Department of Computer Science
Faculty of ICT
University of Malta

Dr Jean-Paul Ebejer
jean.p.ebejer@um.edu.mt
Version 1.2 (November 15, 2019)

CPS3235 Data Science: From Data to Knowledge

Study-Unit Assignment

Monday, 1st November 2019

Necessary Preamble: This document describes the assignment for study-unit *CPS3235 Data Science: From Data to Knowledge*. This assignment is worth **100%** of the total, final mark for this unit. You are expected to allocate approximately 75 hours to complete the assignment. You will be required to demonstrate (and be able to discuss) your working solutions in a 15-minute demo. The deadline for this assignment is **Monday, 20th of January, 2020 at noon**. Late submissions will **not** be accepted. Questions regarding the assignment should **only** be posted in the Assignment VLE forum (and not via personal correspondence with the lecturer of this study-unit).

This is an individual assignment. Under **no** circumstances are you allowed to share the design and/or code of your implementation. You may **not** copy code from internet sources, you will be heavily penalized if you do so. The Department of Computer Science, the Faculty of ICT, and the University of Malta take a very serious view on plagiarism. For more details refer to plagiarism section of the UM website[1].

1 Deliverables

You are to upload all of your code and documentation on the VLE website (submission via email will not be accepted). The following deliverables are expected by the specified deadline. Failure to submit any of these artifacts in the required format will result in your assignment not being graded. Only two files are required for electronic submission. Replace `NAME` and `SURNAME` with your name and surname respectively (doh!). Replace `IDCARD` with your national identity card number (without brackets), e.g. `123400G`. If you are a visiting student, and hence have no Maltese national identity card number, use your passport number instead.

[1] https://www.um.edu.mt/itservices/vle/pds/students/resources

- **201920_CPS3235_SURNAME_NAME_IDCARD_assignment_code.zip -** A zip file containing your assignment code. This needs to be uploaded to VLE. Each task should be located in a top level directory in the `.zip` file named `task1`, `task2`, and `task3`. It is your responsibility to make sure that this archive file has uploaded to VLE correctly (by downloading and testing it). Failure in opening the `.zip` file will result in your assignment not being graded. Your code organization should reflect the tasks' sub-sectioning.
- **201920_CPS3235_SURNAME_NAME_IDCARD_assignment_doc.pdf -** The assignment documentation in `.pdf` format. The documentation has to be uploaded to VLE, together with your code. A hard-copy should be submitted to the secretary's office (Mr Kevin Cortis) at the Computer Science department by the stipulated deadline. The report should **not** be longer than 20 pages (including figures and references) and it should be sectioned according to the tasks' questions. The report should also contain textual answers to any questions asked in the task's description (these should be answered explicitly). It should **not** include code listings or implementation details (these should be discussed as code comments or markdown cells if using Jupyter Notebook). Code snippets are allowed if explicitly requested. It is encouraged you use the University of Malta LaTeX template[2] for your documentation (follow the FAQ entry for how to adapt it for an assignment).
- **Signed copy of the plagiarism form[3] -** This should be submitted to the secretary's office at the Department of Computer Science.

Note that you do not need include the original datasets in your submissions, but, rather, you are required to list a set of instructions describing how to reproduce your results given a directory containing the original data files.

2 Technical Specification

The code you supply will be run on Linux (distribution: Ubuntu 18.04.3 LTS). You are required to implement the tasks assigned using Python (version 3). This assignment will be assessed using the Anaconda Python environment file (`cps3235.yml`) made available on VLE. It is strongly recommended you make use of this environment. Common data science Python libraries such as `numpy`, `scipy`, `pandas`, `scikit-learn`, `nltk`, `matplotlib`, `seaborn`, and `py2neo` are installed in the environment. Please include a `README.txt` file with additional library requirements and installation instructions in the top level of your archive file submission. Use of Jupyter Notebooks is not only allowed, but also encouraged. Apart from the actual code, these notebooks

[2] https://github.com/jp-um/university_of_malta_LaTeX_dissertation_template

[3] https://www.um.edu.mt/__data/assets/pdf_file/0004/415156/PlagiarismDeclarationForm.pdf

should contain markdown cells describing your thought process. If you write stand-alone programs, you should include a brief section in your `README.txt` describing the main functionality, and directions on how to run these (*i.e.* command line arguments). **Note that you should not have any absolute paths hardcoded in your programs.**

3 Tasks

This assignment consists of three tasks. Note that for every task you are required to discuss/explain your answers in the documentation. **Points are awarded both for actual (correct) answers and methodology which got you there.** Each task specification is purposefully kept vague to allow you to explore the data and apply your own sound judgement on the methods to use. You may argue any position, but you are always required to back your argument with evidence (data).

3.1 Data Collection and Storage – Airline Tickets

Data collection is one of the most painful and time consuming processes in any Data Science project. In this task you are required to collect multiple plane ticket data and to determine when is the best time (how much in advance) to buy airline tickets. You can pick any daily flights, but the destination or origin of the flight must be located in Malta.

a) Collect about six weeks worth of plane ticket data (for a single adult on a one-way, direct journey). You can use either an API (*e.g.* skyscanner) or a web scraping library (*e.g.* beautifulsoup). Make sure that these flights run daily (not weekly!). An example of such a flight is KM100 from Malta to London. You are required to collect data on a daily basis (and not collect the next six weeks of data today). Store this data in an appropriate data store. Describe your implementation including timing of data collection (scheduling), the data storage aspect, the schema used, how you cleaned the data *etc.* **(15 marks)**

b) For a given flight path, plot the fluctuation of prices against the number of days to the flight departure. **(5 marks)**

c) Based on the data you collected, when is a good time to buy plane tickets? When was the ticket cheapest during the six weeks? Is this data corroborated by multiple routes? **(4 marks)**

d) Can you notice this trend for other flights in your dataset? Describe, statistically, how to test whether all the air journeys follow the same pricing trends? **(7 marks)**

(Total for Task One: 31 marks)

3.2 Visualization and Statistical Analyses – Chess Games

a) Kingbase is a database of 2.2 million chess games. This dataset is freely available for download[4] in Portable Game Notation (PGN) format. This is a text file format that contains each player's move (and related data) for multiple games.

   i) Using Python and its visualization libraries create a set of **four** visualizations which summarize and explain this database. Make sure to use a combination of different chart types. One of the chart types should be a heatmap. It is up to you to decide on the most interesting aspects to present, but an example of a visualization could be the most popular starting moves which lead to a win. Note that you can use existing Python libraries to read PGN files.

   (20 marks)

   ii) Describe and run a statistical analysis on the data. **(6 marks)**

b) Find a recent, bad visualization from the local media. Critically comment on why you think this is a terrible visualization. Using Python and its visualisation libraries reimplement the visualization, addressing its shortcomings.

   (5 marks)

(Total for Task Two: 31 marks)

[4] http://www.kingbase-chess.net/

3.3 Data Science Project – Dataset Analysis

Together with this specification you will be given an archive file, named `201920_CPS3235_data.tgz`, which contains a number of text files. These text files contain data from a real-world scenario. In this task you will be required to execute a fully-fledged data science project from start to finish. Make sure to explain and document every step of the way.

a) Discuss the many aspects of the data supplied (including, but certainly not limited to, source, collection, quality, quantity, timeliness, interval, correctness, domain etc.). **(5 marks)**

b) Select (and justify) features-of-interest in the data. Extract these to a CSV file. Describe how you handled outliers, missing data, imputed values, etc. **(5 marks)**

c) Make at least four (interesting) statements/conclusions about different aspects of the data. Supply proof to corroborate your statements. **(12 marks)**

d) Can you identify a correlation between two variables in the dataset? **(3 marks)**

e) Run and describe a statistical analyses on the data. **(7 marks)**

f) Build a predictive model. Describe any assumptions you have made in your model. Make sure to describe your training and testing sets. Also, how would you go about testing such a model? **(12 marks)**

(Total for Task Three: 44 marks)

4 Grading Criteria

The following criteria, described in Table 1, will be taken into consideration when grading your assignment.

Table 1: CPS3235 assignment grading criteria.

Overall Considerations	
Functionality	Completeness and adherence to tasks' specification. Also, correctness of solutions provided.
Understanding	Demonstrable and thorough understanding and application of data science concepts. You should make use of most of the ideas presented during lectures.
Evaluation	Ability to critically evaluate your work (shortcomings, assumptions, *etc.*).
Originality	Creativity and originality in solutions.
Documentation	Quality of solutions' documentation. Must be properly written and presented (*e.g.* good use of English, no loose papers, page numbers, table of contents, captions, references, *etc.*)
Coding Practices	Adherence to coding standards, consistency, readability, comments, (no) code warnings, code organization, use of version control, *etc.*
Environment	Use of Linux/Unix setup (including Anaconda) for development.

Any documented extra (cool) functionality will result in bonus points (over and above the 100 marks available in this assessment). Note that **not** submitting one (or more) of the tasks, will severely affect your overall mark.

The End