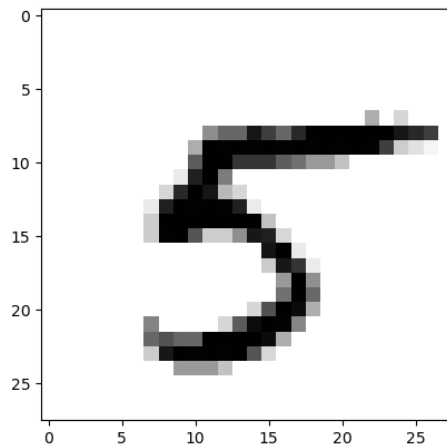


การสร้างโมเดลเพื่อในการคัดรูปตัวเลข0-9

การสร้างโมเดลเพื่อในการคัดรูปตัวเลข0-9ในไฟล์mnist-original.matดังตัวอย่าง



1.เปิดไฟล์mnist-original.matแยกข้อมูลเก็บในx,y

```
mnist_raw=loadmat("mnist-original.mat")
mnist={
    "data":mnist_raw["data"].T,
    "target":mnist_raw["label"][0]p
}
x,y=mnist["data"],mnist["target"]
```

2.เริ่มแรกทำการนำข้อมูลเข้ามาเพื่อแบ่งเป็นส่วนที่train75%ประมาณ1-60000และ25ที่เหลือคือ25%ดังนี้

```
x_train,x_test,y_train,y_test=x[:60000],x[60000:],y[:60000],y[60000:]
```

3.ทำการBinary Classificationเพื่อแยก5ดังนี้

```
y_train_5=(y_train==5)
y_test_5=(y_test==5)
```

4. Stochastic Gradient Descent model ในการแยกกลุ่มของข้อมูลและทำนายผลค่าที่เข้าไปtest

```
sgd_clf=SGDClassifier()
sgd_clf.fit(x_train,y_train_5)
```

5. Cross Validation

```
y_train_pred=cross_val_predict(sgd_clf,x_train,y_train_5,cv=3)
```

6.ทำการเทรนตรวจสอบผลสามารถ การประเมินประสิทธิภาพโดย Confusion Matrix

```
cm=confusion_matrix(y_train_5,y_train_pred)
```

7.แสดงผลผ่านฟังก์ชัน

```
def displayConfusionMatrix(cm,cmap=plt.cm.GnBu):
    classes=["Other Number","Number 5"]
    plt.imshow(cm,interpolation='nearest',cmap=cmap)
    plt.title("Confusion Matrix")
    plt.colorbar()
    tick_marks=np.arange(len(classes))
    plt.xticks(tick_marks,classes)
    plt.yticks(tick_marks,classes)
    thresh=cm.max()/2
    for i , j in itertools.product(range(cm.shape[0]),range(cm.shape[1])):
        plt.text(j,i,format(cm[i,j],'d'),
                horizontalalignment='center',
                color='white' if cm[i,j]>thresh else 'black')

    plt.tight_layout()
    plt.ylabel('Actually')
    plt.xlabel('Prediction')
    plt.show()
plt.figure()
```

8.ให้input function เป็นตัวแปรcmดังด้านล่าง

```
displayConfusionMatrix(cm)
```

9.แสดงผลดั่งภาพต่อไปนี้

Figure 1

