

MACHINE LEARNING - CO3117

HK251 - LO2

Phân loại ảnh chữ số viết tay trên tập dữ liệu MNIST
(Kaggle MNIST Digit Recognizer)

Đoàn Đức Bình - 2210302

Nguyễn Anh Kiệt - 2211758

MỤC LỤC

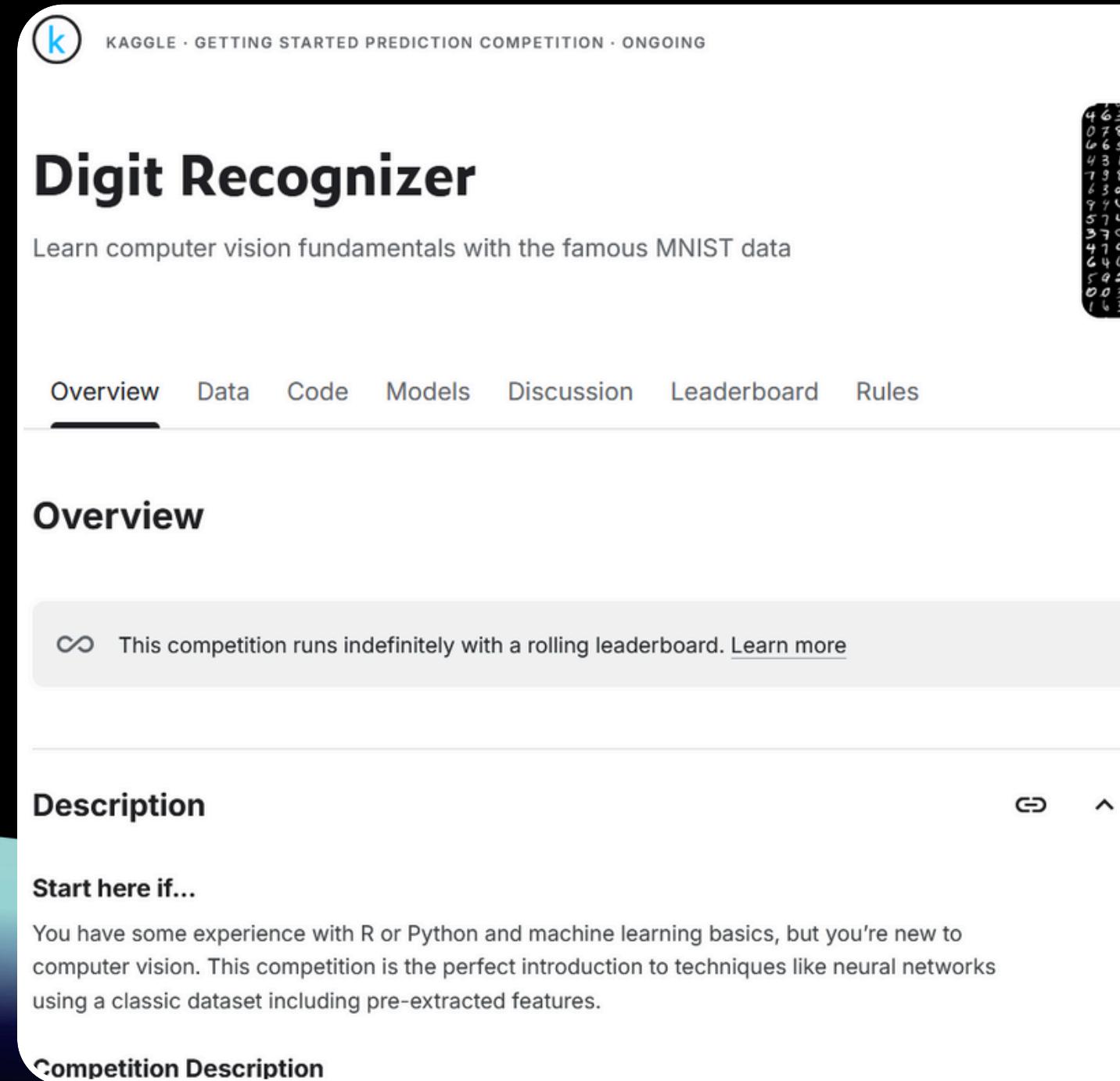
1 - GIỚI THIỆU

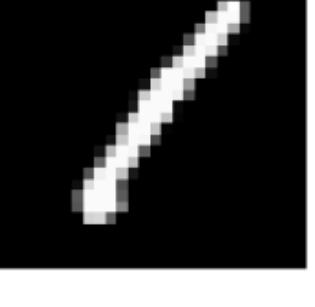
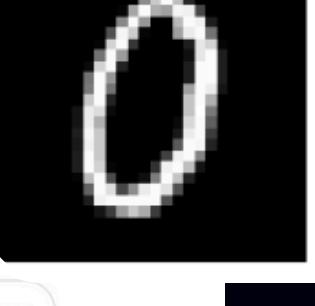
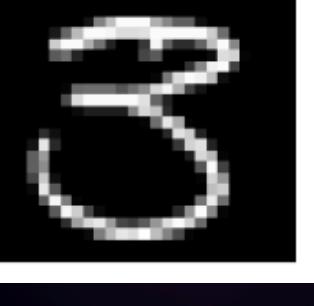
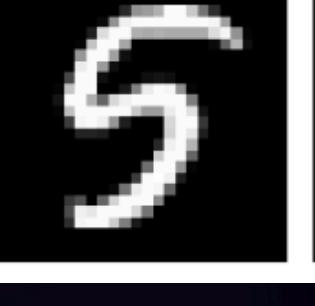
2 - PHÂN TÍCH DỮ LIỆU

3 - THIẾT KẾ THỰC NGHIỆM

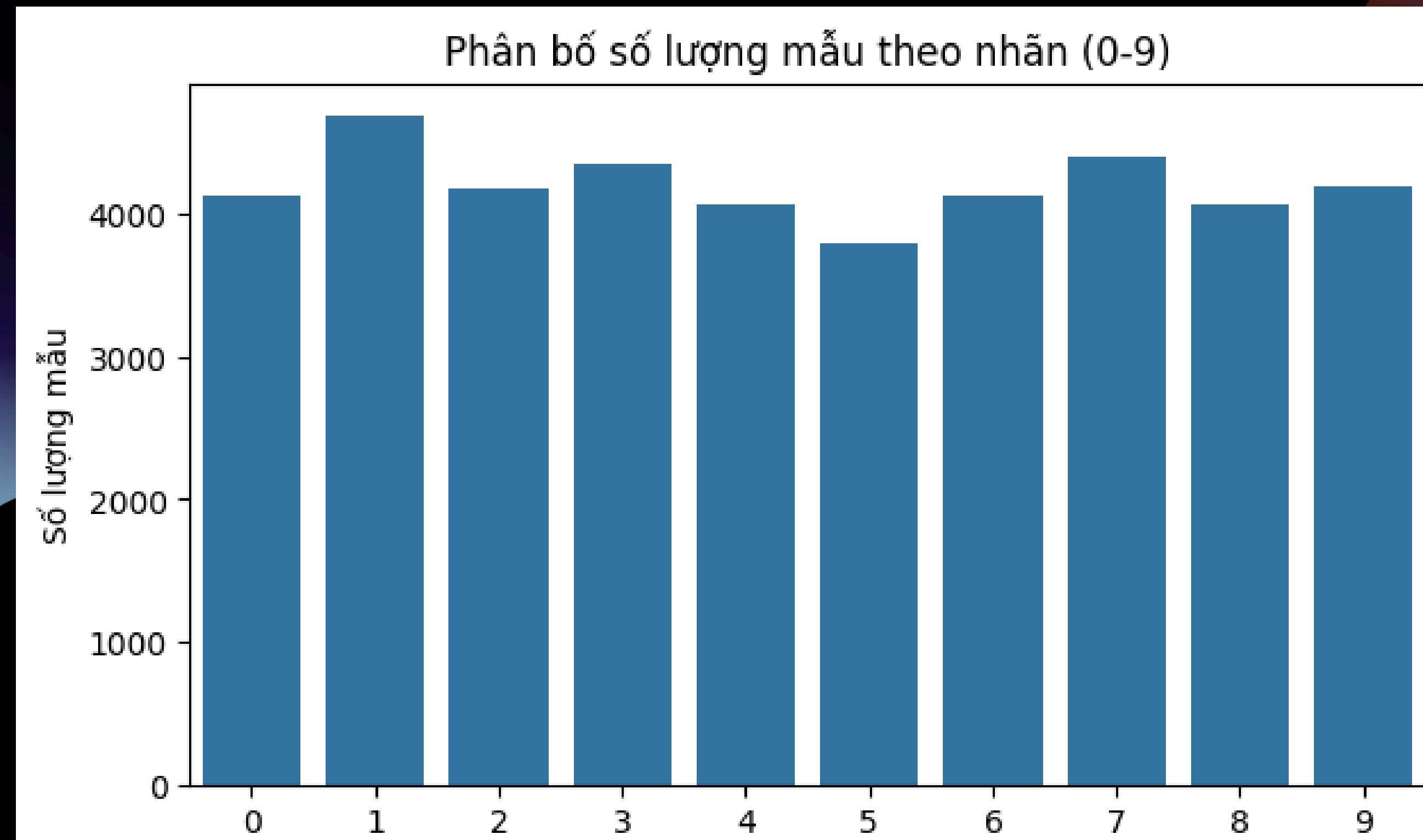
4 - KẾT QUẢ VÀ ĐÁNH GIÁ

1 - GIỚI THIỆU



test.csv - Excel																	Search				
File	Home	Insert	Page Layout	Formulas	Data	Review	View	Help													
YS5	XY	XZ	YA	YB	YC	YD	YE	YF	YG	YH	YI	YJ	YK	YL	YM	YN	YO	YP	YQ	YR	YS
1	I648	I649	I650	I651	I652	I653	I654	I655	I656	I657	I658	I659	I660	I661	I662	I663	I664	I665	I666	I667	pixel668
27979	0	0	0	71	152	253	253	253	253	171	143	29	0	0	0	0	0	0	0	0	0
27980	0	0	0	0	0	0	0	0	37	237	232	31	0	0	0	0	0	0	0	0	0
27981	0	0	41	173	252	253	130	20	0	0	0	0	0	0	0	0	0	0	0	0	0
27982	0	41	190	254	254	254	179	95	10	0	0	0	0	0	0	0	0	0	0	0	0
27983	0	0	0	0	0	48	253	221	6	0	0	0	0	0	0	0	0	0	0	0	0
Submit Prediction		...	0	0	189	253	154	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	22	87	170	183	160	145	66	22	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	26	239	224	0	0	0	0	0	0	0	0	0	0
		168	250	252	250	250	250	250	252	194	83	27	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		253	248	93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	37	246	254	253	183	16	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		0	0	0	0	22	128	101	0	0	0	0	0	0	0	0	0	0	0	0	0
Label: 1					Label: 0					Label: 1					Label: 4					Label: 0	
																					
Label: 0					Label: 7					Label: 3					Label: 5					Label: 3	
																					

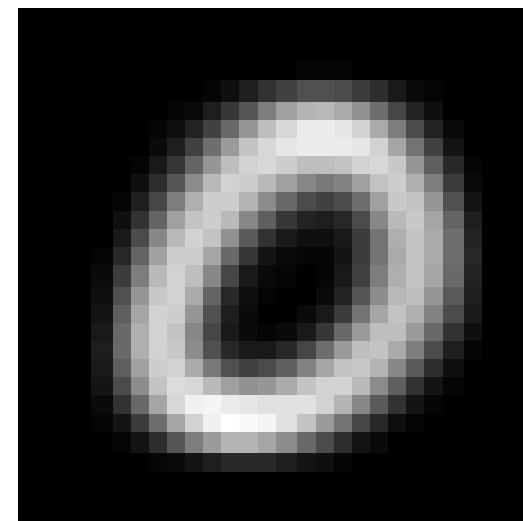
2 - PHÂN TÍCH DỮ LIỆU



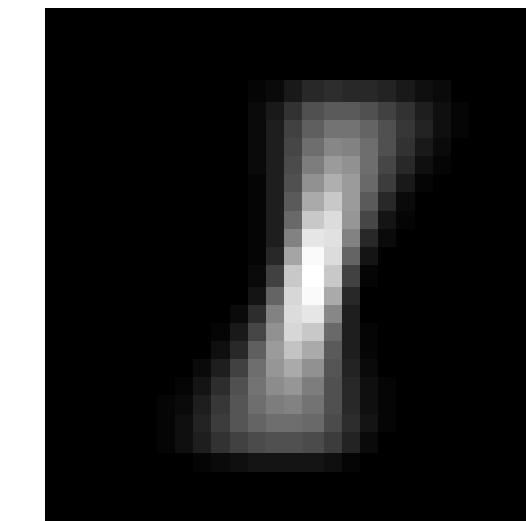
2 - PHÂN TÍCH DỮ LIỆU

Trung bình ảnh mỗi lớp

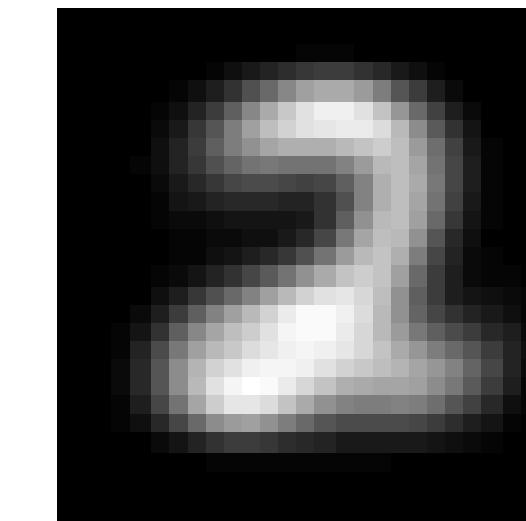
Mean: 0



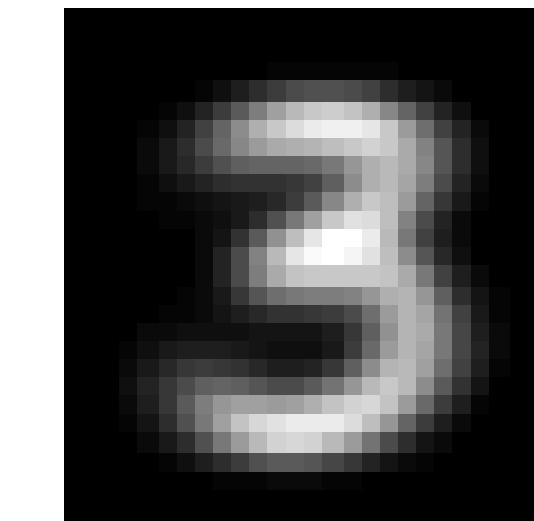
Mean: 1



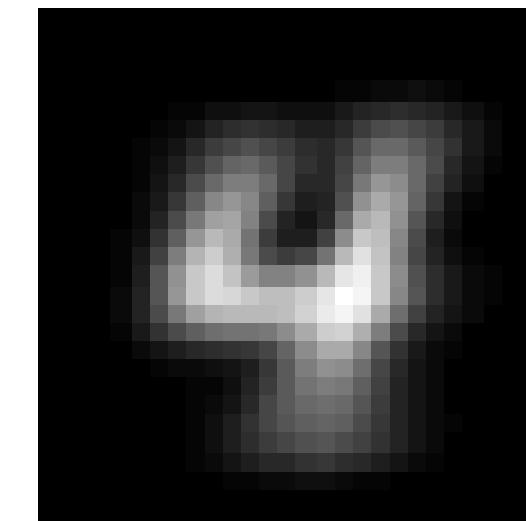
Mean: 2



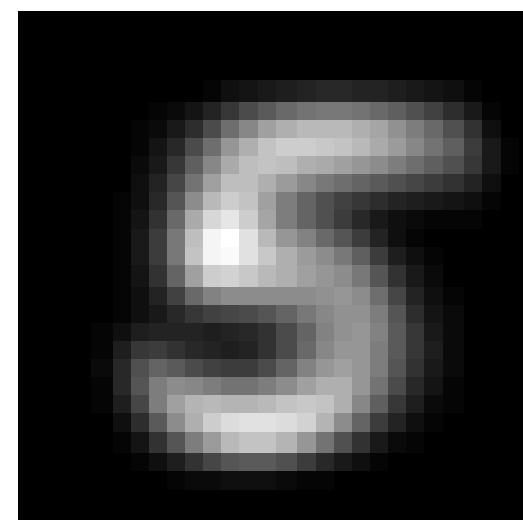
Mean: 3



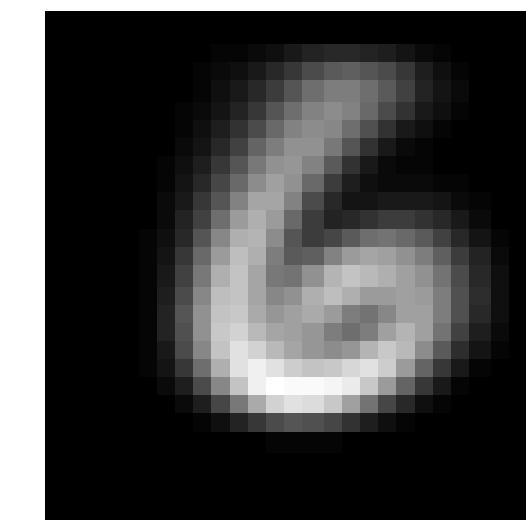
Mean: 4



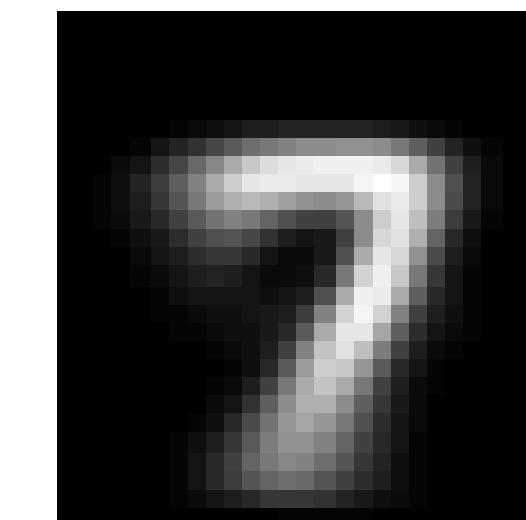
Mean: 5



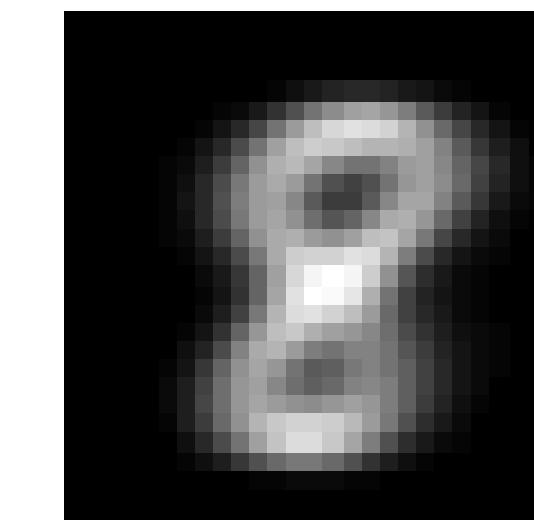
Mean: 6



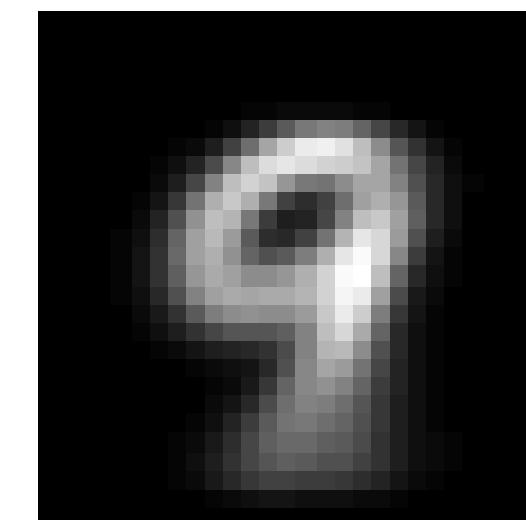
Mean: 7



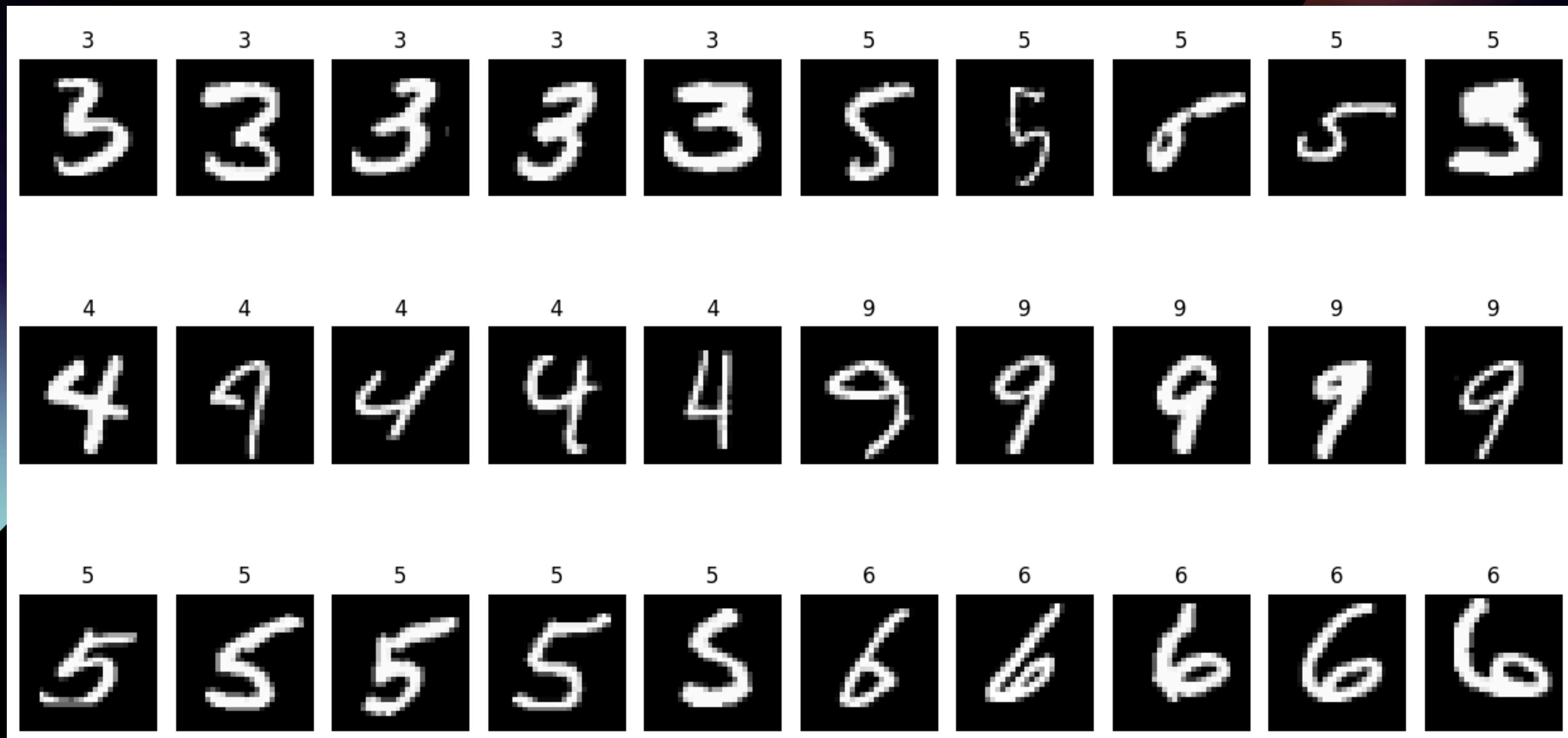
Mean: 8



Mean: 9



2 - PHÂN TÍCH DỮ LIỆU



Nhóm lựa chọn triển khai và so sánh hai mô hình học máy:

- **Logistic Regression** (sklearn)
- **Neural Network** (PyTorch).

Các metric cần đánh giá:

- Accuracy, Precision, Recall, F1-score.
- Confusion Matrix: phân tích chi tiết lỗi nhầm giữa các lớp (các cặp chữ số dễ nhầm lẫn)
- Thời gian: đo thời gian huấn luyện và dự đoán.
- Tài nguyên mô hình sử dụng: dung lượng lưu trữ.

3 - THIẾT KẾ THỰC NGHIỆM

3.1 - LOGISTIC REGRESSION

Logistic Regression là một mô hình tuyến tính được sử dụng phổ biến trong bài toán phân loại.

Với bài toán nhận diện chữ viết tay, mô hình được thiết lập ở chế độ đa lớp (multiclass).

```
# Khởi tạo mô hình Logistic Regression
logreg = LogisticRegression(
    solver='lbfgs',
    max_iter=1000,
    n_jobs=-1
)
✓ 0.0s
```

solver= 'lbfgs' : thuật toán tối ưu phù hợp cho bài toán đa lớp.

max_iter= 1000 : quy định số vòng lặp tối đa khi solver chạy để tối ưu hóa hàm mất mát (loss function).

n_jobs= -1 : cho phép chạy song song trên các CPU cores của máy.

3.1 - LOGISTIC REGRESSION

```
# Đo thời gian huấn luyện
```

```
time0 = time.time()
```

```
logreg.fit(X_train, y_train)
```

```
t_train = time.time() - time0
```

```
print('===== Thời gian huấn luyện =====')
```

```
print(f'Thời gian huấn luyện: {t_train:.2f}s')
```

```
✓ 33.9s
```

```
# Đo thời gian dự đoán
```

```
time1 = time.time()
```

```
y_pred = logreg.predict(X_test)
```

```
t_pred = time.time() - time1
```

```
print('===== Thời gian dự đoán =====')
```

```
print(f'Thời gian dự đoán trung bình: {t_pred:.2f}s')
```

```
print(f'Thời gian dự đoán trung bình: {t_pred:.2f}s')
```

```
✓ 0.0s
```

```
# Độ chính xác mô hình
```

```
acc = accuracy_score(y_test, y_pred)
```

```
prec = precision_score(y_test, y_pred, average='macro')
```

```
recall = recall_score(y_test, y_pred, average='macro')
```

```
f1 = f1_score(y_test, y_pred, average='macro')
```

```
print('===== Độ chính xác mô hình =====')
```

```
print(f'Accuracy : {acc:.4f}')
```

```
print(f'Precision(macro): {prec:.4f}')
```

```
print(f'Recall(macro) : {recall:.4f}')
```

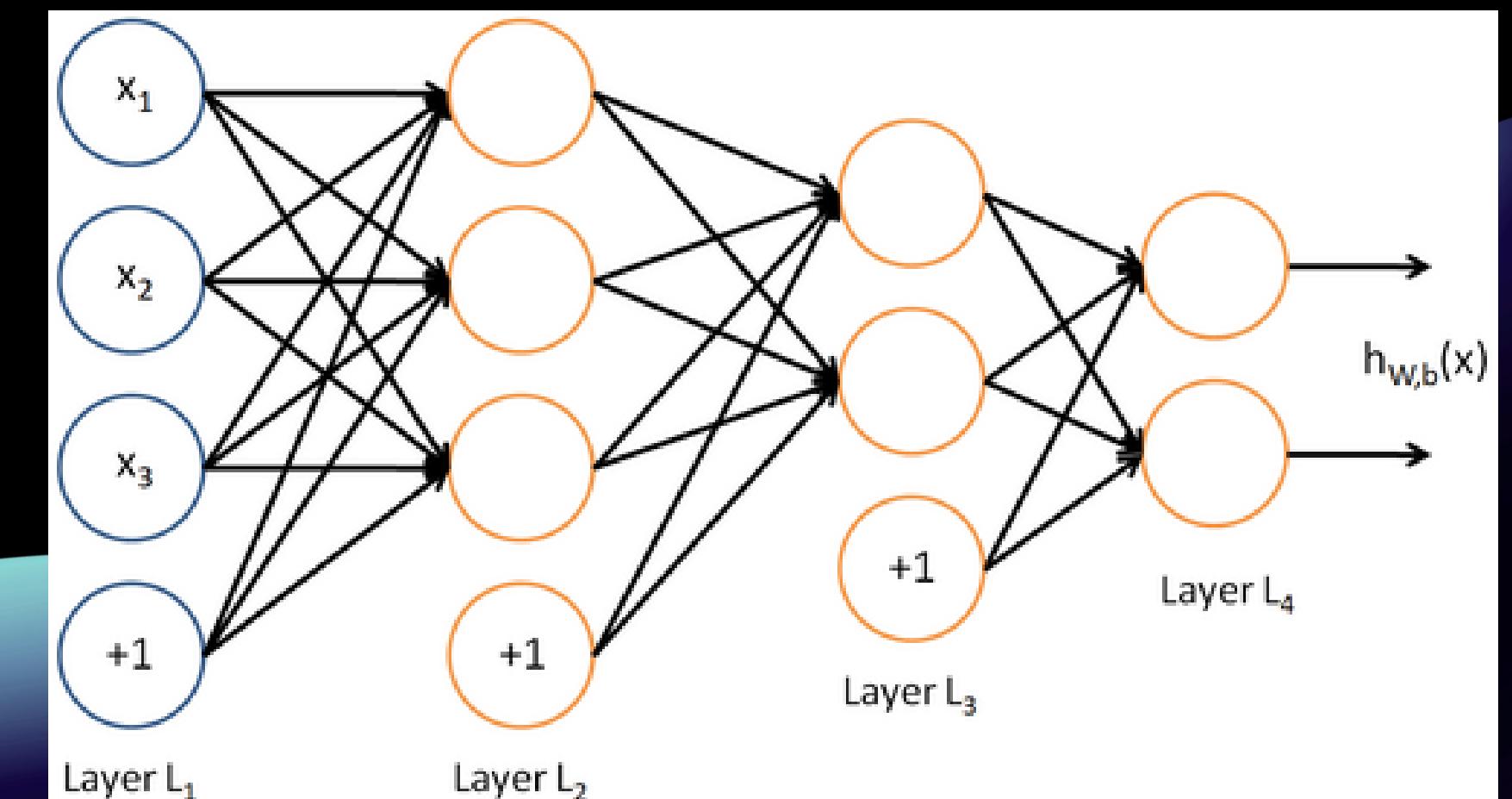
```
print(f'F1-score(macro) : {f1:.4f}')
```

```
✓ 0.0s
```

3.2 - NEURAL NETWORK

Mạng neural nhiều lớp (MLP) là một mô hình học sâu cơ bản, có khả năng học các quan hệ phi tuyến giữa đầu vào và đầu ra.

Trong bài toán nhận diện chữ số viết tay, nhóm xây dựng một MLP với các tầng ẩn và dropout nhằm tăng khả năng biểu diễn và giảm overfitting.



3.2 - NEURAL NETWORK

Mạng neural nhiều lớp (MLP) là một mô hình học sâu cơ bản, có khả năng học các quan hệ phi tuyến giữa đầu vào và đầu ra.

Trong bài toán nhận diện chữ số viết tay, nhóm xây dựng một MLP với các tầng ẩn và dropout nhằm tăng khả năng biểu diễn và giảm overfitting.

```
MLP(  
    (fc_784_to_512): Linear(in_features=784, out_features=512, bias=True)  
    (fc_512_to_256): Linear(in_features=512, out_features=256, bias=True)  
    (fc_256_to_128): Linear(in_features=256, out_features=128, bias=True)  
    (fc_128_to_10): Linear(in_features=128, out_features=10, bias=True)  
    (dropout30): Dropout(p=0.3, inplace=False)  
    (dropout20): Dropout(p=0.2, inplace=False)  
    (dropout10): Dropout(p=0.1, inplace=False)  
    (skip_512_to_128): Linear(in_features=512, out_features=128, bias=True)  
    (reluActivation): ReLU()  
)
```

```
def forward(self, x):  
    x1 = x  
    x1 = self.fc_784_to_512(x1)  
    x1 = self.reluActivation(x1)  
    x1 = self.dropout30(x1)  
  
    x2 = x1  
    x2 = self.fc_512_to_256(x2)  
    x2 = self.reluActivation(x2)  
    x2 = self.dropout20(x2)  
  
    x3 = x2  
    x3 = self.fc_256_to_128(x3)  
    x3 = self.reluActivation(x3)  
    x3 = self.dropout10(x3)  
  
    x4 = x3 + self.skip_512_to_128(x1)  
  
    out = self.fc_128_to_10(x4)  
    return out
```

4 - KẾT QUẢ VÀ ĐÁNH GIÁ

--- LOGISTIC REGRESSION ---

Kích thước Train dataset: (33600, 784)

Kích thước Test dataset : (8400, 784)

Thời gian huấn luyện: 33.975s

Thời gian dự đoán trên Test data : 0.0309s

Thời gian dự đoán trung bình / mẫu: 0.000004s

Accuracy : 0.9129

Precision(macro): 0.9119

Recall(macro) : 0.9115

F1-score(macro) : 0.9116

Confusion Matrix:

```
[[794  0  6  2  3  7 10  0  4  1]
 [ 0 914  1  4  0  2  0  1 14  1]
 [ 5 12 751 11  9  9  3 14 19  2]
 [ 4  7 26 760  0 36  4  9 15  9]
 [ 3  4  8  0 737  3  5  6  7 41]
 [ 8  8  9 32 14 649 10  2 21  6]
 [ 4  3  6  0  5 11 796  0  2  0]
 [ 1  4  4  5  8  1  0 820  6 31]
 [ 7 23  8 26  3 25  9  2 693 17]
 [ 5  4  3  8 23  5  0 29  7 754]]
```

Số tham số của mô hình: 7850

Kích thước của mô hình: 61.33 KB (~0.06MB)

--- NEURAL NETWORK ---

Kích thước Train dataset: 33600

Kích thước Test dataset: 8400

Thời gian huấn luyện: 36.2417

Thời gian dự đoán trên Test dataset: 0.44961

Thời gian dự đoán trung bình mỗi mẫu: 0.0000535251

Accuracy: 0.9752

Precision: 0.9752

Recall: 0.9750

F1 Score: 0.9751

Confusion Matrix:

```
[[816  0  1  2  0  0  4  1  3  0]
 [ 0 926  2  2  1  0  2  1  1  2]
 [ 5  2 811  1  2  3  4  2  2  3]
 [ 2  1  8 840  0 11  0  4  3  1]
 [ 2  1  1  1 797  1  3  1  0  7]
 [ 2  1  2 10  0 730  2  1  6  5]
 [ 0  1  1  0  1  3 821  0  0  0]
 [ 0  1  5  2  2  0  0  863  2  5]
 [ 0  5  1  8  1  2  5  2 787  2]
 [ 4  2  0  9  6  1  0 11  4 801]]
```

Số tham số của mô hình: 633098

Kích thước mô hình: 2.4194 MB

4 - KẾT QUẢ VÀ ĐÁNH GIÁ

Mô hình	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	Train time (s)	Predict time (s)	Params	Model size
Logistic Regression	0.9130	0.9121	0.9117	0.9118	24,964s	0,0269s	7850	66.33 KB
MLP (PyTorch)	0.9719	0.9718	0.9716	0.9717	27,173s	0,31452s	633098	2.4 MB

Hàng: nhãn thật, Cột: nhãn dự đoán

[[793 0 6 2 3 7 10 0 5 1]	[0 914 1 4 0 2 0 1 14 1]	[5 12 751 11 9 9 3 14 19 2]	[4 6 27 762 0 35 4 9 14 9]	[3 4 8 0 736 3 5 6 7 42]	[8 8 10 32 14 649 9 2 21 6]	[4 3 6 0 5 11 796 0 2 0]	[1 4 4 5 8 1 0 819 6 32]	[7 24 8 24 3 24 9 2 695 17]	[5 4 3 8 23 5 0 29 7 754]]
---------------------------	---------------------------	------------------------------	-----------------------------	---------------------------	------------------------------	---------------------------	---------------------------	------------------------------	-----------------------------

Hình 1: Confusion Matrix của Logistic Regression

Confusion Matrix:

[[814 0 1 1 1 1 1 5 1 2 1]	[0 929 3 1 0 1 1 1 1 1 0]	[5 2 811 1 2 1 1 9 3 0]	[1 1 13 831 0 9 0 3 8 4]	[0 2 2 0 781 2 5 3 2 17]	[3 1 0 9 0 733 4 1 5 3]	[3 0 0 0 3 1 817 0 3 0]	[1 4 2 1 3 1 0 864 0 4]	[3 4 3 5 1 1 7 1 786 2]	[3 1 1 5 12 2 0 12 4 798]]
----------------------------	----------------------------	--------------------------	---------------------------	---------------------------	--------------------------	--------------------------	--------------------------	--------------------------	-----------------------------

Hình 2: Confusion Matrix của MLP

4 - KẾT QUẢ VÀ ĐÁNH GIÁ

So sánh hai mô hình:

- **Logistic Regression** cho kết quả ổn định, tốc độ nhanh, phù hợp làm baseline.
- **Neural Network** đạt độ chính xác và F1-score cao hơn đáng kể nhờ khả năng học phi tuyến.

Phân tích nhầm lẫn (Confusion Matrix).

- **Logistic Regression** có xu hướng nhầm nhiều ở các cặp: 3-5, 4-9, 7-9 và chữ số 8.
- **Neural Network** giảm đáng kể các lỗi nhầm so với Logistic Regression.

THANK YOU!

Cảm ơn thầy và các bạn đã lắng nghe!