

Report: Thai Numbers Recognition

1. Data collection

ทำการเก็บรวบรวมข้อมูลเป็นไฟล์รูปภาพที่เก็บตัวเลข ตั้งแต่เลข 0 ถึง 9 (เลขไทย)

2. Data cleansing

2.1 Resize

- ทำการหาขอบของรูปด้วยฟังก์ชัน count up, count down, count left และ count right

```
def Count_Up(img,img_check):
    count_up = 0
    check_up = False
    for i in range(28):
        for j in range(28):
            if img[i,j] < max(img_check):
                check_up = True
                break
        if check_up == True:
            break
        count_up += 1
    return count_up

def Count_Down(img,img_check):
    count_down = 0
    check_down = False
    for i in range(27,0,-1):
        for j in range(28):
            if img[i,j] < max(img_check):
                check_down = True
                break
        if check_down == True:
            break
        count_down += 1
    return count_down

def Count_Left(img,img_check):
    count_left = 0
    check_left = False
    for i in range(28):
        for j in range(28):
            if img[j,i] < max(img_check):
                check_left = True
                break
        if check_left == True:
            break
        count_left += 1
    return count_left

def Count_Right(img,img_check):
    count_right = 0
    check_right = False
    for i in range(27,0,-1):
        for j in range(28):
            if img[j,i] < max(img_check):
                check_right = True
                break
        if check_right == True:
            break
        count_right += 1
    return count_right
```

- ปรับรูปให้อยู่กึ่งกลางเฟรมด้วยฟังก์ชัน change_position

```
def change_position(img,img_check):
    crop_img = img
    dummy = np.full((28,255),dtype='uint8')
    dummy2 = np.full((28,1),dtype='uint8')

    count_array = []
    count_array.append(Count_Up(crop_img,img_check))
    count_array.append(Count_Down(crop_img,img_check))
    count_array.append(Count_Left(crop_img,img_check))
    count_array.append(Count_Right(crop_img,img_check))

    #vertical_change_position
    avg_vertical = (count_array[0] + count_array[1])/2
    while Count_Up(crop_img,img_check) - avg_vertical > 0.5:
        crop_img = crop_img[1:crop_img.shape[0], :]
        crop_img = np.vstack([crop_img,dummy])
    while Count_Down(crop_img,img_check) - avg_vertical > 0.5:
        crop_img = crop_img[0:crop_img.shape[0]-1, :]
        crop_img = np.vstack([dummy,crop_img])

    #horizontal_change_position
    avg_horizontal = (count_array[2] + count_array[3])/2
    while Count_Left(crop_img,img_check) - avg_horizontal > 0.5:
        crop_img = crop_img[:,1:crop_img.shape[1]]
        crop_img = np.hstack([crop_img,dummy2])
    while Count_Right(crop_img,img_check) - avg_horizontal > 0.5:
        crop_img = crop_img[:, 0:crop_img.shape[1]-1]
        crop_img = np.hstack([dummy2,crop_img])

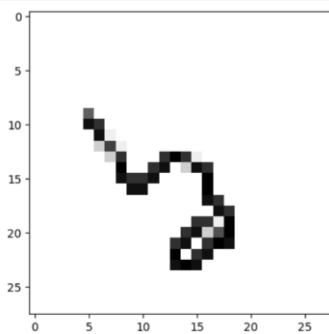
    return crop_img
```

- ทำการ crop รูปออกมาเป็นไฟล์ใหม่ โดยไม่ว่าไฟล์ input จะอยู่ในรูปแบบไฟล์ .png , .jpeg หรือว่ามีขนาดเท่าใดก็ตาม จะถูกปรับให้เป็นไฟล์ .png ขนาด 28x28 pixels

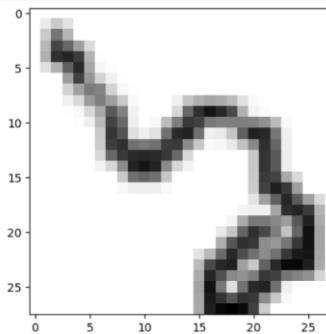
```
for j in range(0,10):
    for i in range(1,41):
        count_array = []
        file = str(j) + "-" + str(i) + ".png"
        img = cv2.imread(file,cv2.IMREAD_GRAYSCALE)
        #img_check = img.flatten()
        img_check = [0,240]
        crop_img = change_position(img,img_check)
        count_array.append(Count_Up(crop_img,img_check))
        count_array.append(Count_Down(crop_img,img_check))
        count_array.append(Count_Left(crop_img,img_check))
        count_array.append(Count_Right(crop_img,img_check))

        crop_img = crop_img[min(count_array):crop_img.shape[0], :]
        crop_img = crop_img[0:crop_img.shape[0]-min(count_array), :]
        crop_img = crop_img[:, min(count_array):crop_img.shape[1]]
        crop_img = crop_img[:, 0:crop_img.shape[1]-min(count_array)]
        crop_img = cv2.resize(crop_img,(28,28))

        newfile = "new_" + str(j) + "-" + str(i) + ".png"
        cv2.imwrite(newfile,crop_img)
```



(1)



(2)

ตัวอย่างรูปก่อน (1) และหลัง (2) ทำการ Resize

2.2 Convert to csv

ทำการ convert ไฟล์รูปภาพที่ resize เรียบร้อยแล้วแปลงเป็น csv เพื่อทำการสร้าง model ในขั้นตอนต่อไป

```
import cv2
# numpy กับ matplotlib เองก็ใช้ตลอดด้วย แนะนำให้ import ไปด้วยทุกครั้ง
import numpy as np
import matplotlib.pyplot as plt
import csv
import pandas as pd

lst = []
col = ['y']
for i in range(1,785):
    name = 'pixel'+str(i)
    col.append(name)
newfile = "pixel_num_new.csv"
with open(newfile, 'a', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(col)
for k in range(0,10):
    for i in range(1,41):
        file = "new " + str(k) + "-" + str(i) + ".png"
        rup = cv2.imread(file,cv2.IMREAD_GRAYSCALE)
        rup = rup.flatten()
        lst = []
        lst.append(str(k))
        for j in rup:
            lst.append(j)
        with open(newfile, 'a', newline='') as f:
            writer = csv.writer(f)
            writer.writerow(lst)
```

3. EDA

```
[1] import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.sparse import csr_matrix
```

นำเข้าข้อมูลทั้งหมดจากไฟล์ pixel_num_new.csv และแสดงข้อมูล 5 แถวแรก และดูรายละเอียดของข้อมูล

```
train = pd.read_csv("/content/pixel_num_new.csv")
print(train.head())
print(train.info())
print("\n Shape of the dataset:", train.shape)
```

```
y pixel1 pixel2 pixel3 pixel4 pixel5 pixel6 pixel7 pixel8 pixel9 \
0 0 255 255 255 255 255 255 255 255 254
1 0 255 254 254 253 250 245 252 254 253
2 0 255 255 255 255 255 255 255 255 255
3 0 251 253 254 255 255 255 255 254 252
4 0 252 252 253 253 253 252 250 250 253
```

```
... pixel775 pixel776 pixel777 pixel778 pixel779 pixel780 pixel781 \
0 ... 189 230 255 253 253 254 255
1 ... 255 255 254 251 249 252 254
2 ... 254 254 254 254 255 255 255
3 ... 255 255 255 255 255 255 255
4 ... 210 237 255 254 252 251 255
```

```
pixel782 pixel783 pixel784
0 253 253 255
1 253 253 254
2 255 255 255
3 255 255 255
4 253 253 254
```

```
[5 rows x 785 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Columns: 785 entries, y to pixel784
dtypes: int64(785)
memory usage: 2.4 MB
None
```

Shape of the dataset: (400, 785)

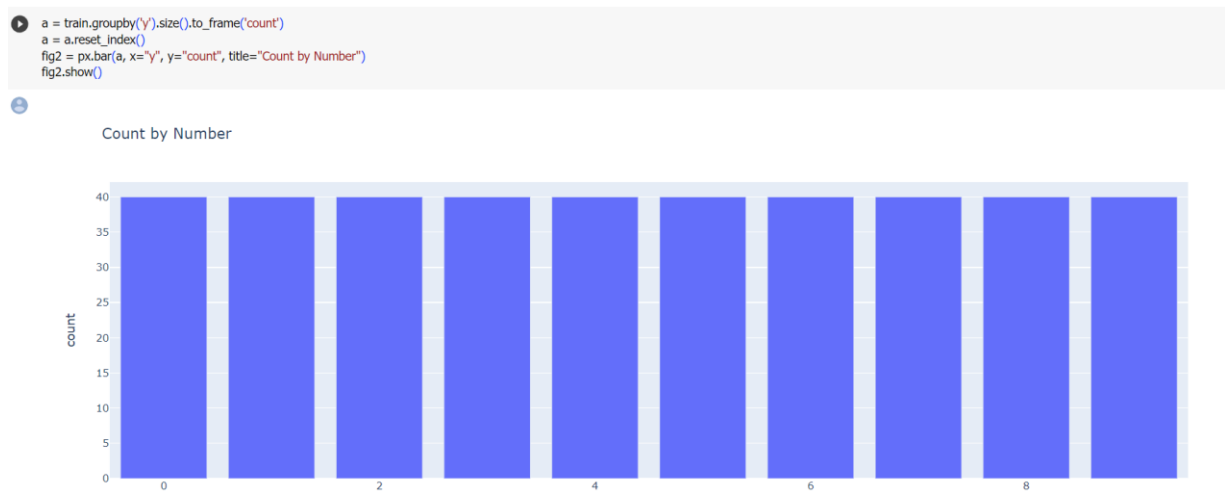
ดูข้อมูลทางสถิติของข้อมูลทั้งหมด

```
[ ] train.describe()
```

	y	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	...	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	4.500000	253.625000	250.7075	247.650000	245.027500	244.770000	247.130000	250.737500	252.940000	253.367500	...	234.425000	240.502500	244.842500	248.145000	248.325000	248.8
std	2.875878	13.682009	22.9640	29.830473	33.729229	36.270387	28.633333	17.595243	10.907053	9.695486	...	53.827296	45.077043	38.418128	30.783161	29.695352	26.8
min	0.000000	64.000000	53.0000	40.000000	48.000000	39.000000	51.000000	131.000000	147.000000	137.000000	...	0.000000	0.000000	0.000000	6.000000	11.000000	16.0
25%	2.000000	254.750000	254.0000	254.000000	254.000000	254.000000	254.000000	254.000000	254.000000	254.000000	...	253.000000	254.000000	254.000000	254.000000	254.000000	254.0
50%	4.500000	255.000000	255.0000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	...	255.000000	255.000000	255.000000	255.000000	255.000000	255.0
75%	7.000000	255.000000	255.0000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	...	255.000000	255.000000	255.000000	255.000000	255.000000	255.0
max	9.000000	255.000000	255.0000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	255.000000	...	255.000000	255.000000	255.000000	255.000000	255.000000	255.0

8 rows x 785 columns

สร้างกราฟแท่งที่นับจำนวนตัวเลขแต่ละตัวจากข้อมูลทั้งหมด



4. Training model

เมื่อได้ csv ที่รวมรูปทั้งหมดแล้วอยู่ในไฟล์ชื่อ pixel_num_new.csv

```
data_unseen.reset_index(inplace=True, drop=True)
print('Data for Modeling: ' + str(data.shape))
print('Unseen Data For Predictions: ' + str(data_unseen.shape))
```

5 rows x 785 columns

Data for Modeling: (320, 785)
Unseen Data For Predictions: (80, 785)

และจะเริ่มสร้างโมเดล โดยใช้ Pycaret และต้อง install และ import ก่อนที่จะเริ่มใช้

```
[ ] !pip install pycaret
```

```
[ ] from pycaret.classification import *
```

และสร้างตัวแปรใหม่ที่ชื่อ numeric_feature ที่ใช้เก็บชื่อ feature ทั้งหมด

```
[ ] numeric_features = data.columns.tolist()
numeric_features.remove('y')
numeric_features[:5]
```

['pixel1', 'pixel2', 'pixel3', 'pixel4', 'pixel5']

การเริ่มการใช้งาน Pycaret ต้องเริ่มจาก Setup ข้อมูลหรือเตรียมข้อมูลโดยกำหนดข้อมูลที่ใช้กับชื่อ column ที่ต้องการจะ predict เพื่อนำไปสร้างโมเดลในขั้นต่อไป และจะแบ่ง training set และ testing set เป็นอัตราส่วน 70/30 อัตโนมัติและจะบอกรายละเอียดการตั้งค่าออกมาดังนี้

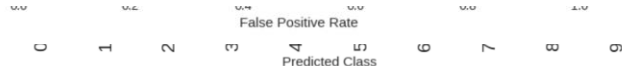
18	USI	ab71
----	-----	------

ต่อมาจะเป็นการสร้างโมเดลโดยจะใช้ compare_models()

เพื่อดูว่าโมเดลอันไหนที่เหมาะสมกับข้อมูลชุดนี้มากที่สุด โดยดูจาก Model Evaluation ของแต่ละโมเดล

dummy	Dummy Classifier							
	0.0893	0.5000	0.0893	0.0080	0.0147	0.0000	0.0000	0.1340
7	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
9	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	
Mean	0.9692	0.9983	0.9692	0.9735	0.9652	0.9656	0.9680	
Std	0.0397	0.0033	0.0397	0.0416	0.0460	0.0443	0.0411	

เมื่อได้โมเดลมาแล้วก็นำมาสร้าง ROC Curves และ Confusion matrix



ต่อมานำโมเดลมา predict กับข้อมูล training set และ testing set จะได้ค่าที่ predict ออกมาและคะแนนการ predict ของแต่ละค่า

255	255	255	255	255	255	...	255	255	255	255	255	255	255	9	9	0.59
255	255	255	255	255	255	...	255	255	255	255	255	255	255	9	9	0.57

และจะสรุปโมเดลด้วย finalize_model() และจะได้ออกมาเป็น workflow หรือ pipeline ในการสร้างโมเดล

`randomforestclassifier`

และ save โมเดลด้วย save_model()

`'thainumber_rf.pkl')`