RA2311047010040

Index

மரம் வளர்ப்போம்
மண் வளம் காப்போம்

Superior® Note Books

Name : Vishva. S

Subject : DL

Std. : ..... Sec. : ..... Roll No. .....

School : .....

No.	Date	Title	Marks	Signature
1	24/7/25	Exploring the DL ~~platforms~~.		
2	31/7/25	Implement a classifier using open source dataset		
3	7/8/25	Study the classifier with respect to statistical parameters.		

31/7/25

2. Implement a classifier using open source data

Aim:

To build a machine learning classifier to predict the species of Iris flowers using the Decision Tree algorithm.

Objective:

* Understand & apply Decision Tree classification on the Iris dataset.
* Preprocess the dataset & split it into training & testing sets.
* Train the model & evaluate its performance using accuracy metrics.
* Interpret the results & observation from the model.

Source code:

```
!pip install pandas scikit-learn seaborn
from sklearn.datasets import load_iris
from sklearn.model_selection import
                    train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

```
iris = load_iris()          # load the
x = iris.data                 Iris dataset.
y = iris.target

# Split data into train & test

x_train, x_test, y_train, y_test = train_test_split
                    (x, y, test_size=0.2, random_
                                state = 42)

# Decision Tree
clf = DecisionTreeClassifier(random_state=42)
clf.fit(x_train, y_train)

y_pred = clf.predict(x_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Decision Tree accuracy on Iris
        dataset : {accuracy*100:.2f}%")
```

Observation:

* The Decision Tree classifier successfully learned patterns from the training data & accurately predicted the species of Iris flowers on the test data.

* The model achieved a high accuracy indicating that the dataset is well suited for decision tree classification.

* The simplicity of the iris dataset & clear separation among classes contribute to the high performance of the model.

* The use of random state ensured reproducible results for consistent evaluation.

Accuracy : 1.00

Precision : 1.00

Recall : 1.00

F1-Score : 1.00

Result:

Successfully Implemented the Iris dataset with the help of the Decision tree classifier.

lab2.ipynb | lab3.ipynb

C: > Users > lenovo > Downloads > lab2.ipynb > !pip install pandas scikit-learn seaborn

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Detecting Kernels

```
!pip install pandas scikit-learn seaborn
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

[5] Python

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (2.3.1)
Requirement already satisfied: scikit-learn in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (1.7.1)
Requirement already satisfied: seaborn in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (0.13.2)
Requirement already satisfied: numpy>=1.22.4 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from pandas) (2.2.5)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/tljh/user/lib/python3.10/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.8.0 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from seaborn) (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /opt/tljh/user/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.0)
Requirement already satisfied: pillow>=8 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /home/jupyter-ra2311047010040/.local/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.2.3)
Requirement already satisfied: six>=1.5 in /opt/tljh/user/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: pip install --upgrade pip
```

```
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

[6] Python

```
iris = load_iris() #load the dataset for iris
X = iris.data
y = iris.target
```

[7] Python

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[8] Python

lab2.ipynb | lab1.ipynb

C: > Users > lenovo > Downloads > lab2.ipynb > !pip install pandas scikit-learn seaborn

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Python 3.11.

[12]
```
from sklearn.datasets import load_breast_cancer
```
Python

[13]
```
# Load Breast Cancer Wisconsin dataset
breast_cancer= load_breast_cancer()
X = breast_cancer.data
y = breast_cancer.target
```
Python

[14]
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```
Python

[15]
```
# Initialize and train the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```
Python

...

DecisionTreeClassifier

▶ Parameters

[16]
```
# Predict on the test set
y_pred = clf.predict(X_test)
```
Python

[17]
```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Decision Tree accuracy on Breast Cancer dataset: {accuracy * 100:.2f}%")
precision = 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(f"Decision Tree accuracy on Iris dataset: {accuracy * 100:.2f}%")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")
```
Python

lab2.ipynb | lab3.ipynb

C: > Users > lenovo > Downloads > lab2.ipynb > !pip install pandas scikit-learn seaborn

Generate | Code | Markdown | Run All | Clear All Outputs | Outline | ... | Python 3.11.9

[15]
```
# Initialize and train the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```
Python

```
DecisionTreeClassifier
▶ Parameters
```

[16]
```
# Predict on the test set
y_pred = clf.predict(X_test)
```
Python

[17]
```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Decision Tree accuracy on Breast Cancer dataset: {accuracy * 100:.2f}%")
precision = 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(f"Decision Tree accuracy on Iris dataset: {accuracy * 100:.2f}%")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1-score: {f1:.2f}")
```
Python

```
Decision Tree accuracy on Breast Cancer dataset: 94.74%
Decision Tree accuracy on Iris dataset: 94.74%
Precision: 0.94
Recall: 0.94
F1-score: 0.94
```

[ ]
Python