3688	3939	1496	1505	1015	1080	773	889	21	23	377	434	0	0	0	0	5
73	2563	93	2430	2661	489	488	896	961	632	754	35	46	374	408	0	0	0	0	2
74	2563	94	4563	4777	1009	1012	2061	2135	834	904	56	59	598	661	0	0	0	0	1
75	2563	95	3326	3679	330	335	2029	2182	616	755	65	74	282	328	3	4	0	0	0
76	2563	96	4231	4535	911	918	2258	2384	857	1006	15	17	185	205	2	2	0	0	1

77 rows × 43 columns

ตรวจสอบค่า Missing ของตาราง 3 ตารางที่เชื่อมกันแล้ว

```
In [52]: merge_3table.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง สรุปคือในนี้คือคอลัมน์ไหนที่มีค่า missing
```

Out[52]:

ปีงบประมาณ	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเส้น (คดี)	False
จำนวนผู้ต้องหาคดีเส้น (คน)	False
จำนวนคดีครอบครัว (คดี)	False
จำนวนผู้ต้องหาคดีครอบครัว (คน)	False
จำนวนคดีครอบครัวเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีครอบครัวเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาคดีจำหน่าย (คน)	False
จำนวนคดีผิด (คดี)	False
จำนวนผู้ต้องหาคดีผิด (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหาคดีนำเข้า (คน)	False
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมทบ (คดี)	False
จำนวนผู้ต้องหาคดีสมทบ (คน)	False
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False
ข้อเท็จจริง	False
เรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรมทั้งหมด (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรมทั้งหมด (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรมทั้งหมด (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม พื้นที่ (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม พื้นที่ (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม พื้นที่ (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม บุคคล (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม บุคคล (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสอาชญากรรม บุคคล (%)	False
จังหวัด	False
ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)	False
ชนิดของกลางยาเสพติด	False
ยาบ้า (กิโลกรัม)	False
ไอซ์ (กิโลกรัม)	False
กัญชาแห้ง (กิโลกรัม)	False
เฮโรอีน (กิโลกรัม)	False
ดีดามีน (กิโลกรัม)	False
โคเคน (กิโลกรัม)	False
กระเทียม (กิโลกรัม)	False
ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)	False
dtype: bool	

In [53]: # เชื่อมต่อตาราง data_03 (จำนวนคดี ผู้ต้องหาคดีอาชญากรรม) กับ data_01 (รับเรื่องร้องเรียน/แจ้งเบาะแสอาชญากรรม) กับ data_02 (ปริมาณของกกลางยาเสพติด) กับ data_04 ข้อมูลทั่วไปของจังหวัด โดยใช้ ชื่อจังหวัด
merge_4table = merge_3table.merge(data_04,how='left', left_on='ชื่อจังหวัด',right_on='ชื่อจังหวัด')
merge_4table

Out[53]:

	ปีงบประมาณ	รหัสจังหวัด	จำนวนคดีทั้งหมด (คดี)	จำนวนผู้ต้องหาทั้งหมด (คน)	จำนวนคดีเสร็จ (คดี)	จำนวนผู้ต้องหาคดีเสร็จ (คน)	จำนวนคดีครอบครอง (คดี)	จำนวนผู้ต้องหาครอบครอง (คน)	จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีครอบครองเพื่อจำหน่าย (คน)	จำนวนคดีจำหน่าย (คดี)	จำนวนผู้ต้องหาคดีจำหน่าย (คน)	จำนวนคดีผลิต (คดี)	จำนวนผู้ต้องหาคดีผลิต (คน)	จำนวนคดีนำเข้า (คดี)	จำนวนผู้ต้องหาคดีนำเข้า (คน)	จำนวนคดีส่งออก (คดี)	จำนวนผู้ต้องหาคดีส่งออก (คน)	จ ค ส ()
0	2563	10	21550	22641	9724	9754	6760	7142	4136	4395	352	373	489	899	34	18	0	0	4
1	2563	11	6990	7627	1816	1817	2646	2793	1906	2026	136	142	469	828	2	2	2	1	1
2	2563	12	3432	3632	937	937	1188	1231	1096	1184	80	87	120	177	0	0	0	0	1
3	2563	13	4235	4509	1192	1193	1378	1396	1369	1426	69	83	213	364	0	0	0	0	1
4	2563	14	4199	4355	1745	1747	940	945	1217	1323	167	198	119	129	0	0	0	0	9
...
72	2563	92	3688	3939	1496	1505	1015	1080	773	889	21	23	377	434	0	0	0	0	5
73	2563	93	2430	2661	489	488	896	961	632	754	35	46	374	408	0	0	0	0	2
74	2563	94	4563	4777	1009	1012	2061	2135	834	904	56	59	598	661	0	0	0	0	1
75	2563	95	3326	3679	330	335	2029	2182	616	755	65	74	282	328	3	4	0	0	0
76	2563	96	4231	4535	911	918	2258	2384	857	1006	15	17	185	205	2	2	0	0	1

77 rows x 46 columns

ตรวจสอบค่า Missing ของตาราง 4 ตารางที่เชื่อมกันแล้ว

```
In [55]: merge_4table.isnull().any() # ดูว่าคอลัมน์ไหนมีค่า missing บ้าง สรุปคือไม่มีคอลัมน์ไหนที่มีค่า missing
```

```
Out[55]:
```

ชิงพระมาต	False
รหัสจังหวัด	False
จำนวนคดีทั้งหมด (คดี)	False
จำนวนผู้ต้องหาทั้งหมด (คน)	False
จำนวนคดีเสีย (คดี)	False
จำนวนผู้ต้องหาเสีย (คน)	False
จำนวนคดีครอบครอง (คดี)	False
จำนวนผู้ต้องหาครอบครอง (คน)	False
จำนวนคดีครอบครองเพื่อจำหน่าย (คดี)	False
จำนวนผู้ต้องหาครอบครองเพื่อจำหน่าย (คน)	False
จำนวนคดีจำหน่าย (คดี)	False
จำนวนผู้ต้องหาจำหน่าย (คน)	False
จำนวนคดีผลิต (คดี)	False
จำนวนผู้ต้องหาผลิต (คน)	False
จำนวนคดีนำเข้า (คดี)	False
จำนวนผู้ต้องหานำเข้า (คน)	False
จำนวนคดีส่งออก (คดี)	False
จำนวนผู้ต้องหาคดีส่งออก (คน)	False
จำนวนคดีสมทบ (คดี)	False
จำนวนผู้ต้องหาสมทบ (คน)	False
จำนวนคดี 5 ข้อหาหลัก (คดี)	False
จำนวนผู้ต้องหาคดี 5 ข้อหาหลัก (คน)	False
ข้อเท็จจริง	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด พื้นที่ (%)	False
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (เรื่อง)	False
การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (เรื่อง)	False
เปอร์เซ็นต์การดำเนินการต่อเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด บุคคล (%)	False
จังหวัด	False
ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)	False
ชนิดของกลางยาเสพติด	False
ยาบ้า (กิโลกรัม)	False
ไอซ์ (กิโลกรัม)	False
กัญชาแห้ง (กิโลกรัม)	False
เฮโรอีน (กิโลกรัม)	False
โคคาอีน (กิโลกรัม)	False
โคเคน (กิโลกรัม)	False
กระพ้อม (กิโลกรัม)	False
ปริมาณของกลางยาเสพติดทั้งหมดที่พบมากที่สุด (กิโลกรัม)	False
จำนวนประชากร (คน)	False
พื้นที่ (ตร.กม.)	False
รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)	False
dtype: bool	



ปัญหา

คือ เพื่อทำนายชนิดของยาเสพติด โดยใช้จำนวนประชากร, พื้นที่, รายได้เฉลี่ยต่อเดือนของครัวเรือน, จำนวนเรื่องร้องเรียน/แจ้งเบาะแสยาเสพติด, ปริมาณของกลางยาเสพติดและจำนวนคดีผู้ต้องหาคดียาเสพติด

เลือกคอลัมน์ที่ต้องการนำมาใช้งาน

```
In [56]: data = merge_4table[['ชื่อจังหวัด','จำนวนประชากร (คน)','พื้นที่ (ตร.กม.)','รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)','ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)','เรื่องร้องเรียนและ  
แจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)','จำนวนคดีทั้งหมด (คดี)','ชนิดของกลางยาเสพติด']]  
data
```

Out[56]:

	ชื่อจังหวัด	จำนวน ประชากร (คน)	พื้นที่ (ตร.กม.)	รายได้เฉลี่ยต่อเดือน ของครัวเรือน (บาท)	ปริมาณของกลางยาเสพติด ทั้งหมด (กิโลกรัม)	เรื่องร้องเรียนและแจ้งเบาะแ สยาเสพติดทั้งหมด (เรื่อง)	จำนวนคดี ทั้งหมด (คดี)	ชนิดของกลาง ยาเสพติด
0	กรุงเทพมหานคร	5,682,415	1,568.70	39459.36	3722.06515	2568	21550	กระช่อม
1	สมุทรปราการ	1,310,766	1,004.10	24729.42	403.45432	519	6990	กระช่อม
2	นนทบุรี	1,229,735	622.3	37502.22	741.87948	526	3432	กระช่อม
3	ปทุมธานี	1,129,115	1,525.90	46977.74	1215.42900	519	4235	กระช่อม
4	พระนครศรีอยุธยา	808,360	2,556.60	30589.95	1598.89086	376	4199	กระช่อม
...
72	ตรัง	643,072	4,917.50	26004.22	2032.81350	122	3688	กระช่อม
73	พัทลุง	524,857	3,424.50	20084.91	1317.79318	91	2430	กระช่อม
74	ปัตตานี	709,796	1,940.40	22903.84	6683.02730	44	4563	กระช่อม
75	ยะลา	527,295	4,521.10	16588.15	4242.57840	34	3326	กระช่อม
76	นราธิวาส	796,239	4,475.40	17716.50	8095.38658	35	4231	กระช่อม

77 rows x 8 columns

```
In [57]: data.shape
```

Out[57]: (77, 8)

กำหนดค่าของตัวแปร X, y หลังจากนั้นทำการตรวจสอบชนิดของข้อมูล

```
In [59]: # กำหนด X และ y
X = data[['จำนวนประชากร (คน)', 'พื้นที่ (ตร.กม.)', 'รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)', 'ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)', 'เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)', 'จำนวนคดีทั้งหมด (คดี)']]
y = data[['ชนิดของกลางยาเสพติด']]

In [60]: X.dtypes # ตรวจสอบชนิดของข้อมูล พบว่าข้อมูลของจำนวนประชากรและข้อมูลของพื้นที่ของแต่ละจังหวัดเป็น Object และจำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)
# และจำนวนคดีทั้งหมด (คดี) ชนิดของข้อมูลเป็น int จึงทำการเปลี่ยนให้เป็น Float เพื่อให้ข้อมูลเป็นข้อมูลชนิดเดียวกันทั้งหมด

Out[60]: จำนวนประชากร (คน)                object
พื้นที่ (ตร.กม.)                object
รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)    float64
ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม) float64
เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง) int64
จำนวนคดีทั้งหมด (คดี)            int64
dtype: object
```

เปลี่ยนชนิดของข้อมูลตัวแปร X

```
In [61]: import re
X = X.replace("[^\\d.]", "", regex=True).astype(float) # เปลี่ยนชนิดของข้อมูล
X
```

Out[61]:

	จำนวนประชากร (คน)	พื้นที่ (ตร.กม.)	รายได้เฉลี่ยต่อเดือนของครัว เรือน (บาท)	ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)	เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติด ทั้งหมด (เรื่อง)	จำนวนคดีทั้งหมด (คดี)
0	5682415.0	1568.7	39459.36	3722.06515	2568.0	21550.0
1	1310766.0	1004.1	24729.42	403.45432	519.0	6990.0
2	1229735.0	622.3	37502.22	741.87948	526.0	3432.0
3	1129115.0	1525.9	46977.74	1215.42900	519.0	4235.0
4	808360.0	2556.6	30589.95	1598.89086	376.0	4199.0
...
72	643072.0	4917.5	26004.22	2032.81350	122.0	3688.0
73	524857.0	3424.5	20084.91	1317.79318	91.0	2430.0
74	709796.0	1940.4	22903.84	6683.02730	44.0	4563.0
75	527295.0	4521.1	16588.15	4242.57840	34.0	3326.0
76	796239.0	4475.4	17716.50	8095.38658	35.0	4231.0

77 rows x 6 columns

```
In [62]: y.dtypes # ตรวจสอบชนิดของข้อมูล
```

Out[62]: ชนิดของกลางยาเสพติด object
dtype: object

หลังจากนั้นทำการแบ่งข้อมูลออกเป็น 2 ส่วน คือ Train – Test และ Train – Validation

เพื่อนำข้อมูลดังกล่าวไปใช้ในการหา model มาใช้ทำนายคำตอบที่เราต้องการ
นั่นคือ ชนิดของยาเสพติด

```
In [63]: from sklearn.model_selection import train_test_split
```

Train - Test

```
In [64]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=6)
```

```
In [65]: X_train.shape
```

```
Out[65]: (38, 6)
```

Train - Validation

```
In [66]: X_train2, X_val, y_train2, y_val = train_test_split(X_train, y_train, test_size=0.3, random_state=6)
```

```
In [67]: X_train2.shape
```

```
Out[67]: (26, 6)
```



01

Classification

Decision Tree



PAGE 30

Classification

Decision Tree

Mytree 2

```
In [113...  
# Define  
mytree2 = DecisionTreeClassifier(max_leaf_nodes=50,criterion='gini')  
# Train  
mytree2.fit(X_train2,y_train2)  
# Test  
mytree2_result = mytree2.predict(X_val)  
accuracy_score(y_val,mytree2_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

Out[113... 0.25

Mytree 1

```
In [119...  
# Define  
mytree1 = DecisionTreeClassifier(random_state=1) # กำหนดพารามิเตอร์ให้กับ model  
# Train  
mytree1.fit(X_train2,y_train2) # ฝึกสอนตัวแบบ  
# Test  
mytree1_result = mytree1.predict(X_val)  
accuracy_score(y_val,mytree1_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

Out[119... 0.3333333333333333

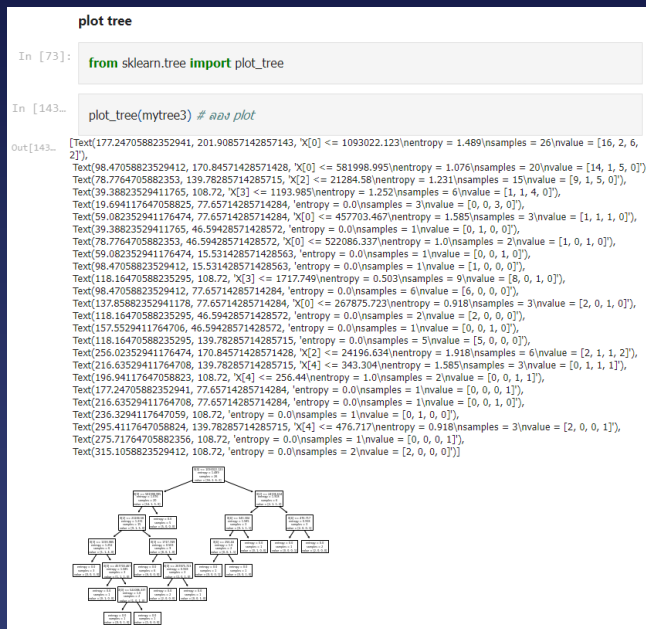
Mytree 3

```
In [138...  
# Define  
mytree3 = DecisionTreeClassifier(criterion='entropy', splitter='random',random_state=1) # กำหนดพ  
# Train  
mytree3.fit(X_train2,y_train2)  
# Test  
mytree3_result = mytree3.predict(X_val)  
accuracy_score(y_val,mytree3_result) # นั่นคือ มีความแม่นยำ 58.33 %
```

Out[138... 0.4166666666666667

Classification

Decision Tree





02

Classification

KNN / K-Nearest Neighbors



PAGE 33



Classification

KNN / K-Nearest Neighbors

KNN

Import

```
In [77]: from sklearn.neighbors import KNeighborsClassifier # เรียกใช้อัลกอริทึมที่เราต้องการ
```

KNN 1

```
In [78]: # Define
neigh1 = KNeighborsClassifier(n_neighbors=3, weights='uniform') # ถ้าเพื่อนบ้านที่ใกล้สุด 3 คนและเชื่อทุกคนเท่ากัน
# Train
neigh1.fit(X_train2, y_train2)
# Test
KNN1_result = neigh1.predict(X_val)
accuracy_score(y_val, KNN1_result) # นั่นคือ มีความแม่นยำ 33.33 %
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
after removing the cwd from sys.path.

```
Out[78]: 0.3333333333333333
```

Classification

KNN / K-Nearest Neighbors

KNN 2

In [122...]

```
# Define
neigh2 = KNeighborsClassifier(n_neighbors=3, weights='distance') # ถ้าเพื่อนบ้าน 3 คน และเชื่อเพื่อนบ้านที่ใกล้ที่สุดมากกว่าเพื่อนบ้าน
# Train
neigh2.fit(X_train2,y_train2)
# Test
KNN2_result = neigh2.predict(X_val)
accuracy_score(y_val, KNN2_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
after removing the cwd from sys.path.

Out[122...]

0.4166666666666667

Classification

KNN / K-Nearest Neighbors

KNN 3

In [80]:

```
# Define
neigh3 = KNeighborsClassifier(n_neighbors=1) # เลือกเพื่อนบ้านที่ใกล้ที่สุดแค่คนเดียว
# Train
neigh3.fit(X_train2,y_train2)
# Test
KNN3_result = neigh3.predict(X_val)
accuracy_score(y_val, KNN3_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
after removing the cwd from sys.path.

Out[80]:


0.4166666666666667



03

Classification

ANN / Neural Network



PAGE 37



Classification

ANN / Neural Network

Neural Network

Import

```
In [81]: from sklearn.neural_network import MLPClassifier # เรียกใช้โมเดลที่เรากำลังต้องการ
```

ANN 1

```
In [133... # Define
perceptron1 = MLPClassifier(random_state=6, max_iter=50, hidden_layer_sizes=1, learning_rate_init=0.01) # กำหนดพารามิเตอร์ไว้
# Train
perceptron1.fit(X_train2, y_train2)
# Test
perceptron1_result = perceptron1.predict(X_val)
accuracy_score(y_val, perceptron1_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:571: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (50) reached and the optimization hasn't converged yet.
  % self.max_iter, ConvergenceWarning)
```

```
Out[133... 0.08333333333333333
```

Classification

ANN / Neural Network

ANN2

In [150...

```
perceptron2 = MLPClassifier(random_state=6, max_iter=2000,hidden_layer_sizes=10,learning_rate_init=0.005)
#Train
perceptron2.fit(X_train2,y_train2)
#Test
perceptron2_result = perceptron2.predict(X_val)
accuracy_score(y_val,perceptron2_result) # นั่นคือ มีความแม่นยำ 41.67 %
```

/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().

y = column_or_1d(y, warn=True)

Out[150...

0.4166666666666667

Classification

ANN / Neural Network

ANN3

```
In [85]: perceptron3 = MLPClassifier(random_state=6, max_iter=20000, hidden_layer_sizes=10, learning_rate_init=0.001)
          # Train
          perceptron3.fit(X_train2, y_train2)
          # Test
          perceptron3_result = perceptron3.predict(X_val)
          accuracy_score(y_val, perceptron3_result) # นั่นคือ มีความแม่นยำ 16.67 %
```

/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

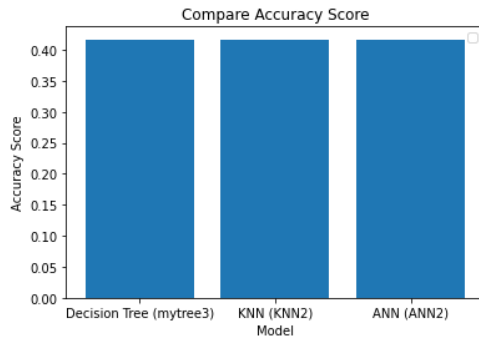
```
Out[85]: 0.16666666666666666
```


เปรียบเทียบความแม่นยำของการทำ classification ด้วย model ต่างๆ

In [142...]

```
# กราฟเปรียบเทียบความแม่นยำของการวัดผลการทำ classification ด้วย model ต่างๆ
from matplotlib import pyplot as plt
labels = ["Decision Tree (mytree3)", "KNN (KNN2)", "ANN (ANN2)"]
accuracy = [accuracy_score(y_val, mytree3_result), accuracy_score(y_val, KNN2_result), accuracy_score(y_val, perceptron2_result)]
#plot
plt.bar(labels, accuracy)
plt.ylabel("Accuracy Score")
plt.xlabel("Model")
plt.title("Compare Accuracy Score")
plt.legend();
```

No handles with labels found to put in legend.



พบว่า จากการวัดผลการทำ classification ด้วย model ต่างๆ Model Decision Tree(Mytree 3), KNN(KNN 2 , KNN 3) และ ANN (ANN 2) มีค่า Accuracy มากที่สุด คือ 0.4166666666666667 นั่นคือมีความแม่นยำ 41.67 %

Evaluation

Retain & Evaluation

เนื่องจากค่า Accuracy ของ Mytree 3, KNN 2 , KNN 3 และ ANN 2 มีค่าเท่ากัน คือ 0.416666666666667 จึงทำการ Retrain ทั้ง 4 ตัวแบบเพื่อหาตัวแบบที่เหมาะสมที่สุด

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix # เรียกใช้อัลกอริทึมที่เราต้องการ
```

Decision_Tree

เลือกตัวแบบ mytree3 แล้วนำมา Train ใหม่

```
In [144]: # Define
mytree3_final = DecisionTreeClassifier(criterion='entropy', splitter='random', random_state=1) # กำหนดพารามิเตอร์ให้มัน model
# Train
mytree3_final.fit(X_train, y_train)
# Test
mytree3final_result = mytree3_final.predict(X_test)
accuracy_score(y_test, mytree3final_result) # มันคือ มีความแม่นยำ 53.85 %

Out[144]: 0.5384615384615384
```

Neural_Network

เลือกตัวแบบ ANN 2 แล้วนำมา Train ใหม่

```
In [151]: # Define
perceptron2_final = MLPClassifier(random_state=6, max_iter=2000, hidden_layer_sizes=10, learning_rate_init=0.005)
# Train
perceptron2_final.fit(X_train, y_train)
# Test
perceptron2final_result = perceptron2_final.predict(X_test)
accuracy_score(y_test, perceptron2final_result) # มันคือ มีความแม่นยำ 71.79 %

/usr/local/lib/python3.7/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:934: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, 1), for example using ravel().
y = column_or_1d(y, warn=True)

Out[151]: 0.717948717948718
```

KNN / K-Nearest Neighbors

เลือกตัวแบบ KNN 2 แล้วนำมา Train ใหม่

```
In [145]: # Define
neigh2_final = KNeighborsClassifier(n_neighbors=3, weights='distance') # กำหนดเพื่อนบ้าน 3 คน และถ่วงน้ำหนักให้ใกล้เพื่อนบ้านการันตีความแม่นยำ
# Train
neigh2_final.fit(X_train, y_train)
# Test
KNN2final_result = neigh2_final.predict(X_test)
accuracy_score(y_test, KNN2final_result) # มันคือ มีความแม่นยำ 28.21 %

/usr/local/lib/python3.7/dist-packages/pykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, 1), for example using ravel().
after removing the cod from sys.path.
0.28205128205128205
```

เลือกตัวแบบ KNN 3 แล้วนำมา Train ใหม่

```
In [146]: # Define
neigh3_final = KNeighborsClassifier(n_neighbors=1) # เลือกเพื่อนบ้านใกล้ที่สุดเพื่อนบ้าน
# Train
neigh3_final.fit(X_train, y_train)
# Test
KNN3final_result = neigh3_final.predict(X_test)
accuracy_score(y_test, KNN3final_result) # มันคือ มีความแม่นยำ 35.90 %

/usr/local/lib/python3.7/dist-packages/pykernel_launcher.py:4: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, 1), for example using ravel().
after removing the cod from sys.path.
0.358974358974359
```

Evaluation

Neural_Network

```
In [88]: confusion_matrix(y_test,perceptron2final_result)

Out[88]: array([[12, 0, 0, 0],
               [ 0, 0, 1, 0],
               [ 5, 0, 16, 2],
               [ 1, 0, 2, 0]])

In [89]: ref = classification_report(y_test,perceptron2final_result)
print(ref)
```

	precision	recall	f1-score	support
กระเทียม	0.67	1.00	0.80	12
กะหล่ำปลี	0.00	0.00	0.00	1
ยาสูบ	0.84	0.70	0.76	23
ไอซ์	0.00	0.00	0.00	3
accuracy			0.72	39
macro avg	0.38	0.42	0.39	39
weighted avg	0.70	0.72	0.70	39



Data Visualization



Data_Visualization

Data_Visualization

ดูการกระจายชนิดของกลางยาเสพติดที่พบมากที่สุดในแต่ละจังหวัด ว่ามีกี่จังหวัดที่พบของกลางยาเสพติดแต่ละชนิดบ้าง

```
In [154]: # ทำรายการกระจายชนิดของกลางยาเสพติดที่พบมากที่สุดในแต่ละจังหวัด ว่ามีกี่จังหวัดที่พบของกลางยาเสพติดแต่ละชนิดบ้าง
data_02.groupby(['ชนิดของกลางยาเสพติด']).count().iloc[:,1]
```

```
Out[154]: ชนิดของกลางยาเสพติด
กระเทียม      33
กัญชาแห้ง      4
ยาบ้า      34
ไอซ์      6
Name: จังหวัด, dtype: int64
```

พบว่า ชนิดของกลางยาเสพติดที่พบมากที่สุด คือ ยาบ้า พบทั้งหมด 34 จังหวัด รองลงมาคือ กระเทียม พบทั้งหมด 33 จังหวัด ไอซ์ พบทั้งหมด 6 จังหวัด และกัญชาแห้งพบทั้งหมด 4 จังหวัด

```
In [146]: # จำนวนของจังหวัดที่พบของกลางยาเสพติดที่พบมากที่สุดและชนิด
กระเทียม = data_02.groupby(['ชนิดของกลางยาเสพติด']).count().iloc[0,1] # จำนวนของจังหวัดที่พบของกลางยาเสพติดที่พบมากที่สุด คือ ก
กัญชาแห้ง = data_02.groupby(['ชนิดของกลางยาเสพติด']).count().iloc[1,1] # จำนวนของจังหวัดที่พบของกลางยาเสพติดที่พบมากที่สุด คือ ก
ยาบ้า = data_02.groupby(['ชนิดของกลางยาเสพติด']).count().iloc[2,1] # จำนวนของจังหวัดที่พบของกลางยาเสพติดที่พบมากที่สุด คือ ยา
ไอซ์ = data_02.groupby(['ชนิดของกลางยาเสพติด']).count().iloc[3,1] # จำนวนของจังหวัดที่พบของกลางยาเสพติดที่พบมากที่สุด คือ ไอซ์
```

```
In [93]: # ทำการติดตั้ง font ภาษาไทยเพื่อนำมาใช้ในการ plot กราฟ
import matplotlib as matplotlib # ทำการเปิด version ของตัว library ๑ ซึ่งเข้าใจว่าต้องเป็น version 3.2 ขึ้นไปเท่านั้นถึงจะใช้ได้ ถ้า version
matplotlib.__version__
```

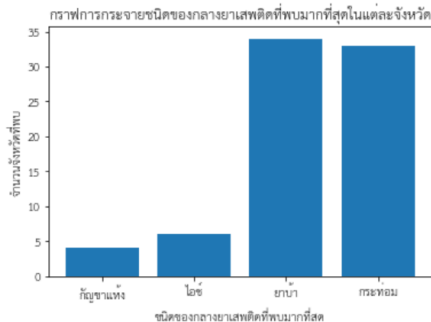
```
Out[93]: '3.2.2'
```

```
In [94]: # ทำการดาวน์โหลด font ในที่นี้จะใช้โค้ดด้านล่างนี้ โหลด font ที่ชื่อว่า thsarabunnew
!wget -q https://github.com/Phonbopit/sarabun-webfont/raw/master/fonts/thsarabunnew-webfont.ttf
```

```
In [95]: matplotlib.font_manager.fontManager.addfont('thsarabunnew-webfont.ttf') # เหม่ font ที่เรา download มาเข้าไปใน font manager ของ
matplotlib.rc('font', family='TH Sarabun New')
```

In [152...]

```
# กราฟการกระจายชนิดของกลางยาเสพติดที่พบมากที่สุดในแต่ละจังหวัด ว่ามีกี่จังหวัดที่พบของกลางยาเสพติดแต่ละชนิดบ้าง
from matplotlib import pyplot as plt # เรียกใช้ไลบรารีที่เกี่ยวกับการ
ชนิดของกลางยาเสพติดที่พบมากที่สุด = ['กัญชาแห้ง','ไอซ์','ยาบ้า','กระเทียม']
จำนวนจังหวัดที่พบ = [กัญชาแห้ง,ไอซ์,ยาบ้า,กระเทียม]
#plot
plt.bar(ชนิดของกลางยาเสพติดที่พบมากที่สุด,จำนวนจังหวัดที่พบ)
plt.ylabel('จำนวนจังหวัดที่พบ')
plt.xlabel('ชนิดของกลางยาเสพติดที่พบมากที่สุด')
plt.title('กราฟการกระจายชนิดของกลางยาเสพติดที่พบมากที่สุดในแต่ละจังหวัด')
plt.show();
```



Data_Visualization

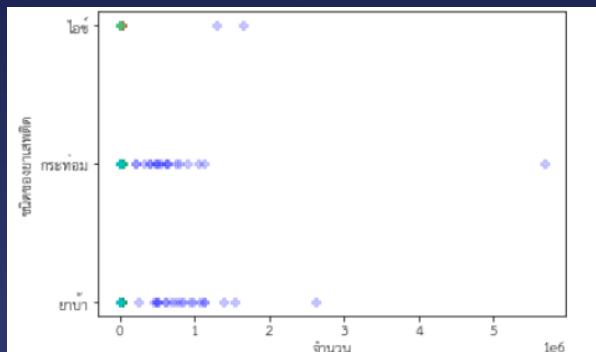
In [95]: `perceptron2final_result` # ค่าของ y หรือผลการทำนายชนิดของยาเสพติดที่ได้จาก `model` ผลลัพธ์ที่ได้เป็นดังนี้

Out[95]: `array(['ยาบ้า', 'ยาบ้า', 'กระต่อม', 'กระต่อม', 'ยาบ้า', 'กระต่อม', 'ยาบ้า', 'ยาบ้า', 'กระต่อม', 'กระต่อม', 'ไอซ์', 'ยาบ้า', 'ยาบ้า', 'กระต่อม', 'กระต่อม', 'กระต่อม', 'ยาบ้า', 'กระต่อม', 'ยาบ้า', 'กระต่อม', 'ยาบ้า', 'กระต่อม', 'ยาบ้า', 'ยาบ้า', 'ยาบ้า', 'ยาบ้า', 'ยาบ้า', 'กระต่อม', 'กระต่อม', 'ไอซ์', 'กระต่อม', 'กระต่อม', 'ยาบ้า', 'กระต่อม'], dtype='<U9')`

In [96]: `X_test` # ค่าของ X หรือข้อมูลที่เรานำมาใช้ในการทำนายหาชนิดของยาเสพติด

Out[96]:

	จำนวนประชากร (คน)	พื้นที่ (ตร.กม.)	รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)	ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)	เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)	จำนวนคดีทั้งหมด (คดี)
29	1544786.0	11730.3	25766.53000	164.12746	324.0	5175.0
34	982578.0	6946.7	19088.97000	384.13460	215.0	4407.0
23	539542.0	4161.7	17768.39000	61.40210	78.0	3035.0

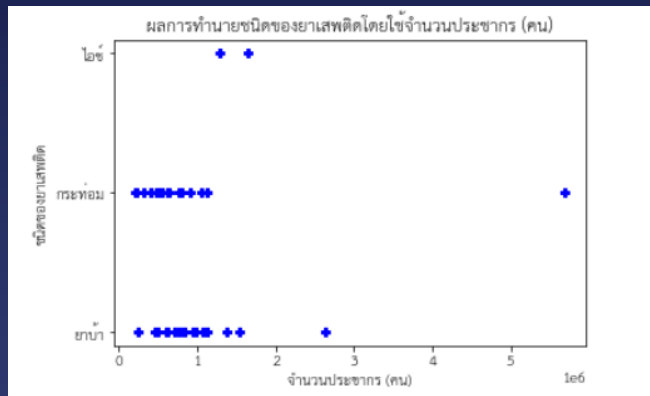


กราฟผลการทำนายชนิดของยาเสพติด

โดยใช้จำนวนประชากร, พื้นที่, รายได้เฉลี่ยต่อเดือนของครัวเรือน, จำนวนเรื่องร้องเรียน/แจ้งเบาะแสยาเสพติด, ปริมาณของกลางยาเสพติดและจำนวนคดีต้องหาคดียาเสพติด

In [100]: `from matplotlib import pyplot as plt # เรียกใช้ไลบรารีที่เรากำลังต้องการ`

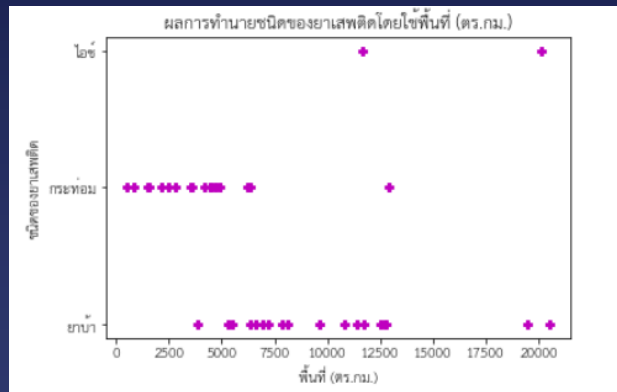
In [101]: `# กราฟผลการทำนายชนิดของยาเสพติด โดยใช้จำนวนประชากร, พื้นที่, รายได้เฉลี่ยต่อเดือนของครัวเรือน, จำนวนเรื่องร้องเรียน/แจ้งเบาะแสยาเสพติด, ปริมาณของกลางยาเสพติดและจำนวนคดีต้องหาคดียาเสพติด`
`plt.plot(X_test['จำนวนประชากร (คน)'],perceptron2final_result,'b',alpha = 0.2)`
`plt.plot(X_test['พื้นที่ (ตร.กม.)'],perceptron2final_result,'r',alpha = 0.8)`
`plt.plot(X_test['รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)'],perceptron2final_result,'g',alpha = 0.1)`
`plt.plot(X_test['ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)'],perceptron2final_result,'m',alpha = 0.8)`
`plt.plot(X_test['เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)'],perceptron2final_result,'y',alpha = 0.8)`
`plt.plot(X_test['จำนวนคดีทั้งหมด (คดี)'],perceptron2final_result,'c',alpha = 0.3)`
`plt.xlabel('จำนวน')`
`plt.ylabel('ชนิดของยาเสพติด')`
`plt.title('ผลการทำนายชนิดของยาเสพติด โดยใช้จำนวนประชากร, พื้นที่, รายได้เฉลี่ยต่อเดือนของครัวเรือน, จำนวนเรื่องร้องเรียน/แจ้งเบาะแสยาเสพติด, ปริมาณของกลางยาเสพติดและจำนวนคดีต้องหาคดียาเสพติด')`
`plt.show();`



กราฟผลการทำนายชนิดของยาเสพติด โดยใช้จำนวนประชากร (คน)

```
In [102]: # กราฟผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนประชากร (คน)
plt.plot(X_test["จำนวนประชากร (คน)"],perceptron2final_result,'Pb')
plt.xlabel("จำนวนประชากร (คน)")
plt.ylabel("ชนิดของยาเสพติด")
plt.title("ผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนประชากร (คน)")
plt.show();
```

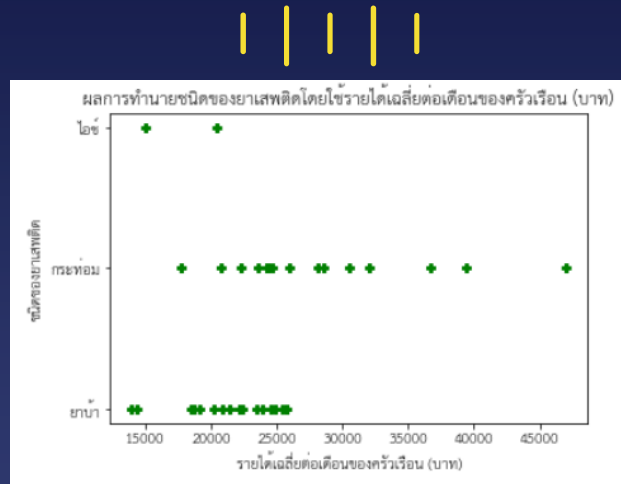
คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนประชากร (คน) พบว่า ถ้าสมมติว่าจังหวัดนั้นมีจำนวนประชากรประมาณ 150,000 คน ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า , สมมติว่าจังหวัดนั้นมีจำนวนประชากรประมาณ 160,000 คน ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ และสมมติว่าจังหวัดนั้นมีจำนวนประชากรประมาณ 580,000 คน ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระท่อม เป็นต้น



กราฟผลการทำนายชนิดของยาเสพติด โดยใช้พื้นที่ (ตร.กม.)

```
In [103]: # กราฟผลการทำนายชนิดของยาเสพติดโดยใช้พื้นที่ (ตร.กม.)
plt.plot(X_test['พื้นที่ (ตร.กม.)'],perceptron2final_result,'Pm')
plt.xlabel('พื้นที่ (ตร.กม.)')
plt.ylabel('ชนิดของยาเสพติด')
plt.title('ผลการทำนายชนิดของยาเสพติดโดยใช้พื้นที่ (ตร.กม.)')
plt.show();
```

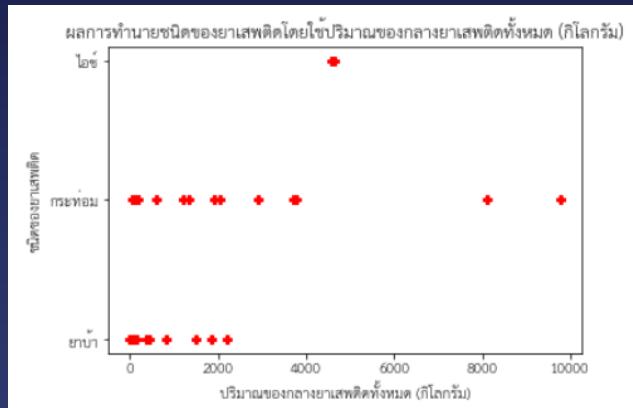
คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้พื้นที่ (ตร.กม.) พบว่า ถ้าสมมติว่าจังหวัดนั้นมีพื้นที่ประมาณ 7,500 ตร.กม.ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า , สมมติว่าจังหวัดนั้นมีพื้นที่ประมาณ 2,500 ตร.กม. ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระท่อม และสมมติว่าจังหวัดนั้นมีพื้นที่ประมาณ 20,000 ตร.กม. ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ เป็นต้น



กราฟผลการทำนายชนิดของยาเสพติด โดยใช้รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)

```
In [104]: # กราฟผลการทำนายชนิดของยาเสพติดโดยใช้รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)
plt.plot(X_test['รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)'], perceptron2final_result, 'Pg')
plt.xlabel('รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)')
plt.ylabel('ชนิดของยาเสพติด')
plt.title('ผลการทำนายชนิดของยาเสพติดโดยใช้รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท)')
plt.show();
```

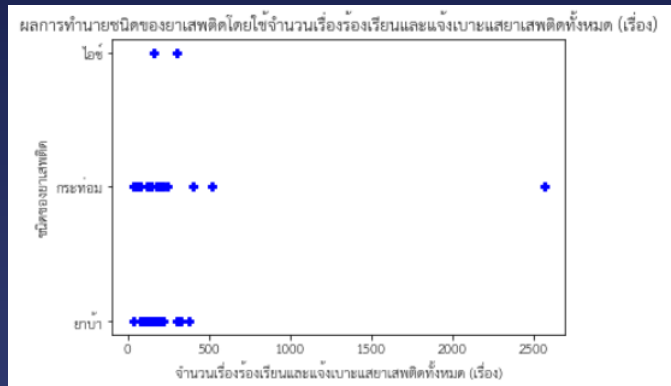
คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้รายได้เฉลี่ยต่อเดือนของครัวเรือน (บาท) พบว่า ถ้าสมมติว่า จังหวัดนั้นมีรายได้เฉลี่ยต่อเดือนของครัวเรือนประมาณ 15,000 บาท ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ , สมมติว่าจังหวัดนั้นมีรายได้เฉลี่ยต่อเดือนของครัวเรือนประมาณ 30,000 บาท ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระเทียม และสมมติว่าจังหวัดนั้นมีรายได้เฉลี่ยต่อเดือนของครัวเรือนประมาณ 25,000 บาท ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า เป็นต้น



กราฟผลการทำนายชนิดของยาเสพติด โดยใช้ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)

```
In [105]: # กราฟผลการทำนายชนิดของยาเสพติดโดยใช้ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)
plt.plot(X_test['ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)'],perceptron2final_result,'Pr')
plt.xlabel('ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)')
plt.ylabel('ชนิดของยาเสพติด')
plt.title('ผลการทำนายชนิดของยาเสพติดโดยใช้ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม)')
plt.show();
```

คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้ปริมาณของกลางยาเสพติดทั้งหมด (กิโลกรัม) พบว่า ถ้าสมมติว่าจังหวัดนั้นพบปริมาณของกลางยาเสพติดทั้งหมดประมาณ 1,200 กิโลกรัม ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า , สมมติว่าจังหวัดนั้นพบปริมาณของกลางยาเสพติดทั้งหมดประมาณ 3,000 กิโลกรัม ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระต้อม และสมมติว่าจังหวัดนั้นพบปริมาณของกลางยาเสพติดทั้งหมดประมาณ 4,500 กิโลกรัม ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ เป็นต้น

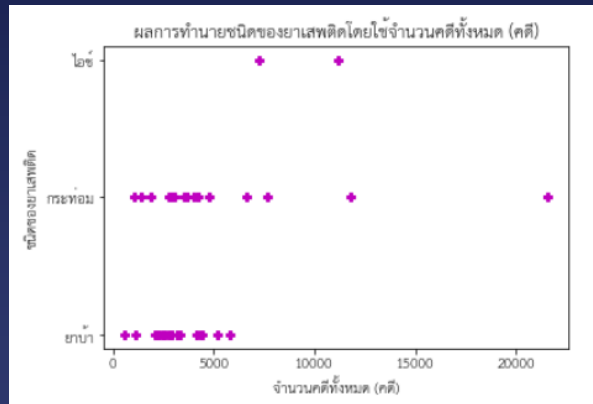


กราฟผลการทำนายชนิดของยาเสพติด

โดยใช้เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)

In [106]: `# กราฟผลการทำนายชนิดของยาเสพติดโดยใช้เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)
plt.plot(X_test['เรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)'],perceptron2final_result,'Pb')
plt.xlabel("จำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)")
plt.ylabel("ชนิดของยาเสพติด")
plt.title("ผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง)")
plt.show();`

คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมด (เรื่อง) พบว่า ถ้าสมมติว่าจังหวัดนั้นมีจำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมดประมาณ 400 เรื่อง ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า , สมมติว่าจังหวัดนั้นมีจำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมดประมาณ 500 เรื่อง ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระถ่อม และสมมติว่าจังหวัดนั้นมีจำนวนเรื่องร้องเรียนและแจ้งเบาะแสยาเสพติดทั้งหมดประมาณ 300 เรื่อง ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ เป็นต้น



กราฟผลการทำนายชนิดของยาเสพติด โดยใช้จำนวนคดีทั้งหมด (คดี)

```
In [107]: # กราฟผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนคดีทั้งหมด (คดี)
plt.plot(X_test['จำนวนคดีทั้งหมด (คดี)'], perceptron2final_result, 'Pm')
plt.xlabel('จำนวนคดีทั้งหมด (คดี)')
plt.ylabel('ชนิดของยาเสพติด')
plt.title('ผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนคดีทั้งหมด (คดี)')
plt.show();
```

คำอธิบาย : จากกราฟผลการทำนายชนิดของยาเสพติดโดยใช้จำนวนคดีทั้งหมด (คดี) พบว่า ถ้าสมมติว่าจังหวัดนั้นมีจำนวนคดีทั้งหมดประมาณ 7,500 คดี ชนิดของยาเสพติดที่พบมากที่สุดก็คือ กระตอม , สมมติว่าจังหวัดนั้นมีจำนวนคดีทั้งหมดประมาณ 5,000 คดี ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ยาบ้า และสมมติว่าจังหวัดนั้นมีจำนวนคดีทั้งหมดประมาณ 7,000 คดี ชนิดของยาเสพติดที่พบมากที่สุดก็คือ ไอซ์ เป็นต้น

รายชื่อสมาชิกในกลุ่ม

ชื่อกลุ่ม : เอกาไรสติ

นางสาวกัญญาวิร์ ศรีเทียมเงิน 623020511-1

นางสาวชลริชา ศาลางาม 623020518-7

นางสาวสุภาภรณ์ บุตุธรรม 623020543-8

นางสาวจิราพร กลบรัตน์ 623020762-6

นางสาววิภาณดา หงษ์บุญมี 623020764-2

***Data Warehouse &
Data Mining 2021***

