Cardiff School of Computer Science and Informatics

Module Code: CMT655
Module Title: Manipulating and Exploiting Data
Lecturer: Dr Luis Espinosa-Anke
Assessment Title: Course Portfolio
Assessment Number: 1
Date Set: February 14th 2020
Submission Date and Time: Friday May 7th at 9:30am.
Return Date: June 7th 2019

This assignment is worth **70%** of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:

https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

This coursework consists of a portfolio divided in two assessments.

> Assessment (1) consists of **a set of exercises**. The final deliverable consists of **one PDF file and three python (.py) scripts**.
>
> Assessment (2) consists in implementing a **machine-learning powered web service** where students implement database, data analysis and machine learning logic in the context of a Natural Language Processing (text classification) exercise. The deliverable is **a zip file with the application and a PDF with a written summary** (up to 2,000 words) describing solutions, design choices and a reflection on the main challenges and ethical considerations addressed during development.

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Assessment 1	**Compulsory**	One PDF (.pdf) file	assessment1_normalisation_[student number].pdf
Assessment 1	**Compulsory**	One Python (.py) script	assessment1_sql_[student number].py
Assessment 1	**Compulsory**	One Python (.py) script	assessment1_corpus2mysql_[student number].py
Assessment 1	**Compulsory**	One Python (.py) script	assessment1_clf_[student number].py
Assessment 2	**Compulsory**	One ZIP (.zip) file containing the webapp code	assessment2app_[student number].zip
Assessment 2	**Compulsory**	One PDF (.pdf) file for the reflective report	assessment2report_[student number].pdf

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

Assignment

In this portfolio, students demonstrate their familiarity with the topics covered in the module via two separate assessments.

Deliverable

Assessment 1

The deliverable for Assessment 1 consists of 4 separate files. First, a PDF file with answers to a theoretical DBMS question (see sect. **Assessment 1**). For the practical exercises, *four Python scripts* with starter code are provided. Students should adhere strictly to the submission format and only write code inside the functions provided (specifically, under the line `#your code here` in each function). See the scripts for further details.

Assessment 2

The deliverable for Assessment 2 will be a zip file containing the webapp code and a *requirements.txt* file with the dependencies the app requires. An optional README file with how to run the pipeline can be included. For example, a submission may have two parts. One, where a program converts raw text data into a MySQL/MongoDB database. And second, the flask app that “speaks to” the database created in the previous step.

Assessment 1

In Assessment 1, students solve three main types of challenges, and provide a sound justification in their own words of the decisions made for their solutions. These challenges are: (1) Relational database design; (2) SQL; and (3) Data analysis and machine learning.

The exercises are:

DATABASE SYSTEMS AND DATA ANALYSIS (20 Marks)

1. **DB normalisation (5%) |** `assessment1_normalisation_[student number].pdf`

   A job agency helps clients, who are looking for employment, to find new jobs. The agency stores data about their clients in an Excel spreadsheet. The information stored includes clients' personal and employment details. Personal details include first name and surname, DOB, gender, and phone number. The employment details include employer name, phone, employer's address, job title, start date, end date, reason for leaving, and final yearly salary. The agency also keeps record of a set of qualifications for each client. Qualification data includes awarding body, level and year. Currently, qualification data are stored in one cell. Each client is assigned to one consultant within the agency who negotiates with the client and helps to find a suitable employment. **Normalise the database up to 3NF and write a short justification of your design choices for each NFX → NFX+1 step.**

2. **SQL (2%) - Cardiff Drinks Database |** `assessment1_sql_[student number].py`

   The following two questions require you to connect to the **Cardiff Drinks** database, provided in Learning Central.

   JOIN STATEMENTS

   2.1 Retrieve all purchases which happened in 2015. Provide the results in a 4-column table: `order_date`, `client_name`, `total_bottles`, and `total_GBP_spent`.

   SUBQUERIES

   2.2 Write a query to obtain the name of the salesperson in each region with the largest amount of total gbp sales.

3. **MySQL text corpus (8%) |** `assessment1_corpus2mysql_[student number].py`

   3.1 (6%) - You are given a collection of **IMDB reviews classified as positive and negative** (in Learning Central). Turn this raw text data into a relational database

with the format below. The tables are: **corpus, dictionary and sentiment**, and the columns:

-*dictionary*: **wordID** (ID for the word) and **word** (the actual string).
-*sentiment_corpus*: **textID** (ID for the document); **sentimentValue** (sentiment value as per its IMDB dataset classification).
-*mentions*: **hitID** (unique identifier of the mention of word wordID in text textID in position *offset*), **textID** (ID for the document); **offset** (offset of the word in the document); and **wordID** (ID for the word).

You can (and probably should) pre-select a vocabulary based on some criterion (e.g., the 100 most frequent words only) to make your solution more manageable:

dictionary

wordID	word
10	movie
84	actor
5	great

sentiment_corpus

textID	sentimentValue
1	pos
2	pos
3	neg

mentions

hitID	textID	offset	wordID
1	1	112	10
2	1	120	84
3	2	15	5

3.2 (2%) - Use this database to write a query that retrieves the top 100 most frequent words, as well as whether each of them is more frequent in positive or negative documents. The resulting table should have 100 rows and the following four columns: **wordId, word, overall_frequency, more_frequent_as.**

4. **Machine Learning (5%) |** `assessment1_clf_[student number].py`

   Write a Python program that takes the **train portion** of the same IMDB dataset, trains a classifier, runs predictions on the **test portion** of the same IMDB dataset, and saves the results (the output of sklearn's **classification_report** method) to a file called 'results.txt' (in the same directory as where the script is located).

Assessment 2

1. **Web Application (50%) |** `assessment2app_[student number].zip`

In Assessment 2, the goal is to set up a web service based on Flask. Students are provided with starter text dataset, and a code skeleton for both data processing and setting up flask. Based on these skeletons, the task is to develop a machine learning-powered solution which:

→Reads in text data (e.g., user-provided or from an API), vectorizes it and applies predictions in a binary classification setting.

→ Then, these predictions are rendered to the user.

→ The user should be able to either (1) Provide feedback on the prediction, and if the prediction is correct, the app should populate its "user data" database; or (2) Obtain the *performance* of the app's model in a held-out test dataset (also provided).

→ The app should re-train its model with its newly acquired data given some criteria (e.g., every 10 instances), and store an updated version.

The app should have a minimal functionality (i.e., it should work and perform a minimal set of actions), but the students are encouraged to try different routes for improving their final deliverable (see Criteria for assessment for detailed mark allocation).

Learning Outcomes Assessed

This coursework covers the 7 LOs listed in the module description. Specifically:
Assessment 1: LO2, LO3, LO4 and LO7
Assessment 2: LO1, LO5, LO6 and LO7

Criteria for assessment

Credit will be awarded against the following criteria.

Assessment 1 (20 Marks) The main criteria for assessment in based on correctness of normalisation and justification (Q1) and performance (Q2, Q3 and Q4, i.e., the SQL/Python code works and returns the desired output).

Assessment 2 (50 Marks) There are four main items of assessment:

- **[1] Database creation (10 Marks)**: a pipeline is in place for converting noisy data (i.e., text) into a relational or non relational database. The choice of the database (MySQL or MongoDB) is up to the student, as marking criteria will be based on providing a principled justification of DB choice and design in the written report (see **[4]**). A solution worth of *distinction* would implement at least one of the following: (1) procedural sql, (2) normalised (and justified) db design, (3) a discussion on performance in the written report (with measurable tests).

- **[2] Web app (10 Marks)**. The requested web application should *work,* i.e.*,* it should minimally be able to take user text input, serve a prediction, take user feedback and store that feedback (if applicable) in a database. A solution worth of *distinction* would also update the app's machine learning model with user's feedback or any other source (an external API).

- **[3] Evaluation (5 Marks)**: Implementation and critical justification for a sound machine learning lifecycle with regards to model performance. The user should be able to request model performance on a held-out test set. A solution worth of *distinction* would be able to link model version and performance and store it in a dedicated table/collection in the database.

- **[4] A final report of up to 2,000 words (25 Marks)** summarizing the work carried out for this assessment, discussing challenges and problems encountered and the solutions implemented to overcome them. Also a critical reflection of how the deliverable could be improved in the future. The mark will be divided between 3 expected sections:
  [4.1] (10 Marks) Database creation (DB choice, design, etc.);
  [4.2] (10 Marks) User interaction (how text is handled and preprocessed, how user feedback is obtained, how re-training and evaluation is performed, and how the DB is updated);
  [4.3] (3 Marks) Machine Learning solution. A brief description of which model was used and why.
  [4.4] (2 Marks) Ethics and bias in data-driven solutions, where a student discusses potential legal, social and ethical issues arising from the creation and maintenance of the app.

The below table illustrates the marking criteria for the **programming** part of both assessments.

Criteria	Distinction (70-100%)	Merit (60-69%)	Pass (50-59%)	Fail (0-50%)
Functionality	**Fully working application** that demonstrates an **excellent** understanding of the assignment problem using a relevant approach.	**All required functionality** is met, and the applications are working, probably with some **minor errors**.	**Some of the functionality** (database and/or data analysis approach) developed, but with **incorrect output or major errors**.	**Faulty application** with wrong implementation and **wrong or non existent database**.

The below table illustrates the marking criteria for the **written report.**

Criteria	Distinction (70-100%)	Merit (60-69%)	Pass (50-59%)	Fail (0-50%)
Correctness and reflection.	**Excellent report, with clear 'what', 'how' and 'why' claims.** Critical reflection on strengths and limitations across all the stages of the pipeline and clear understanding of the underpinning technologies in each module.	**All required topics addressed sensibly**, i.e., the report adheres to the implementation details, and critically discusses strengths and weaknesses of each design choice and solution.	**All required topics addressed**, but with incorrect statements regarding the actual implementation . Design choices are discussed but poorly justified.	**Non existent** report or with major or **fundamental mistakes** or gaps.

The grade range is divided in:

- Distinction (70-100%)
- Merit (60-69%)
- Pass (50-59%)
- Fail (0-50)

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned between June 3rd and June 7th via Learning Central. There will be opportunity for individual feedback during an agreed time.

Feedback for this assignment will be useful for subsequent skills development, such as database design, SQL and NoSQL, data analysis, machine learning and client-motivated deliverables involving predictive analysis.