THE REPORT AND SOULUTIONs FOR MINIST LAB.

Lab 1' Requirment: The given pyhton code is use to predict number 5 in the MINIST data set, student have to transfer some code to predict number 7.

*Solution for Lab 1: **Binary Classification Setup**

Step 1: We chose number 7 instead number 5

In the original code:

y_train_5 = (y_train == 5)

y_test_5 = (y_test == 5)

I rewrite and implement it as follow:

y_train_7 = (y_train == 7)

y_test_7 = (y_test == 7)

Step 2: Updating the valuable "some_digit" to predict image of number 7

From: some_digit = X[0]

To: some_digit = X[15]

Step 3: Retrain the model with lable of number 7 to predict the result

sgd_clf = SGDClassifier(random_state=42)

sgd_clf.fit(X_train, y_train_7)

sgd_clf.predict([some_digit])

Step 4: ***We also update all the variable*** *y_train_5 or y_test_5* ***to become*** *y_train_7 or y_test_7*

Finally, the result is:

Now you can use it to detect images of the number 7

```
[41]  sgd_clf.predict([some_digit])
      ✓ 0.0s

...   array([ True])
```

****Performance Evaluation***

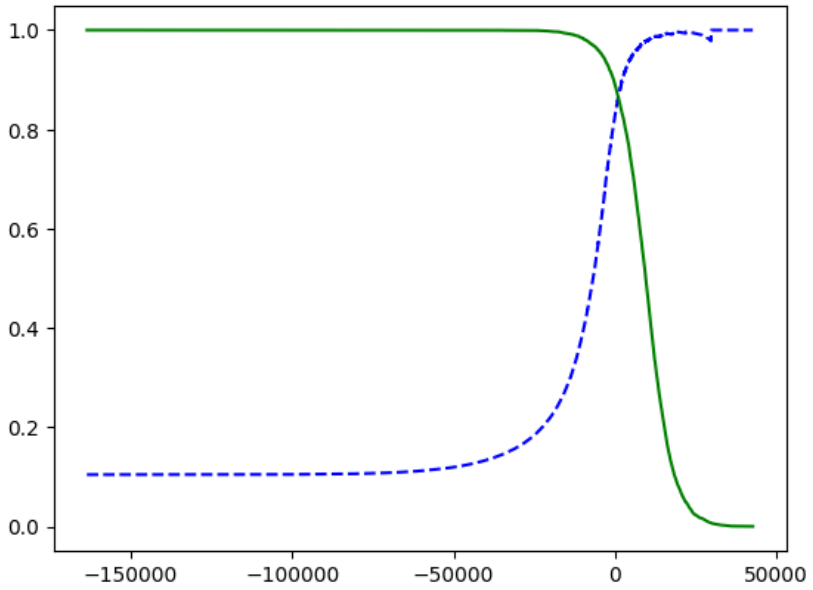
- **Used:**
  - Confusion matrix to observe true/false positives and negatives.

```
[44]  from sklearn.metrics import confusion_matrix
      confusion_matrix(y_train_7, y_train_pred)

...   array([[52581,  1154],
             [  688,  5577]])
```

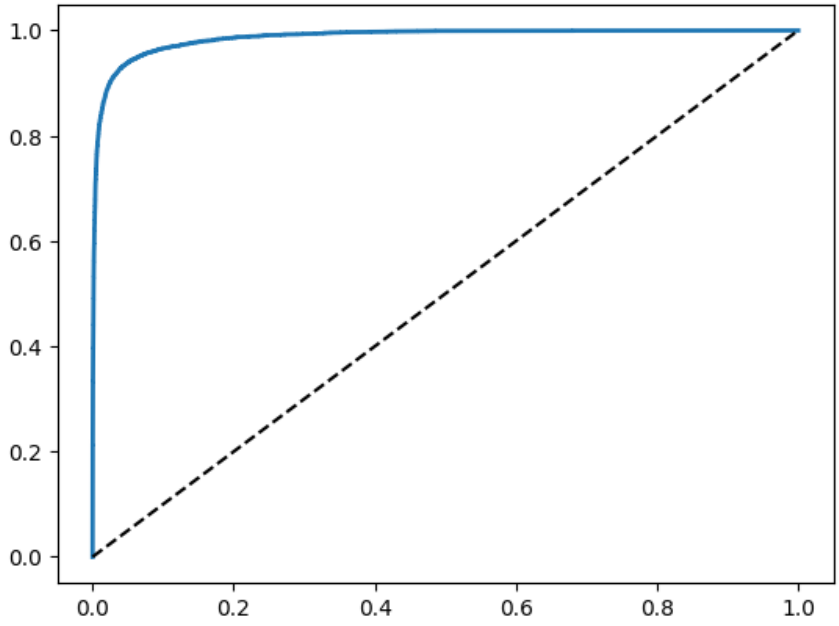
  - Precision, Recall, and F1-score to evaluate the classifier's performance.

```
[45]  from sklearn.metrics import precision_score, recall_score
      precision_score(y_train_7, y_train_pred)

...   0.8285544495617293

[46]  recall_score(y_train_7, y_train_pred)

...   0.8901835594573024

[47]  from sklearn.metrics import f1_score
      f1_score(y_train_7, y_train_pred)

...   0.858264081255771
```

- Precision-Recall curve



- ROC curve to analyze the precision-recall tradeoff and overall classification ability.



***Alternative Models

- Implemented a RandomForestClassifier as an alternative model.
- Compared prediction and probability outputs with SGDClassifier.

****Scaling and Improved Performance

- Scaled the training data using StandardScaler.
- Observed a significant accuracy boost after scaling.

To make it easy for understanding and visulization, please click on this link of my github, you will see full of my python code, which is written in Jupyter Notebook.

https://github.com/Phuc75nguyen/Machine-Learning/tree/main/MNIST_LAB