

47772200

7-09-2025

Exp 2 : IMPLEMENT A CLASSIFIER

USING OPEN-SOURCE DATASET.

AIM:

To implement a classifier using Open-Source dataset.

ALGORITHM:

- ⇒ Load the Iris dataset.
- ⇒ Split the dataset into training and testing sets.
- ⇒ Preprocess the data if necessary.
- ⇒ Train a K-NN classifier on the training data.
- ⇒ Predict the class labels on the test data.
- ⇒ Evaluate the classifier using accuracy score.

$(\text{test_p}, \text{test_x})$ of test

$(\text{test_x}) \text{ belong_to } = \text{best_p}$

$(\text{best_p}, \text{test_p}) \text{ accuracy_score} = \text{accuracy}$

RESULT

Accuracy of K-NN classifier is 1.0

(process)

using

Accuracy of K-NN classifier is 1.0

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.

Code:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

```
iris = load_iris()
```

```
x = iris.data
```

```
y = iris.target
```

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

```
Scaler = StandardScaler()
```

```
x_train = Scaler.fit_transform(x_train)
```

```
x_test = Scaler.transform(x_test)
```

```
knn = KNeighborsClassifier(n_neighbors = 3)
```

```
knn.fit(x_train, y_train)
```

```
y_pred = knn.predict(x_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Accuracy of K-NN classifier on Iris dataset:", accuracy)
```

Output:

Accuracy of K-NN classifier on Iris dataset : 1.0

92EN DOCEDE : BEGIN
load dataset
split data into training and testing sets
standardize the features
Initialize classifier (logistic regression, K-NN, SVM, etc.)
Train classifier with training data
Predict on test data
Calculate Accuracy, Precision, Confusion matrix
Store results
END FOR
Display and compare results
END

RESULT:

Therefore the implementation of a classifier using open-source dataset is completed successfully.

Untitled.ipynb x opensource.ipynb x +

Notebook Python 3 (ipykernel)

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-ra2311047010008/.local/lib/python3.10/site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /home/jupyter-ra2311047010008/.local/lib/python3.10/site-packages (from pandas) (2025.2)
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)
Note: you may need to restart the kernel to use updated packages.

[4]:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

iris = load_iris()
X = iris.data
y = iris.target

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

clf = KNeighborsClassifier(n_neighbors=5)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of KNN classifier: {accuracy:.2f}")
```

Accuracy of KNN classifier: 1.00

(ipykernel) | Idle Mem: 314.98 MB