Predictive Analysis of Gas Hold-up in a Bubble Column

Name: Anand Mangal

Roll No.: 210107009

Submission Date: April 25, 2024

Final Project submission

Course Name: Applications of Al and ML in chemical engineering

Course Code: CL653

Contents

1 Executive Summary

Overview: This project focuses on predicting gas hold-up in a bubble column, a critical parameter influencing various chemical engineering processes. The project involves data preprocessing steps such as data cleaning, exploratory data analysis (EDA), outlier analysis, and data scaling. Six different regression algorithms are employed to build predictive models, and their performance is evaluated based on the root mean square error (RMSE).

Problem Statement: The gas hold-up in a bubble column is a crucial factor affecting mass transfer rates, reaction kinetics, and overall process efficiency in chemical engineering applications. Predicting gas hold-up accurately can lead to optimized design and operation of bubble columns, enhancing process control and product quality.

Proposed Solution: The project proposes using machine learning regression algorithms to predict gas hold-up based on multiple input features. These algorithms include linear regression, decision tree regression, random forest regression, support vector regression, gradient boosting regression, and neural network regression.

Methodologies: Initially, thorough data preprocessing is conducted, addressing missing values and outliers to ensure data quality. Exploratory data analysis (EDA) is then performed to gain insights into the relationships between input features and the target variable, gas hold-up. Following this, six regression algorithms were trained and hyperparameter tuning was performed. Finally, a comprehensive comparison and analysis of RMSE values from each algorithm were conducted, enabling the identification of the most effective model for accurately predicting gas hold-up in a bubble column.

Expected Outcomes: The project aims to identify the most accurate regression algorithm for predicting gas hold-up in a bubble column, providing valuable insights into the relationships between input features and gas hold-up. These insights aid in optimizing processes, contributing significantly to predictive modeling advancements in chemical engineering. Ultimately, this work facilitates better decision-making and enhances process efficiency within chemical engineering applications.

2 Introduction

Gas Hold-up refers to the volume fraction of gas in the reactor. In a bubble column, gas bubbles continuously rise through a liquid phase, creating a dynamic system with complex fluid dynamics and mass transfer phenomena. Gas hold-up is a crucial parameter that directly influences mass transfer rates, fluid dynamics, mixing efficiency, and overall reactor performance.

The project's core challenge lies in accurately predicting gas hold-up in bubble columns, crucial for optimizing mass transfer rates and achieving efficient gas-liquid contact. This entails developing predictive models using input features like gas flow rate, liquid properties, and operating conditions, addressing complex nonlinear relationships.

Efficient prediction of gas hold-up is crucial for optimizing the design and operation of bubble columns. It directly impacts process efficiency, reactor performance, and product quality in various chemical and biochemical applications. Engineers can make informed decisions regarding reactor parameters by accurately estimating gas hold-ups, such as gas flow rates, liquid levels, and operating conditions. This leads to improved process control, reduced energy consumption, and enhanced product yields.

Objectives:

1. Develop a Predictive Model

2. **Optimize Reactor Design:** This information can be valuable for optimizing reactor design and operating conditions to achieve desired gas-liquid contact and mixing.

3. **Improve Process Efficiency:** By understanding the factors influencing gas hold-up, we aim to improve overall process efficiency and reduce energy consumption.

4. **Provide a Practical Tool:** Ultimately, the goal is to provide engineers and researchers with a practical tool that can reliably predict gas hold-up in the reactor.

References: I have taken help of the research papers posted on the links given below:

- Predictive analysis of gas hold-up in bubble column using machine learning methods - ScienceDirect
- (PDF) Numerical Analysis of Gas Hold-Up of Two-Phase Ebullated Bed Reactor (researchgate.net)

3 Methodology

I have collected my data from an online article posted on the link below: Dataset for: Predictive analysis of gas holdup in bubble column using machine learning methods - Mendeley Data .

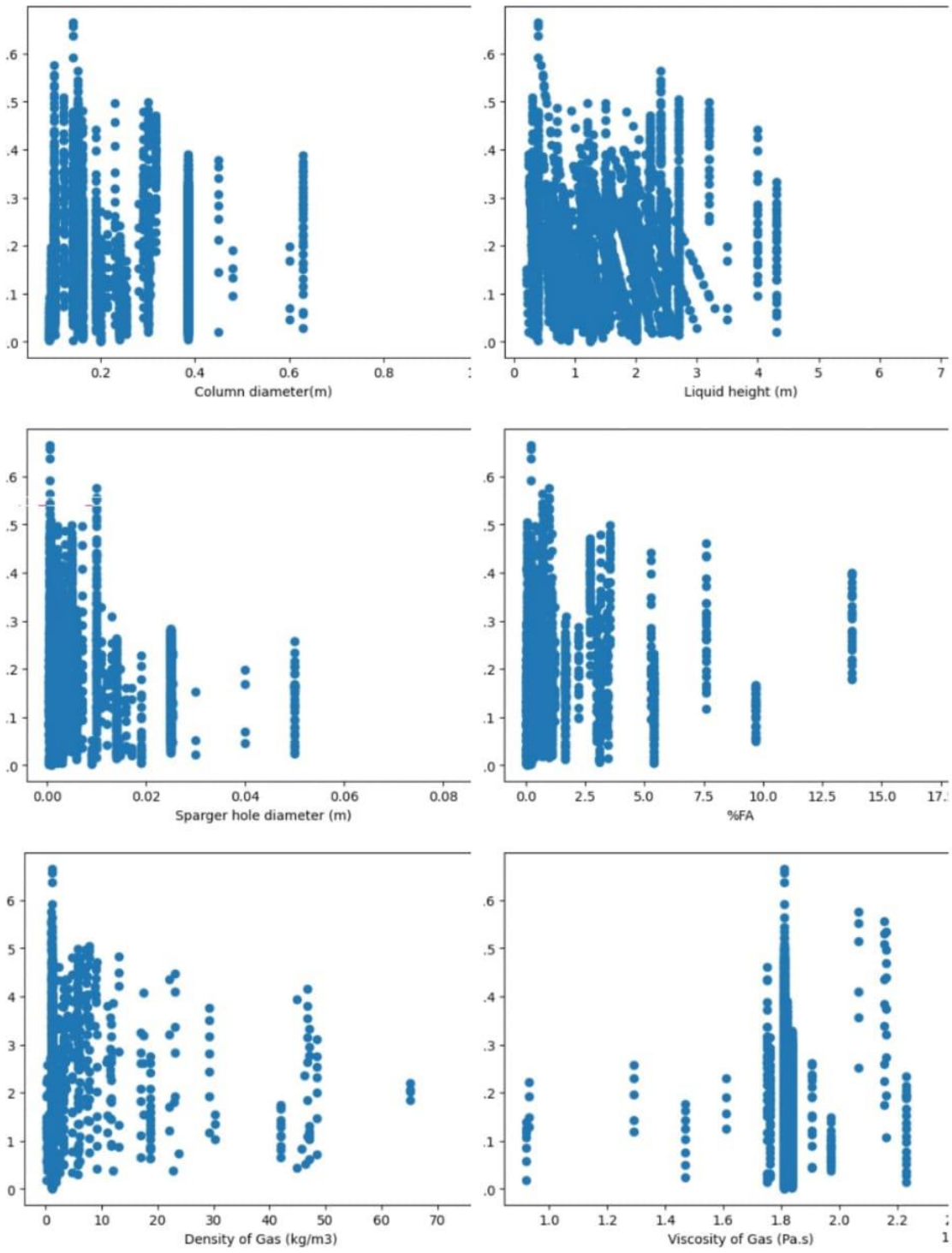
Originally, the data has 4042 rows and 22 columns. The columns names are as follows - 'Authors', 'Gas holdup', 'Column diameter(m)', 'Liquid height (m)', 'Sparger hole diameter (m)', 'Sparger Encode', '%FA', 'Density of Gas (kg/m3)', 'Viscosity of Gas (Pa.s)', 'Molecular Weight of gas (kg/kmol)', 'Density of Liquid (kg/m3)', 'Viscosity of Liquid (Pa.s)', 'Surface Tension of Liquid (N/m)', 'i+ (k.ion/m3)', 'Temperature (K)', 'Pressure (kPa)', 'Superficial gas velocity (m/s)', 'Unnamed: 17', 'Unnamed: 18', 'Unnamed: 19', 'Unnamed: 20', 'Unnamed: 21'.

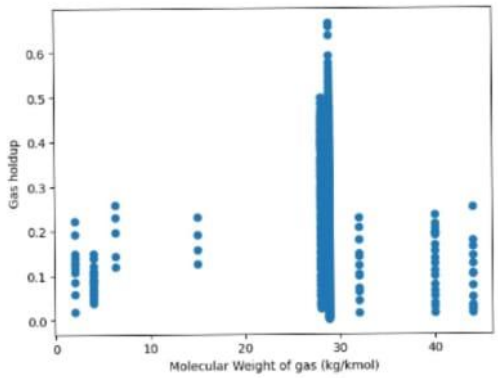
Data Preprocessing -

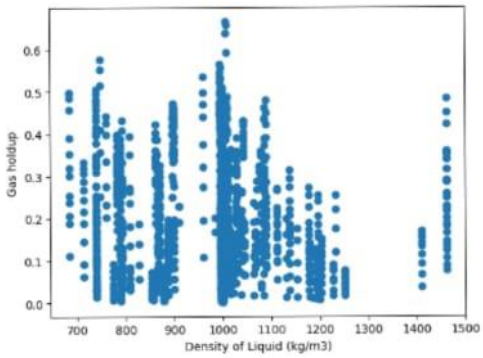
1. **Data Cleaning -** The first step in the preprocessing is data cleaning, which includes removing the redundant columns and making a separate data frame for the Sparger Encoding.

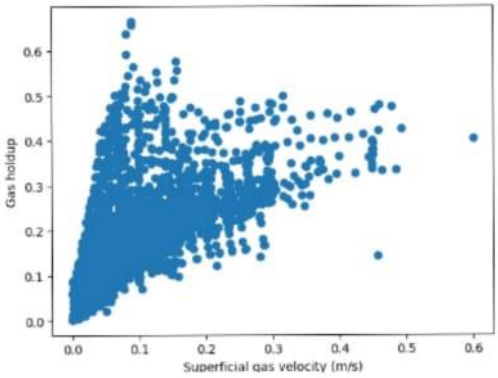
   After this step, the dataset has 4042 rows and 16 columns. The columns left in the dataset are as follows - 'Gas holdup', 'Column diameter(m)', 'Liquid height (m)', 'Sparger hole diameter (m)', 'Sparger Encode', '%FA', 'Density of Gas (kg/m3)', 'Viscosity of Gas (Pa.s)', 'Molecular Weight of gas (kg/kmol)', 'Density of Liquid (kg/m3)', 'Viscosity of Liquid (Pa.s)', 'Surface Tension of Liquid (N/m)', 'i+ (k.ion/m3)', 'Temperature (K)', 'Pressure (kPa)', 'Superficial gas velocity (m/s)'.

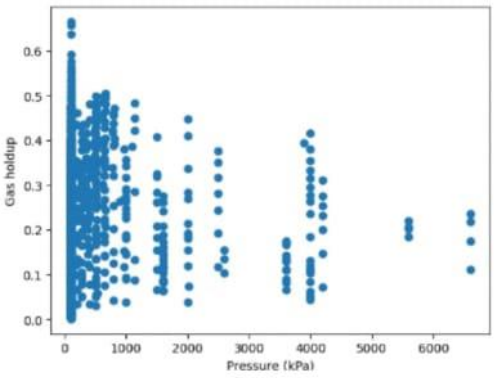
2. **Exploratory Data Analysis (EDA) -** The second step starts with finding the essential data characteristics like its shape, columns, and index list. Further, we describe the overall data by finding the statistical parameters like mean, standard deviation, minimum value, maximum value, and percentile ranges. Then, we move towards checking if the dataset has any null values. We find that it has no missing values but many zeroes in the ion density column, which shows the possibility of the data being preprocessed. Distribution/Scatter plots were plotted of all the input features v/s the target variable.

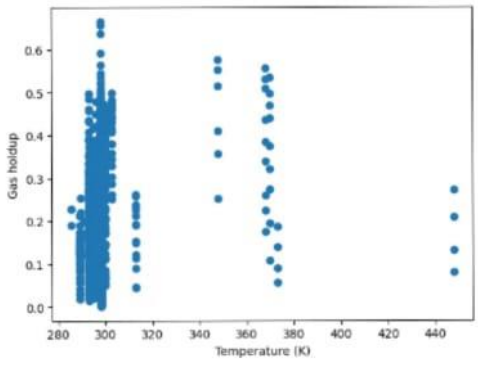

Column diameter(m)
Liquid height (m)
Sparger hole diameter (m)
%FA
Density of Gas (kg/m3)
Viscosity of Gas (Pa.s)

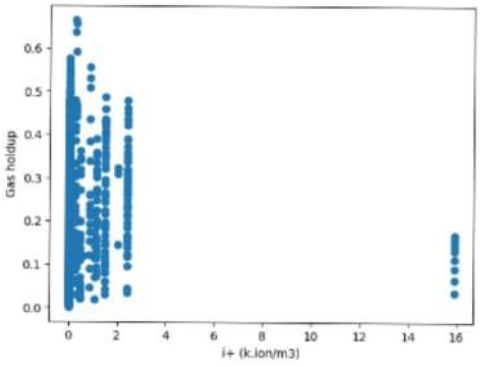


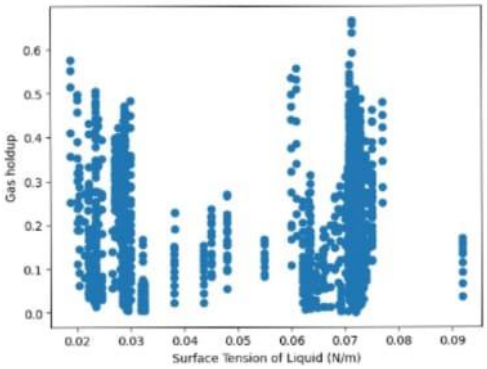


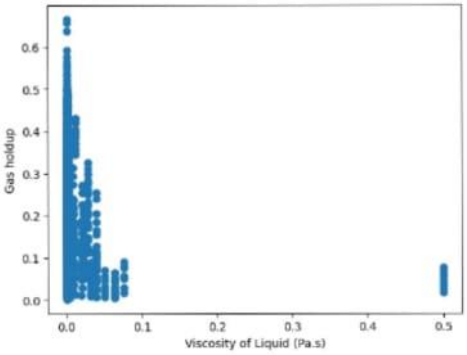






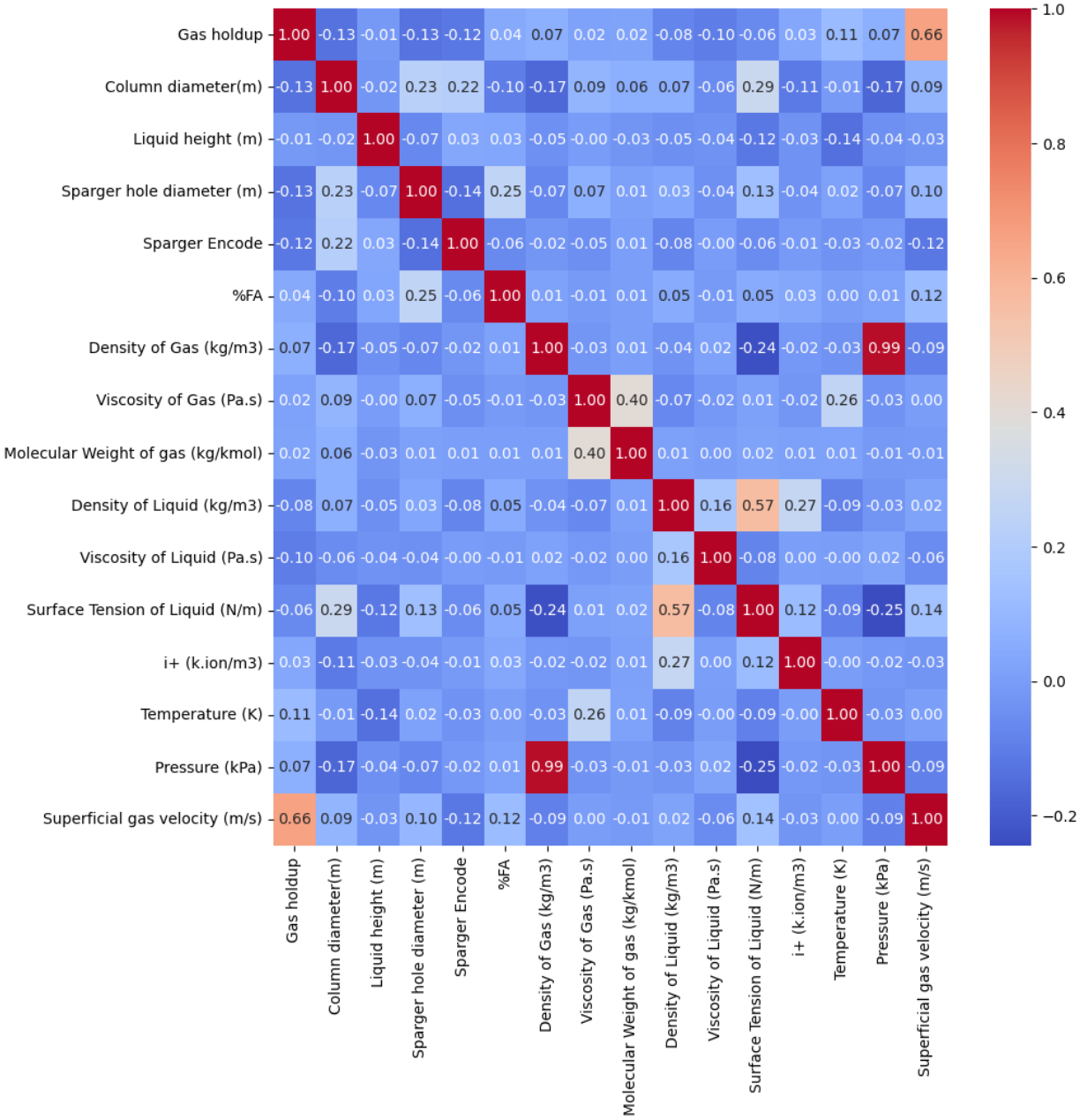






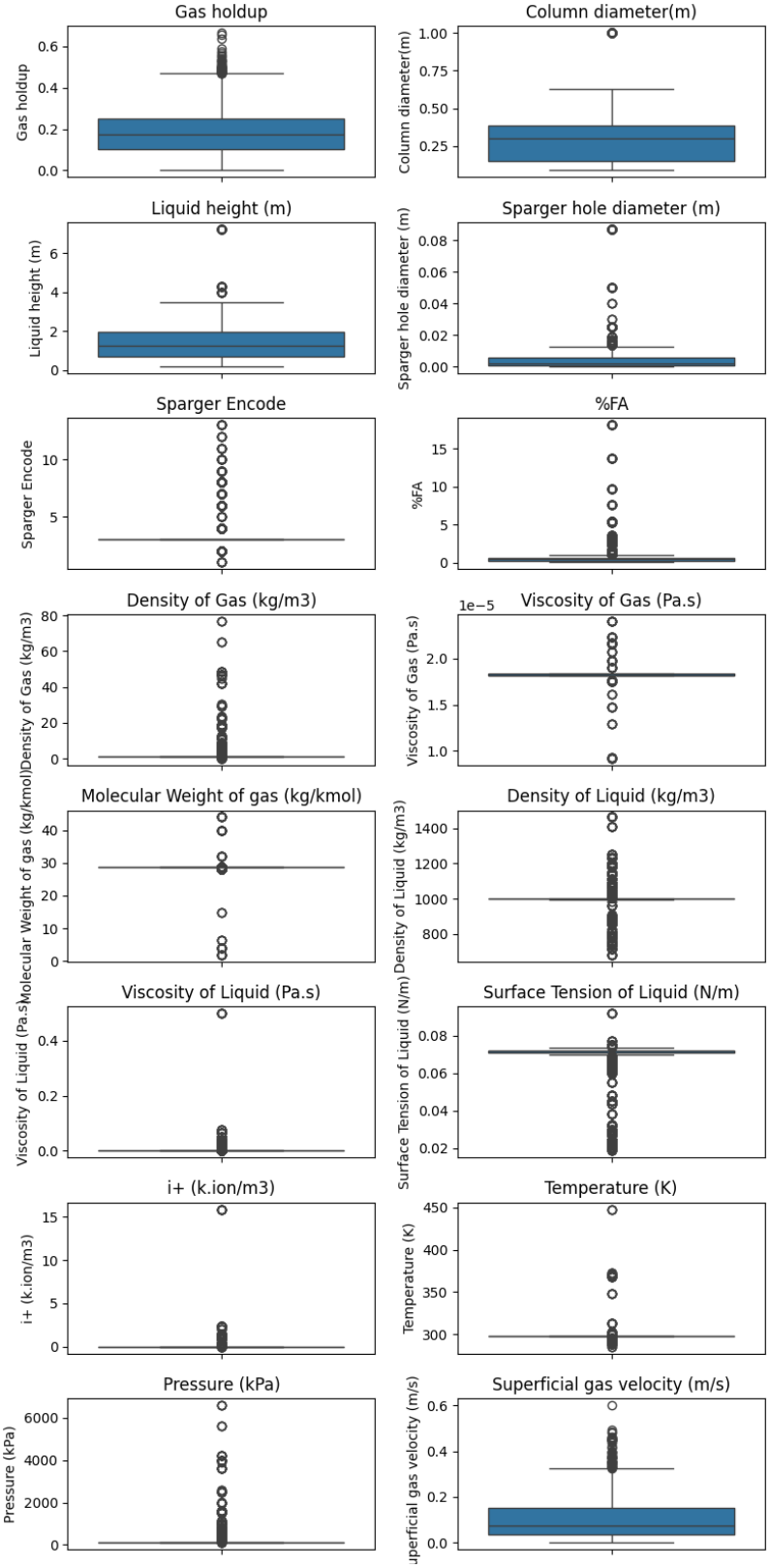
Inference - We can see that only one input feature - Superficial Gas Velocity, is looking to be correlated with the target variable.

Along with it, the heatmap for the correlation matrix was also plotted.



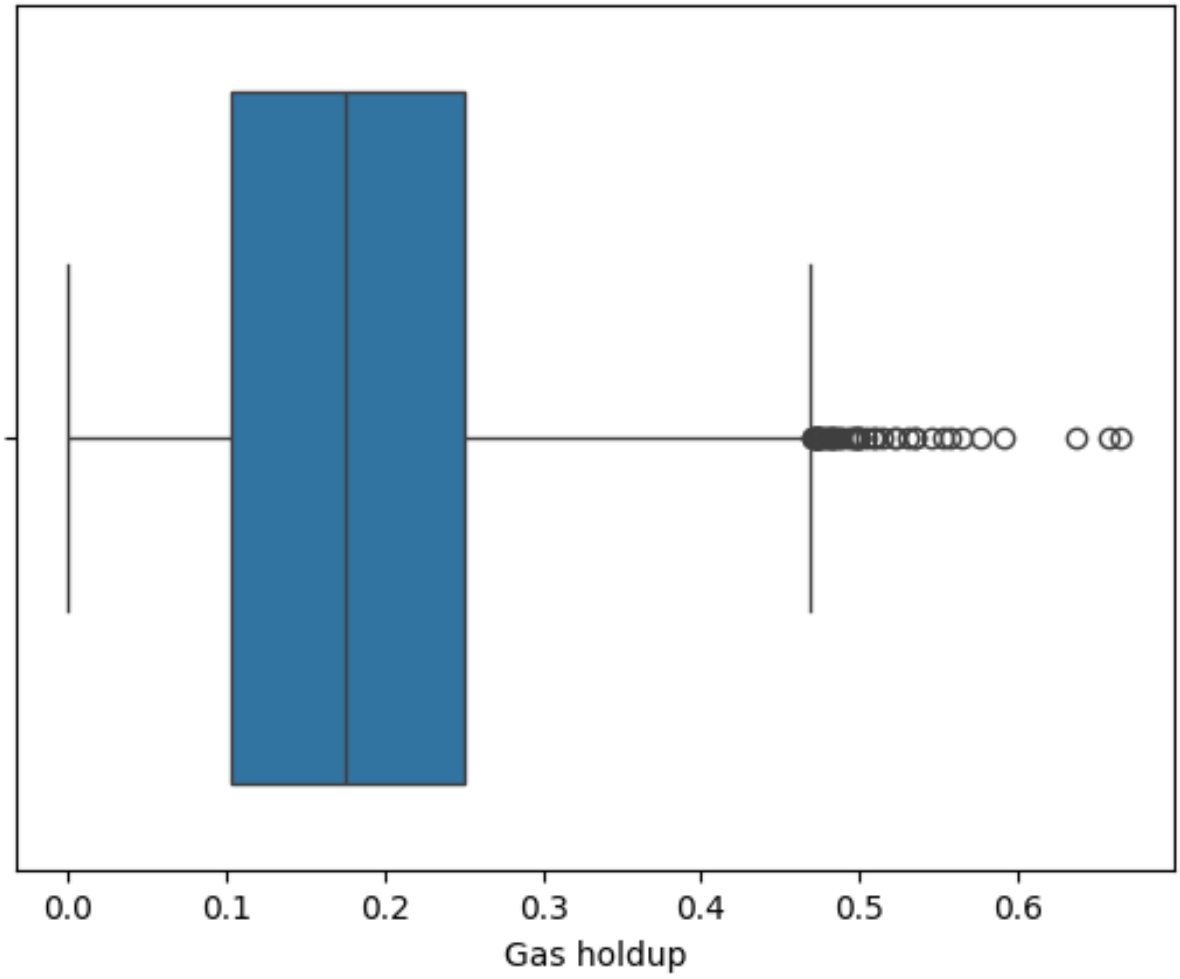
Inference - The same can be observed here. The Superficial Gas Velocity has the highest correlation with our target variable.

The boxplots for all the variables are shown below.



3. **Outlier Analysis and Removal -** Outlier analysis is done to identify and understand data points that significantly deviate from the norm or expected pattern within a dataset. It helps detect data anomalies, potential errors, unusual patterns, or valuable insights that may impact statistical analyses, model accuracy, and decision-making processes**.**

A box plot for the target variable was plotted.

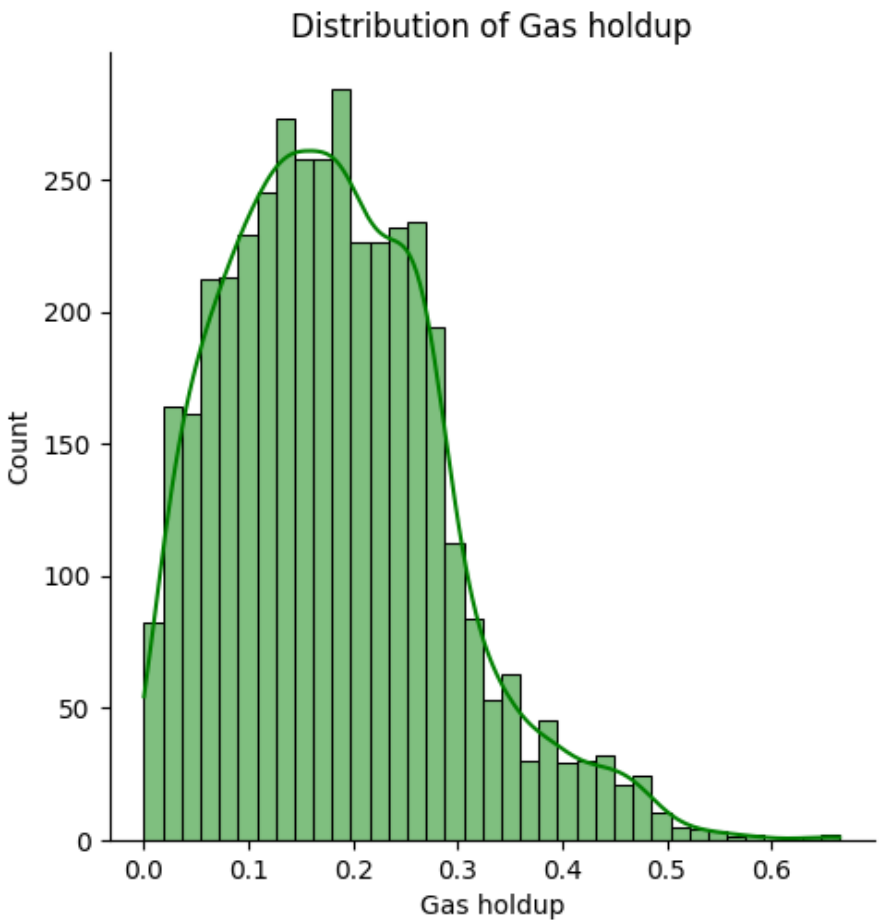


The number of outliers was found using both the Z-Score and IQR methods.

No. of outliers found using the Z-Score method - **23**

No. of outliers found using the IQR method - **49**

It's better to use the IQR method if the data is not normally distributed. We perform the Shapiro-Wilk test to check the target variable's normality. The distribution plot was also plotted for the target variable.



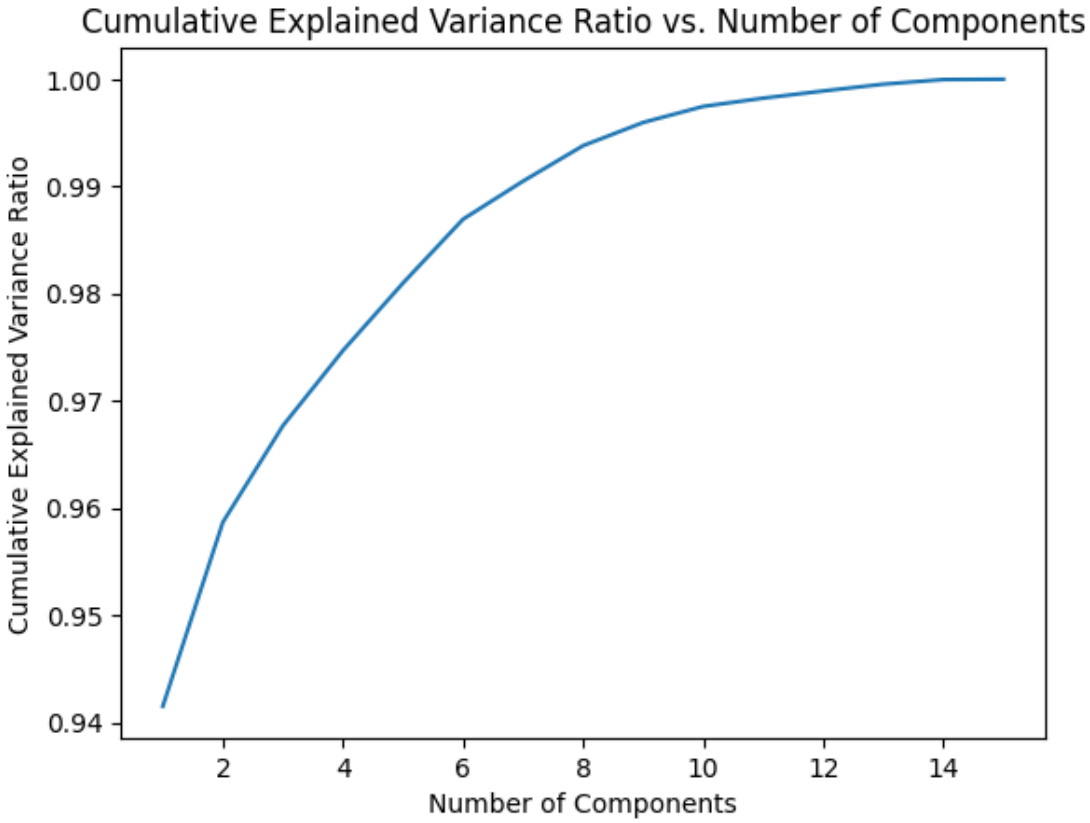
Inference - It can be easily seen that the data is not normally distributed.
Therefore, we use the IQR method for outlier analysis and removing them.

4. **Data Scaling -** Data scaling ensures that all features in a dataset have a consistent range and distribution, preventing features with larger scales from dominating the model's training process. It helps algorithms converge faster, improves model performance, and avoids numerical instabilities caused by differing feature scales.
   As the data is not normally distributed, we use the normalization method of data scaling. The target variable had a mean of 0.183150 and a standard deviation of 0.104527 before scaling. After scaling, the mean changed to 0.381396 with a standard deviation of 0.210227. Similarly, a change in mean and standard deviation was observed for all the input variables as well.
5. **Separating the target variable from the primary dataset -** The target variable - 'Gas Holdup' was separated, and a separate dataset - X, was created for all the input features. This is an important step for further steps in the project.

6. **Feature Engineering -** Feature engineering is done to extract, transform, or create new features from raw data, enhancing the predictive power of machine learning models. It aims to improve model performance, reduce overfitting, and uncover hidden patterns or relationships in the data, leading to more accurate and robust predictions.

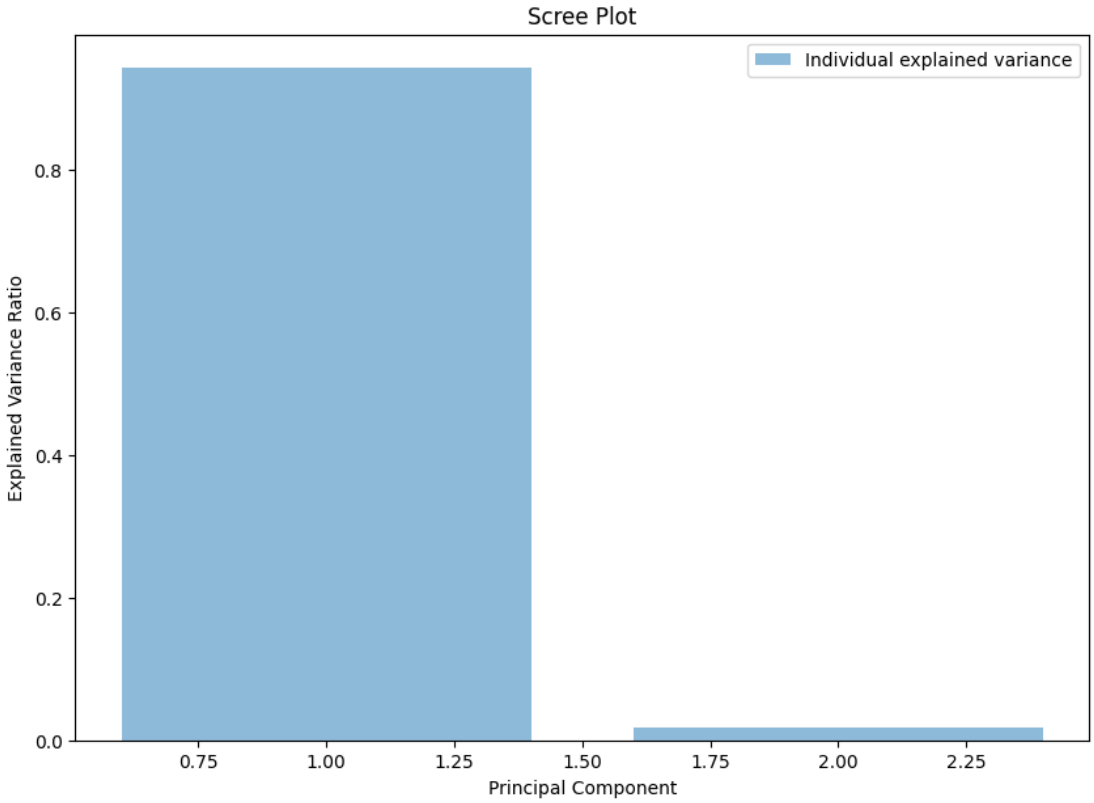
   PCA (Principal Component Analysis) is preferred for feature engineering because it reduces dimensionality while preserving critical information. It captures underlying patterns in data, reduces multicollinearity, and can enhance model performance without losing essential features, making it efficient and effective.

   To find the optimum number of principal components, the Cumulative Experience Variance Plot was plotted. The explained variance ratio tells you the proportion of variance in the original data captured by each principal component.



Inference - We can see that only two principal components will capture more than 95% of the variance in the data. Therefore, we use two principal components.

PCA was applied to the data using only two principal components. The scree plot for the same was plotted.



As seen earlier, only one feature has more than 80% of the variance.

7. **Data Splitting -** Data splitting is done to create separate subsets of data for training, validation, and testing in machine learning. This practice ensures that models are trained on one subset, validated on another to tune hyperparameters, and tested on a third subset to evaluate performance accurately and prevent overfitting.

Model Architecture:

In this project, I have tried to compare six different regression algorithms on the basis of RMSE as the evaluation criteria.

The six algorithms used here are:

1. **Linear Regression:** A simple and interpretable algorithm can provide baseline predictions. It assumes a linear relationship between the input features and the target variable (gas hold-up), making it suitable for initial exploration and understanding of the data.

2. **Support Vector Regression (SVR):** SVR effectively handles nonlinear relationships in data. It works well with high-dimensional datasets like ours, where complex interactions between the features influencing the gas hold-up may exist.
3. **Random Forest Regression:** An ensemble learning method that can handle complex relationships and interactions in the data. It is robust to outliers and can automatically handle feature selection, making it suitable for datasets with multiple features like ours.
4. **Gradient Boosting Regression (e.g., XGBoost, LightGBM):** Gradient Boosting algorithms are powerful ensemble techniques that sequentially build a series of weak learners to make accurate predictions. They are known for their ability to handle complex datasets, feature interactions, and outliers.
5. **Neural Network Regression:** Neural networks can capture complex relationships in the data through their layered architecture and nonlinear activation functions. They can learn intricate patterns and interactions among the features, making them suitable for modeling the gas hold-up prediction problem. However, neural networks may require more data and computational resources for training than other algorithms.
6. **Gaussian Process Regression:** It is a probabilistic model that can capture uncertainty in predictions. It is beneficial when dealing with small datasets or when uncertainty estimation is essential. It can provide point predictions and confidence intervals for the gas hold-up predictions.

Linear Regression was selected to serve as a baseline model as it is not capable of capturing non-linear relationships. Random forest regression and Support vector regression were selected as they can handle nonlinearities well but may require careful tuning to avoid overfitting. Gradient boosting regression can provide strong predictive performance but may require careful regularization to prevent overfitting. Neural network regression, while powerful, may require extensive data and computational resources, as well as careful tuning to avoid overfitting. Gaussian Process Regression is effective in handling small datasets and can provide accurate predictions even with limited data points.

The non-normality of the data does not pose a significant obstacle for the selected algorithms in predicting gas holdup. All the algorithms used here are versatile and robust methods that can handle various data distributions. They focus on capturing underlying relationships and patterns within the data rather than assuming specific distributions like normality.

Linear regression, for instance, primarily requires the residuals to be normally distributed, not necessarily the input features or target variable. Random forest regression, being a non-parametric method, does not rely on distributional assumptions and can effectively model both linear and nonlinear relationships. Support vector regression (SVR) can handle non-normal data distributions by finding the optimal hyperplane regardless of the underlying distribution. Gradient boosting regression and neural network regression are capable of learning complex relationships from non-normal data, with gradient boosting focusing on sequential model improvement and neural networks automatically extracting relevant features. Gaussian process regression (GPR) is a probabilistic method that is robust to different data distributions and provides uncertainty estimates.

Therefore, despite the non-normality of the data, these algorithms remain effective and can provide accurate predictions for gas holdup, as long as other model assumptions such as independence of observations and homoscedasticity of residuals are satisfied.

Tools which were used while the project are as follows:

- **Scikit-learn**: Provides many machine learning tools and algorithms for modeling, evaluation, and data preprocessing.
- **Scipy:** Used for statistical analysis.
- **Pandas:** Offers practical tools for preprocessing, cleaning, and loading structured data.
- **NumPy:** A foundational library for mathematical operations and multi-dimensional arrays in numerical computing.
- Visualization libraries like **Matplotlib** and **Seaborn** allow us to create beautiful plots and charts that are helpful in analyzing data and assessing model performance.

4 Implementation Plan

The project can be divided into 7 stages. The timeline is proposed in the case of working with a huge dataset. These are as follows:

1. **Data Collection and Preprocessing (2 weeks):**
    - Gather relevant data on gas holdup, input features, and operational parameters.

- Perform data cleaning, handling missing values, outliers, and formatting the data for analysis.

2. **Exploratory Data Analysis (1 week):**
    - Conduct EDA to understand data distributions, correlations, and identify important features.
    - Visualize relationships between input features and gas holdup to gain insights.
3. **Algorithm Selection and Setup (1 week):**
    - Choose the regression algorithms (linear regression, random forest regression, SVR, GBR, neural network regression, and GPR) based on their suitability and capabilities.
    - Set up the development environment, install necessary libraries (e.g., Scikit-learn, XGBoost), and prepare for model training.
4. **Model Training and Evaluation (3 weeks):**
    - Split the data into training and testing sets.
    - Train each regression algorithm on the training data and evaluate their performance using RMSE and other relevant metrics on the testing data.
    - Perform hyperparameter tuning using techniques like grid search or randomized search to optimize model performance.
5. **Comparison and Analysis (1 week):**
    - Compare the RMSE values and other metrics of each algorithm to determine the most accurate predictor of gas holdup.
    - Analyze the strengths and weaknesses of each algorithm in terms of computational efficiency, interpretability, and robustness.
6. **Documentation and Reporting (1 week):**
    - Document the entire project, including data preprocessing steps, algorithm selection criteria, model training details, and evaluation results.
    - Prepare a comprehensive report summarizing the project methodology, findings, and recommendations.
7. **Presentation and Feedback (1 week):**
    - Create a presentation summarizing key findings, insights, and recommendations.
    - Present the project to stakeholders, receive feedback, and incorporate any necessary revisions.

This timeline is approximate and may vary depending on the complexity of the data, availability of resources, and any unforeseen challenges encountered during the project. Adjustments may be made to accommodate specific project requirements and deadlines.

Model Training: The models were trained on the training dataset. Libraries and frameworks like scikit-learn, and XGBoost were used.

Performance Metrics: RMSE was used as the performance metric because it provides an interpretable measure of the average magnitude of errors in the same units as the target variable. It penalizes significant errors more than minor ones, making it suitable for models where accuracy is crucial.

Validation: K-fold cross-validation was used to assess each model's performance on the training data and tune hyperparameters to improve model accuracy and generalization.

5 Testing and Deployment

All six algorithms were tested on the unseen test dataset. RMSE was used as the evaluation criteria. Along with it hyperparameter tuning was performed using the validation dataset to ensure the best performance of the algorithm.

The deployment plan for the gas holdup prediction model involves several key steps. Firstly, the model will be containerized using platforms like Docker for scalability and ease of deployment across various environments. Next, a scalable cloud infrastructure, such as AWS or Azure, will host the containerized model, ensuring high performance and availability. Continuous monitoring and automated testing will be implemented for maintenance, with regular updates and retraining scheduled to keep the model accurate and relevant. Lastly, API endpoints will be created for seamless integration with existing systems, facilitating real-time predictions and user interactions in the real-world environment.

Deploying the gas holdup prediction model raises ethical considerations regarding data privacy, fairness, and accountability. Datasets can be anonymized to protect individuals' privacy. Fairness concerns arise in algorithmic decision-making, requiring measures to mitigate biases and ensure equitable outcomes for all stakeholders. Additionally, maintaining transparency about the model's limitations, potential biases, and uncertainties is essential for accountability. Continuous monitoring, bias detection, and regular audits are necessary to address ethical concerns and uphold integrity in deploying the model for real-world use.

6 Results and Discussion

The table for the RMSE values of all the six algorithms used is given below:

Algorithm	RMSE Value
Linear Regression	**0.209568**
Support Vector Regression	**0.193693**
Random Forest Regression	**0.136373**
XGBoost Regression	**0.138431**
Nueral Network Regression	**0.222029**
Gaussian Process Regression	**0.193693**

We can observe that **Nueral Network Regression has the worst performance**. This might be due to the limited data as these often require large amounts of training data to generalize well and avoid overfitting. If the available training data is limited, a simpler linear regression model with fewer parameters may generalize better and provide more stable predictions. While the **Random Forest Regression performed best** for predicting gas holdup due to its ability to handle nonlinear relationships, reduce overfitting, provide insights into feature importance, and maintain robustness to outliers, making it a suitable choice for complex and dynamic datasets in chemical engineering applications.

The gas holdup prediction model's performance was compared against existing benchmarks and solutions in the field. The model has underperformed on comparing it with models developed in various research papers. This can be due to various factors like limited availability of data. Variations in preprocessing steps, feature engineering, and model selection criteria between the research papers and the implemented model may also account for the observed differences in performance.

Challenges faced during the project and limitations of the proposed solutions are as follows:

1. **Limited Data Availability:** One of the primary challenges was the limited availability of high-quality data specifically tailored for gas holdup prediction in bubble columns. This scarcity constrained the diversity and size of the dataset, potentially impacting model generalization.

2. **Complexity of Gas Holdup Dynamics:** Gas holdup in bubble columns is influenced by numerous factors, including fluid properties, column geometry, and operating conditions. Capturing all these complexities accurately in the model was challenging and may have led to simplified representations.
3. **Ethical Considerations:** Ensuring ethical considerations such as fairness, transparency, and accountability throughout during deployment may require careful attention and adherence to ethical guidelines.
4. **Limited Real-time Adaptability:** The proposed solution may face limitations in real-time adaptability to sudden changes or fluctuations in the process parameters or input features, necessitating regular updates and retraining.

7 Conclusion and Future Work

The gas holdup prediction project aimed to develop a robust model using six regression algorithms to predict gas holdup in bubble columns. The project involved data preprocessing, algorithm selection, model training, and evaluation. The key findings revealed that random forest regression performed best due to its ability to handle nonlinear relationships and provide accurate predictions.

The project contributes to chemical engineering by offering a reliable tool for predicting gas holdup, aiding in process optimization and decision-making. Insights into feature importance and model performance provide valuable guidance for future research and industrial applications.

Multiple additions can be done to this project in the future including incorporating advanced machine learning techniques like deep learning to improve model accuracy and capture intricate relationships, expanding the dataset with more diverse and detailed features to enhance model robustness and generalization, conducting real-time monitoring and feedback integration to enable adaptive modeling and enhance operational efficiency, and addressing ethical considerations and transparency in model development and deployment for responsible AI practices.

8 References

Here is the list for the references used to make this project:

- Predictive analysis of gas hold-up in bubble column using machine learning methods - ScienceDirect
- (PDF) Numerical Analysis of Gas Hold-Up of Two-Phase Ebullated Bed Reactor (researchgate.net)
- Dataset for: Predictive analysis of gas holdup in bubble column using machine learning methods - Mendeley Data .
- scikit-learn: machine learning in Python — scikit-learn 1.4.2 documentation
- https://chat.openai.com/
- https://www.sciencedirect.com/science/article/pii/S0009250919301496

9 Auxiliaries

Data Source link:

https://data.mendeley.com/datasets/s3wjhzzdr3/2

Python file link:

https://colab.research.google.com/drive/1MKDP13NY-Ec07oOXirwpB-yA6nK_4JZG?usp=sharing