Sheffield Hallam University

College of Business, Technology and Engineering (BTE)

Programming Concepts and Practice

Academic Year: 2022/2023, Semester 1

ASSIGNMENT 2 Assessment

Module Leader: Dr. Abayomi Otebolaku		Level: 7
Module: Programming Concepts and Practice		Module Code: 55-706555
Assignment Title: PCP Assignment 2		
Academic Year: 2022/2023		
This is an individual task. There should be no collusion or collaboration whilst working on and subsequently submitting the assignment.		
Individual	Weighting: 40%	Wordcount: 1500
Submission date/time: 23-May-2023, 14:59	Blackboard submission: Yes Turnitin submission: Yes	Format: source code, digital media, report.
Planned feedback date: 13-June-2023	Mode of feedback: Written and verbal	In-module retrieval available: No

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

1. Introduction

Activity context recognition is one of the core functions of context-aware computing that deals with automatic determination and inference of contextual information from a set of observations captured by sensors. It enables developing applications that can respond and adapt to user's situations. Thus, developing innovative context recognition capabilities into context-aware systems has received significant attention both in the industry and academia. This has allowed the proliferation of various intelligent applications, such as those for remote health and lifestyle monitoring, intelligent personalized services, etc.

A fitness company has hired you to develop an intelligent model for their mobile fitness application that can automatically recognise people's activities. The company has provided you with labelled historical activity context data of individuals who participated in a data collection process. The data was captured from smartphone built-in sensors namely: orientation, rotation, accelerometer, gyroscope, magnetic, sound, and light sensors.

However, this dataset contains low-level data and thus it is difficult to generalize inference from the data. Extracting meaningful information from the data before feeding it to machine learning algorithms for context recognition purposes can be done using statistical feature extraction methods.

In this assignment, your task is to deliver the activity recognition model of the fitness system, working with the activity context tracking dataset to analyse, design, implement, and evaluate the model.

For this purpose, you will make use of programming concepts such as the use of custom module, function definitions, Object Oriented Programming (OOP) concepts, file processing, and exception handling, use of scientific computing, data analysis, data visualization, and machine learning libraries (such as NumPy, Pandas, Matplotlib, and Scikitlearn) in the implementation.

2. Assignment Key Tasks

The following tasks are to be performed in this assignment:

a) Exploratory Data Analysis

You are required to write code to load the dataset and explore the data, checking missing data points, (and applying appropriate cleaning techniques). All necessary data preprocessing must be executed via relevant python libraries and functions. The EDA should be implemented as a module. The module should also include descriptive statistical analysis of the dataset (such as mean, median, standard

deviation, variance, minimum, maximum, skewness and kurtosis): choose a range of the variables of your interest, find their frequencies and dependencies through bar plots, grouped bar plots, pie-charts, etc. Interpret and report your conclusions.

In addition, explore the dataset to determine if the classes are balanced or not by producing the plot of the class distribution. If the classes are not balanced, use at least 1 relevant technique to address the class distribution issue. Finally, you will split the cleaned dataset into training and test datasets in preparation for training machine learning algorithms.

b). Activity Classification

In this task, you will build classification models by training a set of algorithms using the dataset. You will build minimum of **three** classification models (e.g., Support vector machine, Random Forest classifier, Multi-Layer Perceptron Neural Networks, etc.). You will then evaluate your models using the test dataset and provide the confusion matrix for all models. In addition, report and compare performance of the models in terms of accuracy, precision, recall, and F1-Score. Draw conclusions and provide recommendations.

c) Optional Feature computation using statistical analysis.

In this task, you may optionally (as an extension) design and implement 8 functions for computing statistical features from the dataset in section (3). See table 2 on weight allocated to this optional task and others.

Table 1: Statistical Features

Features	Formula
Variance (Var)	$var = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)$
Median (Med)	$median = l + \left[\frac{\frac{N}{2} - \sum f_0}{f_w}\right] i$
Mean	$mean(\mu) = \frac{1}{n}\sum_{i=1}^{n} x_i$
Standard Deviation (STD)	$STD(\delta) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$
Root Mean Square	$RMS = \sqrt{\frac{x_1^2 + x_2^2 \ldots, x_n^2}{n}}$

Zero Crossing (ZC)	$Zcr = \frac{i}{X-1}\sum_{x=1}^{X-1}\{X_i X_{i-1} < 0\}$
Sum of Squares (SOS)	$SOS = \sum_{i=1}^{n}(x_i - \bar{x})^2$
Covariance	$\sum(x_i - \bar{x})(y_i - \bar{y}) \Big/ (N-1)$

d) Mini-Report

You will write a report of your implementation, summarising your programming processes, implementation decisions, justifications, and pseducode/algorithms for your functions/system. Provide an explanation in your mini report on how to execute your application. The report should not exceed 5 pages. If the main body of your report exceeds 5 pages, there will be a penalty.

3. Sample Dataset

In the folder (data) you will find the dataset "activity_context_tracking_data.csv" containing features as shown in Figure 1.

The dataset contains data from 7 sensors (see section 1). 5 of the sensors have 3 columns representing x, y, z axes of the sensors. Sound and light sensors have only one axis each. Please use all sensing data in your EDA and algorithms training.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	_id	orX	orY	orZ	rX	rY	rZ	accX	accY	accZ	gX	gY	gZ	mX	mY	mZ	lux	soundLevel	activity
2	1	125	-17	2	0.0709966	-0.131696	-0.877469	-0.0383072	2.68151	8.65743	-0.0413156	2.67655	8.64271	-31.2	-35.6	-37.6	5000	49.56	Sitting
3	2	126	-17	2	0.0714863	-0.13148	-0.878024	-0.0383072	2.68151	8.65743	-0.0541957	2.67834	8.64654	-31.2	-36	-37.2	5000	53.38	Sitting
4	3	127	-17	2	0.0714005	-0.131551	-0.878799	0.153229	2.68151	8.65743	-0.0568672	2.68004	8.65088	-31.2	-36	-37.2	5000	53.38	Sitting
5	4	127	-17	2	0.0714005	-0.131551	-0.878799	0.153229	2.68151	8.65743	-0.0568672	2.68004	8.65088	-31.2	-36	-37.2	5000	49.53	Sitting
6	5	127	-17	2	0.0707716	-0.131888	-0.879645	0.153229	2.68151	8.65743	-0.0491283	2.6813	8.65458	-31.2	-35.6	-36.8	5000	49.53	Sitting
7	6	127	-17	2	0.0707716	-0.131888	-0.879645	0.153229	2.68151	8.65743	-0.0491283	2.6813	8.65458	-31.2	-35.6	-36.8	5000	49.53	Sitting
8	7	127	-17	2	0.0691724	-0.132711	-0.881449	0.153229	2.68151	8.65743	-0.0276463	2.682	8.65709	-31.2	-35.6	-36.8	5000	50.25	Sitting
9	8	127	-17	2	0.0691724	-0.132711	-0.881449	0.153229	2.68151	8.65743	-0.0276463	2.682	8.65709	-31.2	-35.6	-36.8	5000	50.25	Sitting
10	9	127	-17	2	0.0671819	-0.133712	-0.882605	0.153229	2.68151	8.65743	0.00535765	2.68224	8.65839	-31.2	-35.6	-36.4	5000	50.25	Sitting
11	10	127	-17	2	0.0671819	-0.133712	-0.882605	0.153229	2.68151	8.65743	0.00535765	2.68224	8.65839	-31.2	-35.6	-36.4	5000	50.25	Sitting
12	11	127	-17	2	0.0648748	-0.134859	-0.883693	0.153229	2.68151	8.65743	0.0449671	2.6822	8.65882	-31.6	-36	-36.4	5000	51.59	Sitting
13	12	127	-17	2	0.0648748	-0.134859	-0.883693	0.153229	2.68151	8.65743	0.0449671	2.6822	8.65882	-31.6	-36	-36.4	5000	51.59	Sitting
14	13	127	-17	2	0.0648748	-0.134859	-0.883693	0.153229	2.68151	8.65743	0.0449671	2.6822	8.65882	-31.6	-36	-36.8	5000	51.59	Sitting
15	14	127	-17	2	0.0625757	-0.135989	-0.884615	0.153229	2.68151	8.65743	0.085264	2.68203	8.65871	-31.6	-36	-36.8	5000	49.93	Sitting
16	15	127	-17	2	0.0625757	-0.135989	-0.884615	0.153229	2.68151	8.65743	0.085264	2.68203	8.65871	-31.6	-36	-36.8	5000	49.93	Sitting
17	16	127	-17	2	0.0625757	-0.135989	-0.884615	0.153229	2.68151	8.65743	0.085264	2.68203	8.65871	-31.6	-36	-36.8	5000	49.93	Sitting
18	17	127	-17	2	0.0625757	-0.135989	-0.884615	0.153229	2.68151	8.65743	0.085264	2.68203	8.65871	-31.6	-36	-36.8	5000	49.93	Sitting
19	18	127	-17	2	0.0605399	-0.13698	-0.885323	0.153229	2.68151	8.65743	0.121494	2.6818	8.65833	-31.6	-36	-36.8	5000	49.93	Sitting
20	19	127	-17	2	0.0605399	-0.13698	-0.885323	0.153229	2.68151	8.65743	0.121494	2.6818	8.65833	-31.6	-36	-36.8	5000	49.93	Sitting
21	20	127	-17	2	0.0605399	-0.13698	-0.885323	0.153229	2.68151	8.65743	0.121494	2.6818	8.65833	-31.6	-36	-37.2	5000	49.93	Sitting
22	21	127	-17	2	0.0591209	-0.137668	-0.885762	0.153229	2.68151	8.65743	0.147023	2.68163	8.65795	-31.6	-36	-37.2	5000	49.05	Sitting
23	22	127	-17	2	0.0591209	-0.137668	-0.885762	0.153229	2.68151	8.65743	0.147023	2.68163	8.65795	-31.6	-36	-37.2	5000	49.05	Sitting
24	23	127	-17	2	0.0591209	-0.137668	-0.885762	0.153229	2.68151	8.65743	0.147023	2.68163	8.65795	-32	-36	-37.2	5000	49.05	Sitting
25	24	127	-17	2	0.0583155	-0.138058	-0.885947	0.153229	2.68151	8.65743	0.161898	2.68151	8.65765	-32	-36	-37.2	5000	49.05	Sitting

Figure 1: samples from the dataset

4. Requirements

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

II. You are required to submit at least two python files. One of that should implement **a custom module** (.py files). An example could be a custom module (saved as EDA.py), which consists of functions/ objects defined to perform the descriptive statistics. Another could be the main. ipynb file, which consists of EDA module imported (to perform exploratory data analysis), and classification solutions.

III. You are required to submit a report (Ms word or PDF file, see 2(d)). The report should provide justifications for your analysis of the solution, design decisions and pseudocode. It should explain the relationships between the modules. A good report should be based on evidence with critical analysis of the implemented system and interpretation of model evaluation/validation results. 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?

IV. You are required to submit a video recording demonstrating the programming and machine learning concepts you have adopted. **Please explain the concepts in detail.**

V. Any evidence of collusion/plagiarism will be penalised 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 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.

VI. Appropriate and intuitive use of variable names for clearer understanding is extremely important.

VII. Adequate commenting of your codes for easier understanding during grading is also desirable.

5. Submission Process

I. Your assignment should be submitted electronically through the module's Blackboard site as a single ZIP file that contains **all your source code, video demonstration, and report**.

II. In addition, a copy of your report should be submitted on Turnitin.

III. If your video is longer than 15 minutes, we will stop watching it during grading at exactly 15th minute and this would result in loss of marks.

IV. Please check your upload to ensure you have submitted the correct files successfully as any issues will not be considered after the deadline. Submission of incorrect files will attract 0

marks.

V. Kindly provide a succinct explanation in your report on how to execute your application.

VI. 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.

VII. Your assignment must be submitted on or before Tuesday, 23rd May 2023 by 2:59 pm.

VIII. Note that late submission will attract penalty. The penalty is capping of your mark to 50%.

6. Suggested Structure of Your Report

i. Cover page with your name, student number and title.

ii. Introduction, which contains a short description of the context & method.

iii. Answers on the stated questions should be **succinctly presented.** For example, approach, and result/findings.

iv. All evaluation results should be presented in a table (screenshots of Python result in Appendix)

v. All plots, figures and graphs must be numbered and clearly labelled.

vi. Provide conclusion and recommendations.

vii. Please include a reference list, especially, to all libraries used in the programming. You are required to use the APA style of referencing.

8. Assessment Criteria

This assignment will be assessed through the report, testing of implementation and video demonstration of the submitted deliverable. The video demo should show how your solution meets the assessment criteria. In general, 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 you have met the LOs. In order to ensure consistent use 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.

Table 2: Assessment Matrix

Assessment 2 Criteria	Weights	Learning Outcome Covered
Clear understanding of relevant programming concepts such as Object-Oriented Programming concepts of classes, objects, methods, inheritance, custom module, functions, parameters, and argument, etc.	10%	LO1, LO2, LO3
Use of python (data science) libraries for exploratory data analysis (EDA)	10%	LO3
Use of python machine learning library training and fitting predictive models.	10%	LO3
Subject knowledge/Video Demonstration/Report/Deliverable.	5%	LO2, LO3
Optional Statistical features and GUI	5%	LO1, LO2, LO3

Note that the University's new grading descriptor (attached) will be used to determine your marks as shown in Table 3. 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 3: Assessment Marking Grid

Fail (<50%)	Pass (50-59)	Merit (60-69)	Distinction (70% +)
Understanding of relevant programming concepts (/10)			
No evidence of understanding and use of OOP (classes, objects, methods, inheritance, etc.), function definition, parameter and argument passing. Nothing is submitted.	Evidence of clear and consistent understanding of the OOP (classes, objects, methods, inheritance, etc.), function definition, parameters, and argument passing. Evidence of practical solution.	Very good and appropriate OOP (classes, objects, methods, inheritance, etc.), definition of functions, parameters and argument passing.	Exceptional understanding and creative use of programming solutions. OOP (classes, objects, methods, inheritance, etc.), function definition, parameter passing etc. Robust implementation of deliverable.

Use of python libraries for exploratory data analysis (EDA) (/10)			
No evidence of the use of Python libraries such as numpy, pandas, matplotlib, seaborn etc. Not able to apply appropriate python libraries. No submission.	Clear and good evidence of the use and application of python libraries (such as numPy, pandas, matplotib and seaborn) with some correct and expected outputs. But some minor issues with outputs.	Very good understanding and good implementation using python libraries to implement some of the functionality of the system with justifications and correct outputs. Program executes and produces expected	Excellent understanding and implementation of relevant python libraries, such as numPy , pandas, matplotib and scikit-learn with outstanding results. excellent user interaction using menu-driven interface or an optional GUI, etc.
Use of python (data science) libraries for building predictive models (/15)			
No evidence of the use of Python libraries such as numpy, pandas, especially scikilearn. Not able to apply appropriate python libraries. Limited or no interpretation of evaluation results. No submission	Clear and good evidence of the use and application of python libraries (such as numPy, pandas, matplotib and scikit-learn) with some correct and expected outputs. But some minor issues with outputs. Good interpretation of evaluation results.	Very good understanding and good implementation using python libraries to implement some of the functionality of the system with justifications and correct outputs. Program executes and produces expected. Very good interpretation of evaluation results.	Excellent understanding and implementation of relevant python libraries, such as numPy , pandas, matplotib and scikit-learn with outstanding results. excellent user interaction through GUI, etc. Excellent interpretations of evaluation results.
Knowledge of subject//Video Demonstration/Report/Deliverable (/5)			
Report lacking good structure, no personal reflection, no description of the deliverable or explanation and justification of decisions. Poor use of language. No/Poor video demonstration.	Good structure, evidence of personal reflection on what went well or not. Good recommendations were made. Good justification for design and implementation decisions. Good use of language. Good video demonstration.	Very good structure, evidence of personal reflection on what went well or not. Very good recommendations were made. Very good justification for design and implementation decisions. Very good video demonstration. Good use of language.	Excellent structure, excellent personal reflection on what went well or not. Excellent recommendations were made. Good justification for design and implementation decisions. Excellent video demonstration. Excellent use of language. Evidence of innovation in the deliverable, e.g., excellent user interaction using user friendly menu-driven user interface or an optional GUI, etc.

****

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 submit **a demonstration video (not more than 15 minutes)** showing your running prototype system and explaining how it meets the assessment criteria. Please look at the marking criteria and prepare your demonstration accordingly.

****

WARNING

All work must be yours. If evidence of collusion/copying/plagiarism 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