Introduction to data science - Assignment #4

The 'kidney_disease.csv' data file attached to the assignment file (available at Sakai) is taken from the Kaggle website [1].

The Kidney Disease Data reports the age, blood pressure, and the results of the blood test factors for healthy people and kidney patients.
This data includes the following 9 variables:

- 'age': Age of the individual (numeric: from 2 to 90)
- 'bp': Blood pressure (numeric: from 50 to 180)
- 'sod': Blood sodium level test result (numeric: from 104 to 163)
- 'pot': Blood potassium level test result (numeric: from 2.5 to 47)
- 'hemo': Hemoglobin blood test result (numeric: from 3.1 to 17.8)
- 'pcv': Packed cell volume test result (numeric: from 9 to 54)
- 'wc': White blood cell test result (numeric: from 2200 to 26400)
- 'rc': Red blood cell test result (numeric: from 2.1 to 8)
- 'CKD': Chronic kidney disease (binary: 0 for healthy individuals and 1 for kidney patients)

Note. You must put the CSV files in the same folder as your code file. If you use Jupyter notebook it should be in the address: 'C:/Users/YOUR-USER-NAME'.
You can also read the file with its address, for example:

```
f = open('C:/files/sample-file.txt')
```

Question 1

The following code is to

1. train decision tree model that predicts chronic kidney disease based on other variables in the Kidney Disease Data,
2. plot the trained tree and save it in the file 'tree.pdf'.

```
import pandas as pd
from sklearn import tree
import matplotlib.pyplot as plt

data = pd.read_csv('kidney_disease.csv')

X, y = data.drop(['ckd'],axis=1), data[['ckd']]

feature_names = list(X.columns)

################## your code here ###################
## you need to
## 1. train a decision tree classifier to predict 'ckd'


#########################################################


fig = plt.figure(figsize=(40,20))
fig = tree.plot_tree(model, feature_names=feature_names,
class_names=['0','1'], filled=True)
plt.savefig('tree.pdf')
```

Complete the code and then use the plot to determine whether or not the following two cases have kidney disease.

- Person (A) with
  - age: 25
  - rc: 4
  - wc: 6600
  - bp: 70
  - pot: 4.2
  - pcv: 38
- Person (B) with

- age: 62
- rc: 5
- wc: 7200
- bp: 80
- pot: 2.5
- pcv: 40

Note: In the plotted tree, the right branch corresponds to the answer 'no' and the left branch corresponds to the answer 'yes' to the question of the decision node.

Question 2

The following code loads the diabetes dataset from `sklearn` package and scales the target variable to be in the range (0,1).
Features in this dataset are age, sex, BMI, blood pressure, and six blood serum measurements all scaled in range (-2,2), and the target variable is the progression of diabetes in patients.

1. Complete the code to train a neural network model with one hidden layer including 8 neurons. Use 70% of the data for training the model and set the 'epochs' argument to 10.

2. Is the 'relu' activation function appropriate for the output layer of this model?

3. Compare the model performance with and without using the 'relu' activation function on the testing set. You can use `mean_absolute_error`.

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.datasets import load_diabetes

# load dataset
diabetes = load_diabetes()
X = diabetes.data
y = diabetes.target


scaler = MinMaxScaler()
```

```
y = y.reshape(-1,1)
y = scaler.fit_transform(y)

################## your code here ##################
## you need to
## 1. train a neural network model to predict diabetes
## progression with or without the 'relu' activation function
## for the output layer
## 2. Measure the performance of the models on the
## testing set


#####################################################
```

Question 3

The 'iris_model.h5' file (available at Sakai) is a neural network model trained to predict the iris type based on features in iris data in `sklearn` package. Write a code to load this model and determine its accuracy on iris data using `evaluate()` method.

Question 4

The `mnist` dataset in the `tensorflow` package is an image dataset similar to the `digits` dataset, which contains images of handwritten digits, but with different sizes (28x28 pixels).

The following code loads the training and testing sets of this dataset. Each instance of X_train (X_test) is a list of numbers representing the color of image pixels. y_train (y_test) includes the label for each instance (a digit between 0-9).

1. Complete the code to create and train a neural network model using the training set to predict the digits in images and then report the accuracy on the testing set.
2. Save the model in a file with the name 'digits_model.h5'.

```
from tensorflow.keras.datasets import mnist

# load dataset
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```
5
6       number_of_training_instances = X_train.shape[0]
7       number_of_testing_instances = X_test.shape[0]
8
9       # reshape dataset
10      X_train = X_train.reshape((number_of_training_instances, 28*28))
11      X_test = X_test.reshape((number_of_testing_instances, 28*28))
12
13
14      ################## your code here ##################
15      ## you need to
16      ## 1. Train a neural network model to predict digits
17      ## with an arbitrary number of hidden layers and neurons
18      ## 2. Measure the accuracy on the testing set
19      ## 3. Save the model in 'digits_model.h5'
20
21
22      ########################################################
```

References

[1] Chronic kidney disease dataset. https://www.kaggle.com/datasets/mansoordaku/ckdisease?resource=download.