COMP3010 Machine Learning Assignment

@ Computing, Curtin University

Last updated: 21st March, 2025

Weighting:
This assignment is worth 100 points, which weighs 40% of the final mark.

Submission:
You need to submit your prediction to Kaggle (see Section 5.1). You also need to submit everything in **a single ZIP** file to Blackboard. Name the file as `<studentID>_<name>_assignment.zip`. The due date is **04 May 2025 11:59 PM**.

Academic Integrity:
This is an **individual** assignment so any form of collaboration is not permitted. This is an **open-book** assignment so you are allowed to use external materials, but make sure you properly **cite the references**.

1 Introduction

Globally, a substantial volume of oil and gas products, including hazardous materials, is transported daily through various means. Frequently, this transportation occurs in densely populated areas, posing significant risks to nearby structures and residents. In particular, the road transport of Liquefied Petroleum Gas (LPG) is common in industrialised countries and has raised public safety concerns. A critical hazard associated with such transport is the occurrence of Boiling Liquid Expanding Vapour Explosions (BLEVEs)—intense explosions triggered by complex, nonlinear physical processes. These events can cause severe damage to infrastructure and pose serious threats to human life. However, predicting the intensity of such explosions remains challenging using conventional methods. For further details, refer to relevant studies [1, 2].

This project aims to address this challenge by applying data-driven machine learning techniques to predict the overpressure generated by blast waves resulting from BLEVEs.

2 Problem Description

In this assignment, your task is to perform predictive analysis on the peak pressure generated by Boiling Liquid Expanding Vapour Explosions (BLEVEs). Consider a scenario where a BLEVE occurs inside a rectangular tank located within a three-dimensional environment, as illustrated in Figure 1. A rigid wall, simulating a building structure, is placed at a certain distance from the BLEVE source. This wall interacts with the blast wave, causing reflections and deflections that increase the complexity of the pressure distribution.

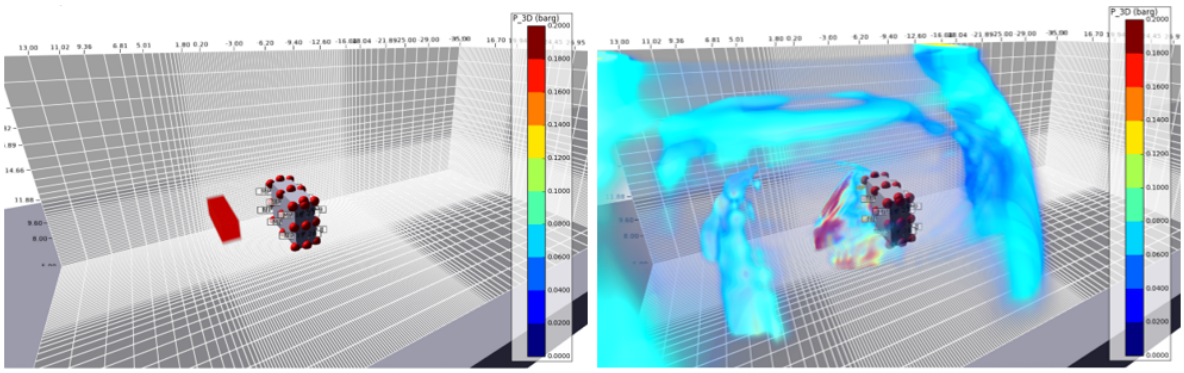

Figure 1: BLEVE blast wave propagation in an environment with an obstacle.

The objective is to accurately predict the peak pressure in the vicinity of the obstacle. To support this task, 27 sensors are strategically placed around the obstacle walls—nine each on the front, back, and side walls—as shown in Figure 2. These sensors are used both for training and testing the predictive model.

The 3D environment includes a variety of physical variables relevant to BLEVE scenarios, such as temperature, pressure, gas-to-liquid ratio, and the dimensions of both the tank and the obstacle. The full list of input features is as follows:

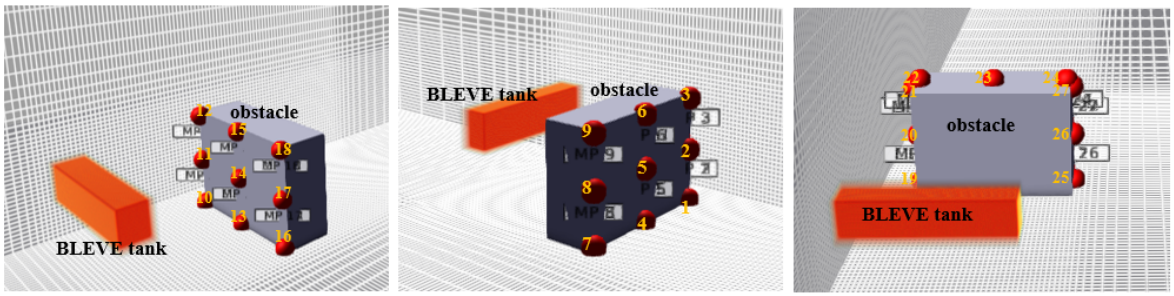

Figure 2: Sensor locations on the obstacle. Left to right: front wall, back wall, and side wall.

- `Tank Failure Pressure`: internal pressure at the time of BLEVE (in bar)
- `Liquid Ratio`: ratio of liquid in the tank (coexistence of liquid and vapour)
- `Tank Width`: tank width (in metres)
- `Tank Length`: tank length (in metres)
- `Tank Height`: tank height (in metres)
- `Vapour Height`: vapour column height inside the tank (in metres)
- `BLEVE Height`: height of the tank above ground level (in metres)
- `Vapour Temperature`: temperature of vapour (in K)
- `Liquid Temperature`: temperature of liquid (in K)
- `Obstacle Distance to BLEVE`: distance between the obstacle and BLEVE source (in metres)
- `Obstacle Width`: obstacle width (in metres)
- `Obstacle Height`: obstacle height (in metres)
- `Obstacle Thickness`: obstacle thickness (in metres)
- `Obstacle Angle`: angle between the line connecting the centres of the obstacle and BLEVE and the horizontal axis (in degrees)
- `Status`: liquid state, either `subcooled` or `superheated`
- `Substance Critical Pressure`: pressure required to liquefy vapour at the critical temperature (in bar)
- `Substance Boiling Temperature`: boiling point at atmospheric pressure (in C)

- `Substance Critical Temperature`: temperature above which vapour cannot be liquefied regardless of pressure (in C)
- `Sensor ID`: unique identifier of the sensor (ranging from 1 to 27)
- `Sensor Position Side`: side of the obstacle where the sensor is located
- `Sensor Position x`: x-coordinate of the sensor (in metres)
- `Sensor Position y`: y-coordinate of the sensor (in metres)
- `Sensor Position z`: z-coordinate of the sensor (in metres)
- `Target Pressure`: peak pressure to be predicted (in bar)

3 The Tasks

In this assignment, you are provided with two datasets: train.csv and test.csv, containing the training and testing data, respectively. The training set includes the target variable (peak pressure), while the testing set does not. Your objective is to train a machine learning model using the training data and to predict the peak pressure for the testing data.

You are required to complete three major tasks: data preprocessing, model development, and report writing.

3.1 Data Preprocessing

Effective data preprocessing is crucial for successful machine learning applications. This stage includes selecting relevant features, removing redundant or irrelevant ones, and potentially creating new features. You are also expected to adapt data formats and types based on the requirements of your chosen model. A particular emphasis should be placed on data cleaning, which may include the following:

- **Identifying and Handling Missing Values**: Examine the dataset for missing or incomplete entries. Choose appropriate strategies such as imputation or deletion, depending on the context and potential impact on model performance.
- **Outlier Detection and Treatment**: Detect and address any outliers that may distort your analysis. Apply suitable statistical techniques to correct or exclude these anomalies.
- **Duplicate Removal**: Ensure dataset integrity by identifying and removing duplicate records, which can otherwise bias the model.
- **Correcting Inaccurate Entries**: Carefully inspect the data for incorrect values and rectify them to maintain dataset quality.

- **Feature Selection**: Identify and retain only the features that exhibit meaningful correlation with the target variable. Consider building a "sparse" model using a reduced set of the most informative features.
- **Feature Engineering**: You are encouraged to derive new features that could improve model performance. For instance, the ratio $\frac{\texttt{Tank Width}}{\texttt{Tank Length}}$ may serve as a useful additional feature.
- **Data Type Conversion**: Convert features to appropriate data types as required by your model. For example, categorical variables may need to be encoded into numerical formats using one-hot encoding or similar techniques.
- **Feature Scaling**: Apply normalization or standardization where appropriate, particularly for models sensitive to input magnitudes.
- **Data Augmentation**: To improve model robustness, consider techniques for increasing the number of training samples, such as synthetic data generation.
- **Other Preprocessing Steps**: You may apply any other preprocessing methods that you find beneficial.

Note: Some preprocessing steps may influence one another. Carefully consider the sequence in which you apply these steps to ensure optimal results.

3.2 Model Development

In this phase of the assignment, your objective is to explore and compare a diverse set of machine learning models to determine the most effective approach for predicting peak pressure in BLEVE scenarios. A key requirement is to evaluate **at least three fundamentally different types of machine learning models**. These models must be assessed using at least two different performance metrics.

- **Model Selection**: You must examine at least three distinct types of machine learning models. Examples include linear regression models, support vector regression (SVR), decision tree-based models (e.g., Random Forest, XGBoost), and neural networks. Note that models of a similar nature—such as Random Forest and Gradient Boosted Decision Trees (GBDT)—are not considered sufficiently distinct and will not fulfil the requirement on their own.
- **Hyperparameter Tuning**: Each model has associated hyperparameters that significantly influence its performance. You are expected to tune these hyperparameters using appropriate strategies such as hold-out validation or cross-validation. Techniques such as grid search or random search may be used to identify optimal parameter values.
- **Evaluation Metrics**: All models must be evaluated using at least two metrics. The use of **Mean Absolute Percentage Error (MAPE)** and R^2 is compulsory, both of

which are available in the Scikit-learn library. Additional metrics such as Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE) may be included for a more comprehensive analysis.

- **Model Ensembling (Optional)**: After training individual models, you are encouraged to explore ensemble methods to potentially enhance prediction accuracy. Ensembling involves combining predictions from multiple models, such as through averaging or weighted voting schemes.

Once you have selected your final model, use it to generate predictions for the test dataset. Refer to Section 5.1 for information on submitting predictions to the associated Kaggle competition.

3.3 Report

As part of this assignment, you are required to submit a comprehensive report that documents the entire workflow, from data preparation to model evaluation. The report should provide a clear rationale for all methodological choices and reflect on your findings and experiences throughout the project.

The report must address the following components:

- **Data Cleaning**: Describe the data issues encountered (e.g., missing values, outliers, duplicates, incorrect entries) and the specific steps taken to resolve them.
- **Data Processing**: Detail the preprocessing methods applied, such as normalization, feature engineering, and data type conversion. Provide justifications for each step.
- **Model Selection**: Discuss the models considered and your reasoning for selecting or rejecting each one. Emphasise the diversity of the models chosen and their suitability for the task.
- **Hyperparameter Tuning**: For each model, report the hyperparameters tuned, the search strategy used (e.g., grid search, random search), the range of values considered, and the final values selected.
- **Prediction Performance**: Summarise each model's performance using appropriate metrics. Include training, validation, and testing results in tabular form. For example, a table showing MAPE on training, validation, and test sets is sufficient.
- **Self-Reflection**: This section allows you to reflect on your learning experience. Discuss the challenges encountered, insights gained, and how you might improve your approach in future projects. You may also share any additional thoughts or observations.

Your report should be concise, well-organised, and informative. The final report must **NOT exceed 10 A4 pages**.

4 Python Environment

You will use `Python` for this assignment and you can use any library you like. You can conduct experiments with your local python environment but the final submission has to be a **Jupyter Notebook that can be run on Google Colab**. Note that the notebook you submitted should contain necessary comments or markdown cells to briefly explain what you are doing.

When saving the notebook for submission, make sure **it contains the cell output**. Colab does save cell outputs by default. If you are not sure, double-check the notebook setting and make sure the "Omit code cell output when saving this notebook" is disabled.

5 Submission

5.1 Kaggle Submission

A private Kaggle competition has been created for this assignment to allow you to evaluate your model's performance on the test set. The test set is divided into two equal parts: one half is used for the **public leaderboard**, and the other half for the **private leaderboard**.

The public leaderboard enables you to track your model's performance during development. You may submit predictions before the assignment deadline and view your MAPE score, along with the scores of other participants. **Note: this is not a competition in the traditional sense**, but rather a tool to help you monitor progress and validate your model.

When joining the competition, you **must use your STUDENT ID as your team name**. Submissions from teams without a valid student ID as their team name will be deleted, and the student will receive **zero marks for the 'model performance'** component (worth 20 marks).

Each team is allowed up to **five submissions per day**. On the assignment's due date, Kaggle will automatically compute your final score based on the best public submission, evaluated on the private portion of the test set. This **private leaderboard score** will be used for grading the model performance (see Section 6).

While daily submissions are not mandatory, you **must submit at least one final prediction before the assignment deadline** to receive a score on the private leaderboard. We strongly recommend submitting regularly to monitor your progress and avoid discrepancies between training and test performance, especially in case of last-minute errors.

The Kaggle competition link is:
`https://www.kaggle.com/t/b5fc642379a04e07b5d617f412b010cb`

5.2 Blackboard Submission

In addition to the Kaggle submission, you must make a final submission to Blackboard. Submit a single `.zip` file containing the following materials:

- main.ipynb: Your Jupyter notebook with all source code, comments, cell outputs, and your Kaggle team name and final private leaderboard score.
- report.pdf: Your written report (see earlier instructions), with a screenshot of your Kaggle private leaderboard score displayed at the top of the first page.
- prediction.csv: Your final test set prediction file, in the same format as sample_prediction.csv provided on Kaggle.
- **Signed Declaration Form**: A completed and signed form declaring that the work submitted is your own.
- (Optional) README.txt or README.md: Any additional information that does not fit well in the notebook (e.g., installation notes, external package dependencies).

6 Marking

This assignment carries a total of **100 marks**, distributed across the following components:

- **Satisfactory Submission [10 marks]**: Assesses compliance with submission requirements, including:
  - Inclusion of all required files (code, report, prediction file, declaration form, etc.)
  - Proper naming and organisation of files, as per submission instructions
- **Data Preprocessing [20 marks]**: Awarded based on the depth and effectiveness of preprocessing steps, such as:
  - Identification and treatment of data issues (e.g., missing values, outliers, duplicates)
  - Correct handling of data types and feature scaling
  - Use of advanced preprocessing techniques, such as meaningful feature engineering
- **Model Development [30 marks]**: Evaluates the thoroughness and rigour of the modelling process, including:
  - Exploration of a diverse range of machine learning models (minimum three fundamentally different types)
  - Systematic hyperparameter tuning using techniques like hold-out validation or cross-validation
  - Balanced and fair comparison of model performance
- **Prediction [20 marks]**: Focuses on the final model's performance and related analysis:

- MAPE score of the final model on the Kaggle **private leaderboard** (see Table 1)
- Comparison between training, validation, and test results, with reflections on generalisation
- **Bonus: The top 10 teams on the leaderboard will receive an additional 10 marks**

Table 1: Marking guide based on final testing MAPE (Private Leaderboard)

MAPE	**Mark(s)**
< 0.20	20
0.20 – 0.23	18
0.23 – 0.25	15
0.25 – 0.30	12
0.30 – 0.40	8
0.40 – 0.50	4
> 0.50	0

- **Report [20 marks]**: Evaluates the clarity, completeness, and insightfulness of your report. This includes:
  - Inclusion of all required sections (data cleaning, preprocessing, model development, results, self-reflection, etc.)
  - Clear explanation and justification of decisions made throughout the project
  - Thoughtful analysis, including interpretation of results and lessons learned

Note: While some aspects are not specifically marked, they can substantially influence your scores across sections. For example, the readability and clarity of your Jupyter notebook—including both code and explanatory text—are critical. Poor documentation, confusing code, or unclear analysis may lead to a loss of marks. Aim for clear, well-structured, and well-documented work throughout.

This marks the end of the assignment specification. Good luck, and enjoy the process!

References

[1] Jingde Li, Qilin Li, Hong Hao, and Ling Li. Prediction of bleve blast loading using cfd and artificial neural network. *Process Safety and Environmental Protection*, 149:711–723, 2021.

[2] Qilin Li, Yang Wang, Yanda Shao, Ling Li, and Hong Hao. A comparative study on the most effective machine learning model for blast loading prediction: From gbdt to transformer. *Engineering Structures*, 276:115310, 2023.