FIT5149 S2 2024 Assessment 1

Stock Volatility Prediction

Marks	25% of all marks for the unit
Due Date	11:55 PM Monday, 9 September 2024 (Week 8)
Extension	Extensions and other individual alterations to the assessment regime will only be considered using the University Special Consideration Policy. Students should carefully read the Special Consideration website, especially the details about required formal documentation. All special consideration requests should be made using the Special Consideration Application. Note that students will no longer seek extensions from the teaching team/Chief Examiner. All extensions will be via the special consideration form process and extensions will be approved by SEBS (not the Chief Examiner).
Lateness	For all assessment items handed in after the official due date without an agreed extension, a 5% penalty applies to the student's mark for each day after the due date (including weekends and public holidays) for up to 5 days. Assessment items handed in after 5 days will not be considered/marked.
Group Member	This assignment is a **group assignment**, and you are highly encouraged to form **a group with two or three members, including yourself**. A group with more than three members (4 or more) is not allowed. You may complete this assessment by yourself if you have difficulty forming a group.
Authorship	The final submission must be **identifiably your group's own work**. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary actions.
Submission	Each group is required to submit two files: one PDF file containing the report, and a ZIP file containing the implementation and other required files. The two files must be submitted via Moodle by one group member. All group members are required to log in to Moodle to accept the terms and conditions on the Moodle submission page and Click Submit Button. A draft submission won't be marked.
Programming language	Either R or Python

Note: Please read the description carefully from start to finish before you begin your work! Given that this is a group assessment, **each group should evenly distribute the work among all group members**.

1 Introduction

Stock market volatility, a crucial measure of market risk and uncertainty, plays a pivotal role in financial decision-making. Accurate prediction of stock volatility is valuable for multiple reasons: it enables investors and fund managers to better assess and manage portfolio risk, facilitates more precise options pricing, enhances the performance of certain trading strategies, helps companies optimize the timing of stock issuances or buybacks, and provides insights into economic uncertainty for policymakers and economists. Given its wide-ranging implications, the ability to forecast volatility effectively is a highly sought-after skill in the financial industry.

Stock volatility is a statistical measure that represents the degree of variation in a stock's trading price over a specific period of time. It is a key indicator of the risk associated with a particular stock or the overall market. Higher volatility implies greater price fluctuations and thus higher risk, while lower volatility suggests more stable price movements.

For this assignment, we focus on monthly volatility. The monthly volatility is calculated as follows:

1. Calculate daily returns: For each trading day, compute the percentage change in closing price using the formula:

$$\text{Daily Return} = \frac{\text{Closing Price}_t - \text{Closing Price}_{t-1}}{\text{Closing Price}_{t-1}} \cdot 100\%$$

2. Calculate monthly volatility: Compute the standard deviation of these daily returns over the month and multiply by 100 to express as a percentage.

This measure of volatility provides a standardized way to quantify and compare the price variability of different stocks or the same stock over different time periods. By predicting this metric, investors and financial professionals can make more informed decisions about risk management, portfolio allocation, and trading strategies.

This assignment focuses on developing the skill of predicting monthly stock volatility using historical market data and various factors that influence stock price movements. You will work with a dataset containing fundamental trading features and financial report features for 613 stocks over a 22-month period. Your task will be to develop regression models that can accurately forecast the volatility for the 23rd and 24th months based on the historical data.

Through this task, you will gain practical experience applying data analysis and machine learning techniques to a problem with significant real-world applications in finance and investment. You will engage in exploratory data analysis, feature engineering, model selection, and hyperparameter tuning. Additionally, you'll need to interpret your results, identifying key features that influence stock volatility and providing statistical evidence to support your findings.

This assignment allows you to apply theoretical knowledge to a practical financial problem and simulates the kind of work done by quantitative analysts and data scientists in the financial industry. By completing this task, you'll develop valuable skills in financial data analysis, predictive modeling, and results interpretation—all of which are highly relevant in today's data-driven financial world.

2 Task Description

This assessment aims to practice your exploratory data analysis and machine learning skills by building regression models for predicting monthly stock volatility. You will work in groups of up to 3 members to complete the following tasks:

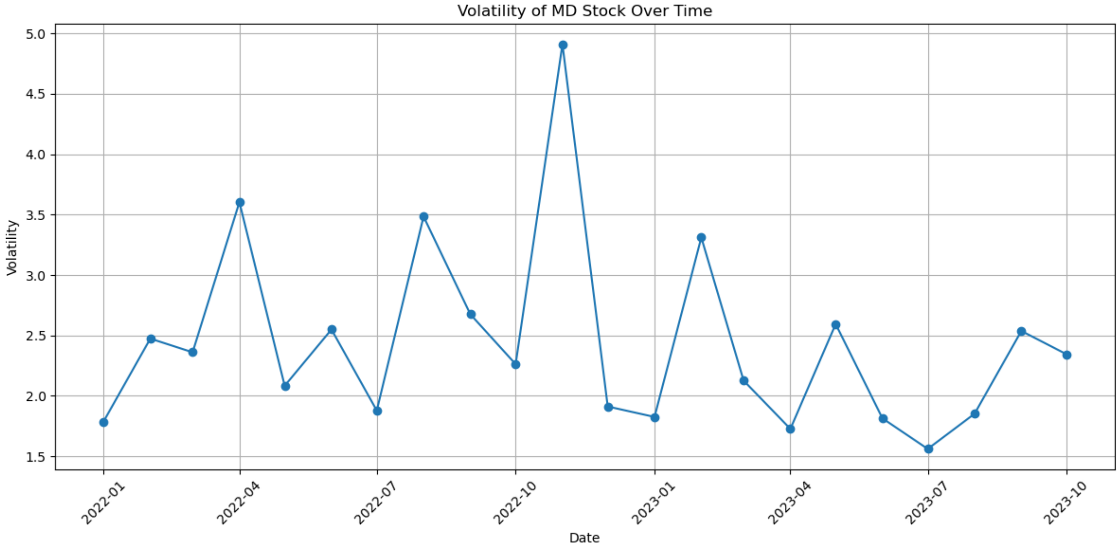

Figure 1: Volatility of the stock, MD, over time

2.1 Prediction Task

- Develop at least three different types of regression models to predict stock volatility for the 23rd month using various features extracted from the last 22 months' historical stock data.
- Select the best-performing model for final predictions and submission.
- Apply appropriate data preprocessing methods, feature selection, or feature extraction techniques.
- Tune hyperparameters of the models to optimize performance.

Note: Different input variables for the same algorithm (e.g., linear regression) are considered the same type of model. Linear regression and KNN regressor, for example, are considered different types.

2.2 Analysis Task

- Identify key input variables and/or outliers that significantly influence the predictive outcomes.
- Provide statistical evidence to support your findings, such as feature importance scores.
- Explain which features (including newly created ones) are particularly useful for estimating stock volatility and why.

2.3 Report

Write a report documenting your approach and results, including:

- Description of at least *three* different models you tried.
- Explanation of your data preprocessing and feature engineering techniques.
- Analysis of model predictions and performance.
- Evidence-based discussion of feature importance and their relationship to stock volatility.

2.4 Kaggle Challenge (Optional)

You are encouraged to participate in the Kaggle challenge to predict the 24th month's stock volatility. For the Kaggle competition, there are no restrictions on programming languages or machine learning tools.

2.5 Evaluation Metric

Given a test set $\{(\mathbf{x}_1, y_i), ..., (\mathbf{x}_n, y_n)\}$ and the corresponding predicted values $\{\hat{y}_1, ...\hat{y}_n\}$, the key metric to evaluate your final model is Root Mean Squared Error (RMSE), defined as:

$$\text{RMSE} \quad = \quad \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- RMSE in Python can be computed as the square root of Mean Squared Error (MSE)[1];
- RMSE in R can be computed by following the lecture or the tutorial[2].

3 Dataset

The dataset provided contains monthly stock market data for 613 stocks. For each stock, the data is structured as follows:

- Training set: 22 months of historical data
- Submission test set: Data for the 23rd month
- Kaggle test set: Data for the 24th month

3.1 Dataset Features

The dataset includes two types of features: Fundamental Trading Features and Financial Report Features.

Fundamental Trading Features These features are derived from monthly trading activities:

- Date: The first day of each trading month (format: YYYY-MM-DD). This represents the period for all other metrics in that record.
- Open: The stock's opening price on the first trading day of the month (unit: USD).
- Close: The stock's closing price on the last trading day of the month (unit: USD).
- High: The stock's highest price reached during the month (unit: USD).
- Low: The stock's lowest price reached during the month (unit: USD).
- Volume: The total number of shares traded during the month (unit: number of shares). This is a measure of trading activity and liquidity.
- Amount: The total monetary value of all shares traded during the month (unit: USD). This is calculated as $(Volume \times Price)$ for each trade, summed over the month.
- Avg_Price: The average stock price for the month (in USD), calculated as the mean of daily closing prices throughout the month.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

[2] https://www.geeksforgeeks.org/root-mean-square-error-in-r-programming

- Return: The monthly return of the stock (unit: percentage, %). This is typically calculated as $\frac{(Close\_current-Close\_previous)}{Close\_previous} * 100$.
- Volatility: The target variable for prediction, representing the stock's price variability over the month (in percentage). It is calculated as the standard deviation of daily returns multiplied by 100. Please note that we do not provide daily data, only the final calculated monthly volatility.

Financial Report Features These features are extracted from quarterly and annual financial reports. The frequency of updates and the nature of the data (cumulative vs. snapshot) vary depending on the specific metric:

- **Revenue, Net Income, and Gross Profit:**
  - **Revenue:** Total money earned from primary business activities.
  - **Net Income:** Profit after subtracting all expenses from revenue.
  - **Gross Profit:** Revenue minus the cost of goods sold.
  - Updated quarterly. Values remain constant for three consecutive months, then update.
  - Reported as both quarterly figures and year-to-date cumulative figures
  - Reset to zero at the beginning of each new fiscal year
- **EPS (Earnings Per Share):**
  - Measure of profitability on a per-share basis (Net Income / Outstanding Shares).
  - Updated quarterly. Values remain constant for three consecutive months, then update.
  - Reported as both a quarterly figure and a trailing twelve months (TTM) figure
  - The TTM figure is recalculated each quarter, representing the sum of the last four quarters
- **Total Assets, Total Liabilities, Total Equity, and Cash and Cash Equivalents:**
  - **Total Assets:** Sum of all company-owned resources with economic value.
  - **Total Liabilities:** Sum of all financial obligations.
  - **Total Equity:** Assets minus Liabilities; represents shareholders' stake.
  - **Cash and Cash Equivalents:** Most liquid assets, including cash and short-term investments.
  - Updated quarterly. Values remain constant for three consecutive months, then update.
  - Represent snapshot values as of the end of each quarter
  - Not cumulative; each reported value stands on its own
- **Operating Cash Flow, Investing Cash Flow, and Financing Cash Flow:**
  - **Operating Cash Flow:** Cash generated from core business operations.
  - **Investing Cash Flow:** Cash used in or generated from investment activities.
  - **Financing Cash Flow:** Cash from activities like issuing stock, paying dividends, or managing debt.
  - Updated quarterly. Values remain constant for three consecutive months, then update.
  - Reported as both quarterly figures and year-to-date cumulative figures
  - Reset to zero at the beginning of each new fiscal year

Important Considerations for Analysis When working with this dataset, students should consider:

- **Temporal Misalignment:** Financial report features update quarterly, while stock data is monthly. This creates a pattern where financial data remains constant for three-month periods.
- **Data Nature:**
  - Cumulative within fiscal year: Revenue, Net Income, Gross Profit, Cash Flows
  - Snapshot values: Balance sheet items (Assets, Liabilities, Equity)
- **Preprocessing Strategies:**
  - Convert cumulative figures to quarterly values
  - Create indicators for financial report release months
  - Normalize features to address scale differences
- **Temporal Aspects:**
  - Avoid using future information in predictions
  - Consider lagged features to capture delayed financial report impacts
- **Feature Engineering:** Derive new features from existing ones to potentially enhance predictive power.

3.2 Dataset Statistics

Task	# historical examples	# examples for submission	# examples for Kaggle
Stock Volatility Prediction	13486	613	613

Table 1: Overview of the datasets.

We provide the following datasets for this assignment (Table 1):

- **A1_stock_volatility_labeled.csv**: This file contains 13,486 labeled instances, representing historical records from January 2022 to October 2023 for 613 stocks. Each stock has 22 months of data. This is the only labeled dataset available for training and evaluating your regression model.
- **A1_stock_volatility_submission.csv**: This file contains test examples for 613 stocks. The *volatility* values in this dataset are set to a default value of **0**. Your task is to estimate the most probable volatility value for each stock in this test set. Specifically, you need to predict the volatility values for November 2023.
- **A1_stock_volatility_kaggle.csv**: This file also contains data for 613 stocks and is intended for the Kaggle challenge. Similar to the submission file, the *volatility* values are set to a default of **0**. For the Kaggle challenge, you will need to predict the volatility values for December 2023.

Note: For the main assessment, you should use only the A1_stock_volatility_labeled.csv file to train your regression models. The A1_stock_volatility_submission.csv file should be used solely for generating your final predictions.

4 Mini Tutorial on Supervised Learning in Practice

4.1 Overview

Supervised learning in practice typically consists of three major steps: feature extraction, model selection, and application of the selected model to predict values for unseen data. Feature extraction aims to transform the data into representations that are best suited for the models. As the datasets already provide initial features, you are expected to apply preprocessing and feature engineering techniques to further improve data representations. Model selection involves identifying appropriate models and tuning their hyperparameters to maximize performance. Both feature extraction and model selection are essential for achieving optimal performance. While the prediction result is largely determined by the selected model, there are various approaches for improving model performance at this stage. However, these advanced prediction techniques are not the focus of this assessment.

4.2 Recommended Practice

We introduce the following practices not just for successfully completing this assessment, but also to provide a general guideline for tackling supervised learning problems with handcrafted features. You are encouraged to try most of these steps and familiarize yourself with them. Given a labeled training dataset and an unlabeled test dataset, the following steps are recommended to develop an optimal model:

(i) Split the labeled dataset into training and hold-out sets. Begin by using the training set to tune hyperparameters of models, for example, through cross-validation. Multiple models may show only minor performance differences. You can then apply these candidate models to the hold-out set to determine which one to choose for the final submission. The final model should be trained on all labeled data, including the hold-out set. Additionally, if you are new to machine learning toolkits, the hold-out set helps ensure you can generate predictions in the correct format required by the assessment and the Kaggle challenge. This step is optional if you are already familiar with model selection techniques such as cross-validation.

(ii) Explore and represent the data properly. You may use different R or Python packages to understand the most appropriate way to represent each input variable with respect to the classifiers of your choice. This dataset includes binary, nominal, ordinal and numeric variables.

- Categorical (Nominal) variable: In this dataset, the ‘Stock’ variable is a nominal variable, representing different stock symbols. It has multiple distinct categories (different stocks) with no inherent order. For machine learning models:
  - In R, you can use `dummyVars` from the *caret* package to convert 'Stock' into binary variables.
  - In Python, `pandas.get_dummies()` or `sklearn.preprocessing.OneHotEncoder` can be used for this purpose.
- Date variable: The ‘Date’ in your dataset can be treated as an ordinal variable or converted to numerical features:
  - Extract ordinal features like month number (1-12), quarter (1-4), or year.
  - Create cyclical features to capture seasonality, e.g., $\sin\left(\frac{2\pi\cdot\text{month}}{12}\right)$ and $\cos\left(\frac{2\pi\cdot\text{month}}{12}\right)$.
  - Generate binary features like 'Is_Quarter_End' or 'Is_Fiscal_Year_End'.
- Numerical variables: Most variables in your dataset are numerical (e.g., Open, Close, Volume, Revenue, EPS). These often require preprocessing due to varying scales:
  - Standardization: Transform a variable x to standard normal distribution using $\frac{x-\mu_x}{\sigma_x}$, where μ_x is the mean and σ_x is the standard deviation.
    - In R: `preProcess(data, method = c("center", "scale"))` from *caret*.
    - In Python: `sklearn.preprocessing.StandardScaler()`.

- Normalization: Rescale variables to a specific range, often [0,1]:
    - In R: `preProcess(data, method="range")` from *caret*.
    - In Python: `sklearn.preprocessing.MinMaxScaler()`.
- Log Transformation: Useful for highly skewed financial data like Volume or Revenue:
    - In R: `log1p(x)` (adds 1 before taking log to handle zeros).
    - In Python: `np.log1p(x)`.
- Binning: Convert continuous variables to categorical. This can be useful for variables like company size based on Total Assets:
    - In R: `cut()` function or `dplyr::case_when()`.
    - In Python: `pandas.cut()` or `numpy.digitize()`.

(iii) Select the best-performing model by using, e.g. cross-validation, hold-out test, etc. (covered in Week 4) and tuning hyper-parameters, e.g. the regularization coefficient for Lasso (Week 6). The goal of this step is to identify the model that performs the best on unseen data. There are various techniques for tuning hyper-parameters of a model implemented in both R and Python, such as

- Expert search. For each experiment, you carefully select a value for each hyper-parameter based on your experience and evaluate them with a model evaluation method, such as cross-validation. You may have multiple trials for the same model and select the next set of hyper-parameters based on the previous experimental results.
- Grid search. Often, you only know a set of values may be good for a hyperparameter, but you want to try all possible combinations of the values of all hyperparameters for a model. The grid search tries to find all possible combinations of those values and selects the best model according to the results of a selected model evaluation method.
- Random search. Grid search is expensive because it aims to try all combinations. Given a limited computational budget, you may be able to try model evaluation at most n times, e.g., K-fold cross validation. Therefore, the alternative strategy is to randomly draw n distinct combinations of hyper-parameter values, and evaluate them sequentially or in parallel. In practice, this method often achieves comparable performance as grid search if the computational budget is not too low.

The above hyper-parameter search methods are often not used in isolation but are combined together in practice. You may use expert search to determine which values are good for a model and apply grid search or random search to identify which combinations of the hyperparameters perform the best on your data. In R, the package *caret* provides such a functionality through the methods `trainControl` and `train`, see http://topepo.github.io/caret/model-training-and-tuning.html for details. The scikit-learn provides also the same support, see https://scikit-learn.org/stable/modules/grid_search.html# for further details.

Steps (ii) and (iii) are often applied iteratively because changes in features (input variables) may lead to different choices of models. A significant change in Step (ii) often requires redoing certain parts of model selection. Additionally, ML practitioners often consider feature selection after Step (ii) or as a regularization technique during training, such as the L1 regularizer (Week 6).

4.3 Group Forming

This is a group-based assessment. We will create an ”Assessment 1 group selection” on Moodle. Please follow these guidelines:

- Form a group of up to 3 members. You may choose members from any tutorial class.
- Select an ID for your group on Moodle.
- Ensure all members join the same group on Moodle.

Note:

- There is NO restriction on whether group members are from the same tutorial class.
- The recommended and maximum number of group members is 3. Single-member groups are allowed under special circumstances.

4.4 Project Management Requirements

To ensure effective teamwork and fair distribution of responsibilities, each group must adhere to the following project management guidelines:

- Designate a project leader who will rotate on a weekly basis.
- The project leader's responsibilities include:
  - Coordinating team meetings and activities
  - Assigning and overseeing tasks
  - Ensuring timely completion of deliverables
  - Documenting group progress and individual contributions
- All team members are expected to:
  - Actively participate in project activities
  - Contribute meaningfully to assigned tasks
  - Communicate effectively with teammates
- Plan and monitor the weekly activities using a tool of your preference, such as Trello and Google Spreadsheet, and maintain a project log that includes:
  - Clear task assignments and their completion status (e.g. To do, ongoing, or done)
  - Specific individual contributions to the project
- Include a final project management report based on the project log as the last section of your submission. This report should cover:
  - The group's workflow and collaboration methods
  - Weekly leadership rotations
  - Each member's contributions throughout at least a four-week period

The report should include a table summarizing the project management and brief contribution descriptions over the four-week period, as shown below.

Member	**Week 1**	**Week 2**	**Week 3**	**Week 4**
Alice	L, Project management	Feature selection	Data analysis	L, Project management
Bob	Data cleaning	L, Cross-validation	Model tuning	Results compilation
Charlie	Data preprocessing	Visualization	L, Report writing	Code cleaning

Note: L = Leader for the week.

Table 2: Weekly Roles and Contributions of Project Team Members

4.5 Kaggle Data Competition (Optional and Not Marked)

We will launch a Kaggle Data Competition for this assessment. The link will be available in the Assessment section of Moodle.

- Each group should use the same user ID for submitting model predictions on Kaggle.
- This competition provides feedback on your model's predictions and allows you to compete with classmates.
- Although Kaggle's results are not marked, we highly encourage participation. It will be a lot of fun!

Note: The Kaggle results will not be marked. Please refer to Section 5 for the required submission details and check the marking rubrics.

4.6 Submission Guidelines and Hints

Based on past submissions, we offer the following advice:

- Use relative paths in your Jupyter Notebook (e.g., "./A1_stock_volatility_labeled.csv"). Place data files in the same folder as your notebook.
- Provide meaningful interpretation and discussion of your results, especially for plots and statistics.
- Choose appropriate visualizations and statistics to convey relevant information.
- Clearly document your model development process, including parameter selection.
- Be precise in using various tools and in your discussion.
- Avoid overly long notebooks or Rmd files. Focus on key information and analyses.
- Document your methodology and logic clearly.
- Use markdown cells for discussion to demonstrate your reasoning skills.
- Utilize the discussion forum and consultations to clarify any doubts.
- Before final submission, ensure your Jupyter Notebook or Rmd file runs without errors. Use "Kernel → Restart & Run All" to verify.

5 Submission

To finish this data analysis challenge, all the groups are required to submit the following files:

- **"pred_values.csv"**, where the predicted values on the testing set (A1_stock_volatility_submission.csv) is stored.
  - In your "pred_values.csv", there will be two columns: the first column is the **stock** column from A1_stock_volatility_submission.csv. The second column should include **predicted values** for stock volatility prediction.
  - The "pred_values.csv" file must be reproducible by the assessor using your submitted R/Python code. To ensure reproducibility, please use a fixed random **seed** in your code. Without a fixed seed, your results might not be reproducible.

- The **R/Python implementation** of your **final** model with A README file that tells the assessor how to set up and run your code. If you use R, the R version should be either R 3.64, R 4.2 or above. The output of your implementation must include the predictions, which are stored in the file **"pred_values.csv"**. The use of Jupyter notebook or R Markdown is **not required**. All the files that are required for running your implementation must be compressed into a **zip** file, named as "**groupName_ass1_impl.zip**". Please note that the unnecessary code must be excluded from your final submission. For example, if you tried two different types of models, say linear regression and regression splines (Week 7), and your group decides to submit regression splines as the final model. You should remove the code for the other models from the submission. **The discussion of the comparison should be included in your report.** *However, you should keep a copy of the implementation used for the model comparison just in case if an interview is scheduled.*

- **A PDF report**, where you should document in detail the development of the submitted model. **The maximum number of pages allowed is 5**. The report must be in the PDF format, named a **"groupdName_ass1_report.pdf"**. The report must include (but not limited to)

  - The discussion of how the data preprocessing/features extraction has been done.
  - The development of the submitted model: To choose an optimal regression model for a task, we often carry out empirical comparisons of multiple candidate models with different feature sets. In your report, you should include a comprehensive analysis of how you compare different models and why you choose the final one. For example, the report can include (but not limited to)
    - A brief description of the model(s) considered in your comparison.
    - The detailed experimental settings, which can include, for example, the discussion of how the cross-validation is set up, how the hyper-parameters are tuned for the model considered (if applicable).
    - Regression performance in terms of RMSE with comprehensive discussion.
    - The justification of the final model submitted.
    - Analysis of the key features you identified for the target task and provde statistical evidence about why they are important.
    - The Kaggle user ID if you participate the Kaggle challenge.
  - Project Management Summary:
    - Include a brief overview of your group's workflow and collaboration methods.
    - Provide a table summarizing the weekly leadership rotation and contribution percentages for each team member over the four-week period.

  Warning: If a report exceeds the page limit, the assessment will only be based on the first 5 pages.

  Generative AI Usage Guideline: You may use AI solely for grammar checking or paraphrasing. However, using generative AI for creating new content is prohibited. If you use AI or generative AI for grammar checking or paraphrasing, please include a brief acknowledgment in the Appendix (not included in the 5-page limit) specifying which parts involved AI assistance.

- **A signed group assignment cover sheet** (link provided on Moodle), which will also be included in your zip file.
  Warning: typing name is not counted as a signature in the cover sheet.

6 How to submit the files?

The Moodle setup allows you to upload only two files

- "**groupdName_ass1_report.pdf**": A pdf report file, which will be submitted to Turnitin.

- “**groupName_ass1_impl.zip**”’: a **zip** file includes
  - the implementation of the final submitted model
  - “pred_values.csv”, where the predictions on the testing data is stored.
  - the signed grouped assignment cover sheet

While submitting your assignment, you can ignore the Turnitin warning message generated for the ZIP file.

Please note that

- **Only one group member needs to upload the two files. But all the group members have to login in to their own Moodle page and click the submit button in order to make the final submission.** If any member does not click the submit button, the uploaded files will remain as a draft submission. A draft submission won’t be marked.
- **The two files must be uploaded separately.**

7 Academic integrity

Please be aware of University’s policy on academic integrity. Monash University takes academic misconduct[3] very seriously. You can learn from the above materials and understand the principle of how the analysis was done. However, you must finish this assessment with your own work.

[3] https://www.monash.edu/students/study-support/academic-integrity