STD.: _____ DIV./SEC.: _____ SUBJECT: _____ ROLL NO.: 008

S. No.	Date	Title	Page No.	Teacher's Sign/Remarks
1.	24.07.25	Exploring The Deep Learning Platforms.		
2.	7.08.25	Implement a classifier using open-Source Dataset.		
3.	7.08.2025	Study of the classifier with respect to statistical parameters.		14/8/25
4.	14.08.2025	Build a single Feed Forward neural network to recognise handwritten characters.		

7.08.2025

Exp 3: STUDY OF THE CLASSIFIERS WITH RESPECT TO STATISTICAL PARAMETERS

AIM:

To Study of the classifiers with respect to Statistical Parameters.

ALGORITHM:

⇒ Load an open-source dataset.

⇒ Split the dataset into training and testing sets.

⇒ Standardize the dataset.

⇒ Initialize multiple classifiers.

⇒ Train each classifier using the training data.

⇒ Predict on the test set using each model.

⇒ Evaluate each model using:
- Accuracy
- Precision.
- Recall
- F1-score
- Confusion Matrix.

⇒ Compare the results.

PSEUDOCODE:

BEGIN
Load dataset.
Split data into training and testing sets.
Standardize the features.
Initialize classifiers (Logistic Regression, k-NN, SVM, Decision Tree)

FOR each classifier:
Train classifier with training data.
Predict on test data.
Calculate Accuracy, Precision, Confusion Matrix
Store results.
END FOR
Display and compare results
END.

OBSERVATION

Classifier	DecisionTime	SVM	Logistic Regression
Accuracy	84.5 %.	98.6 %.	97.4 %.
Avg precision	85.3 %.	98.72 %.	97.67 %.
Avg Recall.	83.98 %.	98.66 %.	97.65 %.
Avg F1-score.	84.3 %.	98.68 %.	97.65 %.
weighted F1-score.	84.72 %.	98.61 %	97.51 %.

* SVM provides the best overall classification performance on the digits dataset.

* Logistic Regression is very close in performance and significantly better than decision tree.

* Decision tree shows some signs of overfitting in or misclasification in some cases, especially shapes.

Code:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report,
                            confusion_matrix

data = load_iris()
X = data.data
y = data.target
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.3, random_state = 42)

Scaler = StandardScaler()
X_train = Scaler.fit_transform(X_train)
X_test = Scaler.transform(X_test)
classifiers = {
    "Logistic Regression": LogisticRegression(),
    "K-Nearest Neighbors": KNeighborsClassifier(n_neighbors=3),
    "Support Vector Machine": SVC(),
    "Decision Tree": DecisionTreeClassifier()
}
```

```
for name, clf in classifiers.items():
    clf.fit(x_train, y_train)
    y_pred = clf.predict(x_test)
    print(f"=== {name} ===")
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
    print("Classification Report:")
    print(classification_report(y_test, y_pred,
        target_names=data.target_names))
    print("\n")
```

OUTPUT:

Accuracy: 0.8472222.

Confusion matrix:

```
array([[29, 0, 1, 0, 1, 1, 0, 1, 0, 0],
       [0, 22, 1, 0, 1, 0, 1, 1, 1, ],
       [0, 0, 0, 29, 0, 1, 0, 1, 2, ]])
```

29/8/23

RESULT:

Therefore the study of the classifiers with respect to statistical parameters is completed successfully.

PSEUDOCODE:

BEGIN

Load the Iris dataset.

Split the dataset into training and test sets

Initialize the k-NN classifier with k neighbors

Train the classifier using the training set.

Predict the class labels on the test set.

Compute the accuracy of the model.

Print the accuracy.

END.

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DL / WEEK-3 /

Name	Last Modified
Untitled.ipy...	next year

Untitled.ipynb | Untitled.ipynb | opensource.ipynb

Code

Notebook | Python 3 (ipykernel)

[8]:
```
!pip install sklearn
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting sklearn
  Using cached sklearn-0.0.post12.tar.gz (2.6 kB)
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> [15 lines of output]
      The 'sklearn' PyPI package is deprecated, use 'scikit-learn'
      rather than 'sklearn' for pip commands.

      Here is how to fix this error in the main use cases:
      - use 'pip install scikit-learn' rather than 'pip install sklearn'
      - replace 'sklearn' by 'scikit-learn' in your pip requirements files
        (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
      - if the 'sklearn' package is used by one of your dependencies,
        it would be great if you take some time to track which package uses
        'sklearn' instead of 'scikit-learn' and report it to their issue tracker
      - as a last resort, set the environment variable
        SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

      More information is available at
      https://github.com/scikit-learn/sklearn-pypi-package
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> See above for output.

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
```

Simple | 0 | 3 | Python 3 (ipykernel) | Idle | Mem: 521.74 MB

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DL / WEEK-3 /

Name	Last Modified
Untitled.ipy...	next year

Untitled.ipynb | Untitled.ipynb | opensource.ipynb

Code | Notebook | Python 3 (ipykernel)

[6]:
```
from sklearn.datasets import load_digits
```

[7]:
```
d=load_digits()
```

[9]:
```
x=d.data
y=d.target
```

[10]:
```
x
```

[10]:
```
array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ..., 10.,  0.,  0.],
       [ 0.,  0.,  0., ..., 16.,  9.,  0.],
       ...,
       [ 0.,  0.,  1., ...,  6.,  0.,  0.],
       [ 0.,  0.,  2., ..., 12.,  0.,  0.],
       [ 0.,  0., 10., ..., 12.,  1.,  0.]], shape=(1797, 64))
```

[11]:
```
y
```

[11]:
```
array([0, 1, 2, ..., 8, 9, 8], shape=(1797,))
```

[12]:
```
from sklearn.model_selection import train_test_split
```

[13]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.2,random_state=42)
```

[14]:
```
from sklearn.tree import DecisionTreeClassifier
```

[15]:
```
clf=DecisionTreeClassifier()
```

[16]:
```
clf.fit(x_train,y_train)
```

[16]:
```
DecisionTreeClassifier
▶ Parameters
```

Tabs Settings Help

Untitled.ipynb | Untitled.ipynb | opensource.ipynb

Code | Notebook | Python 3 (ipykernel)

[17]:
```
array([6, 9, 3, 7, 2, 1, 5, 3, 5, 7, 2, 7, 4, 0, 4, 2, 3, 7, 8, 8, 4, 3,
       9, 7, 5, 6, 3, 5, 6, 3, 4, 9, 1, 4, 4, 6, 9, 4, 7, 6, 6, 9, 1, 3,
       6, 1, 3, 0, 6, 5, 5, 1, 4, 5, 6, 0, 3, 0, 0, 8, 5, 4, 8, 2, 4, 5,
       7, 0, 7, 5, 9, 9, 5, 4, 7, 0, 4, 5, 5, 9, 9, 0, 2, 3, 8, 0, 6, 4,
       4, 3, 1, 2, 5, 3, 9, 2, 9, 4, 4, 7, 4, 3, 4, 3, 4, 3, 5, 9, 4, 2,
       7, 7, 4, 6, 1, 9, 2, 7, 8, 3, 2, 6, 9, 4, 0, 7, 2, 7, 5, 8, 7, 5,
       7, 9, 0, 6, 6, 4, 2, 8, 0, 9, 4, 6, 9, 9, 6, 9, 0, 3, 5, 6, 6, 0,
       6, 4, 3, 9, 3, 7, 7, 2, 9, 0, 4, 5, 8, 6, 5, 9, 9, 8, 4, 2, 1, 8,
       7, 7, 2, 2, 3, 9, 8, 0, 3, 3, 2, 5, 6, 9, 9, 4, 6, 5, 4, 3, 3, 6,
       4, 8, 5, 9, 5, 7, 1, 9, 4, 8, 1, 5, 4, 4, 9, 6, 1, 8, 6, 0, 4, 5,
       2, 7, 4, 6, 4, 5, 6, 7, 3, 2, 3, 6, 7, 1, 9, 1, 4, 7, 6, 9, 1, 5,
       5, 1, 4, 4, 8, 8, 9, 9, 7, 4, 2, 8, 2, 3, 5, 9, 1, 3, 6, 0, 9, 7,
       4, 0, 1, 0, 4, 5, 1, 5, 3, 6, 0, 7, 1, 0, 2, 3, 6, 5, 9, 7, 7, 5,
       9, 9, 9, 8, 5, 3, 5, 2, 0, 5, 8, 3, 4, 0, 2, 4, 6, 4, 3, 7, 5, 0,
       5, 2, 1, 3, 1, 4, 7, 1, 7, 0, 1, 5, 6, 1, 3, 8, 7, 0, 6, 4, 8, 8,
       5, 1, 8, 4, 5, 9, 7, 9, 8, 6, 0, 4, 2, 0, 7, 9, 8, 9, 5, 2, 7, 4,
       9, 7, 7, 4, 3, 8, 9, 5])
```

[18]:
```
from sklearn.metrics import accuracy_score
```

[19]:
```
accuracy_score(y_test,y_pred)
```

[19]:
```
0.8472222222222222
```

[20]:
```
from sklearn import metrics
```

[21]:
```
confusion_matrix=metrics.confusion_matrix(y_test,y_pred)
confusion_matrix
```

[21]:
```
array([[29,  0,  1,  0,  1,  1,  0,  1,  0,  0],
       [ 0, 22,  1,  0,  1,  0,  1,  1,  1,  1],
       [ 0,  0, 25,  4,  1,  0,  1,  1,  1,  0],
       [ 0,  0,  0, 29,  0,  1,  0,  1,  2,  1],
       [ 0,  0,  0,  0, 41,  1,  1,  3,  0,  0],
       [ 0,  0,  0,  0,  1, 40,  1,  0,  1,  4],
       [ 0,  0,  0,  0,  3,  0, 32,  0,  0,  0],
```

n 3 (ipykernel) | Idle Mem: 526.56 MB Mode: Edit Ln 2, Col 1 Untitled.ipynb 0

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DL / WEEK-3 /

Name	Last Modified
Untitled.ipy...	next year

Untitled.ipynb | Untitled.ipynb | opensource.ipynb

Code

Notebook | Python 3 (ipykernel)

[22]:
```
cm_display=metrics.ConfusionMatrixDisplay(confusion_matrix=confusion_matrix,display_labels=[0.1])
cm_display
```

[22]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x70398c6e1f90>
```

[26]:
```
pip install matplotlib
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting matplotlib
  Downloading matplotlib-3.10.5-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.59.0-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (107 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 107.9/107.9 kB 4.4 MB/s eta 0:00:00
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.9-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl.metadata (6.3 kB)
Requirement already satisfied: numpy>=1.23 in /home/jupyter-ra2311047010008/.local/lib/python3.10/site-packages (from matplotlib) (2.2.6)
Requirement already satisfied: packaging>=20.0 in /opt/tljh/user/lib/python3.10/site-packages (from matplotlib) (24.0)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.3.0-cp310-cp310-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (9.0 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.3-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in /opt/tljh/user/lib/python3.10/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/tljh/user/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Downloading matplotlib-3.10.5-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (8.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.7/8.7 MB 38.7 MB/s eta 0:00:0000:0100:01
Downloading contourpy-1.3.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (325 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 325.0/325.0 kB 17.1 MB/s eta 0:00:00
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.59.0-cp310-cp310-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (4.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.8/4.8 MB 20.4 MB/s eta 0:00:0000:0100:01
Downloading kiwisolver-1.4.9-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 27.0 MB/s eta 0:00:00a 0:00:01
Downloading pillow-11.3.0-cp310-cp310-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl (6.6 MB)
```

Python 3 (ipykernel) | Idle | Mem: 529.40 MB

Mode: Edit | Ln 2, Col 1 | Untitled.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ DL / WEEK-3 /

Name	Last Modified
Untitled.ipy...	next year

Untitled.ipynb | Untitled.ipynb | opensource.ipynb

Code

Notebook | Python 3 (ipykernel)

```
[notice] To update, run: pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

[27]:
```
import matplotlib.pyplot as plt
```

[28]:
```
plt.show()
```

[29]:
```
plt.matshow(d.images[5], cmap="viridis")
plt.colorbar()
plt.show()
```