CSCI 333 Final Project

100 points + 10 bonus points

Note: This is an individual assignment. Each student **MUST** complete the work on his/her own. Any code sharing/plagiarism is not tolerated.

Overview

This project consists of three tasks. The goal is to apply what we have learned to solve real problems in Data Science and Machine Learning. Glance at "What to Submit" when you start working on a task so that you know what information to provide from each task.

Submission Example

```
csci333-project-XX
  csci333-project-XX.doc
  Task1XX.py
  namelist.txt
  patientList.txt
  task2XX.py
  task3XX.py
  README.txt
```

What to Submit

1. One doc file "csci333-project-*XX*.doc" including the text source code and screenshots of the outputs of all programs. Please replace *XX* with your first name and last name. You can copy/paste the text source code from Pycharm or other IDEs into the doc file. Hopefully, based on the screen snapshots of the output, you can show that your programs passed tests and were well.

2. Python files for all programs. In well-defined programs, proper comments are required. For programs without comments, they will be deducted greatly in grade.

3. Note that if any program or code does not work, you can explain the status of the program or code and then attach your explanation and description in a file "README.txt".

4. Optional. Anything you want to attract the attention of instructor in grading.

Task 1 (20 points): (Intro to Data Structure and Data Science: Survey Response Statistics) Write a program that create, calculate, and display the survey Response. Five hundred (500) people were asked to quantify their pain by using a numerical rating scale (NRS) from 0 to 10. Zero means no pain; one to three (inclusive) means mild pain; four to six means moderate pain; seven to nine means severe pain; and ten mean the worst pain, as shown in Fig. 1.

Based on the pain scale, write a program by performing the following subtasks:

Perform the following subtasks:

(a) Create a patientlist variable to select 500 names sequentially from the file "namelist.txt" based on your last four digits of CWID. For example, the last four digits of your CWID are 5678. The first 5 names are "Suzy" in the line 5678, "Suzannah","Sully","Sulema", and "Sueann".

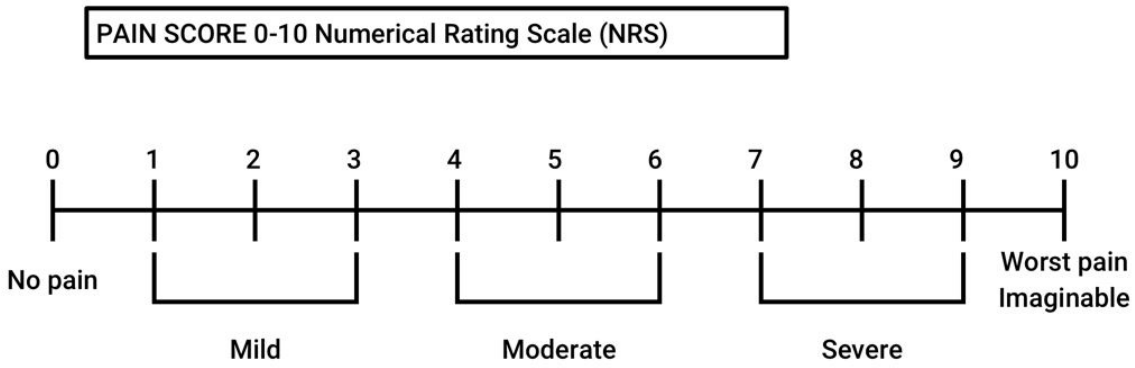

Figure 1: Pain Scale

(b) Use random.randint() or numpy.random.randint() to generate 500 responses for 500 patients in a list.
(c) Create a function to include (a) and (b) and create a file "patientList.txt" by saving the patients from the paitentlist and responses from the responselist in the format "patient" and "response" per line. For instance, "Suzy 9".

```
Suzy 9
Suzannah 6
Sully 2
... ...
```

(d) Determine and display the frequency of each pain value i from 0 to 10.

(e) Use the built-in functions, *statistics* module functions and *NumPy* or *Panda* functions covered in the course materials to display the following response statistics: minimum, maximum, range, mean, median, variance and standard deviation.

(f) Display a bar chart showing the response frequencies and their percentages of the total responses. The x-axis should show 11 pain values while the y-axis should show each pain value's relative frequency in %.

(g) Test your function and display each pain with its relative frequency.

Grading Rubric

- 5 points for defining functions.
- 5 points for finishing Task1(a)-(g).
- 5 points for a runnable python program with correct data visualization.
- 5 points for appropriate comments and screenshots of the program

Task 2 (30 points): (Intro to Data Science: Pandas-dateframes) Write a program that does the following tasks with pandas DataFrames (as shown in the slides "09-02-Data Science.pdf"):

(a) Create a dictionary "patients" by reading all patients from the file "patientList.txt" created by Task 1.
(b) Create a DataFrame named patientData from a dictionary "patients".
(c) Recreate the DataFrame patientData in Part (a) with custom indices using the 'Name keyword argument and 'Pain'.
(d) Select from patientData the column of temperature readings for 'Name'.
(e) Select from patientData the row of 'Pain' readings.

(f) Based on pain values, insert a new column "Level" with 5 possible values "No pain", "Mild", "Moderate", "Severe", or "Worst" by referring to Figure 1. For each patient, specify its level based on the pain value. For instance, if the pain value is 9, the level should be "Severe".
(g) Use the describe() method to produce patientData's descriptive statistics.
(h) Transpose patientData (One example can be found at https://www.geeksforgeeks.org/python-pandas-dataframe-transpose/).
(i) Display a bar chart showing the pain level frequencies and their percentages of the total number of records. The x-axis should show 5 levels while the y-axis should show each pain level's relative frequency in %.

Grading Rubric

- 10 points for defining functions.
- 5 points for finishing Task2(a)-(i).
- 5 points for appropriate comments and necessary screenshots of the program.
- 10 points for a runnable python program with correct data visualization.

Task 3 (50 points): (Classification with k-Nearest Neighbors and the Digits Dataset) Read the file "09-02-MachineLearning-Long.pdf" and the python program "CaseStudyDemo.py" to learn the algorithm of k-Nearest Neighbors with the Digits dataset for recognizing handwritten digits.

Re-write the python program by doing the following subtasks:

(a) Write code to display the two-dimensional array representing the sample image at index XY (where XY are the last two digits of your TAMUC CWID) and the numeric value of the digit the image represents.

(b) Write code to display the image for the sample image at index XY of the Digits dataset.

(c) For the Digits dataset, what the number of samples would the following statement reserve for training and testing purposes?

```
X_train, X_test, y_train, y_test = train_test_split(digits.data,
      digits.target, random_state=11, test_size=(XY%10)/10)
```

(d) Write code to get and display the number of training examples and the number of testing examples.

(e) Using the predicted and expected arrays, calculate and display the prediction accuracy percentage.

(f) Display and explain th Xth row of the confusion matrix presented in the example we have studied in the "Intro-to-MachineLearning-Part-II.mp4", where X is the last digit of your CWID.

(g) Rewrite the list comprehension in snippet [34] using a for loop. Hint: create an empty list and then use the built-in function "append".

```
# In[34]:
names = [str(digit) for digit in digits.target_names]
```

Grading Rubric

- 15 points for finishing Task3(a)-(g).
- 5 points for appropriate comments.
- 20 points for a runnable rewritten python program

– 10 points for screen-shots of the program.

Challenges in This Project

1. For 10% extra credit, you are welcome to explore the design of each task. Note: You still have to finish all tasks required by this project.
2. You should configure your machine and PyCharm properly to facilitate the project development.

Reference: [1] Computer Science. https://en.wikipedia.org/wiki/Computer_science

——————x————— **Good Luck** —————x——————