

RA2311047010005



NAME: Kuberan STD.: A1 SEC.: A ROLL NO.: _____ SUB.: Deep learning

2. Implement a classifier using open source dataset

Aim::

To build a machine learning classifier to predict the species of Iris flowers using decision Tree Algorithm

OBJECTIVE

- * Understands and apply decision Tree classification on the Iris Dataset
- * Preprocess the dataset and splits it into training and testing sets
- * Train the model and evaluate its Performance using accuracy matrices
- * Interpret the results and observation from the model

Pseudo code::

1. START

1. Import necessary libraries
2. Load the Iris dataset
3. Standardize features
4. ~~Split data into training and testing sets~~
5. Initialize Decision Tree classifier
6. Train the model on Training data
7. Predict target on test data
8. Display results

END

Source code

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

data = load_breast_cancer()
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)
models = [
    "Logistic Regression": LogisticRegression(
        max_iter=5000),
    "KNN": KNeighborsClassifier(n_neighbors=5),
    "Decision Tree": DecisionTreeClassifier(
        criterion='gini')
]

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(name)
    print("Accuracy: ", accuracy_score(y_test, y_pred))

    print("Recall: ", recall_score(y_test,
        y_pred, average='macro'))

    print("f1-score: ", f1_score(y_test, y_pred,
        average='macro'))
```

EX: P:8

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:

~~After training classifier using open source dataset~~ successfully implemented a

SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '='?

```
[7]: from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
x, y = load_iris(return_X_y=True)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
model = KNeighborsClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print(confusion_matrix(y_test, y_pred))
```

```
[[14  0  0]
 [ 0 12  1]
 [ 0  1 17]]
```

```
[7]: from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score,confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
data = load_breast_cancer()
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)
models = {"LogisticRegression": LogisticRegression(max_iter = 5000), "KNN": KNeighborsClassifier(n_neighbors = 5), "DecisionTree": DecisionTreeClassifier()}
for name,model in models.items():
    model.fit(x_train,y_train)
    y_pred = model.predict(x_test)
    print(name)
    print("Accuracy :accuracy_score(y_test,y_pred)")
    print("Precision :precision_score(y_test,y_pred)")
    print("Recall :recall_score(y_test,y_pred)")
    print("F1score: f1_score(y_test,y_pred)")
    print("ConfusionMatrix :/n confusion_matrix(y_test,y_pred))
```