3.4.1 Your Task:

• Build from scratch: Develop the functions independently based on the mathematical formula tion.

```
Mean: 4.859999999999999
Standard Deviation: 0.1854723699099141
```

• ✓Loads MNIST CSV file into a Pandas DataFrame.

```
Dataset Preview:
   label  pixel_0  pixel_1  pixel_2  pixel_3  pixel_4  pixel_5  pixel_6  \
0      5        0        0        0        0        0        0        0
1      0        0        0        0        0        0        0        0
2      4        0        0        0        0        0        0        0
3      1        0        0        0        0        0        0        0
4      9        0        0        0        0        0        0        0
```

• ✓Extracts labels (y) and image pixel values (X).

```
y = df.iloc[:, 0]
X = df.iloc[:, 1:]
```

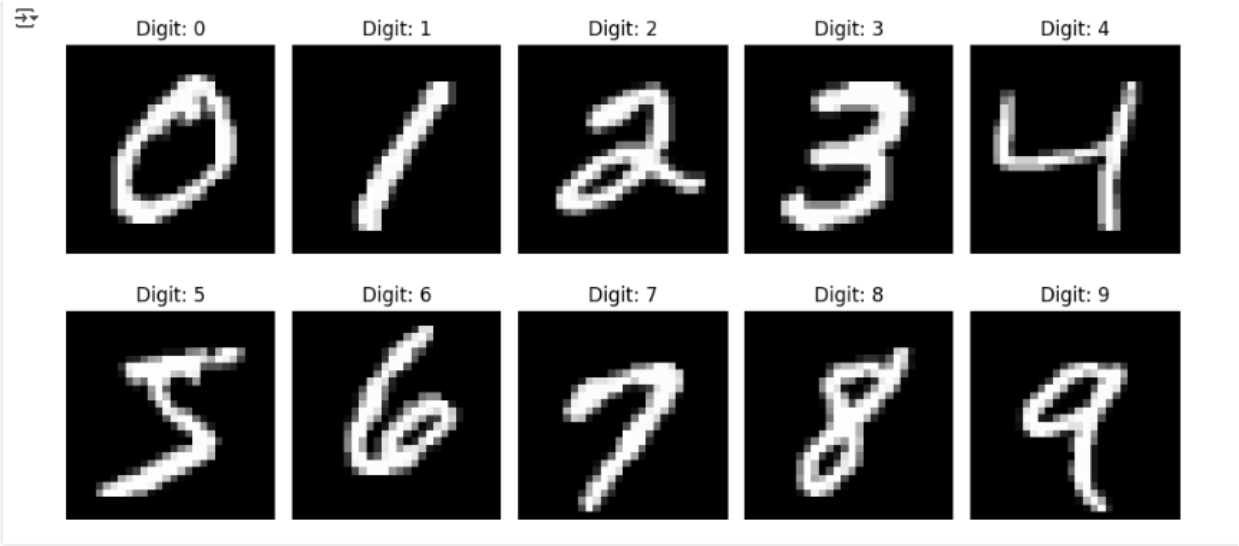
• ✓Normalizes pixel values (optional, but helps convergence)

```
23] # Normalize pixel values (scale to 0-1)
    X = X / 255.0
```

• ✓Splits data into training and test sets (default: 80% train, 20% test)

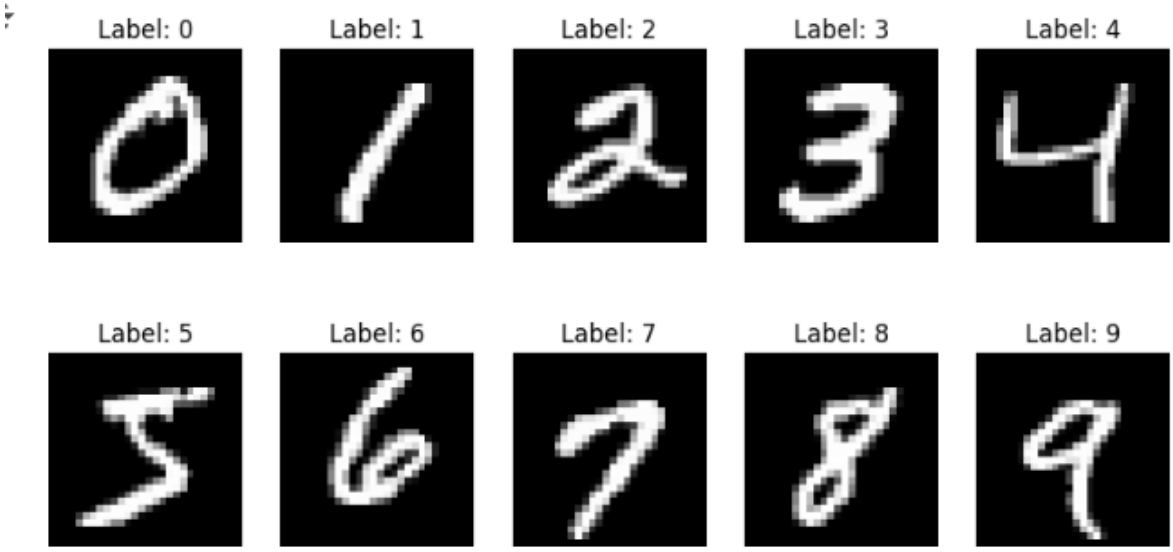
```
(X, y, test_size=0.2, random_state=42, stratify=y)
```

• ✓Plots one example image per digit (0-9) for visualization



---

1. Starter- Code- load and prepare mnist:

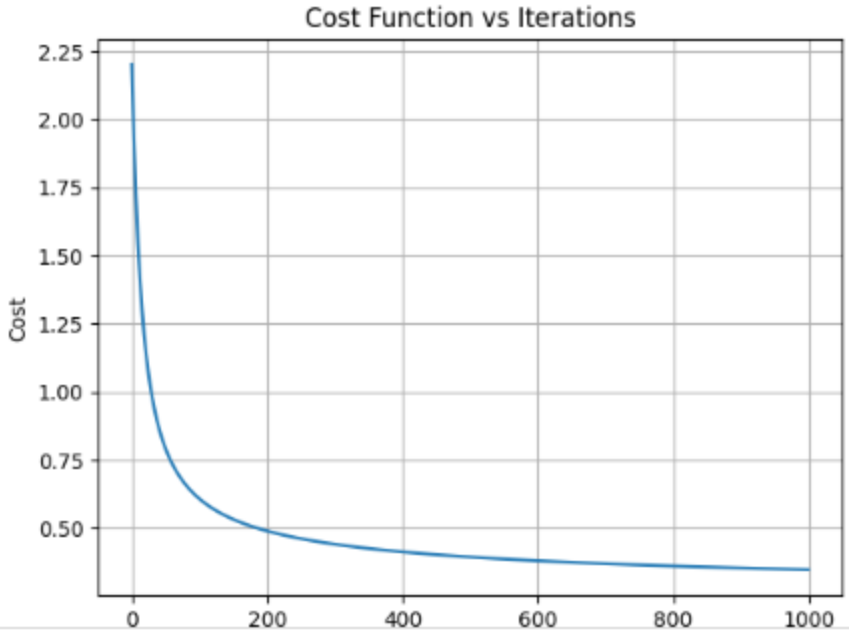


---

2. Starter-Code-plotsampleimages:

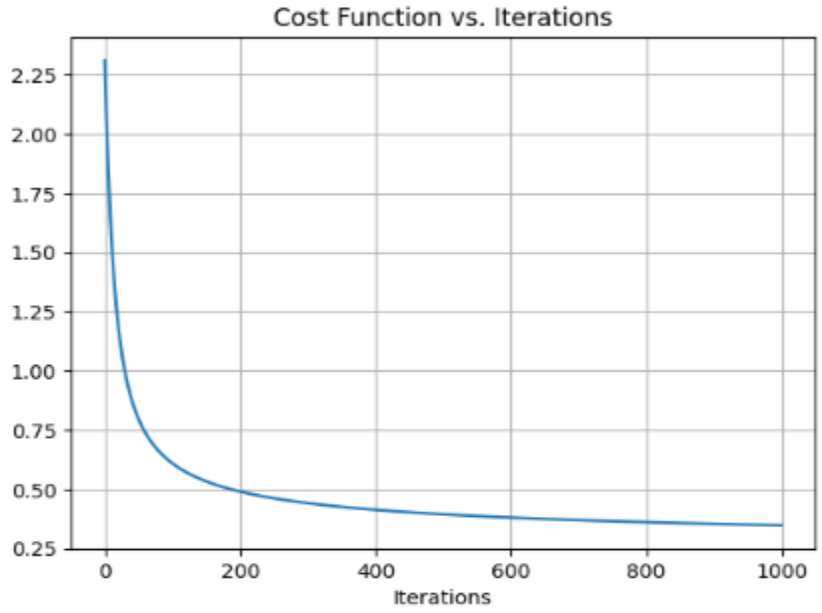
TrainingoftheSoftmaxRegressionModel.

```
Iteration 0: Cost = 2.2003
Iteration 100: Cost = 0.6070
Iteration 200: Cost = 0.4893
Iteration 300: Cost = 0.4408
Iteration 400: Cost = 0.4128
Iteration 500: Cost = 0.3939
Iteration 600: Cost = 0.3801
Iteration 700: Cost = 0.3694
Iteration 800: Cost = 0.3608
Iteration 900: Cost = 0.3537
```



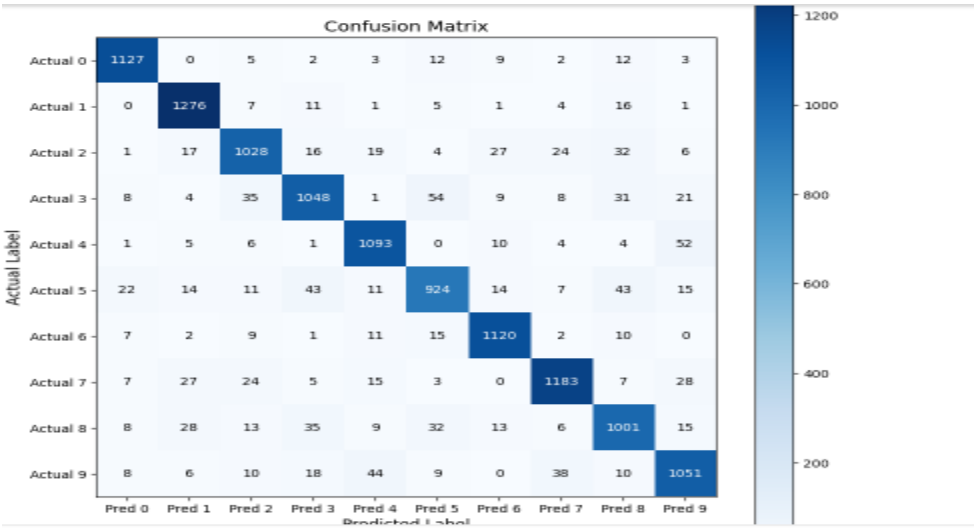
✓ 6m 15s

3.6 EvaluatingYourModelPerformance:



Iterations

```
Confusion Matrix:
[[1127    0    5    2    3   12    9    2   12    3]
 [   0 1276    7   11    1    5    1    4   16    1]
 [   1   17 1028   16   19    4   27   24   32    6]
 [   8    4   35 1048    1   54    9    8   31   21]
 [   1    5    6    1 1093    0   10    4    4   52]
 [  22   14   11   43   11  924   14    7   43   15]
 [   7    2    9    1   11   15 1120    2   10    0]
 [   7   27   24    5   15    3    0 1183    7   28]
 [   8   28   13   35    9   32   13    6 1001   15]
 [   8    6   10   18   44    9    0   38   10 1051]]
Precision: 0.90
Recall: 0.90
F1-Score: 0.90
```



Exercise-LinearSeparabilityandLogisticRegression.