

RA231104T010028.



NAME: Pooja STD: AI SEC: "A" ROLL NO. RAD018 SUB: DLT (LAB)

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	21/07/25	Exploring the deeplearning platform		<del>ATK 14/8/25</del>
2.	21/07/25	Implement a classifier using open source code		
3.	07/08/25	Study of classifier with respect to Statistical Parameter		<del>ATK 14/8/25</del>
4.	22/08/25	Build a Sample feed forward neural network to recognize handwritten character		<del>ATK 14/8/25</del>

21/07/20

## Q. Implement a classifier using Open Source Code

Aim: Implementing a classifier using a open source dataset

Objective: Applying machine learning

1. Load and Explore Iris dataset

2. Preprocess the data

3. Apply logistic regression for classification

4. Evaluate the model using accuracy & other classification metrics



## Pseudo code

1. Import required libraries

- Sklearn, Pandas, numpy,

matplotlib

2. Load iris dataset using

sklearn datasets

3. Explore iris dataset

- Features: Sepal length, Sepal width, Petal length, Petal width

Target : 3 classes

4. Split data:

Train & test split

5. Train logistic regression

model on training data

6. Predict labels on test

data

7. Evaluate performances:

- Accuracy

Pytorch:

organization: Facebook AI Research (FAIR) (2016)

Main features:

- Dynamic Computational graph
- Native Python syntax
- Strong GPU accelerating support

Popular use cases: Research and Academic Projects NLP models.

fast model prototyping

Graph type: dynamic

Result: successfully explored DLT Platform

### Observation:

#### 1. Dataset

- Iris dataset contains 150 samples, equally divided into 3 classes
- Each sample has 4 features

#### 2. Model Performance.

- Logistic regression achieved accuracy approximately.

~~effe~~ ✓

Result: Successfully implemented a classifier using open source dataset.

Output - 1 barrier approach

Accuracy: 1.0

Precision recall f1-score Support

setosa	1.00	1.00	1.00	1.00	50
versicolor	1.00	1.00	1.00	1.00	9
virginica	1.00	1.00	1.00	1.00	11
accuracy				1.00	30
macro avg	1.00	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	1.00	30.

The screenshot shows a Jupyter Notebook interface with multiple tabs open. The current tab is titled "Logistic regression w X" and contains Python code for a logistic regression model. The code includes importing metrics from sklearn, calculating a confusion matrix, and displaying its accuracy. The notebook also lists other files in the directory, such as datasets, EX1.ipynb, L2.ipynb, L3.ipynb, and Logistic reg... . The bottom status bar shows the mode as "Command", the kernel as "Python 3 (ipykernel)", and memory usage as 283.83 MB.

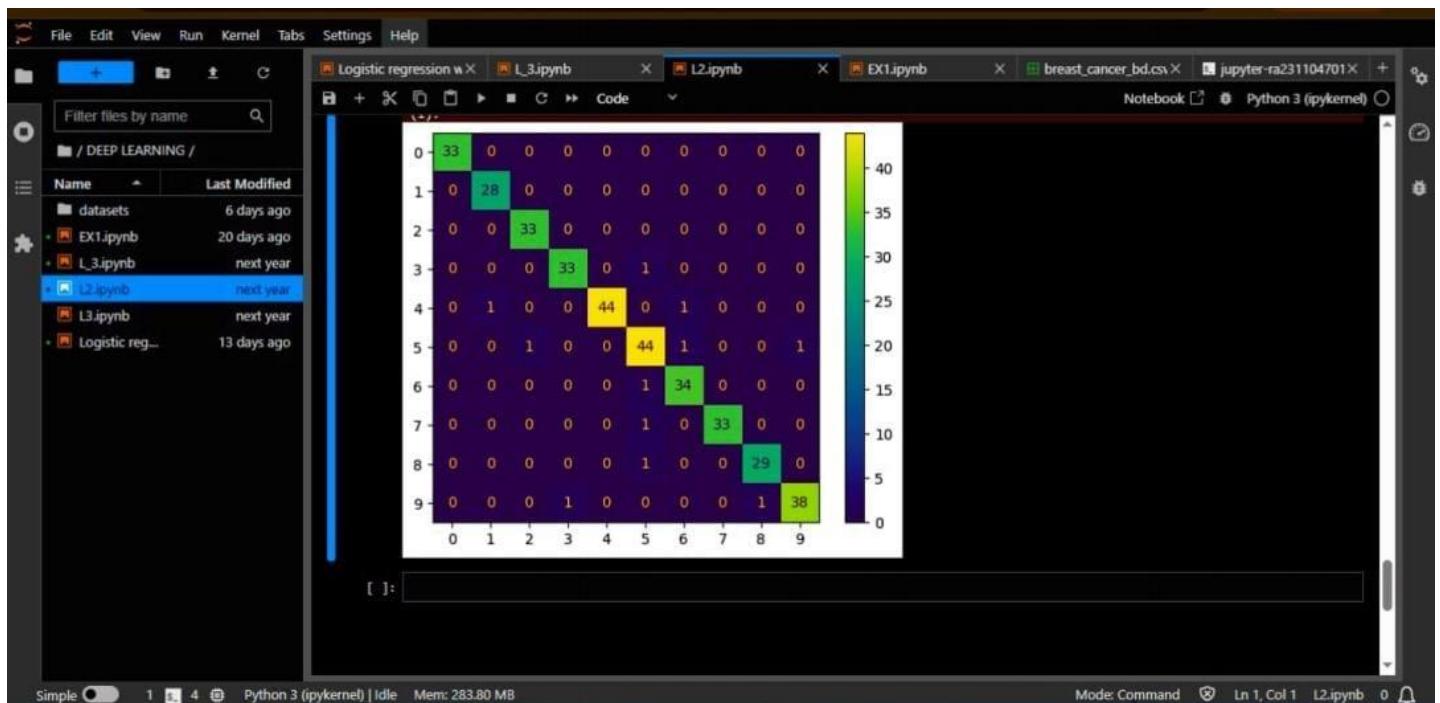
```
File Edit View Run Kernel Tabs Settings Help
Logistic regression w X L_3.ipynb L2.ipynb EX1.ipynb breast_cancer_bd.csv jupyter-ra231104701X +
Notebook Python 3 (ipykernel) ○
Filter files by name
/ DEEP LEARNING /
Name Last Modified
datasets 6 days ago
EX1.ipynb 20 days ago
L_3.ipynb next year
L2.ipynb next year
L3.ipynb next year
Logistic reg... 13 days ago
[25]: Logistic Regression
Accuracy: 0.9694444444444444

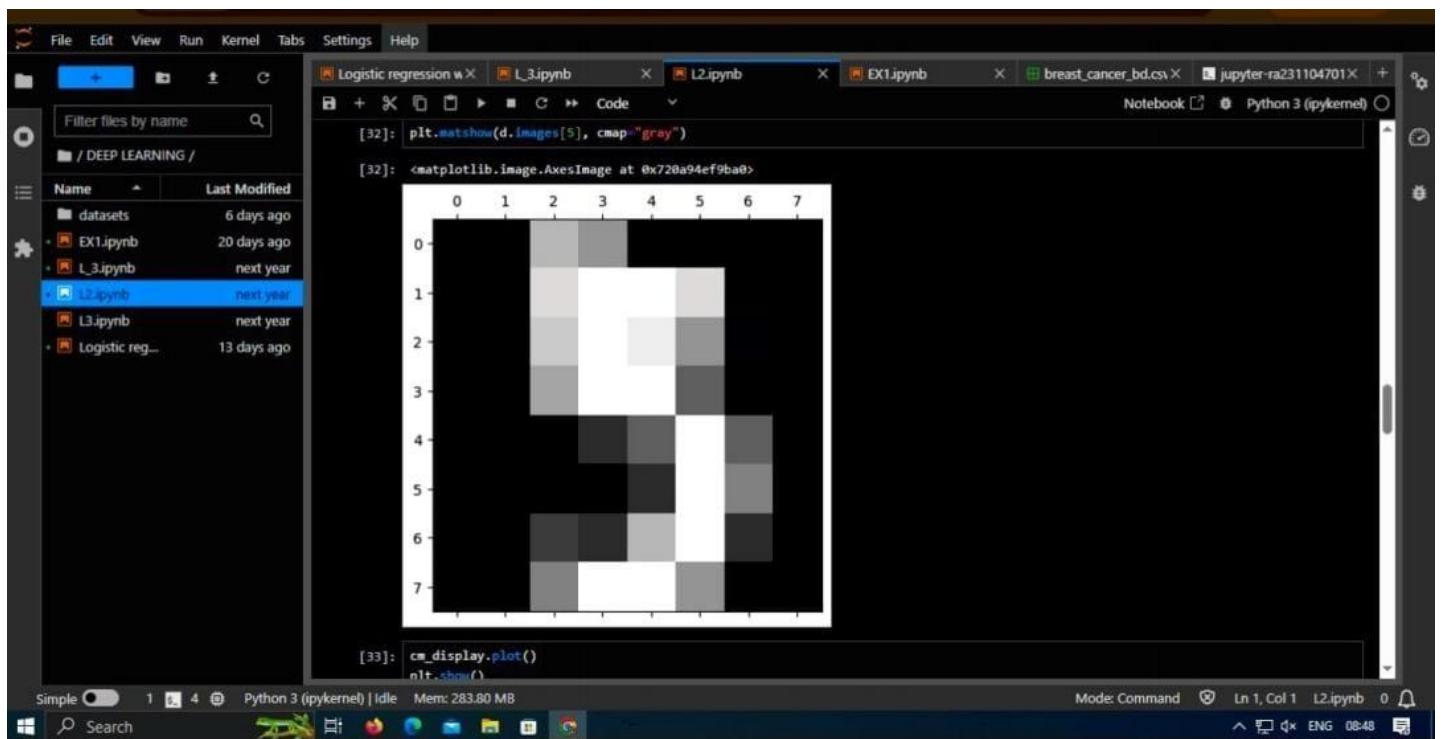
[26]: From sklearn import metrics
confusion_matrix=metrics.confusion_matrix(y_test,y_pred)
confusion_matrix

[26]: array([[29,  0,  0,  0,  2,  1,  0,  0,  0,  1],
   [ 0, 22,  1,  1,  0,  0,  1,  1,  1,  1],
   [ 0,  1, 27,  2,  0,  0,  0,  1,  2,  0],
   [ 0,  0,  1, 38,  0,  0,  0,  0,  3,  0],
   [ 0,  0,  0,  1, 41,  0,  0,  4,  0,  0],
   [ 0,  0,  0,  0,  1, 43,  1,  0,  1,  1],
   [ 0,  0,  0,  0,  2,  0, 32,  0,  1,  0],
   [ 0,  0,  0,  2,  2,  0,  0, 30,  0,  0],
   [ 0,  2,  0,  2,  1,  2,  0,  0, 20,  3],
   [ 0,  0,  0,  2,  1,  0,  0,  3,  0, 34]])
```

```
[27]: confusion_matrix=metrics.confusion_matrix(y_test,y_pred)
confusion_matrix

[27]: array([[29,  0,  0,  0,  2,  1,  0,  0,  0,  1],
   [ 0, 22,  1,  1,  0,  0,  1,  1,  1,  1],
   [ 0,  1, 27,  2,  0,  0,  0,  1,  2,  0],
   [ 0,  0,  1, 38,  0,  0,  0,  0,  3,  0],
   [ 0,  0,  0,  1, 41,  0,  0,  4,  0,  0],
   [ 0,  0,  0,  0,  1, 43,  1,  0,  1,  1],
   [ 0,  0,  0,  0,  2,  0, 32,  0,  1,  0],
   [ 0,  0,  0,  2,  2,  0,  0, 30,  0,  0],
   [ 0,  2,  0,  2,  1,  2,  0,  0, 20,  3],
   [ 0,  0,  0,  2,  1,  0,  0,  3,  0, 34]])
```





The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, View, Run, Kernel, Tabs, Settings, Help.
- Sidebar:** Shows a file tree under "/ DEEP LEARNING /". The current directory contains:
  - datasets (6 days ago)
  - EX1.ipynb (20 days ago)
  - L\_3.ipynb (next year)
  - L2.ipynb** (next year) - This is the active notebook.
  - L3.ipynb (next year)
  - Logistic reg... (13 days ago)
- Code Cells:**
  - [40]: knn = KNeighborsClassifier()
  - [41]: from sklearn.linear\_model import LogisticRegression
  - [42]: clf = LogisticRegression()
  - [20]: clf.fit(x\_train, y\_train)

Output: /home/jupyter-ra2311047010011/.local/lib/python3.10/site-packages/scikit-learn/\_linear\_model/\_logistic.py:473: ConvergenceWarning: lbfgs failed to converge after 100 iteration(s) (status=1):  
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT

Increase the number of iterations to improve the convergence (max\_iter=100).  
You might also want to scale the data as shown in:  
<https://scikit-learn.org/stable/modules/preprocessing.html>  
Please also refer to the documentation for alternative solver options:  
[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
  - [20]: LogisticRegression()

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.  
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
  - [21]: y\_pred = clf.predict(x\_test)  
y\_pred
  - [21]: array([6, 9, 3, 7, 2, 1, 5, 2, 5, 2, 1, 9, 4, 8, 4, 2, 3, 7, 8, 8, 4, 3,
- Kernel:** Python 3 (ipykernel)
- Help:** Notebook, Python 3 (ipykernel)