CITS4009 Project 2 — Modelling

Worth 20% of the total assessment

Due: Friday, October 20th, 2023, 11:59pm

1. Project Preamble

This is a group project with at most two students in a group, contributing 20% to the final assessment of this unit.

- The group work is not compulsory, you can attempt this project on your own.
- If you prefer to work in pairs, and
    - **have a partner**, one member needs to register both of the member information in the group registration form (https://forms.office.com/r/Myda2UN5Gq).
    - prefer to be **randomly allocated to a group**, please fill in this Random Allocation EOI (https://forms.office.com/r/udq8qeYi3K) form.
    - if you worked in pairs, upon submission, you need to clearly indicate the percentage of contribution of each member to each task at the preamble of your R Markdown file, and include a digital signature of each member in the submitted file.

In this project, you are to demonstrate your understanding of how modelling is carried out using R machine learning functions.

The submission should be an HTML generated from your R Markdown file and a Shiny App.

2. Data

To demonstrate a full data science life cycle, you are strongly suggested to use the data and domain you have explored in Project 1:

- A specified dataset
- Data from public repositories

Specified data

If you do not have any datasets or domain of interests in mind, we suggest you to use the **YouTube** dataset as described in project 1.

Data from public repositories

Please refer to the links given in Project 1. You can choose a different public dataset if you prefer; however,

- this will require you to spend extra time to go through the EDA process to understand the new dataset;
- you will need to consult Wei or Sirui for permission as we need to check if all learning outcomes can be demonstrated by your chosen dataset.

3. Modelling

Classification

Firstly, study your dataset and choose the response (i.e., target) variable suitable for a classification task. The remaining variables can be your feature variables. You may need to discard some character string and categorical columns. Columns that have a unique value for each row should be discarded, e.g., `rank`, `Youtuber`, etc. If possible, formulate it as a binary classification problem (e.g. `high-earning` / `low-earning`), as multi-class classification is very difficult and is not covered in the lectures.

Your next step is to split the data into a training set and a test set. You can use any meaningful split ratio (90/10, 80/20, etc). You should implement R code for a **Descision Tree Classifier** and choose one different classification techniques to compare with (e.g., *logistic regression classifier*, *Naïve Bayes classifier*, *K nearest neighbours classifier*, etc.) and compare the performance of the two models. Your report should include discussions about attribute and feature selection and their impacts on the models. For example, you may have two attribute selection techniques, then you will build two models for each classifiers.

For example, using the *Youtuber* dataset, we can predict the level of earnings (high/low). You may make your own assumptions of high earnings or low earnings. For example, earnings against the social economic status of the country (e.g. normalised using GDP, or GDP per capita). The downloadable data from Word Bank is available at: https://data.worldbank.org/indicator/NY.GDP.PCAP.CD (https://data.worldbank.org/indicator/NY.GDP.PCAP.CD)

Note: depending on what predication question you'd like to be answered, you may need to tidy the data into the right shape or merging multiple datasets. You can start with single-variate models and then multi-variate models, following the same process demonstrated in the lecture slides. Highly correlated columns and semantic related columns will need to be removed (e.g. only keep one of the `earnings` column.

A good demonstration of the following investigations is expected:

1. Understanding of what a null model would look like in this context.
2. Aggregating, sub-setting, sampling or reshaping the data for better data preparation if necessary.
3. Transforming the categorical variables into numerical for single-variate model selection.
4. Using various measures to select a good combination of variables for multi-variate models.
5. Using **LIME** (see Labweek09) to find the determining feature(s) for the classification of a few instances from the test set. Discuss about your experimental findings in the report.
6. Evaluation of models. This step involves comparing your multi-variate models for the two implemented machine learning techniques on the training set and the test set using various measures (such as ROC plots, confusion matrices, deviances, etc).
7. Different techniques to generate combinations of variables. For example, information gain, principal component analysis, and forward selection. For bonus marks, you can also implement an attribute combination technique from reviewing literature.

Clustering

Choose or compute a set of feature variables; apply a clustering algorithm to these variables; visualise the clustering results and explain the rules discovered.

- Explain how and why you choose the distance measures and how the choices affect your clustering outcome.
- An investigation on the selection of k – the number of clusters.

4. Marking Criteria

- **Data preparation (10%):** Proficient use of data handling functions (e.g. pipes, merge/joins) or packages to construct clean and tidy training and testing datasets for classification. Sensible aggregation, transformation to obtain the right data for modelling, and good handling of missing data.
- **Classification (40%):** Good and thorough comparison of Decision Tree Classifiers against a different type of classification models with sensible interpretations of the performance measures, contextually with the right domain intuition. Good comparison of attribute or feature selection strategies.
- **Clustering (20%):** Good implementation of clustering with adequate investigations, demonstrations and explanation on the effect of hyper-parameters (e.g., k for the number of clusters) in these unsupervised techniques.
- **Shiny App (20%):** Sensible UI design to allow users to interactively viewing single variable model performance, two classification model performance and the clustering results.
- **Overall Report Quality (10%):** The report quality and the professionalism of video presentation will be considered. Treat this as presenting your final product to stakeholders and your client.
- To obtain marks in the HD range (80%-100%), you should aim at having the following elements in your project:
    - Well demonstrated exceptional understandings of the principles for model comparison and selection (NOTE: simply building a few more models will not automatically warrant marks in the HD range).
    - Thorough and effective treatment of data to improve model performance, e.g., imbalanced dataset treatment.
    - Exceptional data provenance for reproducible data science.
    - Exceptional understanding of feature variable treatment and selection.
    - Appropriate use of short text descriptions in diagrams (e.g., annotations).

5. Project Group Registration

To register to form a group with another student, enter both students' family names, given names, and student IDs on the form given in the **Project 2 Team Registration** tab on Teams Project Channel or through this direct link: https://forms.office.com/r/Myda2UN5Gq (https://forms.office.com/r/Myda2UN5Gq).

If you intend to work by yourself on the project, you don't need to do anything.

If you prefer to be randomly allocated to a group, please use this link: https://forms.office.com/r/udq8qeYi3K (https://forms.office.com/r/udq8qeYi3K)

6. Submission

1. Store the Shiny App in an `app.R` file;
2. Create a short 2-3 minutes video demonstrating the Shiny App, and provide a Youtube link to the video in the .Rmd file. Please show your face (or both of the group members' faces) in your presentation. You will also need to use your own voice to explain the best features of your App, no AI voices accepted.
3. Generate an html file with the name **project2.html** from your .Rmd file
4. Zip up the Shiny App file and .html file and submit it to **cssubmit (https://secure.csse.uwa.edu.au/run/cssubmit)**.

You should keep a record of your progressive work towards the final submission and also a copy of your latest work. You are encouraged to submit to *cssubmit* as often as you like (or need). The latest submission will overwrite the previous version.

If you like version controlling your work, then you can keep your working copies on GitHub (https://github.com/) before the final submission to cssubmit; however, do make sure you keep your GitHub repo **private**.

Submission Check List:

- Make sure that you have your name and student number clearly written at the beginning of your R markdown file. To make it easier for us in the marking process,
    - please ensure that you **have your student number written correctly**.
    - please ensure that you **use exactly the same name as shown on LMS**.
    - please put your **surname in uppercase letters**, e.g., Michael CHEN, John SMITH, Xiaolian HUANG.

- **If you have a project partner, then both students' names, student numbers, and percentages of contribution to the project** should be clearly shown. **NOTE: Only one submission is needed from each project group.**
- Submit the generated **.html** file (**not the `.Rmd` file or the `.nb.html` file**) and the Shiny App file. Before submission, check that you can view your **.html** file in a web browser outside Rstudio and that no diagrams are missing.

7. Special Considerations

Please make yourself familiar with the UWA special consideration process. All special considerations are handled centrally at UWA. Unit Coordinators are no longer processing individual request of project extensions. Please fill in the forms if needed:

https://www.uwa.edu.au/students/My-course/Exams-assessments-and-results/Special-consideration (https://www.uwa.edu.au/students/My-course/Exams-assessments-and-results/Special-consideration)

8. Penalty on Late Submissions

See the URL below about late submissions of assignments:
https://ipoint.uwa.edu.au/app/answers/detail/a_id/2711/~/consequences-for-late-assignment-submission (https://ipoint.uwa.edu.au/app/answers/detail/a_id/2711/~/consequences-for-late-assignment-submission)

For example, if you get 70 marks (out of the total 100 marks) before applying the late penalty and if your submission is *two days* late, then you get 60 as your final mark (i.e., 10% of the **total mark** - NOTE not your mark - is deducted).

9. Use of ChatGPT (or other Generative AI tools)

You are permitted to use ChatGPT or other Generative AI tools to generate code and help you learn. However, you must cite and explain how you used it. Generated code and documentations can be easily detected by human markers. To avoid plagiarism and academic misconduct investigations, please do provide citation and explain what changes you have to made to the code to demonstrate your own understanding. Here are a few UWA articles about how to cite and use generative AI in your assessment.

- Can I use ChatGPT and other AI tools in my assessments? (https://ipoint.uwa.edu.au/app/answers/detail/a_id/3432/related/1)
- How do I cite and reference ChatGPT (https://ipoint.uwa.edu.au/app/answers/detail/a_id/3434/~/how-to-cite-and-reference-chatgpt-and-other-generative-ai-tools-in-assessments)
- Guide for using AI Tools at UWA (https://www.uwa.edu.au/students/-/media/Project/UWA/UWA/Students/Docs/STUDYSmarter/Using-AI-Tools-at-UWA.pdf)

The library guide on writing academic papers (https://guides.library.uwa.edu.au/strategicpublishing/draftpapercheck) might also be relevant:

> For example, where relevant, Mathematics and Physical Sciences papers should discuss and reference the following as a minimum, to assign attribution and enable readers to re-create the outcome:

- Source (e.g. OpenAI/Microsoft, Anthropic/Google, NVIDIA etc.)
- Model (e.g. GPT 3)
- Implementation (e.g. davinci-003)
- Fine-tuning (Where the user has fine-tuned the ‘inbuilt’ knowledge of LLMs based on their own libraries of content via APIs or other processes).