Lancaster University

MSCI 517: Introduction to Python

Coursework III (60% of module)

Deadline: 22/3/19, 10AM | *Lent Term* | Maximum Marks: 100

1 Coursework Description

This coursework is composed of two parts. You have been given the flexibility to structure the code in the way that you think is best following the principles of *re-usability*, *maintainability*, *information hiding* and *clarity*.

Part 1 (30 Marks)

You should submit a single .ipynb file containing a working version of all the tasks. The name of this file should be your library number (e.g. 123456789.ipynb, where 123456789 is an example library number). In case you want to import any packages apart from pandas, matplotlib, numpy, math and seaborn, you should contact the module leaders first.

A newly appointed health minister has heard reports that there are some hospitals in England where the average Length of Stay (LoS) of patients is over 1 week, whereas in many others the average LoS is under 1 week. You will need to download the file **patients.csv** from Moodle. The file contains the following data for patients admitted to one of five NHS hospitals in England:

Patient ID	String
Admission type	String
Age	Integer
Height	Integer
Weight	Integer
Hospital	String
LoS in hours	Integer

Task 1: (2 marks) Give the code that creates a DataFrame from the information in **patients.csv**. The DataFrame should be called **df**.

Task 2: (10 marks) LoS in **weeks** can be used to provide the following "Status":

LoS in weeks	**Status**
< 1	"Short LoS"
$= 1$	"Normal LoS"
> 1 and < 3	"Long LoS"
≥ 3	"Too Long LoS"

Give the commands that display the following columns: Patient ID, LoS in weeks, Status, and Flag. If Status is "Long LoS" or "Too Long LoS", then Flag should be set to "True", else it should be set to "False".

Task 3: (6 marks) Give the code that calculates the correlation between the Age and Height (using Pearson's correlation coefficient). Also give the code for producing a scatter-plot of this data.

Task 4: (6 marks) Give the code that creates a histogram of the Height and fit a normal curve over the histogram. Also give the code that calculates the mean, median, standard deviation, range and inter-quartile range of the Height.

Task 5: (6 marks) Give the code that displays only those patients with LoS more than one week. Now give the code that uses this data to calculate the mean Age of patients from each of the five NHS hospitals.

Part 2 (70 Marks)

You should submit a single .py file containing a working version of your program. The name of this file should be your library number (e.g. 123456789.py, where 123456789 is an example library number). In case you want to import any packages apart from pandas, matplotlib, numpy, math and seaborn, you should contact the module leaders first.

In this part, you are to write a program that attempts to solve a machine delivery problem using stochastic optimisation techniques. Specifically your program will read in a problem instance from a file and will attempt to produce a solution using heuristic methods.

It is important to break your code into small, meaningful functions and appropriately comment your code.

Problem description

There are machine requests from n customers that all have to be satisfied. There is one depot location where all the machines are located. Trucks are hired to transport the machines from the depot to the customers. The number of available trucks is $\lfloor\sqrt{n}\rfloor$, where n is the number of customers. All trucks must be allocated exactly $\lfloor n/\sqrt{n}\rfloor$ deliveries (locations). The route of a truck must start and end at the depot. We assume that it does not take any time to load a machine at the depot or to unload a machine at a customer.

The objective is to minimise the total distance travelled by the trucks. In order to determine the travelled distances, integer coordinates are provided for the customer locations. The location of the depot is always $(0, 0)$. The distance between coordinates $(x_1,\ y_1)$ and $(x_2,\ y_2)$ is defined as $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, i.e., the Euclidean distance.

The problem instances, which are available in .csv format, contain the customer IDs and the x and y coordinates of the customer locations (all in integers). An example of a problem instance file I1.csv is shown below.

ID	x	y
0	2	2
1	-2	1
2	0	-1
3	3	1

In this example, we have $n = 4$ customers. The number of available trucks is $\lfloor \sqrt{n} \rfloor = 2$. All trucks must be allocated $\lfloor n/\sqrt{n} \rfloor = 2$ deliveries (locations). The locations of the first, second, third and fourth customers are (2, 2), (-2, 1), (0, -1) and (3, 1), respectively.

Using the problem instance given above, a solution which specifies the routes for the trucks can be stored as follows: [[2, 1], [3, 0]]. Here the route for the first truck is given as [2, 1], and the route for the second truck is given as [3, 0].

Note that all trucks depart from and must return to the depot, so this is not mentioned explicitly in this solution format. The distance travelled by the first truck is 6.064. This can be computed by calculating the straight line distance between the coordinates of the locations (i.e. distance between depot (0, 0) and customer 2 (0, -1) + distance between customer 2 (0, -1) and customer 1 (-2, 1) + distance between customer 1 (-2, 1) and depot (0, 0)). The distance travelled by the second truck in this example is 7.405. Therefore, the total distance travelled by all trucks is $6.064 + 7.405 = 13.469$.

Task 1: (5 marks) On Moodle you will find three problem instances: I1.csv, I2.csv and I3.csv. The first task is to write a Python program that reads the coordinates of the customer locations from a given file.

Task 2: (10 marks) Generate an initial solution to this problem at random. To get the full mark you will need to add some 'intelligence' to the procedure to produce better quality initial solution.

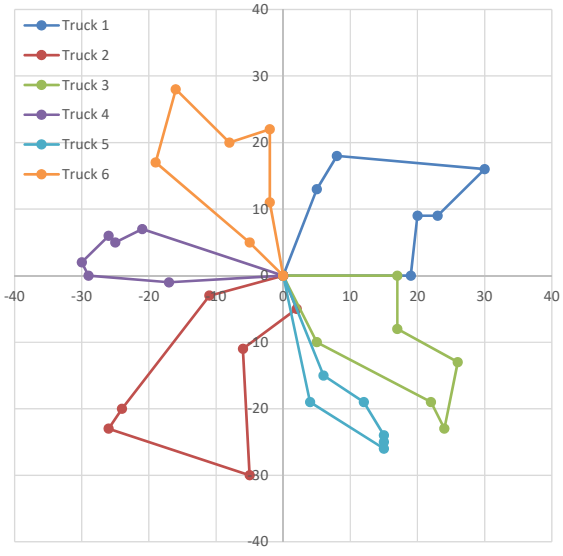
Task 3: (10 marks) Implement a TotalDistance function that calculates the total distance of the solution in the way described above.

Task 4: (30 marks) Implement a function that improves the initially generated solution using e.g. hill climbing method which accepts only non-worsening moves. To do this, you will need to implement an appropriate operator, e.g. swap operator. To get the full mark, you should make your method more efficient and powerful. Here are some ideas: When performing a move, you may notice that only few trucks are altered. Thus, you may be able to speed up your program by only re-evaluating the effected trucks. You may also consider advanced techniques such as simulated annealing, tabu search, or great deluge algorithms (you will need to research these yourselves). You might also consider more than one operators (e.g. swap and insert operators). In any case, you program should be run for a **fixed number of iterations between 100,000 to 1,000,000**.

Task 5: (10 marks) At the end of the run, your best obtained solution should be saved in a .csv file. The file specifies the routes for the trucks, and the last row displays the total distance of the solution. This format is extremely important, as I will be using my own checker software tool to automatically examine your solution and your calculated total distance. The example solution described above would be displayed:

2	1
3	0
13.469	

Task 6: (5 marks) Update your code to provide a visual representation of the best solution. Below an example of a visual representation to a solution produced by solving the problem instance I2.csv.



2 Coursework Submission

The coursework deadline is 10:00AM, Friday 22nd March, 2019. In accordance with University regulations, **marks are deducted** from any coursework which is not submitted by the deadline. However, if an extension is given then the rule applies from the date of the extension.

3 Plagiarism

According to university rules, instances of plagiarism will be treated very seriously. **Penalties** are in line with the institutional framework of the University. It is natural that students will sometimes exchange ideas and help each other out in completing coursework of this nature. However, please be mindful that flagrant instances of copying will result in both parties receiving zero.

Table 1: Version Control System

Version	Date	Author	Changes
1.0	23/02/19	Ahmed Kheiri	-
2.0	27/02/19	Ahmed Kheiri	math package is allowed.
2.1	28/02/19	Ahmed Kheiri	clarify how distance is calculated