Practical 6
Python programs on Numpy, Pandas, Scikit-learn (Level 2)

a. Numpy Example

```
#How to sort the elements in the given array using Numpy?

import numpy as np

array = np.array([
    [3, 7, 1],
    [10, 3, 2],
    [5, 6, 7]
])
print(array)
print()

# Sort the whole array
print(np.sort(array, axis=None))

# Sort along each row
print(np.sort(array, axis=1))

# Sort along each column
print(np.sort(array, axis=0))
```

b. Pandas Example

```
#Creating pandas dataframe using python dictionary

import pandas as pd

# create a dictionary
data = {'Name': ['John', 'Alice', 'Bob'],
        'Age': [25, 30, 35],
        'City': ['New York', 'London', 'Paris']}

# create a dataframe from the dictionary
df = pd.DataFrame(data)

print(df)
```

c. Scikit learn

```
# load the iris dataset as an example
from sklearn.datasets import load_iris
iris = load_iris()

# store the feature matrix (X) and response vector (y)
X = iris.data
y = iris.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
random_state=1)

# training the model on training set
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# making predictions on the testing set
y_pred = knn.predict(X_test)

# comparing actual response values (y_test) with predicted response
values (y_pred)
from sklearn import metrics
print("KNN model accuracy", metrics.accuracy_score(y_test, y_pred))

# making prediction for out of sample data
sample = [[3, 5, 4, 2], [2, 3, 5, 4]]
preds = knn.predict(sample)
pred_species = [iris.target_names[p] for p in preds]
print("Predictions", pred_species)
```