**QBUS6840 2022S1 Group Assignment (25 marks)**

**Marking rubric:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Criteria** |  | **Max** |
| Python Code | * The Python code is executable with the output is the (multi-step ahead) forecast MSE on the test data. * The forecast MSE is close to the benchmark MSE | 5.0  10.0 | 15.0 |
| Report | **Data cleaning/EDA – 1.5 marks**   * The data cleaning step is well-documented * The EDA step is well-documented   **Model selection and training – 7.5 marks**   * Justification on the selection of the model in the report is well-documented * Justification on how to train the selected model is well documented * Discussion on the results is well-documented   **Presentation – 1 mark**   * Clear presentation of text, figures, and formulas, free of spelling and grammar mistakes, etc. | 0.5  1.0  3.0  3.0  1.5  1.0 | 10.0 |
|  |  | **Total** | **25.0** |

**General marking strategy:**

I would suggest the following marking steps:

|  |  |
| --- | --- |
| **Step 1**  **Code checking**  **(15 marks)** | Run the python code to get the forecast MSE on the test data   * **Case 1:**   + If the python script is executable and the output is the **multi-step ahead** forecast MSE, give **5 marks.**   + **The marks for forecast performance will be done later. (Maximum 10 marks)** * **Case 2:**    + If the python script is executable and the output is the **one-step ahead** forecast MSE, give maximum **7 marks**   + If the python script is executable but there is no code to produce final forecast values, and code to calculate MSE on the true test data, give **5 marks.** * **Case 3:**    + If the python script is not executable, give MAXIMUM **5 marks** for the implementation part (depend on how clean, well-structured, sufficient comments, etc.).   + If you cannot run the scripts because of the issues with software installation, data parsing (test data) or some errors that are out-of-control of students then please fix the issues and give the marks according to Case 2 or Case 3. |
| **Step 2**  **Report Content**  **(9 marks)** | Checking the report and give the marks following the marking rubric   * **Data cleaning/EDA: (2 marks)**   + Introduction, business context and summary of the works (**0.5 mark**)   + Checking / handling missing values (if any) (**0.5 mark**)   + Visualizing the time series (**0.5 mark**)   + Discussion on the stationarity or interesting properties of the data (**0.5 mark**). If students use ARIMA models, stationarity must be discussed in the report. * **Selecting and training model: (6 marks)**   + From the EDA step, explain why a particular model is selected. (**1 marks**)     - Students should discuss at least 2 models including a benchmark model and a final model. If not -> **deduct 0.5 mark**   + A sufficient discussion, e.g., mathematical equations, important properties, etc., should be provided. (**1 mark**)   + Justification on how to train the model: (4 **marks)**     - Discussion on the data pre-processing step, if needed. For example, if they use the neuron network models, a re-scaling step should be mentioned.     - Discussion on the model selection step. For example, if student use the ARIMA model, they should discuss how to select the values for (p, d, q).     - A validation/train splitting step is necessary for tuning hyper-parameter of evaluating the model. If students do not use validation/train split, they should explain why. Justification on split ratio must be provided.     - Students should use all training data to estimate the final model. If not, explanation is needed.     - How are the Python functions specified to train the model (training settings)?     - Report the training results such as parameters estimation (for simple models) or validation errors of candidate models. * **Discussion on the results: (1 marks)**   + - Is the result good or bad? Why? (**0.5 mark**)     - What could be implemented to improve the model. (**0.5 mark**) |
| **Step 3**  **(1 marks)** | Checking the overall presentation of the report and give the marks accordingly |