The University of Manchester
Alliance Manchester Business School

Coursework

BMAN73701
Programming in Python for Business Analytics
2024-2025

Lecturers: Xian Yang, Manuel Lopez-Ibanez
GTA: Qian Zhao

Last update: 21/10/2024

Due date: 2pm, 13th Dec 2024

Submission: The electronic version of your report (saved as a single .pdf file) is to be submitted through Blackboard/Turnitin.

Format: Conduct the coding and analysis using appropriate functionalities provided in Python (ideally version 3.10 or later). Then summarize your work using Microsoft Word. All main results have to be included in your report, as further detailed in the brief below. Keep your writing as brief as possible to around **4000** words, excluding figures, tables, references and appendixes. Figures and tables need to be numbered and must be referred to in the text. Copy and paste the Python code at the end of the report as an appendix. You can print the code to a PDF file, and then append the PDF file to your final PDF report file. There is no limit on the length of your code. For the report, use a minimum font size of Arial 11pt. For the code, you can use any font that is readable but fontsize should be also 11pt. Hand-written submissions will NOT be accepted.

While the brief below is broken up into questions/tasks, this is done to help you in guiding your analysis and understanding what tasks must be undertaken. It does not determine the organization of your report. The report should not quote the questions/tasks in the brief. It should be a coherent piece of work, where you describe the problem at hand and you explain the steps taken to analyse it and solve it. It should contain an introduction section and a conclusion section.

Assessment: The final coursework mark will count 70% to your overall module mark; the remaining 30% will be made up of grades you obtained in the two online tests. Ensure that the material presented in your report is correct and reproducible, and that the Python code is correct and readable. In terms of professional presentation, pay attention to things such as layout, font size, keeping to the page limit, spelling, grammar, use of tables and charts, consistent number of decimal places.

Each group MUST submit their report by the deadline indicated above. Late submission will be penalised in accordance with the School's guidelines on late submission (see course description). Each group MUST work on its own. Any evidence of plagiarism will result in a mark of zero.

Each group must keep a record of their meetings, in particular, date, meeting duration, who attended the meeting and the actions assigned to each member of the group after the meeting. This record should be added as an appendix to the group report. Insufficient contribution to groupwork will result in a mark of zero.

Questions: For any questions related to the coursework, please pose your question in **the Coursework discussion board** available on Blackboard. In addition to the discussion board, there will be a FAQ section on Blackboard with the most frequently-asked questions students have asked us and our answers to them.

Brief: The Acme Corporation must complete a *task* every day; these can be e.g. deliveries of goods, project management tasks, or recruitment of staff. For completing each task, Acme must choose one out of sixty-four (64) *suppliers* that provide the resources required to complete the task. The final *cost* of the task depends on how effective the chosen supplier is at performing the particular task. Unfortunately, estimating this cost in advance requires significant resources. Acme has hired your Business Analytics firm to develop a machine learning (ML) approach for selecting suppliers given a new task.

To complete your assignment, Acme has provided you with three datasets:

- '`tasks.csv`': a CSV file that contains one row per task and one column per task feature (TF1, TF2, TF3, …). Each task is uniquely identified by a Task ID (T1, T2, T3, …).
- '`suppliers.csv`': A CSV file that contains one column per supplier and one row per supplier feature (SF1, SF2, …, SF18). Each supplier is uniquely identified by a Supplier ID given in the first column of the file (S1, …, S64).
- '`costs.csv.zip`': a compressed CSV file that contains data collected and/or estimated by Acme about the cost of a task when performed by each supplier. Each row gives the cost value (in millions of dollars, M$) of one task performed by one supplier.

Your goal is to provide Acme with an ML model that, given the task features of a task, selects one supplier among the 64 suppliers available. The selection error made by the ML model will be measured as:

$$Error(t) = min\{c(s,t) | s \in S\} - c(s'_t, t) \tag{1}$$

where t is a task, S is the set of 64 suppliers, s_t' is the supplier chosen by the ML model for task t, and $c(s,t)$ is the cost if task t is performed by supplier s. That is, the Error is the difference in cost between the supplier selected by the ML model and the actual best supplier for this task.

Given a set of tasks for validation, we can assign a score to the ML model by computing the root mean squared error (RMSE) over the tasks:

$$Score = \sqrt{\frac{\sum_{t=1}^{T} Error(t)^2}{T}} \tag{2}$$

where T is the number of tasks and $\mathrm{Error(t)}$ is given in Eq. (1). This score should be minimised.

1. Data preparation: In the data preparation stage, please adopt a comprehensive approach, addressing the following areas:
   1.1. Handle any missing data by identifying and managing incomplete records or missing values to ensure the dataset is ready for further analysis.
   1.2. Perform relevant feature selection by determining which features are most important for your goal. Additionally, apply feature scaling to normalize the data and ensure all features contribute equally to the model, avoiding biases due to differing ranges.
   1.3. Focus on identifying the top-performing suppliers for each task by analysing the cost data. Since not all suppliers perform equally well, you should remove the worst performing suppliers from the dataset to make the following tasks more manageable.

2. Exploratory Data Analysis: (EDA): Consider using various EDA techniques to explore the distribution of feature values, cost data and supplier performance:
   2.1. Use EDA methods to analyse the distribution of task features and interpret emerging patterns.
   2.2. Analyse the cost data to identify patterns and good combinations of tasks and suppliers.
   2.3. Explore how errors (Eq. 1) are distributed across different choices of suppliers, identifying key trends or outliers in their performance. Calculate the RMSE (Eq .2) of each supplier for all tasks.

3. ML model fitting and scoring: In this step, you will need to split the data into input features and output values. In addition, you will create training and testing datasets. One particularity of this case study is that the same task appears in both training and testing, thus data must be grouped by task when creating training and validation sets.

3.1. Combine the task features, supplier features and costs into a single dataset with the following columns:

Task ID	TF1	…	TFn	SF1	…	SFm	Cost

Each Task ID (and its corresponding feature values) will appear multiple times (once for each supplier). The number of rows of this dataset must be the same as the number of rows in '`costs.csv`'. Then split the dataset into `X` (TF1,…,TFn, SF1,…, SFm), `y` (Cost), and `Groups` (Task ID). Note that there will be fewer task features (TF*i*) than in the original data because of the features removed in Step 1 above.

3.2. Randomly select 20 unique 'Task ID' values from `Groups` (let's call this subset `TestGroup`). Split the dataset created in Step 3.1 into four different datasets (*`X_train, X_test, y_train`* and *`y_test`*) as follows. *`X_train, X_test`* contain only the columns TF1,…,TFn, SF1,…, SFm; whereas *`y_train,y_test`* contain only the column 'Cost'. Also, *`X_test, y_test`* contain only rows whose Task IDs are in `TestGroup`; whereas *`X_train, y_train`* contain the remainder rows.

3.3. Train a **regression** ML model of your choice using *`X_train`* and *`y_train`*. Score the performance of the model on the *`X_test`* and *`y_test`* using the score function of the model. When using a regression model, the ML model predicts the cost of each supplier for a given task, thus the supplier selected by the ML model in Eq. (1) is the one with the lowest predicted cost. Note that Eq. (1) uses the actual cost of the selected supplier, not the predicted cost.

3.4. Using Eq. (1) calculate the Error of the trained model for each task in `TestGroup` and using Eq. (2), calculate the RMSE score.

3.5. Compare Errors and RMSE to the values obtained in Step 2.3 and summarise your conclusions.

4. Cross-validation: In Step 3, we have done a simple train-test split grouped by Task ID. Now you will do a Leave-One-Out cross-validation grouped by Task ID. Fortunately, scikit-learn already provides a Leave-One-Group-Out cross-validation type. Starting from the datasets created in Step 3.1, perform a Leave-One-Group-Out cross-validation using the `Groups` dataset to define the groups (read the documentation and examples of Leave-One-Group-Out cross-validation to learn how to specify the groups). In addition, we wish to use, within the cross-validation, our own score function (Eq. 1) for scoring. You will need to use the scikit-learn function '**`make_scorer`**', as shown in the lectures, to create a 'score' function that can be passed to scikit-learn's cross-validation function. Report the scores returned by cross-validation and compute the RMSE of

the scores returned. Compare with the results obtained in Steps 2.3 and 3.5.
Hint: If your computer has multiple CPUs, use the **`n_jobs`** parameter to use all CPUs in parallel.

5. Hyper-parameter optimization: Looking at the documentation for the ML model of your choice, devise a hyper-parameter grid (**`param_grid`**) appropriate for your ML model and apply GridSearch to identify the best hyper-parameter configuration starting from the datasets from Step 3.1. As in Step 4, you must setup Grid Search to use Leave-One-Group cross-validation and the scoring function from Eq. (1). Report the best hyper-parameter configuration found by GridSearch and its RMSE. Compare the results with those obtained in Steps 2.3, 3.5 and 4.

6. Explore additional regression models from Scikit-learn and select one that you believe could perform well (justify your choice in the report). Apply the selected model and repeat Steps 3 to 5. Compare the results between the models and discuss the strengths and weaknesses of each approach. Feel free to experiment with more regression models for comparison, but keep the word limit in mind.

Hints:

a) Implement all steps in Python. Use Python to compute statistics, generate tables and plots. Do NOT use Excel or any other software.

b) Use different Python files to implement independent steps that read/write intermediate data files. For example, `step1-preparation.py` will create a new set of files that will be read by `step-2-exploratory.py` and `step-3-train-test.py`. This is only an example: It is part of the coursework to *think* and decide on the best way to divide the tasks between Python files.

c) Further subdividing the steps into individual files will allow you to distribute the work among the team members and require less computational resources. For example, Steps 3, 4 and 5 are mostly independent once Step 3.1 is completed.

d) All steps must be reproducible without having to edit the Python files. That is, if we gave you new datasets with the same filenames, you should be able to execute all Python files in order and get new results *without having to change anything in the code*.

e) This coursework requires doing your own research. The coursework can be completed using only the Python packages: Numpy, Pandas, Matplotlib and Scikit-learn. However, you DO need to look up in the documentation of the relevant package how to perform the various steps and understand the documentation using the knowledge acquired in the lectures and labs.

f) Document your code.

Coursework Marking Scheme

Your coursework mark will be determined based on the quality of both the report and the group presentation. The final coursework mark will count 70% to your overall module mark; the remaining 30% will be made up of your two online test grades. Ensure that the material presented in your report is correct and reproducible, and that the Python code is correct and readable.

Each group MUST submit their report by the deadline indicated in the coursework brief. Late submission will be penalised in accordance with the School's guidelines on late submission (see course description). Each group MUST work on its own. Any evidence of plagiarism will result in a mark of zero.

- *Q1 :* 5%
- *Q2 :* 5%
- *Q3 :* 15%
- *Q4 :* 15%
- *Q5 :* 15%
- *Q6 :* 15%
- *Professional and visual presentation, writing and clarity:* 15%
  Pay attention to layout, font size, keeping to the page limit, spelling, grammar, use of tables and charts, consistent number of decimal places.
- *Conciseness, correctness and clarity of the code:* 15%
  Avoid repetition of code, good use of Numpy, Pandas and Scikit-learn. Code should be as concise as possible but commented in detail.