RA2311047010028.

INDEX

NAME: Pooja STD.: B.Tech AI SEC.: "A" ROLL NO. RA028 SUB.: DLT (LAB)

31/07/25 Q. Implement a classifier using Open source code

Aim: Implementing a classifier using a open source dataset

objective:

1. Load and Emplore 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 requred 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 trainig data
6. Predict labels on test data
7. Evaluate performances:
    - accuracy
```

Pytorch:-

organization: Facebook AI Research [fair] (2016)

Main feautures: Dynamic Computational graph
Native pythonic syntax
strong GPU accelerating Supp

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.

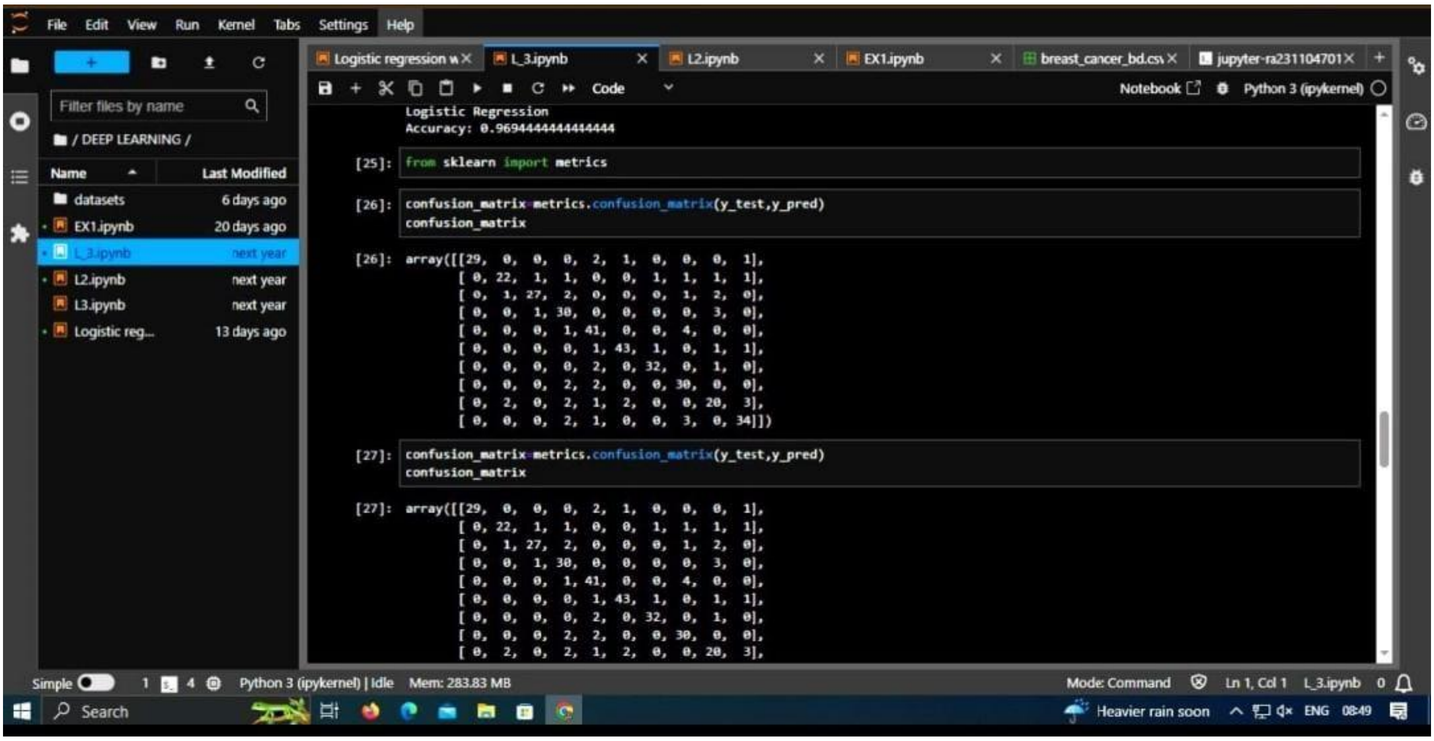
. Logistic regression achieved accuracy approximately.

Result: Successfully implemented a classifier using open Source dataset.

Output :-

Accuracy : 1.0

	Precision	recall	F1-score	Support
setosa	1.00	1.00	1.00	10
versicolor	1.00	1.00	1.00	9
Verginica	1.00	1.00	1.00	11
accuracy			1.00	30
marco avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30.



```
Logistic Regression
Accuracy: 0.9694444444444444
```

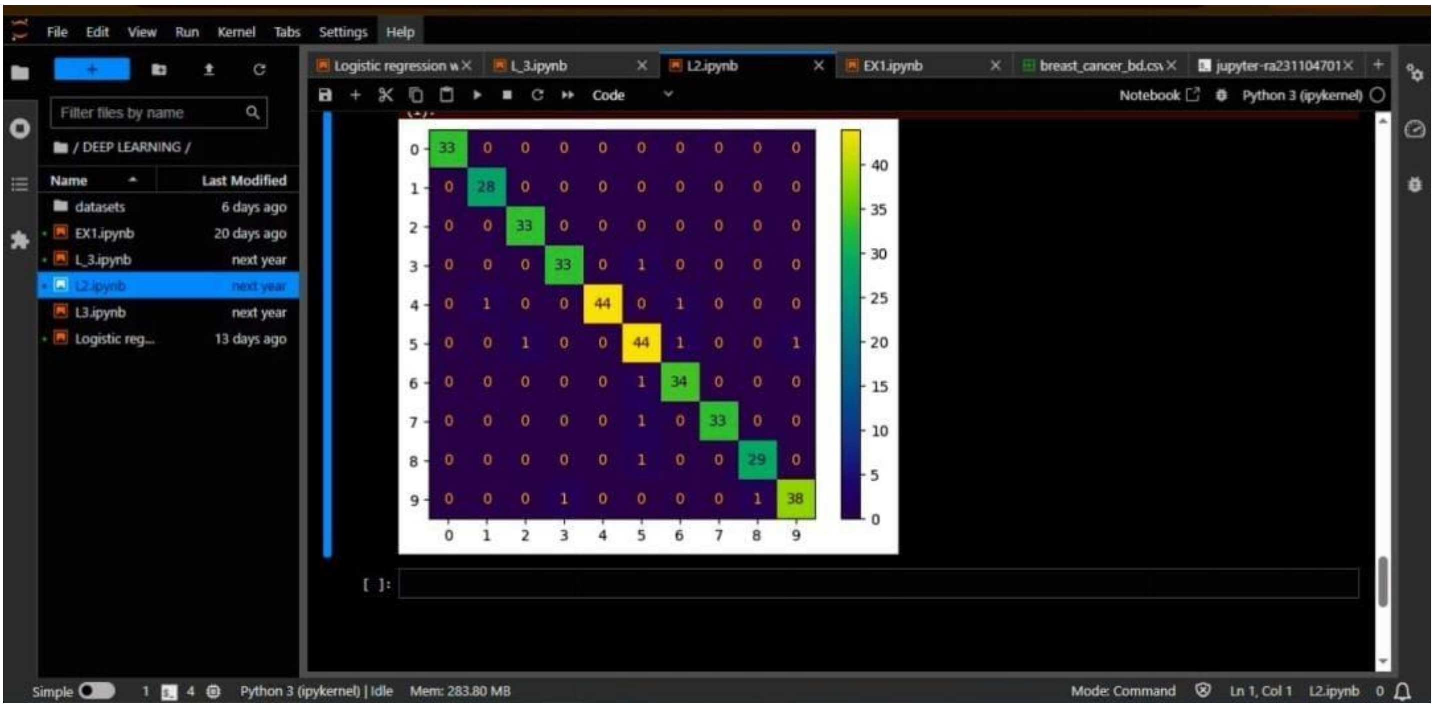
```
[25]: from sklearn import metrics
```

```
[26]: 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, 30,  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]])
```

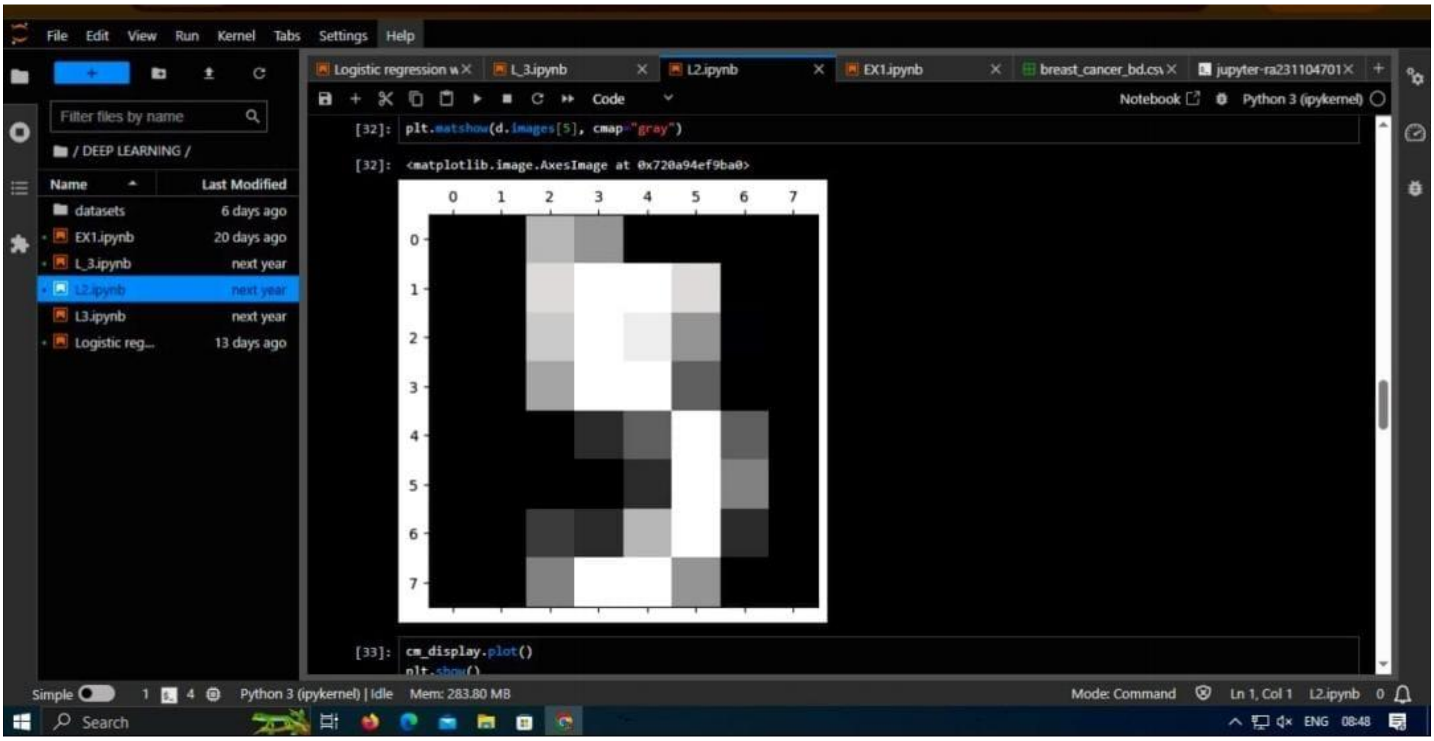
```
[27]: confusion_matrix=metrics.confusion_matrix(y_test,y_pred)
      confusion_matrix
```

```
[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, 30,  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],
```



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

	0	1	2	3	4	5	6	7	8	9
0	33	0	0	0	0	0	0	0	0	0
1	0	28	0	0	0	0	0	0	0	0
2	0	0	33	0	0	0	0	0	0	0
3	0	0	0	33	0	1	0	0	0	0
4	0	1	0	0	44	0	1	0	0	0
5	0	0	1	0	0	44	1	0	0	1
6	0	0	0	0	0	1	34	0	0	0
7	0	0	0	0	0	1	0	33	0	0
8	0	0	0	0	0	1	0	0	29	0
9	0	0	0	1	0	0	0	0	1	38

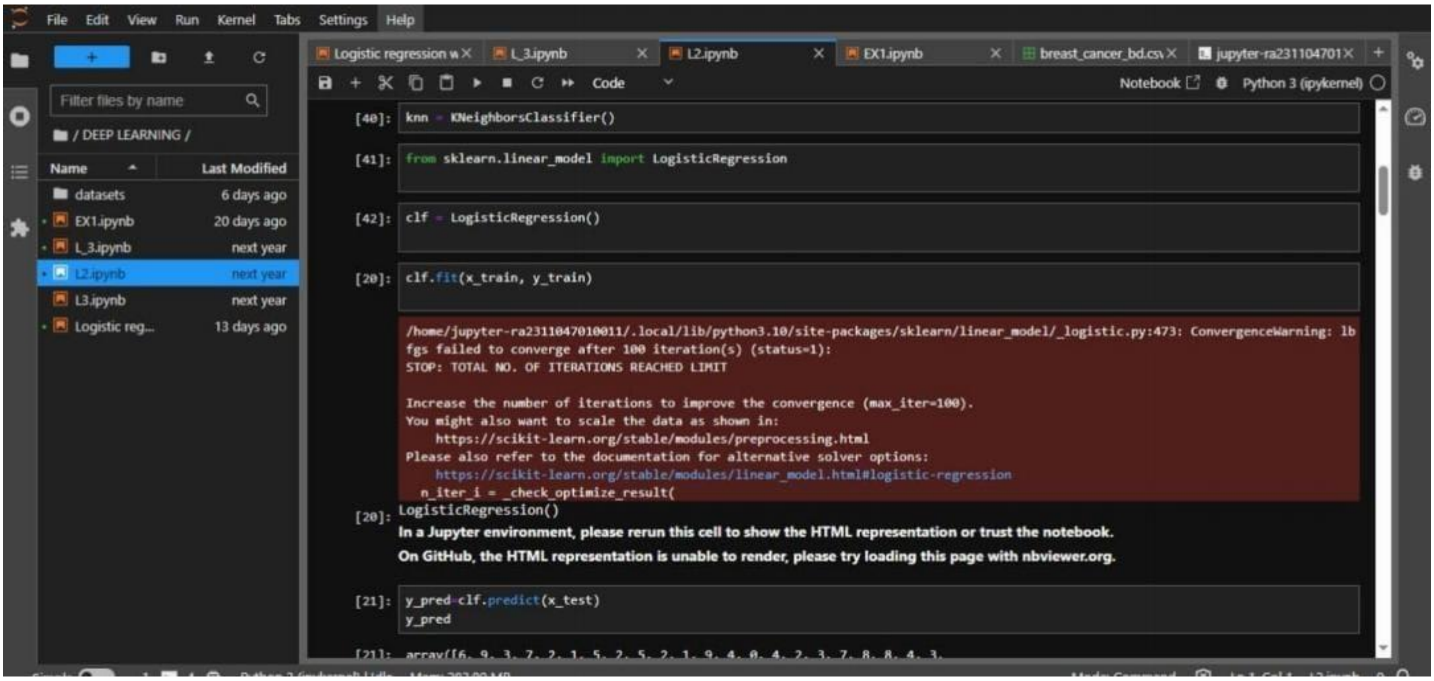


Name	Last Modified
datasets	6 days ago
EX1.ipynb	20 days ago
L_3.ipynb	next year
L2.ipynb	next year
L3.ipynb	next year
Logistic reg...	13 days ago

```
[32]: plt.matshow(d.images[5], cmap="gray")

[32]: <matplotlib.image.AxesImage at 0x720a94ef9ba0>
```

```
[33]: cm_display.plot()
      plt.show()
```



```
[40]: knn = KNeighborsClassifier()

[41]: from sklearn.linear_model import LogisticRegression

[42]: clf = LogisticRegression()

[20]: clf.fit(x_train, y_train)

      /home/jupyter-ra2311047010011/.local/lib/python3.10/site-packages/sklearn/linear_model/_logistic.py:473: ConvergenceWarning: lb
      fgs 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
        n_iter_i = _check_optimize_result(
[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, 0, 4, 2, 3, 7, 8, 8, 4, 3,
```