Sheffield
Hallam
University

College of Business, Technology and Engineering (BTE)

Programming Concepts and Practice

Academic Year: 2022/2023, Semester 2

Assignment 1

Module Leader: Abayomi Otebolaku, PhD		Level: 7
Module Name: Programming Concepts and Practice		Module Code: 55-706555
Assignment Title: PCP Assignment 1		
Individual piece of work.	Weighting: 60%	Magnitude: *wordcount: 2500*
Submission date/time: 17-April-2023, 14:59	Blackboard submission: Yes Turnitin submission: Yes	Format: source code, digital media, report.
Planned feedback date: 8-05-2023	Mode of feedback:Written and verbal feedback during face-to-face demo.	In-module retrieval available: Yes, for those whose marks fall between[45%-49%].

Module Learning Outcomes

- LO1: select appropriate programming techniques and data structures to develop effective software implementations of relatively complex systems using an appropriate programming language
- LO2 apply relevant program design strategies to the implementation of software applications using that programming language.
- LO3: design and implement well-engineered, domain specific software using that programming language

Sheffield
Hallam
University

College of Business, Technology and Engineering (BTE)

Assignment Brief

1. Introduction

In the electronic world, fraud is quite prevalent. Fraudulent transactions can be found in insurance, Internet auctions, benefit claims, and telecommunication services. When you engage in any online based transactions, there is high probability that you are dealing with a fraudster. Thus, the ability to identify when any entity you are engaging with online is not playing by the rules is very important.

Considering the enormous challenges posed by online transactions, you have been hired as a data scientist by a very large bank that issues credit cards.

One of your job functions is to develop a system that uses purchasing transaction data of the bank customers to detect fraudulent behaviours very quickly, possibly in real-time in such a way that appropriate mechanisms for protecting the bank's clients can be activated.

In this assignment, you are provided with datasets consisting of fraudulent and non-fraudulent purchasing transactions. Specifically, you will design and implement the system by analysing the problem of fraud detection systems, designing, and implementing a solution based on the concepts and principles taught in this module.

2. Key Tasks

The following tasks are to be performed in this assignment:

a) **Processing datasets**

In this task, you will develop a suitable module for retrieving data from the provided datasets. The data to be retrieved is a nested dictionary containing user transactions, i.e., users and transactions they have performed (including the location and amount of money spent in that transaction). You are provided with sample datasets, and the information regarding these datasets is in section 4. In module the function for retrieving the required data must handle and address all possible errors and exceptions.

b) **Computing distance between any two transactions of any user**

You will design and develop function for computing distance between any two transaction of any user and another function for computing distance between transactions of any two given users. See section 5 for further details.

c) Computing user transaction statistics

In this task, you will design and implement 12 functions that use the data retrieved in task(a) to compute basic statistics on transactions of any user or of all users. Details of those functions are provided in section 5.

3. Report

You will write a report of your implementation, summarising your implementation processes, decisions and justifications, and detailed instructions on how to execute your application. Your report should also contain other details of your program e.g., design/pseudocode/algorithms for your functions/system. In addition, your report should include a reflection section of your experience while executing this project. The reflection should detail what went well or not and lesson learnt. What would you do differently if you have another chance to execute this project again?

4. Datasets

In the folder (transactions), you will find three datasets. You are required to extract data from relevant dataset(s), namely the *transaction.txt, fraud-description.txt, and descriptions.txt.*

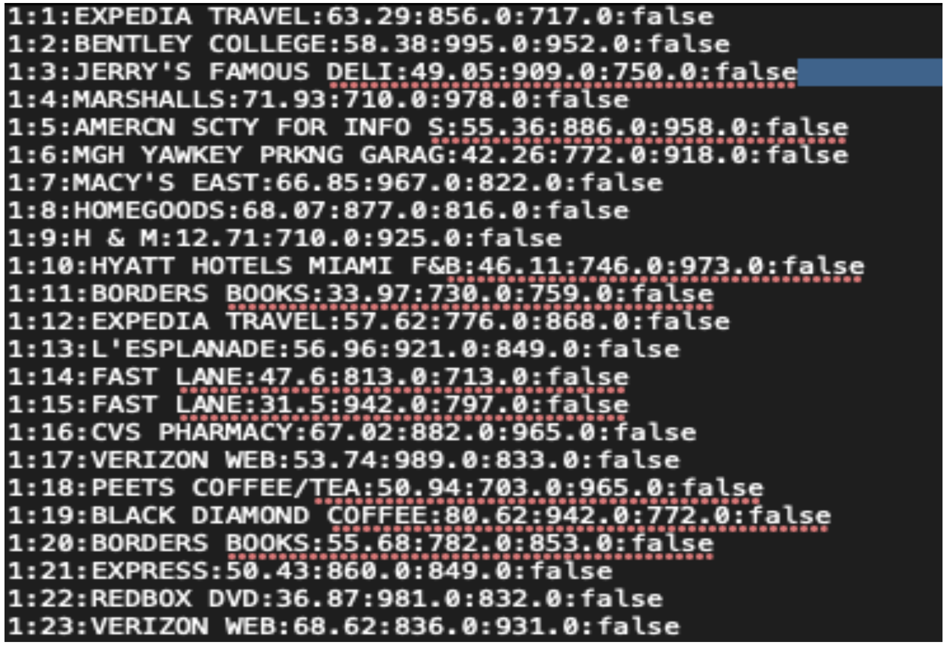
Three datasets are provided. The first dataset(description.txt) contains the description of genuine transactions. The second dataset(fraudulent-transaction.txt) contains the description of fraudulent transactions. The last dataset(transaction.txt) contains the actual transactions(10,000). The transaction dataset contains the description of the transactions, the amount of the transactions and the locations of the transactions. The locations are not real location to avoid tracing any transaction to anyone. The locations are just Euclidean coordinates in x, and y points.

Note that you may not need the three datasets for this assignment. The most important dataset is the ***transaction.txt***

The following are the attributes of the transaction dataset, also see Figure 1.

1. The user id
2. The transaction id
3. The description of the transaction
4. The amount of the transaction
5. The x coordinate of each transaction
6. The y coordinate of each transaction
7. A Boolean label that represents whether the transaction is fraudulent or not.

Figure 1: sample data from the transaction.txt



```
1:1:EXPEDIA TRAVEL:63.29:856.0:717.0:false
1:2:BENTLEY COLLEGE:58.38:995.0:952.0:false
1:3:JERRY'S FAMOUS DELI:49.05:909.0:750.0:false
1:4:MARSHALLS:71.93:710.0:978.0:false
1:5:AMERCN SCTY FOR INFO S:55.36:886.0:958.0:false
1:6:MGH YAWKEY PRKNG GARAG:42.26:772.0:918.0:false
1:7:MACY'S EAST:66.85:967.0:822.0:false
1:8:HOMEGOODS:68.07:877.0:816.0:false
1:9:H & M:12.71:710.0:925.0:false
1:10:HYATT HOTELS MIAMI F&B:46.11:746.0:973.0:false
1:11:BORDERS BOOKS:33.97:730.0:759.0:false
1:12:EXPEDIA TRAVEL:57.62:776.0:868.0:false
1:13:L'ESPLANADE:56.96:921.0:849.0:false
1:14:FAST LANE:47.6:813.0:713.0:false
1:15:FAST LANE:31.5:942.0:797.0:false
1:16:CVS PHARMACY:67.02:882.0:965.0:false
1:17:VERIZON WEB:53.74:989.0:833.0:false
1:18:PEETS COFFEE/TEA:50.94:703.0:965.0:false
1:19:BLACK DIAMOND COFFEE:80.62:942.0:772.0:false
1:20:BORDERS BOOKS:55.68:782.0:853.0:false
1:21:EXPRESS:50.43:860.0:849.0:false
1:22:REDBOX DVD:36.87:981.0:832.0:false
1:23:VERIZON WEB:68.62:836.0:931.0:false
```

5. Implementations

You are expected to design and develop 4 modules namely: *dataset_module*, distance_module, *statistics_module and test_module*.

I. You will implement a ***dataset_module*** with a function that retrieves the above attributes/features and returns a dictionary. You must use python file objects. Please do not use any other libraries for this task. If you do, you will lose the marks for this task.

II. You will implement **distance_module** containing two functions.

a. A function that computes the distance between any two given transactions of a user.

b. And another function should be implemented for computing the distance of transactions of any two users.

III. You will implement a ***statistics_module***. This module will contain the following functions:

a) A function that returns the average transactions of any user and of all users.

b) A function that returns the mode of transactions of any user and that of all users.

c) A function that returns the median of all transactions of a user and that of all users.

d) A function that returns the interquartile range of any user's transactions and of all users.

e) A function that returns the location centroid of any user, based on their transaction locations.

f) A function that computes the standard deviation of any specific user's transactions.

g) A function that determines whether a transaction is fraudulent or not. It should provide details of such transactions.

h) A function that returns an abnormal transaction for any given user.

i) A function that computes the Z score of any user's transactions and for all users' transactions.

j) A function that computes those frequencies of transactions at any given location.

k) A function that returns the outlier of any location and of any user.

l) A function that returns the nth percentiles of transactions of any user and of all users.

IV. Implement the main module(test_module) containing the function that implements the user interface through which users can query and interact with all functions in II and III.

Note: in all your functions, make sure you handle all possible errors/exceptions.

6. Pay attention to the following requirements!

a) This assignment is an individual piece of work, and your submission must be in the form of modules or Jupyter Notebook file. We should be able to open and run your modules on a standard campus computer.

b) You will submit a report. The report should provide analysis of the problem being solved, justification for your design decisions and pseudocodes for the statistical functions and other functions you have developed. It should explain the relationships between the modules. A good report should provide **evidence of critical analysis of the implemented system**. Even if your application does not work correctly, you should still submit the report explaining what you have done, what works and what has not worked. If you have any questions regarding the report, please consult with your tutors.

c) You are required to do a 15 minute-walkthrough of your deliverable. Date and time will be discussed during the assignment brief walkthrough.

d) Please note that any evidence of collusion/plagiarism will be penalised! If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to defend such work, including reasons for the programming decisions taken. You must document with references any use of libraries or existing code in your report.

e) Appropriate use of variable names for clearer understating of your code is important.

f) Adequate commenting of your code for easier understanding during demonstration/grading is also important.

7. Submission Process

- Your assignment should be submitted electronically through the module's Blackboard site as a single ZIP file that contains all source code and a report, not more than 5 pages. Your report should not be more than 5 pages. If it is more than 5 pages, you will lose marks.
- You will prepare for a face-to-face 15-minute demonstration/walkthrough of your deliverable, where you will be provided with verbal and written feedback.
- Check your upload to ensure you have submitted the correct files successfully as any issues will not be considered after the deadline.
  - Note that any wrong submission or submission of wrong file or deliverable will be graded as 0. There is no room for sending deliverable file via email.
  - Please ensure that your correct files are submitted as we will not entertain any request for mistaken submission after the deadline.
- **Note that late submission will attract penalty. The penalty is capping of your mark to 50% or 0 depending on how late the submission is.**
- **You must also check your report's similarity score using Turnitin on the Blackboard before final submission. Please do not submit any report with similarity score higher than 20%. Otherwise, you will be penalized for plagiarism or collusion.**

8. Assessment Criteria

This assignment assesses the module's learning outcomes (LO) in the following way:

- This assignment focuses on designing and implementing python a module for loading and extracting data from real-world datasets and a module for computing basic statistical functions, using appropriate data structures, string manipulation, functions, parameters, argument, iteration, selection, etc. (LO1) and programming design strategies (LO2).
- The coursework will be assessed against the Learning Outcomes (LOs) using a set of assessment criteria. This set of assessment criteria allows assessing how successful a student has met the LOs. In order to ensure consistent application of the relevant criteria, the assessment criteria are summarised in the following assessment matrix and grid. This is an indicator of how the marks will scale across each category of the learning outcomes it covers.

Note that the University's categorical grading descriptor (included in assessment folder) will be used to determine your grade in this assessment. The marking scheme embeds the concept of extended work by rewarding only the highest marks to those who demonstrate evidence of independent investigation, learning, and thought. Thus, to achieve top grades, you will need to go beyond the materials presented in lectures and labs and undertake some of your own research (i.e. read and discuss related materials).

Table 1: Assessment Matrix

Assignment	Assessment Criteria	Marks	Learning Outcomes		
			LO1	LO2	LO3
Assignment 1	Basic Understanding of programming concepts (/10)	10%	X	X	
	Application of programming processes to solving problems(/15)	15%	X		X
	Implementation of software solutions using appropriate programming concepts (/25)	25%	X	X	X
	Quality and usefulness of the submitted mini report (/10)	10%	X	X	

Table 2: Assessment Marking Grid

Fail (<50%)	Pass (50-59)	Merit (60-69)	Distinction (70% +)
Basic Understanding of programming concepts (/10)			
No evidence of understanding of problem being solved. No solution requirements.	Good evidence of understanding of the problem, its definition and analysis. Good understanding of important solution requirements, may lack some	Very good demonstration of knowledge and understanding of the problem. Very good understanding of the system requirements and its analysis.	Excellent and professional demonstration of understanding of the problem being solved. Excellent details of solution requirements and analysis of the programming problem.

	clarity, missing requirements, etc.		
Application of programming processes to solving problems(/15)			
No application of software development processes.	Good use of appropriate software development p[processes . Minor issues, such as lack of use of some software development processes such as missing pseudocodes or architectural figures/flowcharts	Very good use of appropriate software development processes. Use of pseudocodes, algorithms, flowcharts, use of figures e.g. high-level architectural diagram showing key aspects of your solution, etc.	Excellent and professional use of the program development processes.
Implementation of solution using relevant programming concepts (/25)			
Lack/poor use of programming concepts.	Good understanding and use of appropriate programming concepts such as selection, repetition, sequences, functions, and parameters, etc. Use of appropriate data types, data structures, etc.	Very good understanding and use of appropriate programming concepts such as selection, repetition, sequences, functions and parameters, exception handling, etc. Software does not crash when being run. Entering wrong input does	Excellent, professional level use of programming concepts in the implementation, robust application delivered. Application does not crash when wrong input is provided or when running it.

		not crash the program but gracefully handles such exceptions, etc.	
Quality and usefulness of mini-report and face-to-face demo (/10)			
Report lacking good structure, no personal reflection, no description of the deliverable or explanation and justification of decisions. Poor use of language, poor demo, not able to answer questions on the deliverable.	Good structure, evidence of personal reflection on what went well or not. Good justification for design and implementation decisions. Good use of language. Good demo, able to answer basic questions regarding the assessment criteria.	Very good structure, evidence of personal reflection on what went well or not. Very good justification for design and implementation decisions. Good use of language. Very good demonstration. Very good answers to relevant questions on the deliverable and assessment criteria.	Excellent structure, excellent personal reflection on what went well or not. Good justification for design and implementation decisions. Excellent use of language. Evidence of innovation in the deliverable, e.g., excellent user interaction through UI, etc. Excellent demonstration, excellent answers to questions during demo.

Note that your final grade will follow the new University grade descriptor, which is included in the assessment folder.

****

All work must be yours. If evidence of collusion/copying is found, then ***such collusion will be penalised, severely if appropriate!*** *If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to give a detailed explanation of such work, including reasons for the programming decisions taken.*

****

You must prepare for a demonstration /walkthrough (15 minutes) to defend and speak to your running prototype system and explaining how it meets the assessment criteria. Please look at the marking criteria and prepare for your demonstration accordingly. You will receive notification regarding your face-to-face walkthrough/demo.

****

WARNING

All work must be yours. If evidence of collusion/copying/plagiarism is found, then such collusion/plagiarism will be penalised severely if appropriate! If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to give a detailed explanation of such work, including reasons for the programming decisions taken.