Assignment 3

Subject: Machine Learning

Course code: CSE 465

Assignment No: 3

Submitted To	Submitted By
Khan Md. Hasib Lecturer Dept. of CSE BUBT	Ahmed Mahir Shoaib ID: 18192103235 Intake: 41 Section: 7 Dept. of CSE BUBT

> What are the differences between the following data sets?
> 1- Contact Lens data: http://archive.ics.uci.edu/ml/datasets/Lenses
> 2- Iris data: http://archive.ics.uci.edu/ml/datasets/Iris
> From the following algorithms which one is expected to perform
> best on
> the Contact Lens data? Implement those on the Contact Lens data.
> K-Nearest Neighbors
> Decision Tree
> Neural Networks

The Contact Lens data set and Iris data set are different in terms of the number of samples, features and the problem they solve.

1. Contact Lens data set: This data set contains 24 samples and 5 attributes. It is a small data set that describes the attributes of different contact lenses and the corresponding recommendation of a specialist. The attributes include age, spectacle prescription, astigmatism, tear production rate, and the type of contact lens recommended.
2. Iris data set: This data set contains 150 samples and 4 attributes. It is a well-known data set that is commonly used for classification tasks in machine learning. The attributes include sepal length, sepal width, petal length, and petal width, and the target variable is the species of the iris.

From the given algorithms, the Decision Tree algorithm is expected to perform best on the Contact Lens data, as it is a small data set with discrete features. K-Nearest Neighbors and Neural Networks may not perform as well on this data set due to its small size.

Here is the implementation of K-Nearest Neighbors, Decision Tree, and Neural Networks on the Contact Lens data:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

# Download and load dataset
url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/lenses/lenses.data'
columns = ['index', 'age', 'prescription', 'astigmatism', 'tear_rate', 'class']
df = pd.read_csv(url, header=None, names=columns, delim_whitespace=True)

# Drop the index column
df.drop('index', axis=1, inplace=True)

# Split the dataset into training and testing sets
X = df.drop('class', axis=1)
y = df['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

- For KNN

  ```
  from sklearn.neighbors import KNeighborsClassifier
  from sklearn.metrics import accuracy_score

  # Fit and evaluate the K-Nearest Neighbors model
  knn = KNeighborsClassifier(n_neighbors=3)
  knn.fit(X_train, y_train)
  y_pred = knn.predict(X_test)
  knn_acc = accuracy_score(y_test, y_pred)
  print('K-Nearest Neighbors accuracy:', knn_acc)
  ```

- For Decision tree

  ```
  from sklearn.tree import DecisionTreeClassifier

  # Fit and evaluate the Decision Tree model
  dt = DecisionTreeClassifier(random_state=42)
  dt.fit(X_train, y_train)
  y_pred = dt.predict(X_test)
  dt_acc = accuracy_score(y_test, y_pred)
  print('Decision Tree accuracy:', dt_acc)
  ```

- For Neural Network

  ```
  from sklearn.neural_network import MLPClassifier

  # Fit and evaluate the Neural Networks model
  nn = MLPClassifier(hidden_layer_sizes=(5, 2), random_state=42)
  nn.fit(X_train, y_train)
  y_pred = nn.predict(X_test)
  nn_acc = accuracy_score(y_test, y_pred)
  print('Neural Networks accuracy:', nn_acc)
  ```

In this case, we've used the KNeighborsClassifier class from scikit-learn to implement K-Nearest Neighbors, the DecisionTreeClassifier class to implement Decision Tree, and the MLPClassifier class to implement Neural Networks. We've also used the accuracy_score function from scikit-learn to evaluate the accuracy of each model.