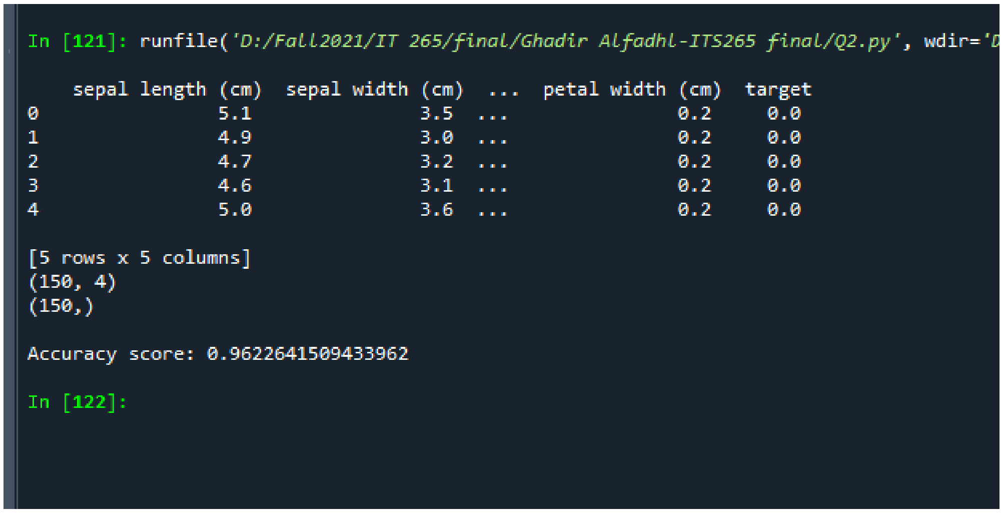
Final Exam ITS 265

Ghadir Alfadhl

Question 2: Supervised Learning: Create a python program using Sklearn to perform iris flower classification on the built-in Sklearn iris dataset using supervised learning. Import the dataset from SKlearn datasets. Use Pandas pd.DataFrame to extract the feature columns (feature_names) and target column (target) into separate input X and output Y datasets. Split the data into training and test sets using SKLearn train_test_split function. Use a split of 35% for testing and 65% for training picking random data items. Use the Naives Bayes model GaussianNB() . Train the model using the training set and test the model using the test set. Print out the accuracy score.



```
In [121]: runfile('D:/Fall2021/IT 265/final/Ghadir Alfadhl-ITS265 final/Q2.py', wdir='D

    sepal length (cm)  sepal width (cm)  ...  petal width (cm)  target
0                 5.1               3.5  ...               0.2     0.0
1                 4.9               3.0  ...               0.2     0.0
2                 4.7               3.2  ...               0.2     0.0
3                 4.6               3.1  ...               0.2     0.0
4                 5.0               3.6  ...               0.2     0.0

[5 rows x 5 columns]
(150, 4)
(150,)

Accuracy score: 0.9622641509433962

In [122]:
```

Code:

import numpy as np

import pandas as pd

from sklearn import datasets

from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score

load the data

```
iris_dataset = datasets.load_iris()

# seprate the input and output
#Use Pandas pd.DataFrame to extract the feature columns
#df = pd.DataFrame(iris_dataset.data, columns=iris_dataset.feature_names)

df = pd.DataFrame(data= np.c_[iris_dataset['data'], iris_dataset['target']],
            columns= iris_dataset['feature_names'] + ['target'])
print("\n", df.head())

x = iris_dataset.data
y = iris_dataset.target


#split train and test data
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.35,
                                                    random_state=42)
'''
print("\n65% train data:")
print(X_train)
print(y_train)
print("\n35% test data:")
print(X_test)
print(y_test)

'''

print(x.shape)
print(y.shape)
```

```
# print('X_train')
# print(x_train)
# print('X_test')
# print(x_test)

 # create gausian naive bayes classifier, train the classifier,
 # predict the values for test data

gnb = GaussianNB()
gnb.fit(x_train, y_train)
y_pred = gnb.predict(x_test)

from sklearn import metrics
accuracy = metrics.accuracy_score(y_test, y_pred) # calculate the accuracy score
print("\nAccuracy score:", accuracy)
```