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

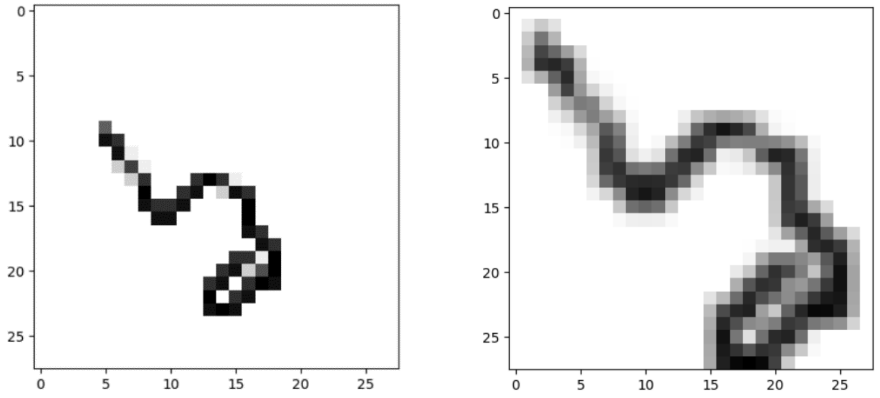
```
def Count_Up(img,img_check):
    count_up = 0
    check_up = False
    for i in range(28):
        for j in range(28):
            if img[i,j] < max(img_check):
                check_up = True
                break
        if check_up == True:
            break
        count_up += 1
    return count_up

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(27,0,-1):
            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],255,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 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 ในขั้นตอนต่อๆ ไป

```
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

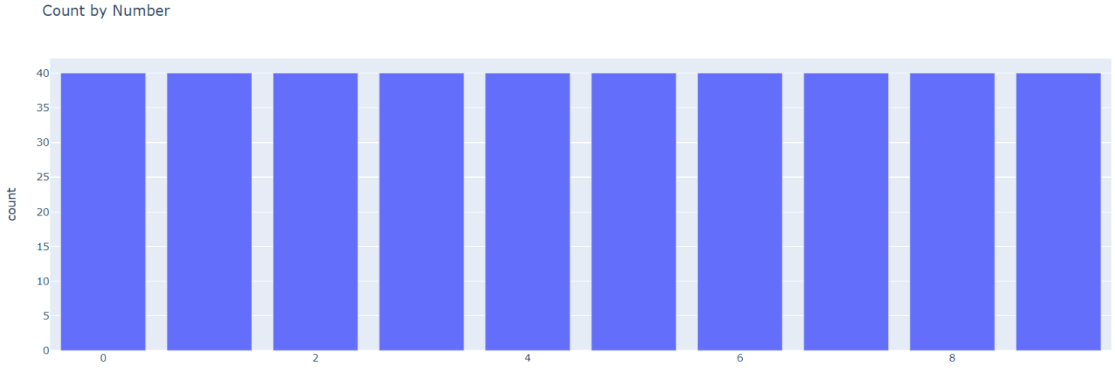
```
[ ] train.describe()
```

	y	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779	pix
count	400.000000	400.000000	400.0000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	...	400.000000	400.000000	400.000000	400.000000	400.000000	400.0
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 × 785 columns

สร้างกราฟแท่งที่นับจำนวนตัวเลขแต่ละตัวจากข้อมูลทั้งหมด

```
a = train.groupby('y').size().to_frame('count')
a = a.reset_index()
fig2 = px.bar(a, x="y", y="count", title="Count by Number")
fig2.show()
```



4. Training model

เมื่อได้ csv ที่รวมรูปทั้งหมดแล้วอยู่ในไฟล์ชื่อ pixel_num_new.csv

```
[ ] # โหลดข้อมูลให้อยู่ใน df_num และ show รายละเอียดข้อมูล
    df_num =pd.read_csv('pixel_num_new.csv')
    df_num.info()
    df_num.head()
```

```
<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
```

	y	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782
0	0	255	255	255	255	255	255	255	255	254	...	189	230	255	253	253	254	255	253
1	0	255	254	254	253	250	245	252	254	253	...	255	255	254	251	249	252	254	253
2	0	255	255	255	255	255	255	255	255	255	...	254	254	254	254	255	255	255	255
3	0	251	253	254	255	255	255	255	254	252	...	255	255	255	255	255	255	255	255
4	0	252	252	253	253	253	252	250	250	253	...	210	237	255	254	252	251	255	253

5 rows × 785 columns

จากนั้นจะทำการแบ่งข้อมูลออกเป็น 2 ส่วน คือ ข้อมูลสำหรับการสร้าง model ที่เก็บอยู่ในตัวแปรชื่อ data และ ข้อมูลสำหรับ prediction ที่เก็บอยู่ในตัวแปรชื่อ data_unseen โดยแบ่งเป็นอัตราส่วน 80/20

```
[ ] # split train and test dataset
    data = df_num.sample(frac=0.8, random_state=42)
    data_unseen = df_num.drop(data.index)
    data.reset_index(inplace=True, drop=True)
    data_unseen.reset_index(inplace=True, drop=True)
    print('Data for Modeling: ' + str(data.shape))
    print('Unseen Data For Predictions: ' + str(data_unseen.shape))

    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

```
[20] exp_name = setup(data = data, target = 'y')
```

	Description	Value
0	Session id	321
1	Target	y
2	Target type	Multiclass
3	Original data shape	(320, 785)
4	Transformed data shape	(320, 785)
5	Transformed train set shape	(224, 785)
6	Transformed test set shape	(96, 785)
7	Numeric features	784
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	ab71

ต่อมาจะเป็นการสร้างโมเดลโดยจะใช้ compare_models() เพื่อดูว่าโมเดลอันไหนที่เหมาะสมกับข้อมูลชุดนี้มากที่สุด โดยดูจาก Model Evaluation ของแต่ละโมเดล

```
[21] compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9692	0.9983	0.9692	0.9735	0.9652	0.9656	0.9680	0.4950
et	Extra Trees Classifier	0.9646	0.9982	0.9646	0.9709	0.9602	0.9606	0.9632	0.4590
lr	Logistic Regression	0.9514	0.9941	0.9514	0.9654	0.9499	0.9458	0.9478	1.2830
svm	SVM - Linear Kernel	0.9287	0.0000	0.9287	0.9363	0.9187	0.9205	0.9249	0.3940
lightgbm	Light Gradient Boosting Machine	0.9020	0.9903	0.9020	0.9102	0.8907	0.8906	0.8954	1.6490
lda	Linear Discriminant Analysis	0.8844	0.9823	0.8844	0.9163	0.8809	0.8711	0.8752	0.1730
xgboost	Extreme Gradient Boosting	0.8437	0.9820	0.8437	0.8457	0.8233	0.8257	0.8334	2.2140
ridge	Ridge Classifier	0.8350	0.0000	0.8350	0.8394	0.8199	0.8159	0.8214	0.1740
knn	K Neighbors Classifier	0.8079	0.9778	0.8079	0.8277	0.7911	0.7857	0.7934	0.5370
nb	Naive Bayes	0.8075	0.9070	0.8075	0.8409	0.7965	0.7854	0.7948	0.2410
gbc	Gradient Boosting Classifier	0.7899	0.9628	0.7899	0.8423	0.7868	0.7658	0.7748	12.1970
dt	Decision Tree Classifier	0.6872	0.8259	0.6872	0.6981	0.6661	0.6511	0.6607	0.3680
qda	Quadratic Discriminant Analysis	0.3178	0.6205	0.3178	0.2898	0.2808	0.2411	0.2498	0.1520
ada	Ada Boost Classifier	0.2816	0.8070	0.2816	0.2160	0.2074	0.2004	0.2359	0.6820
dummy	Dummy Classifier	0.0893	0.5000	0.0893	0.0080	0.0147	0.0000	0.0000	0.1340

```
RandomForestClassifier
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_samples_leaf=1,
                       min_samples_split=2, min_weight_fraction_leaf=0.0,
                       n_estimators=100, n_jobs=-1, oob_score=False,
                       random_state=321, verbose=0, warm_start=False)
```

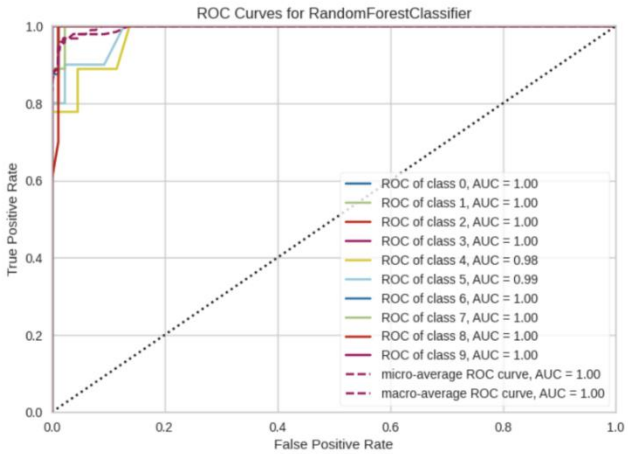
จากการเปรียบเทียบจะเห็นโมเดลที่สร้างจาก Random Forest ค่า Model Evaluation มีค่ามากที่สุดคือโมเดลที่ดีที่สุดกับข้อมูลชุดนี้ จึงสร้างโมเดลจากวิธี Random Forest

```
[29] model = create_model('rf')
```

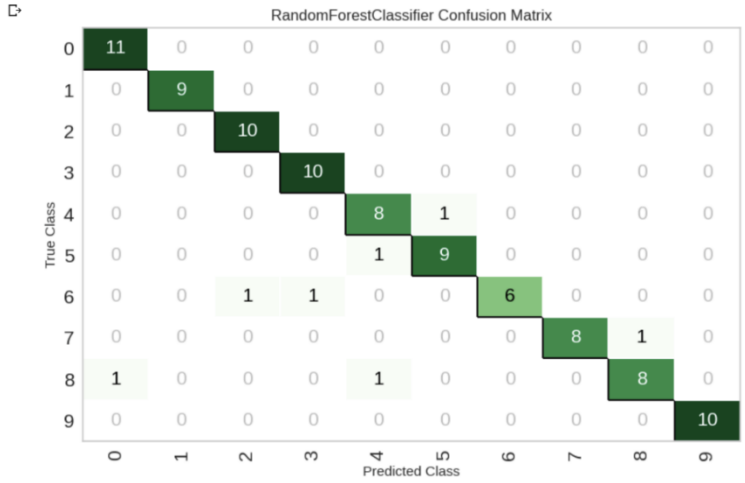
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9130	0.9959	0.9130	0.9565	0.9058	0.9034	0.9110
1	0.9130	0.9891	0.9130	0.9478	0.9123	0.9027	0.9085
2	0.9565	0.9978	0.9565	0.9674	0.9553	0.9515	0.9535
3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
6	0.9091	1.0000	0.9091	0.8636	0.8788	0.8986	0.9070
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

```
[30] plot_model(model)
```



```
plot_model(model, 'confusion_matrix')
```



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

```
[37] predictions = predict_model(model)
     predictions
```

AUC	Recall	Prec.	F1	Kappa	MCC
9958	0.9271	0.9308	0.9260	0.9189	0.9194

pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	...	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783	pixel784	y	prediction_label	prediction_score
255	255	255	255	255	255	...	255	255	255	255	255	255	255	9	9	0.71
255	255	255	255	255	255	...	255	255	255	255	255	255	255	2	2	0.71
255	255	255	255	255	255	...	217	182	195	244	255	255	255	8	8	0.72
255	255	255	255	255	255	...	255	255	255	255	255	255	255	0	0	0.88
255	255	255	255	255	255	...	255	255	255	255	255	255	255	5	5	0.77
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
255	255	255	255	255	255	...	255	255	255	255	255	255	255	1	1	0.69
255	255	255	255	255	255	...	255	255	255	255	255	255	255	7	7	0.48
255	255	255	255	255	255	...	255	255	255	255	255	255	255	7	7	0.51
227	197	229	249	255	255	...	255	255	255	255	255	255	255	6	6	0.64
255	255	255	255	255	255	...	255	255	255	255	255	255	255	5	5	0.56

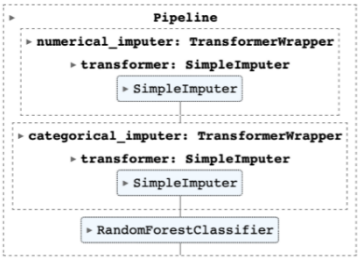
```
[39] pred_unseen = predict_model(model, data = data_unseen)
     pred_unseen
```

AUC	Recall	Prec.	F1	Kappa	MCC
.9955	0.9500	0.9536	0.9490	0.9439	0.9446

pixel5	pixel6	pixel7	pixel8	pixel9	pixel10	...	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783	pixel784	y	prediction_label	prediction_score
250	245	252	254	253	243	...	251	249	252	254	253	253	254	0	0	0.83
254	254	254	254	254	254	...	254	254	254	254	254	254	254	0	0	0.82
255	255	255	255	254	253	...	255	255	255	255	255	255	255	0	0	0.77
255	255	255	255	255	255	...	255	255	255	255	255	255	255	0	0	0.76
255	255	255	255	255	255	...	255	255	255	255	255	255	255	0	0	0.81
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
254	254	254	254	254	254	...	254	254	254	254	254	254	254	9	9	0.71
255	255	255	255	255	255	...	255	255	255	255	255	255	255	9	9	0.35
255	255	255	255	255	255	...	255	255	255	255	255	255	255	9	9	0.66
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

```
[34] final_model = finalize_model(model)
     final_model
```



และ save โมเดลด้วย save_model()

```
[35] save_model(model, 'thainumber_rf')

     Transformation Pipeline and Model Successfully Saved
     (Pipeline(memory=FastMemory(location=/tmp/joblib),
               steps=[('numerical_imputer',
                       TransformerWrapper(exclude=None,
                                          include=['pixel1', 'pixel2', 'pixel3',
                                                   'pixel4', 'pixel5', 'pixel6',
                                                   'pixel7', 'pixel8', 'pixel9',
                                                   'pixel10', 'pixel11', 'pixel12',
                                                   'pixel13', 'pixel14', 'pixel15',
                                                   'pixel16', 'pixel17', 'pixel18',
                                                   'pixel19', 'pixel20', 'pixel21',
                                                   'pixel22', 'pixel23', 'pixel24',
                                                   '...
                       RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
                                              class_weight=None, criterion='gini',
                                              max_depth=None, max_features='sqrt',
                                              max_leaf_nodes=None, max_samples=None,
                                              min_impurity_decrease=0.0,
                                              min_samples_leaf=1, min_samples_split=2,
                                              min_weight_fraction_leaf=0.0,
                                              n_estimators=100, n_jobs=-1,
                                              oob_score=False, random_state=321,
                                              verbose=0, warm_start=False))],
               verbose=False),
      'thainumber_rf.pkl')
```