Predicting LPG BLEVE Blast Wave Pressure using Machine Learning

Doyeon Lee, Omkar Yawale, Miguel Freyermuth

GitHub Repository: https://github.com/omkary14/CHE1147-Final-Project---Group-5/tree/main
Video Presentation: https://www.youtube.com/watch?v=Wt5W_X9MN_s

Abstract

The prediction of Liquified Petroleum Gas (LPG) explosions together with their resulting blast wave remains essential since a single explosion can lead to devastating consequences. Traditional simulation methods based on Computational Fluid Dynamics (CFD) generate remarkable outcomes but often require high computing power and they extend emergency response times. This project develops a machine learning system that efficiently predicts Boiling Liquid Expanding Vapor Explosions (BLEVEs). The model is trained on data from 8000+ instances with 23 features, including dimensions of the obstacle, gas characteristics, and tank-to-obstacle distance. To understand how different factors influence Target Pressure, Exploratory Data Analysis (EDA) was performed using Analysis of Variance (ANOVA) tests and correlation heatmaps. Data cleaning was done by the Interquartile Range (IQR) method to remove outliers, and stratification of train-test split was carried out in three groups. Linear regression and random forest models were used as a baseline, and XGBoost was selected as the final choice, since it handles the nonlinear relationships of the data much better. The XGBoost model performed well, with R^2 values of 0.8822 on the training set, and 0.8601 on the test set, with mean absolute error of 0.0438 and 0.474, and root mean square error of 0.0650 and 0.0706, respectively, confirming that XGBoost was the best model for this problem.

1. Introduction

Liquified Petroleum Gas (LPG), known for its flammable properties, can cause severe explosions and the resulting pressure waves can lead to catastrophic consequences. Although efforts have been made to predict the explosions and their blast wave pressures, reliable and efficient prediction of Boiling Liquid Expanding Vapor Explosions (BLEVEs) remains a challenge in practice. [1] Traditional simulation methods such as Computational Fluid Dynamics (CFD), provide comprehensive insights into the BLEVE behavior, but their high computational demands limit their real-time and large-scale applicability. [2]

This project aims to develop a robust machine learning model with an R^2 value higher than 0.85, capable of accurately predicting blast wave pressures generated by LPG explosions in the presence of an obstacle. The model was trained on a dataset of 8000+ instances with features including obstacle dimensions, gas properties, and distance from the tank. Such a model would enable fast and accurate blast wave pressure predictions, supporting preventive safety measures and emergency planning.

2. Methods

2.1. Dataset, Features, and Target

The dataset “Predicting Blast Wave Pressure from LPG Transport” from Kaggle comprises 10,043 records and 25 attributes. This includes data about the sensor location, the distance of an obstacle from the position of the blast, the pressure of the blast at the position of the sensor, and the dimensions of a wall present between the blast source and the sensor. The model was trained with 23 features, (See Table A.1. in Appendix A for all features). Categorical variables were one-hot encoded, and continuous variables were standardized. The target variable was Target Pressure (bar), and the task was framed as a regression problem.

2.2. Exploratory Data Analysis (EDA)

The analysis of data features and target variable needed EDA to gain a better understanding. Histograms for selected features and the target were plotted as Fig. 1 (See Fig. B.1 in Appendix B for all continuous variables) A near-normal distribution was observed for ‘Sensor Position y’. More importantly, our target variable had a right-skewed distribution, necessitating a stratified train test split for more accurate representation of data.

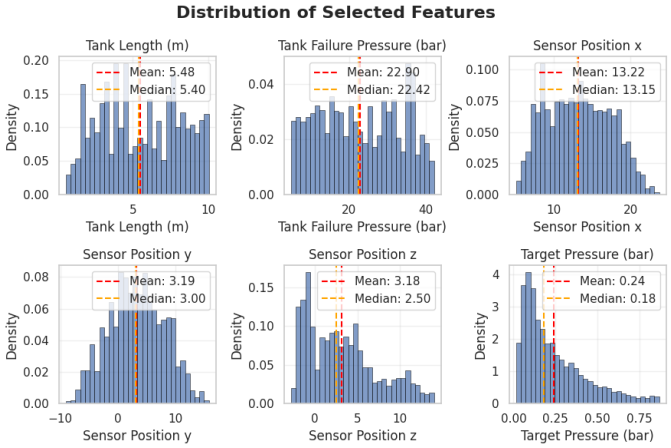

Fig. 1. Distribution of selected features and the target variable.

To visualize and determine the statistical significance of categorical variables, boxplots and Analysis of Variance (ANOVA) were conducted. Fig. 2 shows one example for the status superheated/subcooled along with the ANOVA results in Table 1. (Fig. B.2 and B.3 in Appendix B for boxplots of all categorical variables) The p-value was less than 0.05 for all variables, confirming their statistical significance.

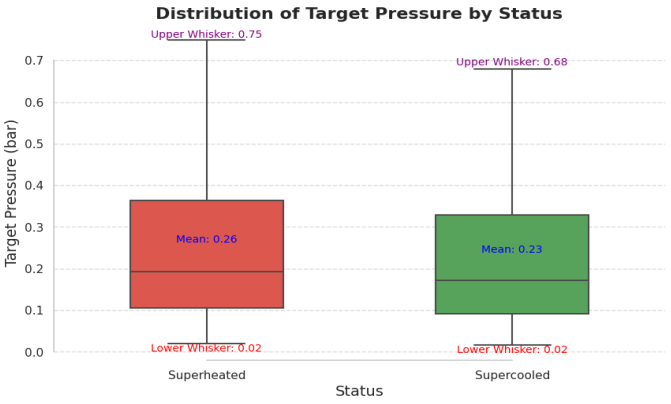

Fig. 2. Status boxplot.

Table 1. ANOVA results for categorical variable significance.

Feature	df	sum_sq	mean_sq	F	PR(>F)
Status	1.0	1.293074	1.293074	36.223233	1.832060e-09
Sensor Position Side	4.0	150.442749	37.610687	2060.033153	0.000000e+00
Liquid Critical Pressure (bar)	1.0	0.544795	0.544795	15.224150	9.620692e-05

Finally, a correlation heatmap was used to measure the linear relationship between the features and the target variable. As seen in Fig. 3 the correlations of the features and the target variable lack linearity, indicating the need for a complex, non-linear model. (Fig. B.4. in Appendix B for complete correlation heatmap)

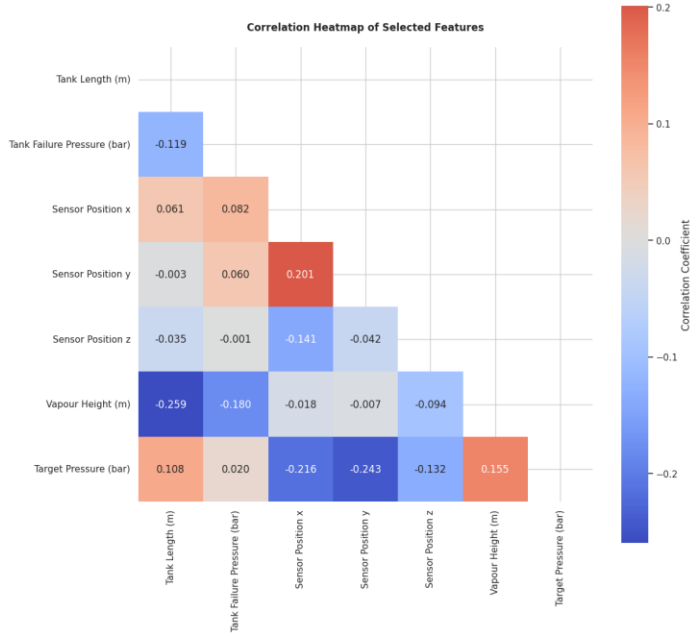

Fig. 3. Correlation heatmap for selected features and target variable.

2.3. Data Cleaning, Encoding, and Train-Test Split

Prior to model training, the dataset underwent a series of cleaning steps. As the dataset was large, rows containing missing entries and rows with duplicate values were dropped. Outliers were identified and removed using the Interquartile Range (IQR) method, where values lying outside the range $[Q1 - 1.5 \cdot IQR,\ Q3 + 1.5 \cdot IQR]$ were considered extreme. This was done to reduce skewness and bias they may introduce and to improve model robustness. Continuous variables were standardized to ensure comparability and prevent biases caused by differences in magnitude among the features.

Categorical variables in the dataset, namely, Liquid Critical Pressure, Liquid Boiling Temperature, Liquid Critical Temperature (all of which contained only two numeric values), Sensor Position (contained five values), and Status (either superheated or subcooled), were transformed using one-hot encoding. After encoding, the feature space expanded from 23 to 30 variables.

Finally, as observed in Fig. 1., to capture the skewness of the target variable in the model, a stratified train-test split (80-20) was carried out with three groups for the target variables in the ranges [0-0.3], [0.3-0.6], and [0.6-max(y)] to reflect the distribution of the data based on visual inspection of the target variable histogram. (See Fig. B.5. in Appendix B)

2.4. Baseline Models

2.4.1. Linear Regression Model

Despite not having strong linear correlation coefficients, a linear regression model was attempted regardless, as it serves as a common benchmark in machine learning research due to its ease of use and interpretability. In addition to providing preliminary insights into feature-target relationships, linear regression offers a clear benchmark for assessing the added value of more complex models. This baseline guarantees that any subsequent performance gains could be attributed to the application of sophisticated modeling techniques.

2.4.2. Random Forest Model

To capture complex non-linear relationships between the features and the target, the second baseline model selected was a random forest model. Hyperparameter tuning was carried out using RandomSearchCV with 10 iterations and 5-fold cross-validation with the following hyperparameter dictionary: param_dist = "n_estimators": [600], "max_depth": [8, 10, 12], "min_samples_split": [40, 45, 50], "min_samples_leaf": [15, 20, 25], "max_features": ["sqrt", 0.6, 0.7, 0.8], "bootstrap": [True]. This ensemble model helps prevent overfitting, resulting in a suitable baseline.

2.5. Final Model – XGBoost

As the final model, XGBoost (Extreme Gradient Boosting) was employed for its optimized speed and accuracy. XGBoost was chosen for its ability to effectively capture nonlinear feature-target relationships, incorporate regularization to reduce overfitting, and scale efficiently to larger datasets. RandomSearchCV was used for hyperparameter tuning with the parameter dictionary as follows: param_dist_xgb = "n_estimators": [400, 500, 600, 700, 800], "max_depth": [6, 7, 8, 10], "learning_rate": [0.02, 0.03, 0.04, 0.05], "subsample": [0.6, 0.7, 0.8, 0.85, 0.9], "colsample_bytree": [0.6, 0.65, 0.7], "gamma": [0.01, 0.2, 0.3], "reg_alpha": [0.25, 0.3, 0.5], "reg_lambda": [1, 1.5, 2, 2.5, 3]. Given its strong record in machine learning research, XGBoost serves as a robust and high-performing model, demonstrating the added value of advanced ensemble methods compared to the previous baseline models.

3. Results and Discussion

3.1. Baseline Models

3.1.1. Linear Regression Model

The linear regression model metrics demonstrated clear underfitting as R^2 of approximately 0.72 indicates that the model could only capture about 72% of the datapoints in the testing set, as shown in Table 2 . (See Fig. B.6. in Appendix B for actual vs. predicted values) This was expected, as the correlation heatmap did not have strong linear correlation coefficients. Overfitting disregarded as the R^2 values for the train and test set were 0.7201 and 0.7269, respectively, which differed by 0.0068, with the test set performing marginally better than the train set. Mean absolute error values (MAE) were 0.0100 and 0.0097, and root mean square error values (RMSE) were 0.1002 and 0,0986 for train and test set, respectively.

The learning curve, Fig. 4., shows convergence in the R^2 values with the increase in the training set size, indicating that any further addition in data would not have resulted in an appreciably better model. The learning curve also exhibited high variance across cross-validation folds, as demonstrated by the wide shaded regions. This suggests that the model's performance was highly sensitive to the specific training subsets, reflecting instability in capturing consistent linear relationships.

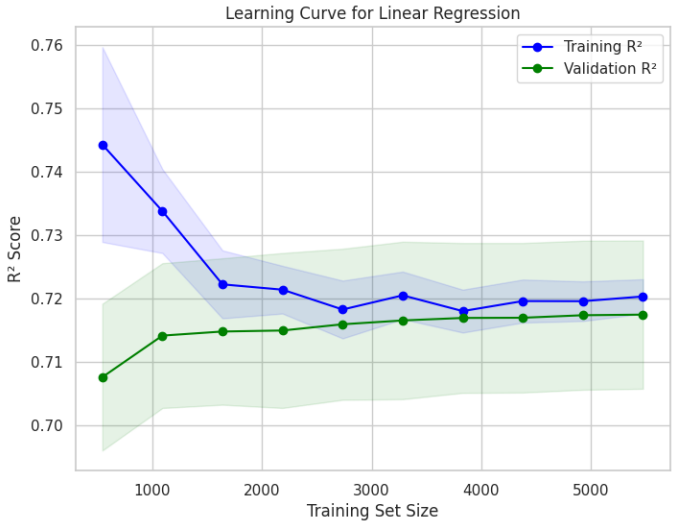

Fig. 4. Learning curve for the linear regression model.

From the bee swarm plot in Fig. 5, the face of the wall on which the sensor is positioned is the most prominent feature for this model, followed by the length of the tank, the tank failure pressure, the y-coordinate of the sensor, and the height of the vapor.

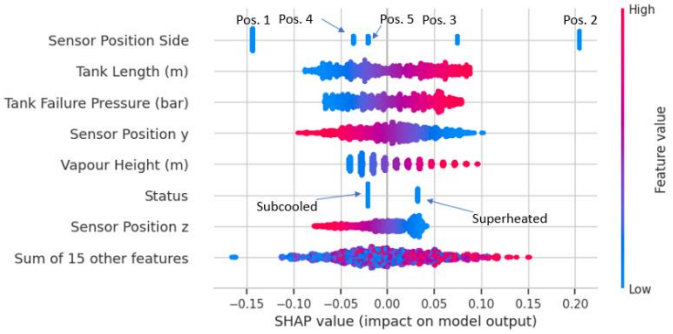

Fig. 5. Feature importance for linear regression.

3.1.2. Random Forest Model

The random forest model metrics indicate better performance than the linear regression baseline as can be seen in Table 2. (See Fig. B.7. for actual vs predicted values) This is due to the model capturing non-linearities better than the previous baseline. Overfitting can be neglected as the R^2 values for the train and test set were 0.8120 and 0.7803, respectively. The difference was 0.0317, with the train set performing marginally better than the test set. MAE values were 0.0557 and 0.0616, and RMSE values were 0.0822 and 0,0884 for train and test set, respectively. The hyperparameters for the model post cross-validation were as follows: 'n_estimators': 600, 'min_samples_split': 50, 'min_samples_leaf': 25, 'max_features': 0.5, 'max_depth': 8.

The learning curve in Fig. 6 shows a lower variance for both the training and the validation set, indicating successful capture of the non-linear complexities in the data. Furthermore, both curves tend to converge but do not quite reach an asymptote, indicating that additional data could provide better result metrics.

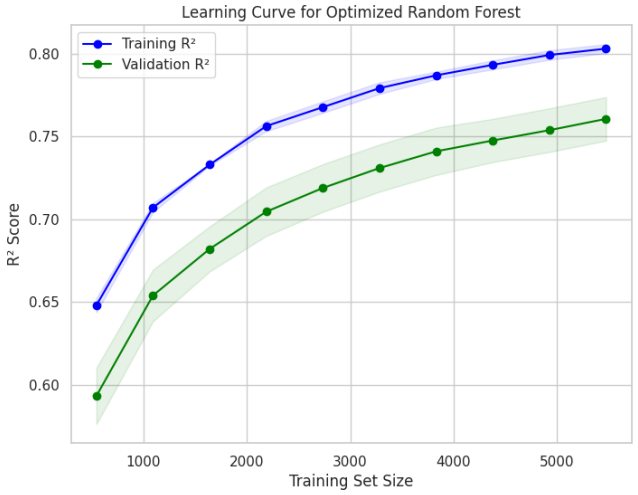

Fig. 6. Learning curve for the random forest model.

The bee swarm plot in Fig. 7 demonstrates that, similar to the linear regression model, the position of the sensor, i.e. the face of the wall on which the sensor is placed, plays the most important role. It is worth noting that the remaining top 7 features differ from the linear model.

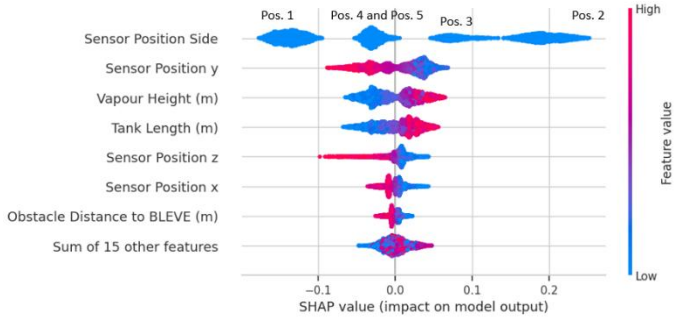

Fig. 7. Feature importance for random forest.

3.2. XGBoost Model

XGBoost model was chosen after establishing baselines with linear regression and random forest due to its superior predictive accuracy, scalability, and integrated regularization, which allowed us to capture complex feature interactions. XGBoost provides the best fit with no overfitting as the R^2 values for the train and test set were 0.8822 and 0.8601, respectively. The difference was 0.0221, with the train set performing marginally better than the test set. MAE values were 0.0438 and 0.0474, and RMSE values were 0.0650 and 0,0706 for train and test set, respectively. (Fig. B.8. in Appendix B for actual vs. predicted values) The hyperparameters used were as follows: "n_estimators": 600, "max_depth": 5, "learning_rate": 0.05, "subsample": 0.6, "colsample_bytree": 0.7, "gamma": 0.1, "reg_alpha": 2, "reg_lambda": 4.

Table 2. Performance metrics for all models

Model	Set	MAE	RMSE	R^2
Linear	Train	0.0100	0.1002	0.7201
	Test	0.0097	0.0986	0.7269
Random Forest	Train	0.0557	0.0822	0.8120
	Test	0.0616	0.0884	0.7803
XGBoost	Train	0.0438	0.0650	0.8822
	Test	0.0474	0.0706	0.8601

The learning curve as indicated in Fig. 8. shows a decrease in variance compared to the random forest model but like random forest, additional data is expected to provide better results.

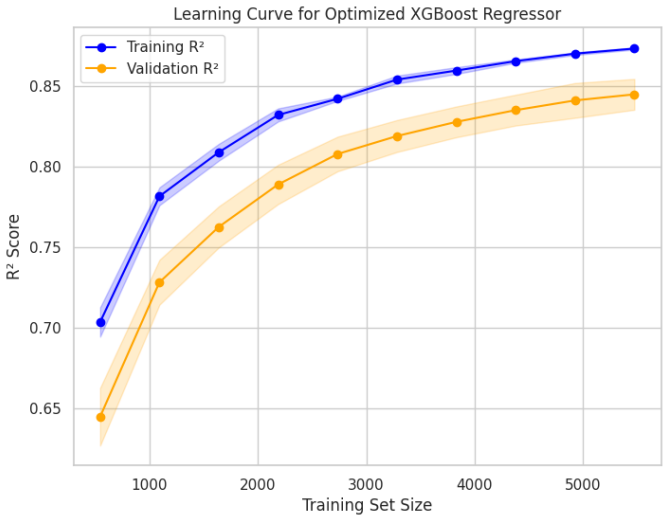

Fig. 8. Learning curve for XGBoost.

The bee swarm plot, as indicated in Fig. 9., shows that the position on the specific wall face where the sensor was mounted again is the most prominent feature, followed by tank length, tank failure pressure, and sensor y-coordinates.

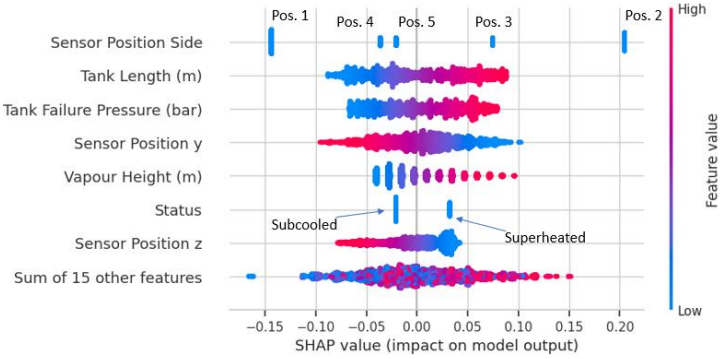

Fig. 9. Feature importance for XGBoost.

4. Conclusion

Upon investigating the impact of 23 different features on blast pressure of LPG's BLEVE and applying XGBoost model to the dataset, the findings demonstrate that the face of the obstacle wall on which the sensor is positioned has the greatest contribution to the value of the blast pressure. This is followed by the length of the tank, the tank failure pressure, and the y coordinate of the sensor, aiding in the appropriate deployment of adequate safety measures around such tanks. Compared to the linear regression model (R^2 = 0.7269) and the random forest model (R^2 = 0.7803), XGBoost achieved an R^2 value of 0.8601 on the test set, thereby achieving our goal of developing a model with an R^2 value higher than 0.85, and validating the effectiveness of choosing XGBoost as the final model. While this study is limited by the number of datapoints, it opens avenues for further research into machine learning model development to predict blast pressures in process plants. We believe that future work could explore the inclusion of additional environmental parameters, testing on real-world BLEVE scenarios, and the application of deep learning methods to further improve prediction accuracy.

5. References

Data Source: ENG. A. Elzoz, “Predicting blast wave pressure from LPG Transport,” Kaggle, https://www.kaggle.com/datasets/engahmedelzoz/volume-of-oil-and-gas-products (accessed Oct. 17, 2025).

[1]: Q. Li, Y. Wang, L. Li, H. Hao, R. Wang, and J. Li, “Prediction of BLEVE loads on structures using machine learning and CFD,” Process Safety and Environmental Protection, vol. 171, pp. 914925, Mar. 2023, Doi: https://doi.org/10.1016/j.psep.2023.02.008.

[2]: Q. Li, Y. Wang, W. Chen, L. Li, and H. Hao, “Machine learning prediction of BLEVE loading with graph neural networks,” Reliability Engineering & System Safety, vol. 241, pp. 109639–109639, Sep. 2023, Doi: https://doi.org/10.1016/j.ress.2023.109639.

Appendix A

Table A.1. Features and their units

Feature	Unit	Feature	Unit	Feature	Unit
Tank Failure Pressure	bar	Liquid Critical Pressure	bar	BLEVE Height	m
Tank Width	m	Liquid Critical Temperature	K	Sensor Position y	m
Tank Length	m	Sensor Position Side	Top, front, etc.	Sensor Position z	m
Tank Height	m	Sensor Position x	m	Obstacle Distance to BLEVE	m
Liquid Ratio	%	Vapor Height	m	Obstacle Angle	
Liquid Temperature	K	Status (Superheated/Subcooled)		Obstacle Thickness	m
Vapor Temperature	K	Liquid Boiling Temp	°C	Obstacle Width	m
Target Pressure	bar	Obstacle Height	m		

Appendix B

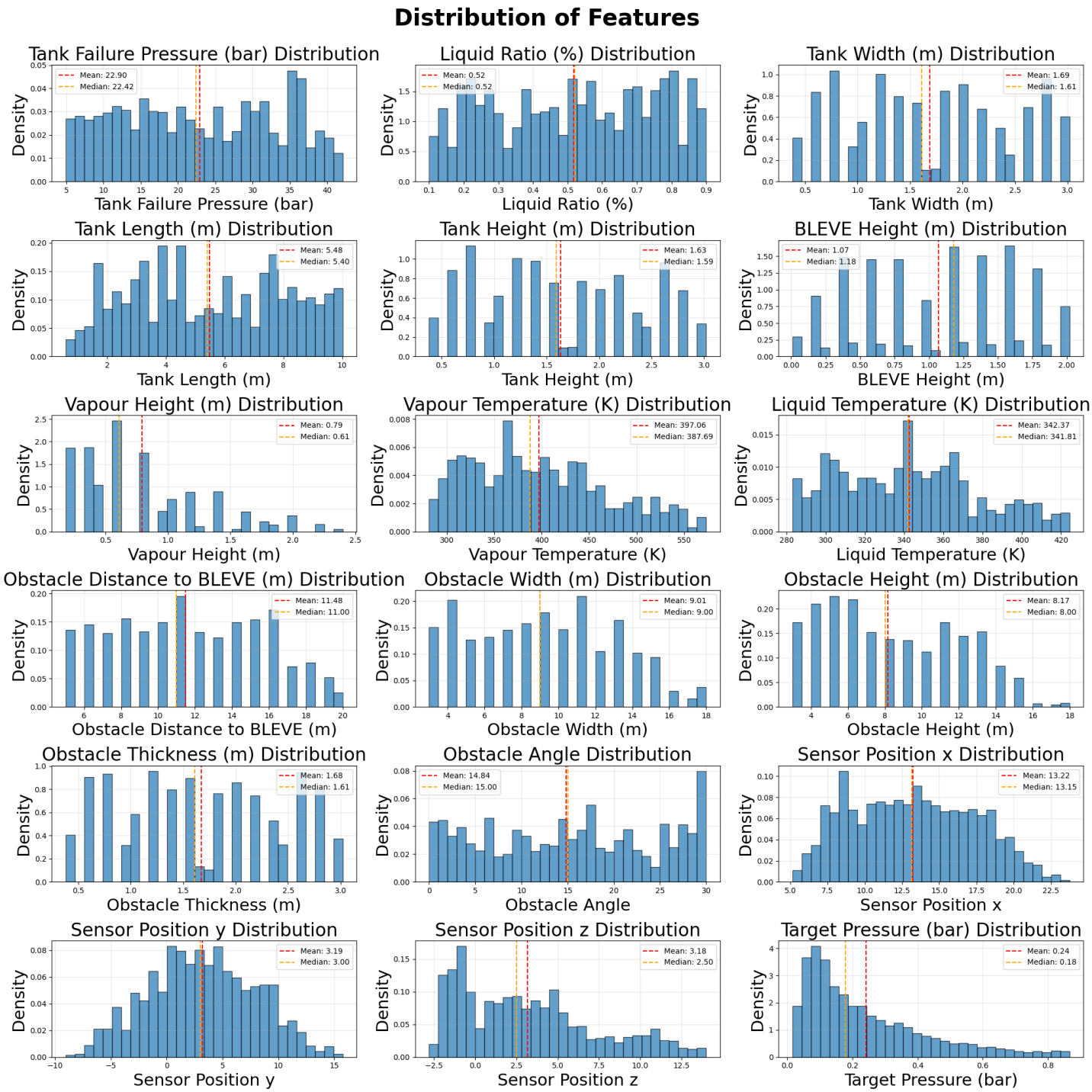

Fig. B.1. Distribution of all continuous features and the target variable.

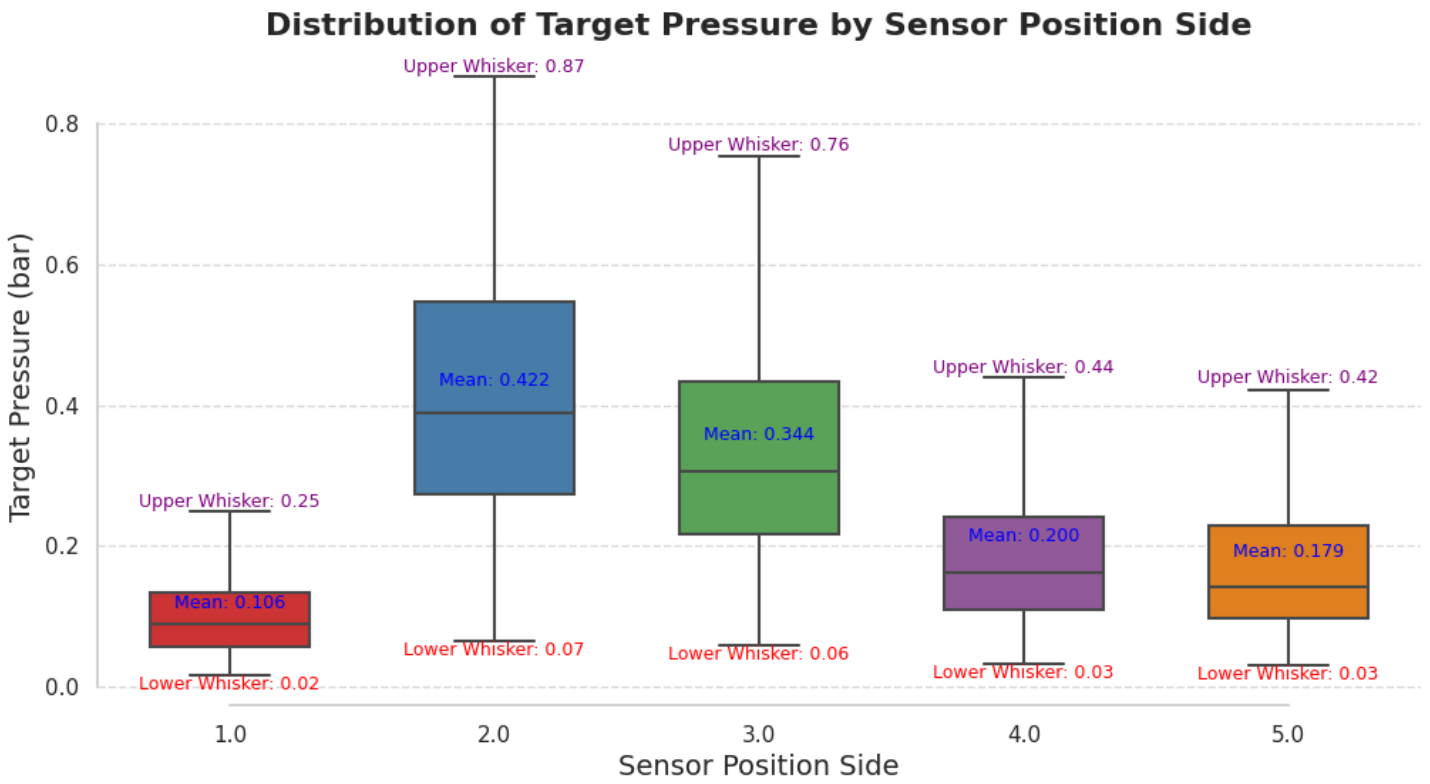

Fig. B.2. Boxplot for Sensor Position Side.

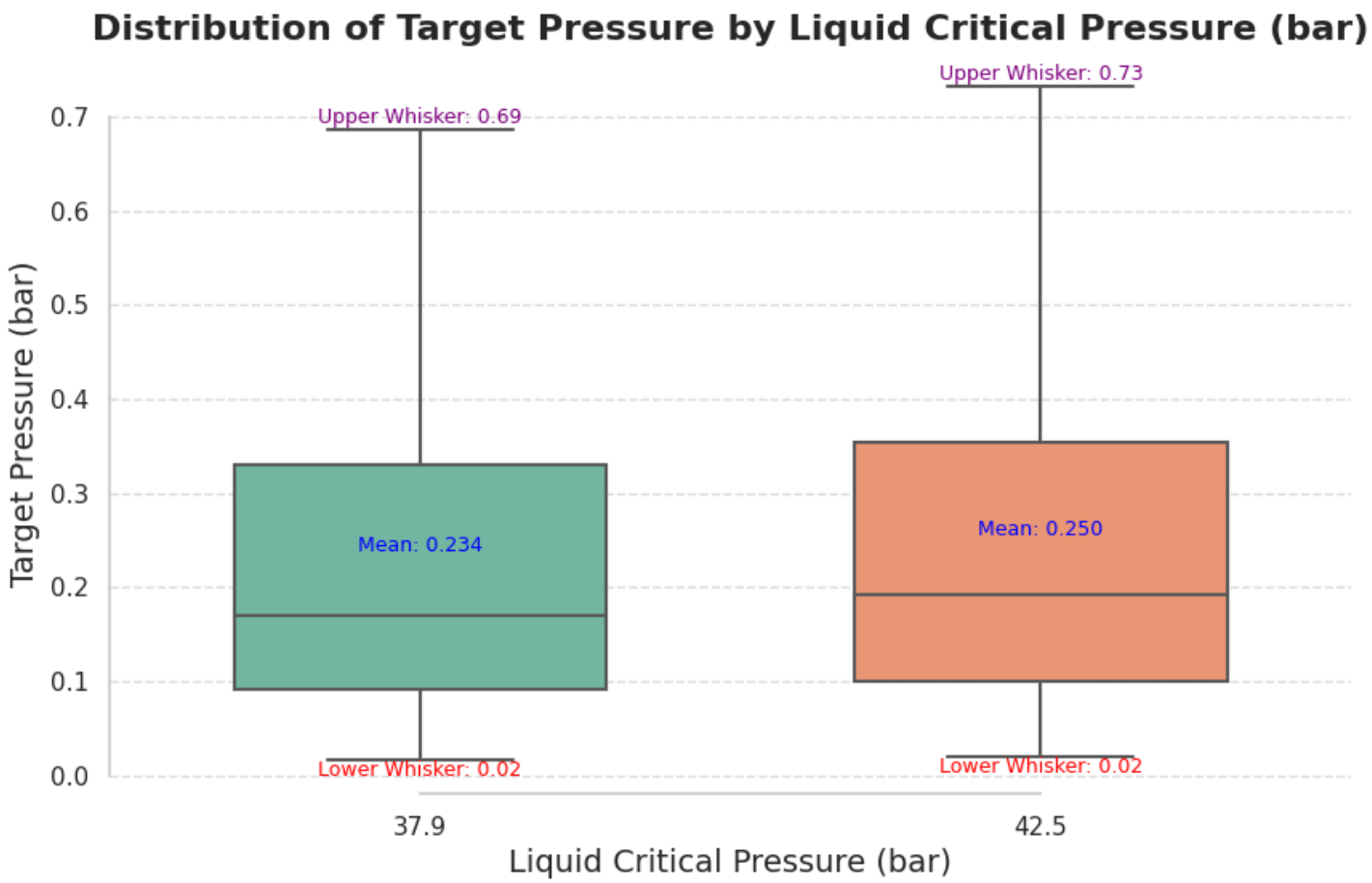

Fig. B.3. Boxplot for Liquid Critical Pressure.

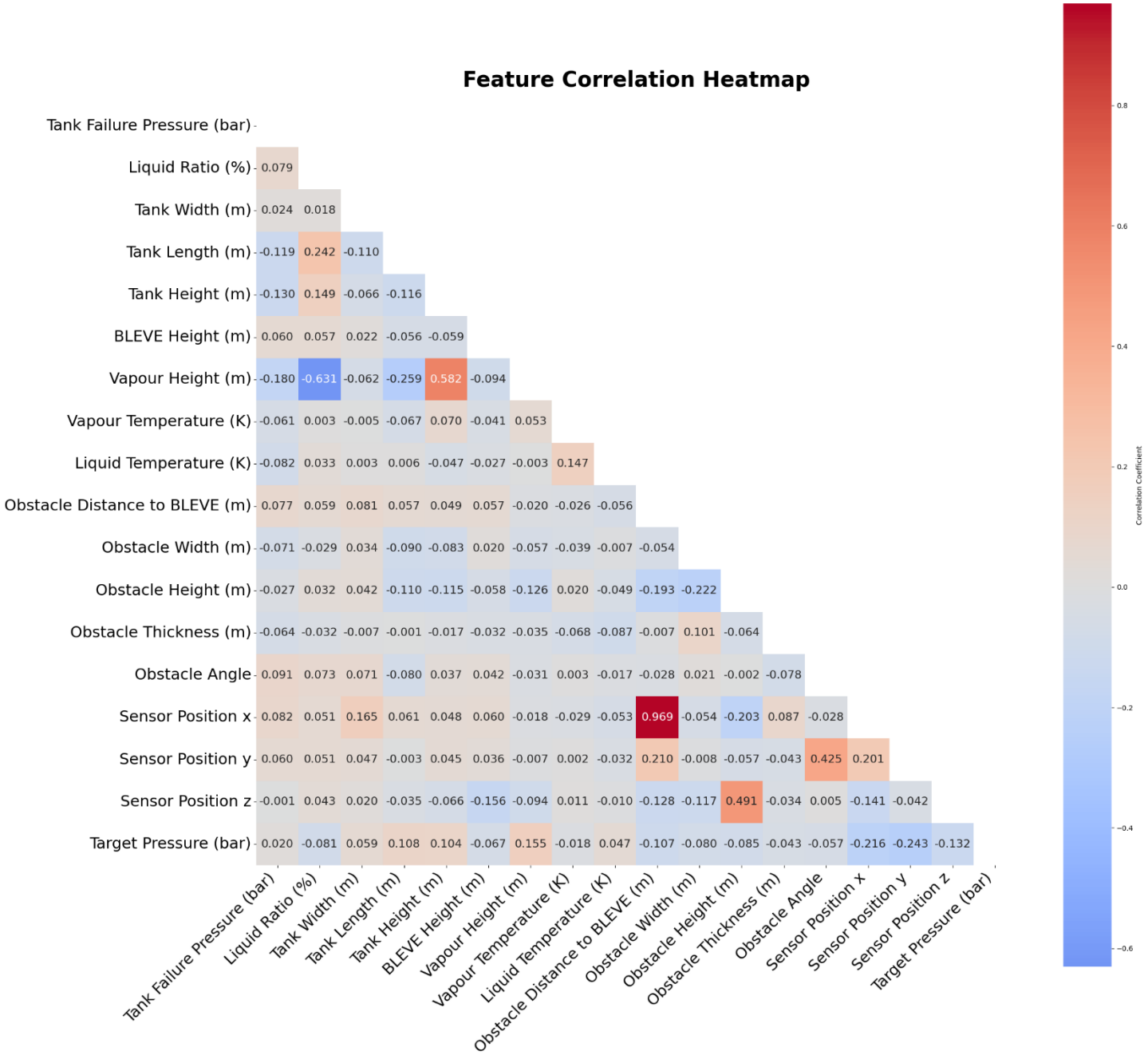

Fig. B.4. Correlation heatmap for all features.

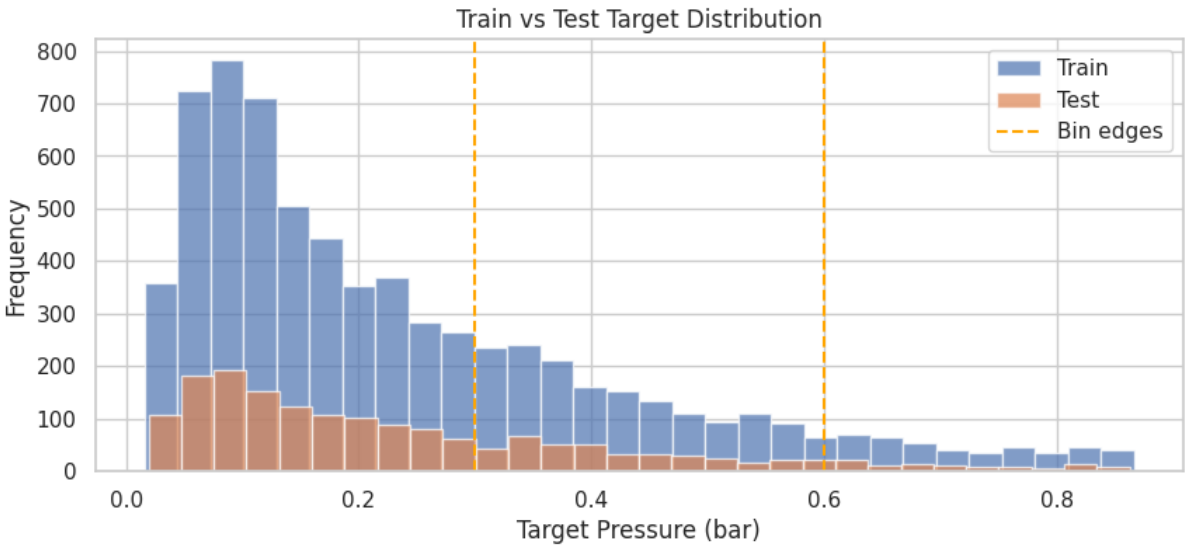

Fig. B.5. Stratified train-test split.

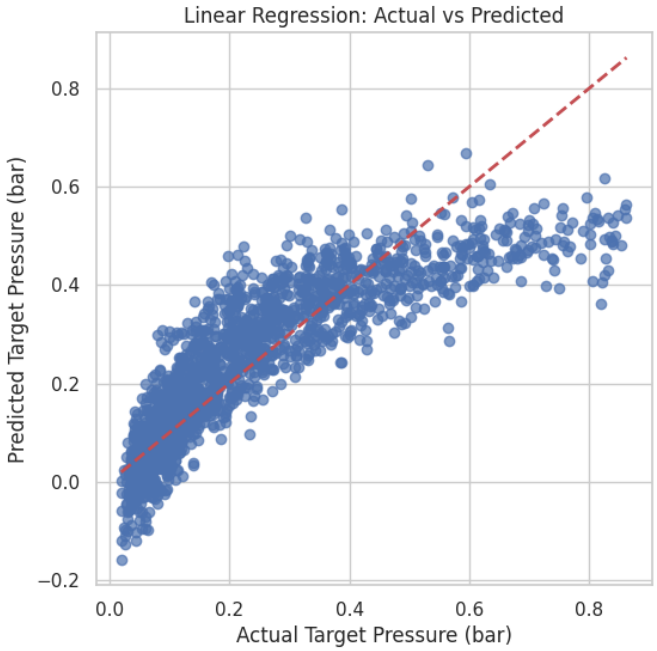

Fig. B.6. Actual vs Predicted values for Linear Regression.

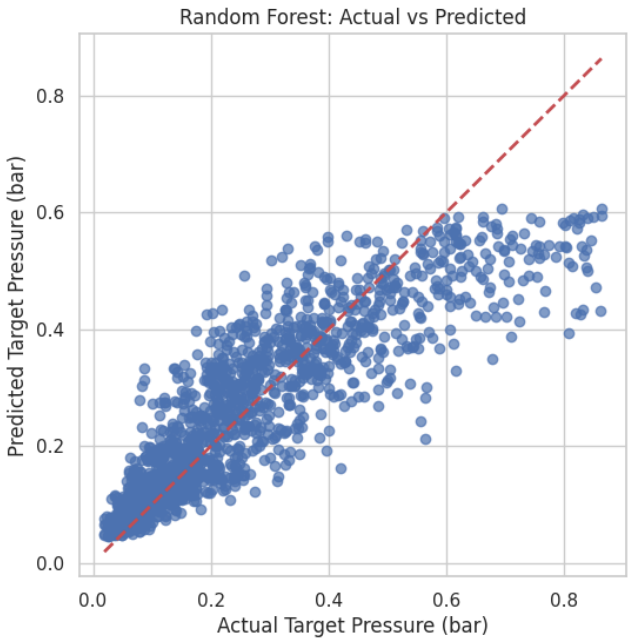

Fig. B.7. Actual vs. Predicted values for Random Forest.

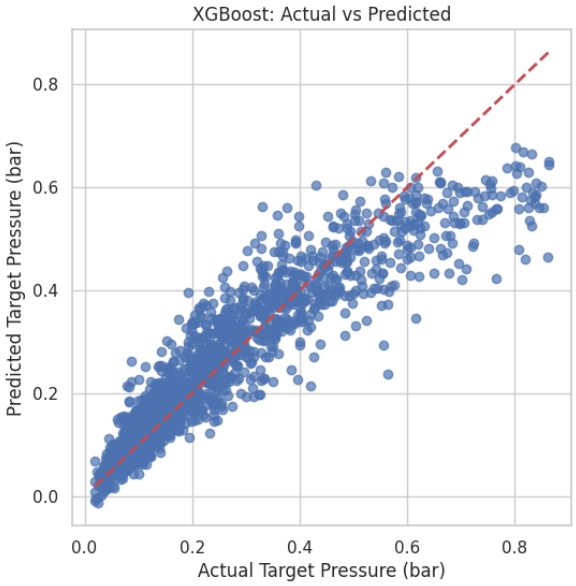

Fig. B.8. Actual vs. Predicted values for XGBoost.