Practical Data Science (with Python)

COSC2670/COSC2738

Assignment 2

Semester 2, 2024

Type	Individual assignment
Due Date	Sunday, 6 October 2024 by 23:59
Weighting	35%
Marking Criteria	See the course Canvas shell >> Assignments
Mark/Feedback	Will be available in Canvas

1. Introduction

This assignment focuses on *data modelling*, a core step in the data science process. You need to apply appropriate machine learning techniques (by using Python) to solve a series of data modelling problems and communicate your solutions and the results in a report.

2. General Requirements

Please read the following requirements and information carefully before you start.

- You must do the assignment in Jupyter Notebook.
- You must choose (by yourself) and use the *appropriate* Python functions (or methods) to complete the tasks.
- You must provide *sufficient (or detailed), precise* comments and/or annotations in your code, explaining what each (key) part does.
- You must use the provided template A2code-Template.ipynb to organise your code.
- Ensure that your submission follows the file naming rules specified in this document. Replace YourStudentNumber with your Student Number (e.g. s1234567) wherever applicable.
- Responsible use of AI tools is allowed. You must appropriately acknowledge and reference the use of any AI tools and their outputs. Failure to reference the use of these tools can result in academic misconduct. For more instructions, see: https://rmit.libguides.com/referencing_AI_tools
- Plagiarism is *never* okay. Relevant tools (like Turnitin) might be used for plagiarism check for this assignment.
- There are rules in place to support you uphold the academic integrity of RMIT. See: https://www.rmit.edu.au/students/my-course/assessment-results/academic-integrity

3. Tasks

In this assignment, you will use the dataset A2data.csv, which is available in Canvas under the *Assignments >> Assignment 2* section of the course Canvas shell. The dataset is related to attributes and quality of wine.

The inputs include physicochemical tests (e.g. PH values) and the output is wine **quality**, which scales between 0 (very bad) and 10 (very excellent). There are 12 variables (i.e., attributes or columns) and 4781 instances (i.e., entries or rows). For description of attributes, check the file Readme-A2data.txt.

Task 1: Regression

Load the CSV data from the provided file (using appropriate Python/Pandas functions). Take a random sample (i.e., subset) of 200 instances from the data set, and ensure that these instances don't have any missing values. Write the random sample into a CSV file and name the file as YourStudentNumber-A2SampleOne.csv. Select data for two variables: **alcohol** and **density**.

Display/show the relationship between these two variables in an appropriate graph (i.e., chart). In the report, describe any interesting relationships (or lack of relationships) that you observe from the visualisation.

Build a linear model (i.e., Simple Linear Regression) for the two variables, with **alcohol** being dependent variable and **density** as independent variable. Evaluate the model appropriately. Display the data points along with the linear model in an appropriate graph. In the report, present the linear model and interpret the coefficients of the model.

Task 2: Classification

Load the CSV data from the provided file (using appropriate Python/Pandas functions). Take a random sample (i.e., subset) of 500 instances from the data set, and ensure that these instances don't have any missing values. Write the random sample into a CSV file and name the file as YourStudentNumber-A2SampleTwo.csv. In this task, you will classify the wine **quality** based on the other variables.

Implement a kNN (k-Nearest Neighbours) classifier. Choose an appropriate value of *k* and justify your choice (in the report). Evaluate the classifier using appropriate metrics.
Propose a method to modify kNN for better performance. Note that simply tuning parameters will *not* be considered a modified version of the algorithm. Implement and evaluate the modified kNN algorithm in comparison to the (above/original) kNN classifier. Display relevant results (of evaluation metrics) for comparison in an appropriate graph. In the report, describe the proposed method in detail; summarise the results and interpret the findings. Explain (and justify) why the proposed method can lead to better performance in this case.

Implement a Decision Tree classifier. Tune the model by adjusting key parameters. Choose the best value(s) for the parameter(s) and justify your choice. Evaluate the performance of the Decision Tree model using the same metrics as for kNN. Compare the results of the Decision Tree classifier with those of the kNN classifier (not the *modified* kNN). Display relevant results (of evaluation metrics) for comparison in an appropriate graph. In the report, summarise the results and interpret the findings. Discuss the strengths and weaknesses of each model in the context of the given dataset.

Task 3: Clustering

Load the CSV data from the provided file (using appropriate Python/Pandas functions). Take a random sample (i.e., subset) of 300 instances from the data set, and ensure that these instances don't have any missing values. Write the random sample into a CSV file and name the file as YourStudentNumber-A2SampleThree.csv. In this task, you will conduct clustering upon all input variables (not using the output variable **quality** – it can be used for evaluation though).

Implement the k-Means algorithm. Evaluate its performance using multiple appropriate metrics. Explore the impact of the number of clusters (*k*) on the performance of the k-Means algorithm. In

the report, summarise the results and interpret the findings, and justify your choice. Discuss any limitations of k-Means you might observe in this case, and possible solutions.

Implement the DBSCAN algorithm. Choose appropriate values for the epsilon parameter (*Eps*) and the minimum number of points (*MinPts*); justify your choice. Evaluate its performance using multiple appropriate metrics. Analyse and compare the performance of k-Means and DBSCAN; display the results for comparison in an appropriate graph. In the report, summarise the results and interpret the findings, and explain why.

Task 4: Report / Presentation

Write your report using the template (A2report-Template.docx) provided in Canvas. Include all key results (e.g. graphs), required descriptions, explanations, and discussions (see the above three tasks), but *not* code. Your report must be at most 12 pages (including everything). For references, if any, use the APA 7th edition referencing style (see: https://www.lib.rmit.edu.au/easy-cite/).

4. Submission

The following files must be submitted (uploaded one by one, in one submission) in Canvas:

- **The Jupyter Notebook file** named as YourStudentNumber-A2code.ipynb: containing your Python code (and comments).
    - For the Jupyter Notebook code, please make sure to clean them and remove any unnecessary lines of code (cells). Comments are required. Follow these steps before submission:
        - Main menu → Kernel → Restart & Run All
        - Wait till you see the output displayed correctly. You should see all the data outputs printed and graphs displayed.
    - Note that you might obtain different data samples after each run. You may choose to do data sampling once and load the saved data sample files in later runs. It is okay if you comment the data sampling code to avoid unnecessary re-sampling in every run.
- **The written report file in PDF**, named as YourStudentNumber-A2report.pdf.
- **The three random data sample files** named
    - YourStudentNumber-A2SampleOne.csv
    - YourStudentNumber-A2SampleTwo.csv, and
    - YourStudentNumber-A2SampleThree.csv.

Resubmission without penalty is allowed until the deadline. Please do NOT submit any unnecessary files.