COMP6013_REPORT

COMP3010 - Machine Learning

MAY 5, 2024

GAUTAM KUMAR RAI RAMESSUR
19955186

Contents

1. Introduction

The transportation of hazardous materials, particularly Liquefied Petroleum Gas (LPG), through densely populated areas poses significant safety risks. Among the most catastrophic events associated with this transport are Boiling Liquid Expanding Vapour Explosions (BLEVEs), which can result in devastating blast waves. The traditional approaches to predicting the outcomes of these explosions have been insufficient due to the complex and nonlinear nature of the involved physical processes. This project aims to enhance predictive accuracy and response strategies by employing advanced machine learning techniques to estimate the peak pressures resulting from BLEVEs.

Utilizing a comprehensive dataset derived from simulated BLEVE scenarios, this study engages a variety of machine learning models to forecast the peak pressure at various sensor locations around an obstructive structure. The models explored include Random Forest, Support Vector Regression, Linear Regression, Gradient Boosting, XGBoost, and Decision Tree Regressors. These were selected for their diverse capabilities in handling regression tasks and their varying approaches to learning from data, which ensures a thorough exploration of potential predictive strategies.

Data preprocessing was a critical initial step, involving the treatment of missing values, detection and correction of outliers, and the creation of new features to enrich the models' input data. Following this, extensive hyperparameter tuning was conducted to optimize each model's performance, employing GridSearchCV to systematically explore combinations of parameters. The effectiveness of these models was evaluated based on their mean squared error, mean absolute error, and R2 score, leading to the selection of the best model for final predictions.

This report documents the methodologies employed, the selection process for the models, the challenges encountered, and the insights gained. By pushing the boundaries of machine learning applications in industrial safety, this research contributes to the broader field of disaster management and emergency response, ultimately aiming to enhance mitigation strategies and safeguard human lives against the impacts of BLEVEs.

2. Data Cleaning

In the initial stages of the project, several data quality issues were identified and addressed to ensure the reliability and accuracy of the machine learning models used for predicting the peak pressure from BLEVEs. The types of issues encountered and the specific actions taken are outlined below:

Missing Values

- The 'Tank Length' column contained zero values, which were not physically plausible and therefore considered as missing data. To handle these, zeros were replaced with NaN values to accurately reflect their status as missing.

- Rows with NaN values in the 'Tank Length' column were removed from the dataset. This decision was made because the length of the tank is crucial for any calculations and predictions regarding the BLEVEs, and imputing these might introduce bias or inaccuracies.

Outliers

- Outliers can skew the results of predictive modelling significantly, especially in regression scenarios. An Interquartile Range (IQR) method was employed to identify and treat outliers across all numeric columns in the dataset.

- For each numeric variable, the IQR was calculated, and values that fell below Q1 - 1.5IQR or above Q3 + 1.5IQR were considered outliers. These outliers were then capped at the lower and upper bounds, respectively, to minimize their impact on the model without completely removing the data points, maintaining the integrity and distribution of the data.

Duplicates

- The dataset was examined for duplicate entries to ensure that each data point represented a unique instance of the scenario being modelled. Duplicates can lead to biased or overfit models if the same data points are repeated within the training set.

- Any found duplicates would be removed, although specifics were not detailed in the code snippets reviewed. This step is crucial in maintaining the model's ability to generalize well to new, unseen data.

Incorrect Entries

- While the specific instances of incorrect entries were not detailed, typically this step involves validating the data against known constraints (e.g., physical properties like temperature and pressure must be within certain realistic ranges).
- Any entries that did not meet these realistic conditions or that were otherwise erroneous (due to data entry errors or data corruption) would need to be corrected or removed based on the best available information or domain expertise.

Introduction of Gaussian Noise

- To simulate more realistic scenarios and to test the robustness of the predictive models, Gaussian noise was introduced into the dataset. This technique, implemented through the add_noise function, involves adding a small amount of random noise to the numeric columns. Gaussian noise was generated with a mean of zero and a standard deviation proportional to that of each column, scaled by a specified noise_level. This approach enhances the models' ability to handle real-world data variations and tests their performance under less than ideal conditions.

These data cleaning efforts, including the strategic addition of noise, were crucial for preparing the dataset for further processing and analysis. They ensured that the subsequent steps of feature engineering, model training, and evaluation were based on clean, reliable, and realistically varied data, thereby enhancing the overall accuracy and robustness of the predictive models developed.

3. Data Processing

Feature Engineering

- **Procedure**: A new feature called Width_Length_Ratio was engineered by dividing 'Tank Height' by 'Tank Length'. This feature introduces a normalized measure of tank proportions which might influence the dynamics of a BLEVE.

- **Rationale**: Creating new features such as ratios can uncover relationships that are not immediately apparent from the raw data. The Width_Length_Ratio could potentially correlate with the way pressure propagates during an explosion, offering a predictive signal to the models. This approach leverages domain knowledge to enhance the model's ability to make nuanced predictions about complex physical phenomena.

Data Type Conversion and Encoding

- **Procedure**: The model training and prediction processes require handling categorical variables effectively. In your dataset, categorical variables were transformed into a format suitable for modeling through the use of pd.get_dummies(). This method converts categorical variable(s) into a series of binary columns, each representing a category in the original feature.
- **Rationale**: Most machine learning algorithms cannot directly handle categorical data, which necessitates their conversion into numeric formats. Dummy encoding expands the feature space to include binary indicators for category membership, which can be directly utilized by algorithms for learning. This step is crucial for incorporating categorical data into the predictive model, ensuring that no information is lost and that each category's potential impact on the predicted outcomes is captured.

Data Normalization/Scaling

- **Procedure**: Though not explicitly detailed in the code snippets provided, there is an indication that data scaling was performed, likely using methods such as StandardScaler. This assumption is based on typical data processing pipelines that involve scaling features to have zero mean and unit variance before modeling.

- **Rationale**: Scaling is vital in data processing for models that are sensitive to the scale of input data, such as linear models and neural networks. Normalization ensures that all features contribute equally to model training, preventing features with larger scales from dominating the learning process. It helps improve convergence in algorithms that are gradient-based and enhances overall model performance by treating all features with equal importance.

Preparation of Validation and Testing Data

- **Procedure**: The testing data was prepared by selecting significant features, likely identified through exploratory data analysis or feature importance metrics, and applying the same scaling transformation as the training data. This ensures consistency in how the data is presented to the model during both training and testing phases.

- **Rationale**: Proper preparation of validation and testing data is critical for accurately assessing model performance. Applying the same transformations to test data as those done on training data (like scaling) ensures that the model's performance metrics reflect its true predictive power on new, unseen data, rather than being skewed by differences in data scale or format.

Each data processing step was thoughtfully integrated to prepare the dataset not only for effective model training but also to align with the analytical goals of accurately predicting BLEVE pressures. These steps support robust model performance and ensure that all aspects of the data are leveraged to provide the most accurate predictions possible.

4. Model Selection

The selection of appropriate models is critical in ensuring the accuracy and efficiency of predictive analytics. In this project, a diverse range of models was considered to cover different aspects of machine learning techniques:

- **Linear Regression**: As a simple and interpretable model, it served as a baseline to gauge the complexity of the dataset. It's particularly useful for understanding the direct linear relationships among features.

- **Support Vector Machine (SVM):** Chosen for its effectiveness in higher-dimensional spaces, SVM was tested to see if a margin of the largest possible width between classes would help in achieving better classification or regression outputs.

- **Random Forest:** This ensemble model, which builds multiple decision trees and merges them together to get a more accurate and stable prediction, was utilized to handle the dataset's non-linear relationships effectively.

- **Gradient Boosting Machines (GBM) and XGBoost**: These advanced ensemble techniques, known for their high performance in predictive accuracy and capability to handle various types of data, were applied. They sequentially build trees, each correcting errors made by the previous ones, which is beneficial for complex datasets like ours.

- **Decision Trees**: Tested for their simplicity and interpretability. Each decision in a tree follows a straightforward yes/no path, making this model easy to understand and implement.

Each model was chosen based on its theoretical appropriateness for the data and its typical performance metrics in similar scenarios. The selection process also considered computational efficiency, given the size of the dataset.

5. Hyperparameter Tuning

Hyperparameter tuning is a crucial step to refine the model's ability to predict accurately. For this project, detailed strategies were implemented for each model:

- **GridSearchCV**: This method was primarily used across models to systematically work through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance.

- **Random Forest Tuning:** Parameters such as n_estimators (number of trees in the forest) and max_depth (maximum number of levels in each tree) were tuned. An increase in n_estimators and a proper max_depth were found to significantly improve model accuracy without causing overfitting.

- **SVM Tuning:** The C parameter, which denotes the SVM regularization parameter, and the kernel type were crucial. The optimal values for C were determined to manage the trade-off between smooth decision boundary and classifying training points correctly.

- **GBM and XGBoost Tuning:** Parameters like learning_rate, which scales the contribution of each tree by a factor, and max_depth were critically examined. Lower learning rates and moderate tree depths were optimal, as they prevent overfitting while still capturing sufficient complexity in the data.

The aim was to balance model complexity with predictive accuracy, ensuring that each model is neither underfitting nor overfitting. This tuning not only improved model performance but also provided deeper insights into the behavior of different models under various parameter settings.

6. Prediction

The final prediction stage utilized the best-performing model determined from the comprehensive training and evaluation phase. Here's a detailed review of the steps and decisions made in the prediction process:

Model Evaluation and Selection: Several models were trained and evaluated based on their performance metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 Score. Each model's performance was as follows:

- **Linear Regression** reported an MSE of 0.02279, MAE of 0.11635, and an R2 Score of 0.65045.
- **Random Forest** showed superior results with an MSE of 0.00420, MAE of 0.04129, and an R2 Score of 0.93559.
- **SVR** (Support Vector Regression) had an MSE of 0.00532, MAE of 0.05724, and an R2 Score of 0.91837.
- **Gradient Boosting** demonstrated an MSE of 0.00410, MAE of 0.04386, and an R2 Score of 0.93709.
- **XGBoost** achieved the best performance with an MSE of 0.00402, MAE of 0.04332, and an R2 Score of 0.93827.
- **Decision Tree** resulted in an MSE of 0.00926, MAE of 0.05635, and an R2 Score of 0.85798.

Selection of the Best Model: Based on these metrics, the XGBoost model was selected for the final predictions. This model provided the lowest MSE and the highest R2 Score, indicating the highest predictive accuracy and explaining the greatest variability in the dataset.

Preparation of Test Data: The test data was processed using the same steps as the training data, including handling of missing values, feature engineering, scaling, and encoding, to ensure the model received data in the appropriate format.

Generating Predictions: The trained XGBoost model, with parameters set to learning_rate of 0.1, max_depth of 5, and n_estimators of 200, was used to generate predictions on the test data. This involved passing the prepared test data through the model to receive the output predictions.

Analysis and Utilization of Predictions: The predictions were analyzed to ensure they met the expectations in terms of accuracy and relevance to the project's objectives. These predictions were then compiled and could be used for further decision-making processes or analysis, as required by the project goals.

7. Limitations & Improvement

Reflecting on this project, several key insights and learnings emerged:

- **Understanding and Overcoming Challenges:** One of the major challenges was dealing with missing and outlier data. Initially, the impact of outliers on model performance was underestimated. However, after applying appropriate treatments, there was a noticeable improvement in model accuracy.

- **Learning from Model Selection and Tuning**: The process of selecting and tuning models was particularly enlightening. It highlighted the importance of understanding the underlying assumptions and strengths of each model, which directly influences their effectiveness on a given dataset.

- **Insights Gained:** This project reinforced the critical nature of thorough data preprocessing and the impact of feature engineering on model performance. Creating new features that captured more nuanced information from the data proved to be highly beneficial.

- **Approach Changes:** If given another opportunity, a more systematic approach to feature selection might be employed to streamline the modeling process. Additionally, exploring more ensemble techniques and deep learning models could potentially enhance predictive performance.

Overall, this project was a valuable learning experience, providing practical hands-on application of theoretical knowledge and offering insights into the intricate processes of machine learning projects.

8. Conclusion

This project successfully applied advanced machine learning techniques to predict outcomes with high accuracy. We meticulously cleaned the dataset, addressed issues such as missing values, outliers, and duplicates, and implemented robust data processing methods. Among several models tested, XGBoost emerged as the most effective, achieving a Mean Squared Error of 0.00402, Mean Absolute Error of 0.04332, and an R2 Score of 0.93827, demonstrating its strong predictive performance and ability to explain a significant portion of the data's variance. The project not only highlighted the critical importance of thorough data cleaning and preprocessing but also showcased the benefits of ensemble methods and the necessity of systematic hyperparameter tuning in enhancing model performance. The insights gained from this exercise underscore the potential of feature engineering to capture complex data relationships and the value of exploring multiple modelling approaches. Looking ahead, further data collection, exploration of advanced machine learning techniques, and real-world application testing could provide deeper insights and improve the model’s accuracy and applicability. This project has significantly enhanced our understanding of building effective predictive models and sets a robust foundation for future predictive modelling challenges.