Aston University
BIRMINGHAM UK

College of Engineering & Physical Sciences
Assignment Brief

AM41DP Data Science Programming	Final Coursework
Philip Trevelyan	p.trevelyan@aston.ac.uk

Assignment Brief / Coursework Content:

- This coursework consists of a real world full data exploration activity, including exploratory data analysis, merging relational data, data pre-processing and clean-up. Freely available data sets from the real world will be provided, together with a set of specific objectives.

- Most of the coursework activities will feel familiar to students who have engaged with the lectures and tutorials. The activities will require some independent exploration and critical thinking, but the core concepts are covered in the lectures.

- The coursework must be completed individually. This does not mean that you cannot discuss with your peers or exchange ideas and insights, but only that your solution and report need to reflect your own thinking and understanding

- The coursework will assess all 4 Learning Outcomes of this module, using both **Python 3** and **SQL** for a variety of data science-related activities.

- The 4 Learning Outcomes are:

LO1 Modify data using Python libraries

LO2 Produce high-quality 2D and 3D data visualisations for exploratory data analysis

LO3 Design and evaluate machine-learning models using Python libraries

LO4 Create datasets from a database using SQL

COVID−19 (70%)

You will need to download the csv file ‘Covid_19.csv’ from Blackboard, which contains data from 3rd January 2020 to 27th September 2023 about COVID−19. There are 67 columns and 344832 rows. The data was obtained from the website:
https://github.com/owid/covid-19-data/tree/master/public/data

Write Python code in Jupyter to do parts (a)-(g) and SQL code to do part (h).

(a) Read in the csv file ‘Covid_19.csv’ to create a dataframe.
Remove the 21 columns named

‘weekly_icu_admissions’,
‘weekly_icu_admissions_per_million’,
‘weekly_hosp_admissions_per_million’,
‘reproduction_rate’,
‘icu_patients_per_million’,
‘hosp_patients_per_million’,
‘total_boosters_per_hundred’,
‘human_development_index’,
‘excess_mortality_cumulative_absolute’,
‘excess_mortality_cumulative’,
‘excess_mortality_cumulative_per_million’.
‘continent’,
‘icu_patients’,
‘male_smokers’,
‘female_smokers’,
‘hosp_patients’,
‘total_boosters’,
‘extreme_poverty’,
‘tests_units’,
‘weekly_hosp_admissions’,
‘excess_mortality’

Remove rows where the ‘iso_code’ equals any of these 19

‘OWID_AFR’, ‘OWID_ASI’, ‘OWID_ENG’, ‘OWID_EUR’,
‘OWID_EUN’, ‘OWID_HIC’, ‘OWID_KOS’, ‘OWID_LIC’,
‘OWID_LMC’, ‘OWID_NAM’, ‘OWID_CYN’, ‘OWID_NIR’,
‘OWID_OCE’, ‘OWID_SCT’, ‘OWID_SAM’, ‘OWID_UMC’,
‘OWID_WLS’, ‘OWID_WRL’, ‘RUS’.

Remove rows where the population is less than 40000000. Without using any built-in functions: create a function to convert a date in the format YYYY-MM-DD (Year, Month, Day) into a single number representing the number of days from 1st January 2020 to the date given. Ensure that the date 1st January 2020 gives an output of 1. Apply your function to the date column of your data frame and put the output in a new column called ‘Days’.
(10 marks)

(b) Create a list containing the countries that remain. Using **Matplotlib**, for each country in your list, do the following:

Let p equal the maximum number of ‘new_cases_smoothed’ divided by the maximum number of ‘new_deaths_smoothed’. Overlay a plot of ‘new_deaths_smoothed’ against ‘Days’ with a plot of ‘new_cases_smoothed’ divide by p. Label the graph, and include the location in the title. Include a legend which gives p rounded to 1 decimal place.

Pick a country and comment on how the ‘new_deaths_smoothed’ and ‘new_cases_smoothed’ compare with each other. (6 marks)

(c) Using **Matplotlib**, for each country in your list, do the following:

Let q equal the median number of ‘total_cases’ divided by the median number of ‘total_deaths’. Overlay a plot of ‘total_deaths’ against ‘Days’ with a plot of ‘total_cases’ divide by q. Label the graph, and include the location in the title. Include a legend which gives q rounded to 1 decimal place. (4 marks)

(d) Obtain the correlation matrix of the data frame. Determine any correlations greater than 0.98 or less than -0.6. Discuss some strong correlations. (4 marks)

(e) Remove the columns ‘location’ and ‘date’. Create a model to predict the ‘life_expectancy’ using the following 8 columns:

‘population_density’, ‘aged_65_older’, ‘aged_70_older’,
‘cardiovasc_death_rate’, ‘median_age’, ‘gdp_per_capita’
‘diabetes_prevalence’, ‘population’.

You should split the data into (80%) training data and (20%) test data. Use an appropriate **linear regression** model from **sklearn** to predict the ‘life_expectancy’. Test your model using the test data set. Obtain the Mean Absolute Error, Mean Squared Error, and R-squared. First calculate them without a built in function, then calculate them using built-in functions. Create a scatter plot of the predicted values against the actual values. Discuss the accuracy of your model using your results. (12 marks)

(f) Make a copy of the dataframe. Create a model to predict the ‘iso_code’ using the 3 columns ‘total_cases’, ‘total_deaths’ and ‘Days’. Only use the following 12 ‘iso_code’s:

‘AFG’, ‘DZA’, ‘BGD’, ‘COL’, ‘EGY’, ‘ETH’,
‘IND’, ‘IDN’, ‘IRN’, ‘PAK’, ‘KOR’, ‘ZAF’.

Only use the rows where ‘total_cases’ and ‘total_deaths’ are greater than or equal to 1. You should split the data into (80%) training data and (20%) test data. Use an appropriate **classification** model from **sklearn** to predict the ‘iso_code’. Test your model using the test data set and obtain the accuracy. Create a scatter plot of the predicted ‘iso_codes’ against the actual ‘iso_codes’. Discuss the accuracy of your model using your results. (10 marks)

(g) You are going to predict the ‘iso_code’ using the 4 columns ‘total_cases’, ‘total_deaths’, ‘total_vaccinations’ and ‘Days’. Only use the following 12 ‘iso_code’s:

‘AFG’, ‘BRA’, ‘CHN’, ‘FRA’, ‘IND’, ‘ITA’,
‘MEX’, ‘PAK’, ‘ZAF’, ‘GBR’, ‘USA’, ‘VNM’.

Only use the rows where ‘total_cases’ and ‘total_deaths’ and ‘total_vaccinations’ are greater than or equal to 1. Normalise the 4 columns ‘total_cases’, ‘total_deaths’, ‘total_vaccinations’ and ‘Days’ using the max-min scalar. You now should split the data into (80%) training data and (20%) test data. Create any classification model you like using **PyTorch** to predict the ‘iso_code’; select an appropriate criterion, optimisation algorithm, and learning rate. Train the model and report the training accuracy. Create a scatter plot of the predicted ‘iso_codes’ against the actual ‘iso_codes’. Discuss the accuracy of your model using your results. (16 marks)

(h) Using **SQLite Studio**. Read in the csv file ‘Covid_19.csv’ to create a table. Remove the 21 columns named

‘weekly_icu_admissions’,
‘weekly_icu_admissions_per_million’,
‘weekly_hosp_admissions_per_million’,
‘reproduction_rate’,
‘icu_patients_per_million’,
‘hosp_patients_per_million’,
‘total_boosters_per_hundred’,
‘human_development_index’,
‘excess_mortality_cumulative_absolute’,
‘excess_mortality_cumulative’,
‘excess_mortality_cumulative_per_million’.
‘continent’,
‘icu_patients’,
‘male_smokers’,
‘female_smokers’,
‘hosp_patients’,
‘total_boosters’,
‘extreme_poverty’,
‘tests_units’,
‘weekly_hosp_admissions’,
‘excess_mortality’

Remove rows where the ‘iso_code’ equals any of these 20

‘OWID_AFR’, ‘OWID_ASI’, ‘OWID_ENG’, ‘OWID_EUR’,
‘OWID_EUN’, ‘OWID_HIC’, ‘OWID_KOS’, ‘OWID_LIC’,
‘OWID_LMC’, ‘OWID_NAM’, ‘OWID_CYN’, ‘OWID_NIR’,
‘OWID_OCE’, ‘OWID_SCT’, ‘OWID_SAM’, ‘OWID_UMC’,
‘OWID_WLS’, ‘OWID_WRL’, ‘OWID_WRL’, ‘RUS’.

Remove rows where the population is less than 40000000. Create a column called ‘Days’ which is defined as the final using the ‘date’ column where

$$\text{‘Days’} = 365 \times (\text{fourth digit in year}) + 30 \times (\text{month} - 1) + \text{day}.$$

(8 marks)

Other solution Guidelines:

Use both **Python** and **SQL** for this coursework. Please submit the Python parts using **Jupyter Notebook** (i.e. a file ending in ipynb) and submit the part in SQL as a text file. Carry out your CW solution process systematically, and add comments to explain your steps.

Marking:

The weighting for this coursework are as follows:

Description	Weight
COVID − 19	70%
Coding (correctness, efficiency and comments)	25%
Quality of report (presentation, description and graphics)	5%

Coding (25 marks)	Marks
Running (Does the code run?)	8
Accuracy (Is the code doing what it is supposed to?)	7
Efficiency (How fast is the code?)	6
Comments (Are the codes well commented?)	4

Don't forget to label the axes and give a title to each figure.

Key Dates:

Thursday 8th February 2024	Coursework set
2pm UK time Wednesday 20th March 2024	Submission date
TBC	Expected feedback return date (individual coursework feedback summary, available on Blackboard.)

Submission Details:

Coursework files must be submitted electronically on Blackboard, using the link available in the 'Assessment Submission' section. Late submissions will be given a 10% penalty per day.

Report submission guide:

- Please clearly indicate your name on the report.

- The report will be submitted through a specific link on Blackboard, which will be located under 'Assessment Submission'.

- You need to submit 2 programming files: a Jupyter Notebook (ipynb) file and a text file containing your commands in SQL. Each file name should follow the naming convention Lastname_Firstname_AM41DP_Report.

- The files must be able to be parsed assuming that the original CSV data files are available under the same folder as the report file.

- Reports submitted as any document format other than the ones above will be subject to a flat penalty of 5 marks. Do not compress your files.

- The total length of the report (excluding template section headers, cover sheet and references) must not exceed 50 pages in total. (You can do a 'print preview' to check). A 10 mark penalty will be given if the page count exceeds this.