



Eureka^{*} 8.0

A Eureka Report on : “AI-ML Driven Enhancement of Biomass Pyrolysis Yield ”

Prepared By:

Scarface

Sandesh Wabale : U24CH016

Pratham Saini : U24CH012

Ayush Chaudhari : U24CH089

Zohair Nalawala : U24CH037

Mentors :

Om Prajapati

Krishna Karnani

Advanced Process Engineering and System Design

Topic Name:

**AI-ML Driven Enhancement of Biomass Pyrolysis
Yield**

Problem Statement:

**Developing a Machine Learning Framework using Artificial
Intelligence for Predicting and Optimizing Biomass Pyrolysis
Yields.**

Team Name: Scarface

November 11, 2025

Abstract

The depletion of fossil fuels and growing environmental concerns have led to increased interest in biomass pyrolysis as a renewable and carbon-neutral way to produce useful energy products like bio-oil, biochar, and pyrolytic gas. However, the efficiency and quality of pyrolysis outputs depend largely on the physical and chemical properties of biomass as well as the operating conditions, making process optimization quite complex when using traditional experimental or analytical methods. These conventional methods are often time-consuming, expensive, and limited in their ability to capture the nonlinear interactions among various influencing factors. To tackle this issue, artificial intelligence (AI), especially machine learning (ML), presents a promising alternative by offering data-driven predictive models that can reveal hidden patterns and guide process optimization. This work aims to explore the potential of ensemble ML algorithms, specifically Random Forest (RF), Gradient Boosting Decision Tree (GBDT), eXtreme Gradient Boosting (XGBoost), and Adaptive Boost (Adaboost), for modeling the relationships among biomass properties, pyrolysis conditions, and product yields. Rather than emphasizing experimental validation, the goal is to create a predictive framework that can assess algorithm performance, identify the most important variables affecting product distribution, and assist in decision-making for optimizing pyrolysis processes. Ultimately, this study hopes to lay the groundwork for integrating AI-based modeling into process engineering and system design, contributing to the development of sustainable bioenergy systems and improving efficiency in future biomass conversion technologies.

Contents

1	Introduction	4
2	Introduction to Biomass Pyrolysis	5
2.1	Overview of Biomass Pyrolysis	5
2.1.1	Types of Pyrolysis	5
2.1.2	Stages in Pyrolysis	6
2.2	Lignocellulosic Biomass Composition	6
2.3	Physicochemical Properties Determining Pyrolysis	7
2.3.1	Proximate Analysis	7
2.3.2	Ultimate Analysis	8
2.3.3	Other Properties	8
2.4	AI and ML in Biomass Pyrolysis	8
3	Methodology	10
3.1	Data Collection and Preprocessing	10
3.1.1	Data Gathering and Source	10
3.1.2	First Look and Data Cleansing	10
3.1.3	Data Cleaning and Validation	11
3.1.4	Transformation and Feature Engineering	13
3.1.5	Exploratory Data Analysis (EDA)	13
3.1.6	Data Partitioning and Cross Validation	15
3.2	Deployment of the AI-Driven Pyrolysis Optimization Framework	16
3.2.1	AI/ML Models Used	16
3.2.2	Web-Based User Interface (UI) for Industrial Application	16
3.3	Optimization Techniques	18
4	Results and Discussion	21
4.1	Performance Metrics	21
4.2	Model Interpretation and Feature Analysis	23
4.2.1	Feature Importance	23
4.2.2	Univariate Partial Dependence (Single-Feature Effects)	25
4.2.3	Analysis of Bivariate Interaction (Effects of Two Features)	27
5	Applications	29
5.1	Industrial Impact	29
6	Conclusion	31
7	References	32

List of Figures

1	Biomass Pyrolysis Flow Diagram	5
2	Monomer units of the main components of Lignocellulosic Biomass.	7
3	Bar chart illustrating the number of missing values for each column.	10
4	Final distributions of key variables after cleaning (N=396).	12
5	Correlation heatmap of input parameters and output yields.	14
6	Data Split Pie Chart.	15
7	Web-Page UI Interface.	17
8	Input Provided to Model.	19
9	Output Obtained from Model.	19
10	Model Prediction Performance for Yields in Solid, Liquid, and Gas Phases . . .	21
11	Feature Importance for models of solids (top), liquids (middle), and gases (bottom). The XGBoost algorithm calculates scores that show how much each feature helps the model's accuracy.	24
12	Univariate Partial Dependence Plot for the Solid phase model.	25
13	Univariate Partial Dependence Plot for the Liquid phase model.	26
14	Univariate Partial Dependence Plot for the Gas phase model.	26
15	Bivariate Partial Dependence Plot for the Solid phase model.	27
16	Bivariate Partial Dependence Plot for the Liquid phase model.	28
17	Bivariate Partial Dependence Plot for the Gas phase model.	28

1. Introduction

The increased and rapid consumption of fossil fuels, as well as the growing environmental issues raised by their use, have compelled scientists to search for cleaner and greener sources of energy. One such alternative is **bioenergy**, a low-pollutant, carbon-neutral, and multifaceted source of energy that has the potential to partially substitute traditional fossil fuels. In contrast to solar or wind energy, bioenergy is weather-independent and therefore a more stable and consistent renewable energy source. Within this category, the thermochemical conversion of biomass—particularly through **pyrolysis**—has attracted significant attention since it has the potential to produce valuable products such as bio-oil, biochar, and syngas.

Pyrolysis is a low-oxygen thermal decomposition process applied to lignocellulosic biomass, wherein the feedstock is converted into lignin-rich bio-oil, carbon-rich biochar, and combustible pyrolytic gases. Each of these products finds useful applications: bio-oil can be upgraded and utilized as a fuel alternative; biochar is used for carbon sequestration, soil enrichment, and as a solid fuel; while pyrolytic gas can be employed as a feedstock for syngas production. The composition and yield of these products are influenced by several parameters, including inert gas flow rate, particle size, temperature, heating rate, and intrinsic biomass properties. However, considering the complex chemical dynamics involved, the optimization of these parameters through conventional experimental or kinetic modeling approaches is often time-consuming, costly, and sometimes ineffective.

Classical formula-based modeling techniques, such as kinetic modeling and computational fluid dynamics (CFD), rely on numerous assumptions regarding transport phenomena and reaction mechanisms. Although these methods provide valuable insights into reactor behavior and design, they are often customized for specific reactor types, computationally intensive, and limited in applicability to other biomass feedstocks. Experimental uncertainties further restrict the development of accurate predictive models for the multifaceted relationships between biomass properties, process conditions, and product yields. In recent years, the rapid advancement of **Artificial Intelligence (AI)** and **Machine Learning (ML)** has provided powerful tools to address these challenges. ML methods can learn from large datasets and uncover complex, nonlinear relationships between input variables and corresponding outputs—without requiring complete mechanistic understanding. Studies have demonstrated that machine learning models such as **Artificial Neural Networks (ANN)**, **Random Forests (RF)**, **Adaptive Boosting (AdaBoost)**, and **Extreme Gradient Boosting (XGBoost)** can accurately predict pyrolysis yields, optimize process conditions, and analyze the influence of biomass characteristics on product distribution. ML offers the capability to handle both numerical and categorical variables, perform large-scale rapid computations, and manage data variability across multiple formats—advantages that are difficult to achieve with traditional methods.

Despite these promising advancements, notable research gaps remain. Most existing studies focus on predicting the yield of a single product phase (solid, liquid, or gas), rather than developing comprehensive models that account for all three simultaneously. The present work aims to address this limitation by developing AI/ML models for predicting biomass pyrolysis yields.

2. Introduction to Biomass Pyrolysis

2.1. Overview of Biomass Pyrolysis

Pyrolysis is the thermal decomposition of organic materials, or biomass, under low- to no-oxygen levels. Compared with combustion, which is an oxygen-based process that yields CO₂ and water, pyrolysis breaks down biomass into less complex compounds without complete oxidation. The process transforms a feedstock of biomass into a mix of solid (biochar), liquid (bio-oil/tars), and a gaseous product (syngas). The quality and amount of products are greatly variable as a function of the process conditions, feedstock and design of the pyrolysis system. Recent reviews point out that pyrolysis is a key independent process and typically the main step for biomass conversion into fuels, chemicals, or energy carriers.

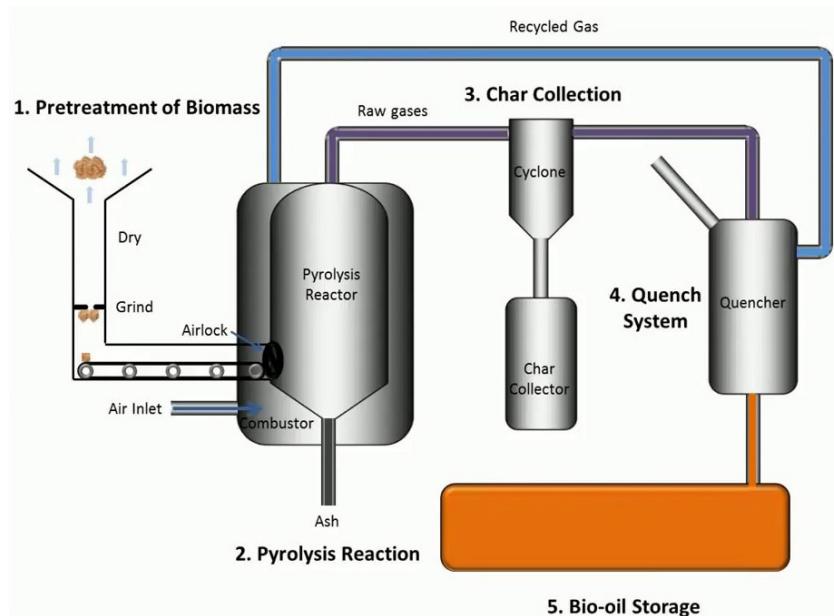


Figure 1: Biomass Pyrolysis Flow Diagram

2.1.1. Types of Pyrolysis

- **Slow pyrolysis** — comprises low rates of heating and long residence times at moderate temperatures. Characterized by preference in char utilization and is primarily applied for charcoal/biochar production.
- **Fast pyrolysis** — 400–600 °C with high rates of heating and short vapor times of residence, with maximum yields of bio-oils and a minimum amount of char.
- **Flash pyrolysis** — has very high heating rates (crude oil/s) and millisecond residence times, commonly achieving very high liquid yields but with sophisticated reactor control.

2.1.2. Stages in Pyrolysis

- **Drying:** Water boils at low temperatures. High moisture can reduce efficiency and affect heat transmission.
- **Devolatilization:** The thermal decomposition of biomass polymers (hemicelluloses, celluloses, lignin, and extractives) into smaller molecules, radicals, and vapors. The mode and rate at which it takes place are governed by the feedstock composition, the heating rate, and the temperature.
- **Secondary reactions:** Volatiles that are released will undergo cracking, polymerization, repolymerization, condensation, and charification, depending on whether residence times are long or perhaps high temperatures are up for grabs. These reactions influence the final composition of the resulting bio-oil, gas, and char.
- **Condensation and collection:** Condensable vapors are cooled and deposited as bio-oil, along with non-condensable gases like CO , CO_2 , H_2 , and CH_4 . Non-volatiles are residuated as biochar. Condenser and quenching system design is required to be efficiently regulated in order to suppress further reactions and achieve maximum yield and quality.

2.2. Lignocellulosic Biomass Composition

Most common pyrolysis feedstock lignocellulosic biomass is mainly consisted of cellulose, hemicellulose, and lignin. Structural feature, thermal stability, and influence on pyrolysis performance of each main component are different.

- **Cellulose** is a linear and crystalline polysaccharide composed by glucose units connected by -1,4 glycosidic bondages. Owing to its crystallinity and regular structure, it beings to breakdown at a restricted temperature region, with a high yield of volatile, condensable products like anhydrosugars (e.g. levoglucosan). Under fast heating, its breakdown is distinct and clear, contributing significantly to the formation of the oiler bio.
- **Hemicellulose** is a branched and amorphous polymer that is made up of a number of sugar monomers (e.g. mannose, xylose, glucose, arabinose). Hemicellulose is less thermally stable than cellulose and is broken down in a wider and typically less hot temperature range. Being an amorphous molecule, hemicellulose typically produces more light oxygenates, gases, and water and more labile heavy fractions.
- **Lignin** is a highly complex and irregular aromatic polymer (with phenolic structures), associated with cellulose and hemicellulose in plant cell walls. It is more thermally stable and degrades across a broad temperature range, frequently coinciding with high temperatures. Lignin tends to produce aromatic, phenolic products, and char and leads to increased char yields and phenolic-rich bio-oil fractions.

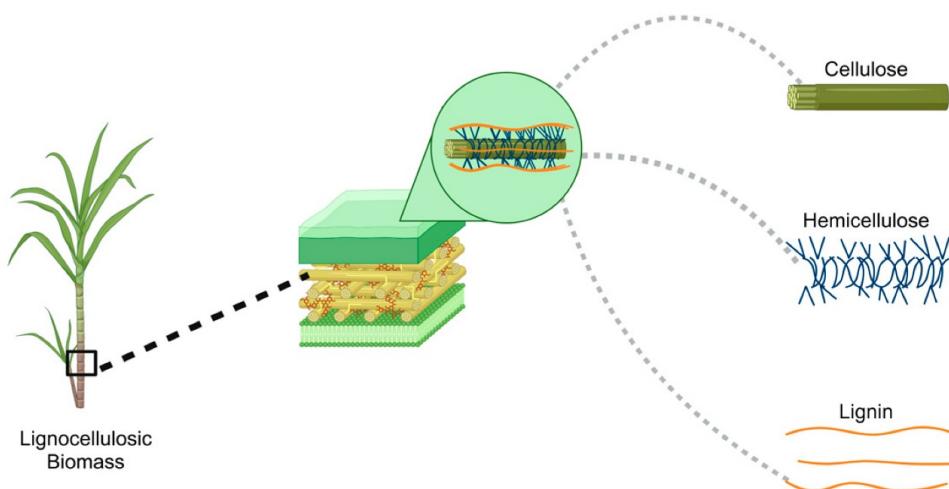


Figure 2: Monomer units of the main components of Lignocellulosic Biomass.

2.3. Physicochemical Properties Determining Pyrolysis

The physicochemical characteristics of biomass significantly defines its thermal breakdown behavior, conversion efficiency for energy, and yields of products during pyrolysis. The characteristics affect how biomass burns, disintegrates, and produces volatiles and solid residues. A transparent comprehension is critical for the maximization of the reactor conditions and for predictive modeling with AI or machine learning methods. The most important affecting factors are achieved by proximate analysis, ultimate analysis, and other physical and calorific properties like heating value, density, and particle size.

2.3.1. Proximate Analysis

Proximate analysis divides biomass into moisture, volatile matter, fixed carbon and ash, providing insight into its basic thermal behavior.

- **Moisture Content:** High moisture reduces heating efficiency, delays decomposition, and reduce biological oil yield since there is additional energy needed for distillation.
- **Volatile Matter (VM):** The portion of biomass that vaporizes upon heating. High VM is desirable for bio-oil production.
- **Fixed Carbon (FC):** Is the stable residue following release of the volatiles. High FC promotes char formation and retarded devolatilization.
- **Ash Content:** Inorganics that catalyze due to inorganic minerals (e.g., K, Ca, Si) reactions. High ash lowers oil yield and can bring about reactor fouling.

Proximate parameters have been commonly used as data-driven model input features to estimate product delivery and power effectiveness.

2.3.2. Ultimate Analysis

Ultimate analysis determines elemental composition — C, H, O, N and S — which directly affects reactivity and heating value.

- **Carbon (C):** Raises HHV and potential of char formation.
- **Hydrogen (H):** Increases energy content and stabilizes bio-oil.
- **Oxygen (O):** High O reduces energy value and bio-oil stability.
- **Nitrogen and Sulfur:** Minor elements that cause gaseous emissions (NOx, SOx).

Elemental ratios such as H/C and O/C are indicators of fuel quality and deoxygenation potential.

2.3.3. Other Properties

- **Higher Heating Value (HHV):** Indicates the cumulative energy content of biomass; higher HHV result in higher yields for volatile and gas.
- **Particle Size:** Smaller particles are faster heating and higher bio-oil producing particles. whereas bigger ones prefer char based on less efficient heat transfer.
- **Density and Porosity:** Affect heat transfer, vapor flow, and residence time; high porosity facilitates effective volatile release.

2.4. AI and ML in Biomass Pyrolysis

Integration with AI and Machine Learning have shown great potential to transform biomass pyrolysis by allowing the prediction of product yields from input feedstock properties and process conditions. Machine learning algorithms, particularly ensemble methods such as Random Forests (RF), Gradient Boosted Decision Trees (GBDT), and XGBoost, have been shown to have high accuracy in yield predictions of bio-oil, biochar, and syngas.

ML helps to model complex, nonlinear relationships between variables and provides feature importance analysis, aiding in process optimization and understanding the influence of biomass and process parameters. AI-based models enable quicker innovation cycles, reduce the need for extensive physical experiments, and can result in real-time process control for enhanced efficiency and sustainability. This ability to model complex, non-linear relationships is not merely a minor academic advantage; it directly addresses the primary failing of traditional pyrolysis modeling. Classical kinetic or CFD models use simplified, first-principle assumptions that fail to accurately account for the simultaneous, competing thermochemical reactions. For example, a classical model will be unable to capture the complex synergistic interaction between high heating rate and large particle size, or the subtle catalytic actions of certain inorganic materials in the biomass ash fraction.

In addition, the diagnostic nature of such models, for example, the feature importance and partial dependence plots utilized in this study , elevate them from a black box to an extremely useful analytical tool. The analysis of feature importance enables researchers and engineers

to quantitatively determine which process parameters have the largest impact—for example , confirming whether Final Temperature is more important than Carbon content in biochar yield, or Heating Rate is truly important in bio-oil production. This knowledge is essential to target process control parameters and make feedstock choices.

Finally, this predictive power is the basis for prescriptive optimization. As demonstrated by the model in this project, a validated ML model can be paired with an optimization algorithm such as Differential Evolution. This combination turns the model from a passive calculator to an active problem-solver, able to traverse the vast, high-dimensional parameter space to find a global optimum. It enables an operator to go beyond “what-if” questions and ask, “What is the maximum possible bio-oil yield that I can get from this particular batch of feedstock, and what are the exact process conditions needed?” This ability is the most important way to make genuinely adaptive, intelligent, and effective biorefinery control systems. This ability to adapt is another major benefit over fixed physical models. A conventional kinetic model tends to be parameterized for a particular, individual feedstock (e.g., “pine wood”). It has to be newly developed and validated again for another feedstock, like wheat straw or miscanthus. An ML model, when learned on a diverse dataset, learns inherently about the impact of feedstock variability. It is built from the ground up to take biomass properties as inputs, so it is a universal tool that is capable of handling feedstock changes on the fly. This is essential for industrial use, where feedstock supply will be dependent upon seasonal and market availability.

Moreover , the speed of computation of these learned models is a critical enabler for actual real-time control. Although the initial training process may be computationally intensive, a fully trained XGBoost or RF model can make a prediction in milliseconds. Such high-speed inference makes it possible to integrate the model directly into a plant’s Distributed Control System (DCS). The system might, theoretically, accept real-time data from analyzers of feedstocks, modify the optimization algorithm’s targets, and deliver new, optimized setpoints (temperature, flow rate, etc.) on a minute-by-minute basis, continuously tuning the reactor for maximum economic yield and process stability.

3. Methodology

3.1. Data Collection and Preprocessing

3.1.1. Data Gathering and Source

The basis of this machine learning model is a dataframe obtained from the Kaggle repository that is titled "Biomass Pyrolysis Data," contributed by user "mustafakeser4". The particular dataframe used, `pyrolysis.csv`, consists of a set of experimental data obtained during biomass pyrolysis runs. This is the conversion of biomass to the product of bio-oil, gas, and char through thermal breakdown. The particular dataframe is very suited to our aim because it contains the important process parameters such as temperature and heat uplift, as well as biomass properties along with resultant product yields. Our aim is to represent these product yields as functions of the operating conditions, so this dataframe is the ideal one to develop a predictive model.

3.1.2. First Look and Data Cleansing

After uploading the data into a pandas DataFrame, the first look that was conducted included the use of the usual functions such as `.info()`, `.describe()`, and `.head()`. The first look showed a dataframe with mostly numerical columns but pointed out the occurrence of missing values within some features. Systemic treatment was provided to the missing values. The number of the missing values in each variable was seen first to realize the extent of the problem.

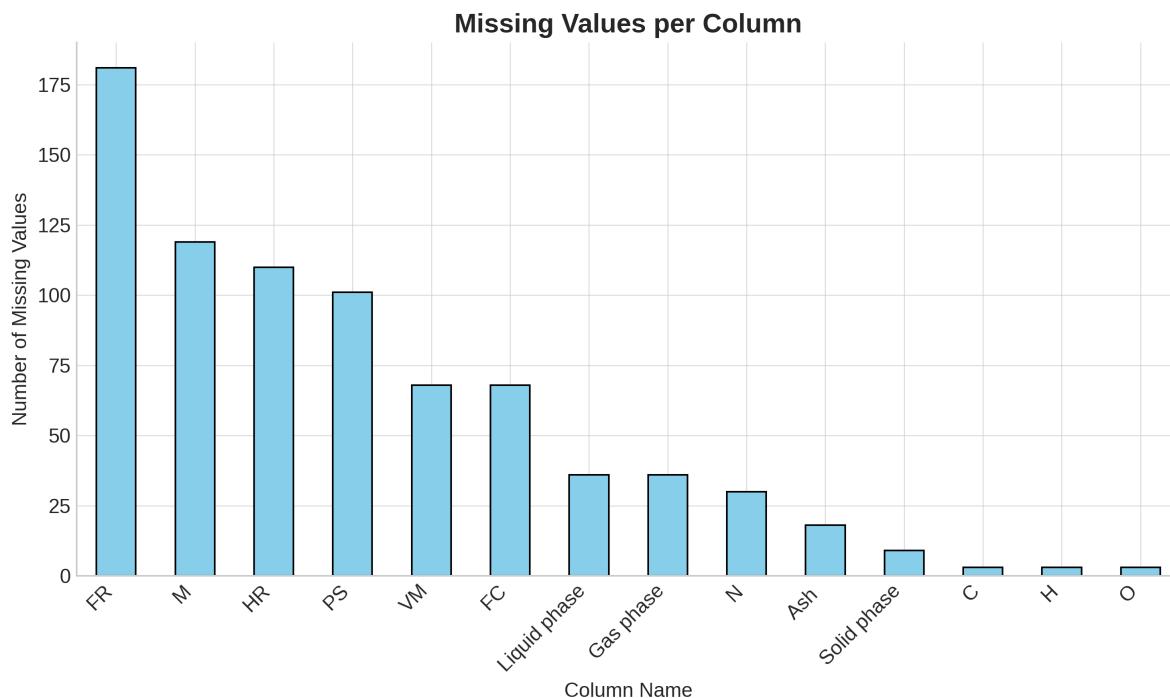


Figure 3: Bar chart illustrating the number of missing values for each column.

As can be seen from Figure 3, columns such as 'FR' (Feed Rate) and 'M' (Moisture) contained a

high number of missing values. Where the missing data represented a minor portion, imputation techniques (mean, median, or KNN) were applied. Duplicated rows were removed to avoid skewing results.

3.1.3. Data Cleaning and Validation

One of the biggest problems with this project was the type of dataset. The data wasn't consistent because it came from hundreds of different research papers (Dong et al., 2022). Because the data came from different places, there was no single standard for entering it, which made the dataset "messy" and mixed. There were two main reasons why a simple method for finding statistical outliers, like an IQR or Z-score, didn't work.

- **Text-Based Data:** Many of the data points were not just numbers. Instead, they were written down as ranges of text, such as "1.5–2.0," or with modifiers, such as "<0.1." Standard statistical methods will not work with this text data. If you try to load this data directly, the software will treat the whole column as a non-numerical "object," which means you can't use it in a regression model or a `StandardScaler`. A custom parsing solution was therefore required for these entries.
- **Valid Extreme Data:** Many of the numerical "outliers," particularly in process conditions such as FT (Final Temperature) and HR (Heating Rate), were not errors. Instead, they represented real, extreme experimental conditions, like slow pyrolysis (e.g., 350 °C, 5 °C/min) versus fast pyrolysis (e.g., 600 °C, 300 °C/min). If we had cut this range down by removing these "outliers," the model would have been very limited and unable to learn the full, complex behaviour of the pyrolysis process. An optimisation model that has not learnt from these extremes cannot be trusted to optimise a process in the real world. Removing these important pieces of information would have made it very difficult for the model to learn actual real-world behaviour.

So, we set up a stronger two-step data integrity process. This approach is implemented in the `clean_and_convert` and `train_and_load_model` functions of our model:

- **Changing Text to Numbers in a Unique Way:** The `clean_and_convert` function was applied to all 12 input features and 3 output yields as a first step. This function uses regular expressions (regex) to intelligently parse text-based entries. It looks for patterns like "num1-num2" and "<num" and converts ranges to their mean value (for example, "1.5–2.0" becomes 1.75) and handles modifiers (for example, "<0.1" becomes 0.1). This crucial step cleaned up the whole dataset and turned it into a fully numerical format, saving hundreds of data points that would have been lost otherwise.
- **Removing Incomplete Samples:** After the conversion, we used `.dropna()` to remove any sample that still had missing values (`NaN`) or was missing a key product yield from the dataset. This is an important second step because it ensures that only samples that were truly incomplete (for example, an experiment that never reported the Nitrogen content) were removed, rather than samples that were just poorly formatted. This establishes a high-quality standard for the

training data, based on the principle that a smaller but cleaner dataset is better than a larger, noisier one.

This focused approach guarantees that the final, verified dataset of 396 samples is both clean and a good representation of the full, valid experimental range. This 396-sample database is significantly better than the original one, even though it is smaller. It no longer contains any text or non-numerical artefacts that would have made it impossible to build a successful model. Figure 4 shows the distributions of the main process variable (FT) and the three product yields. These distributions—especially the wide and multi-peaked one for FT—show that the data covers a broad range of experimental conditions and is now ready for model training.

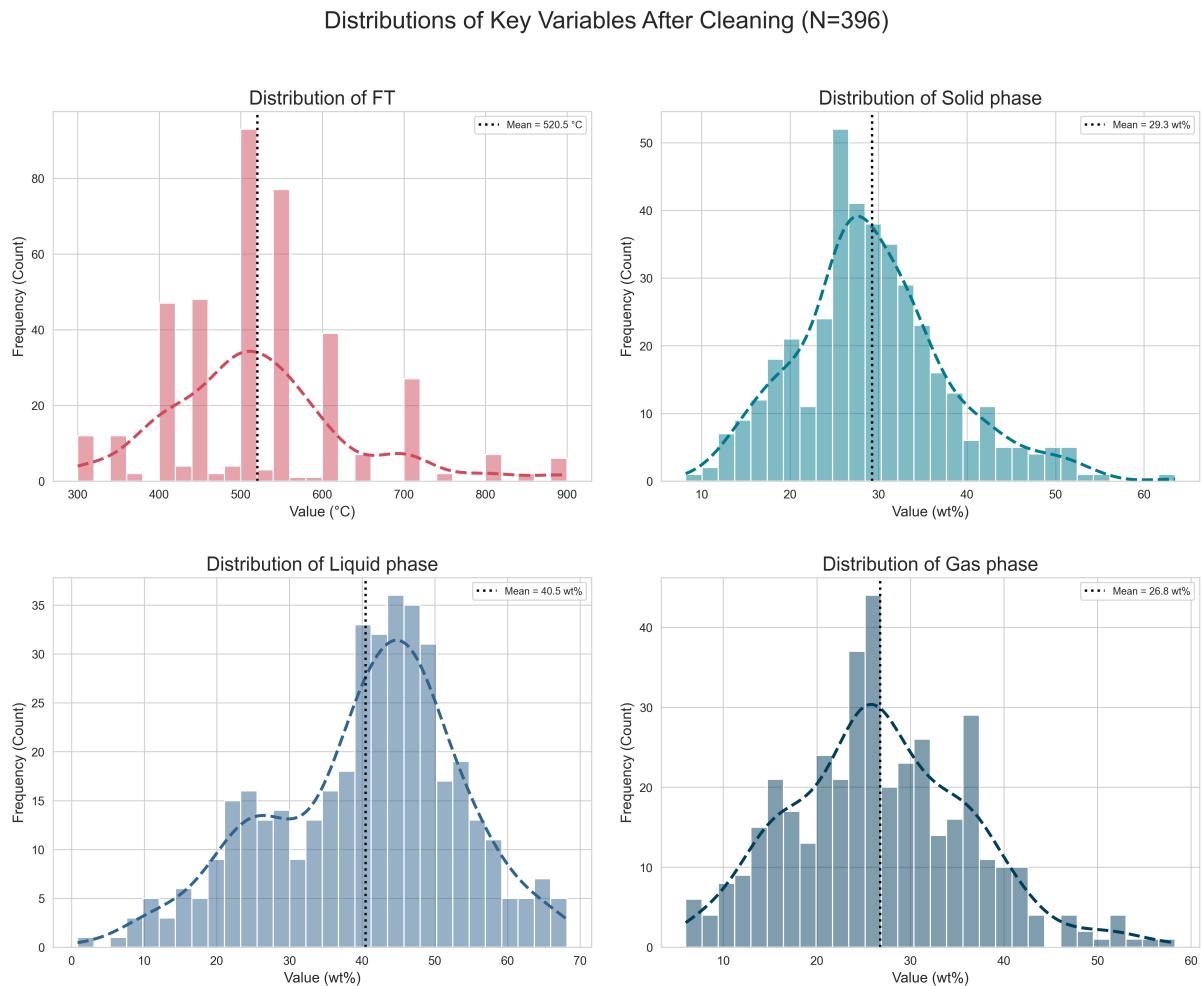


Figure 4: Final distributions of key variables after cleaning (N=396).

3.1.4. Transformation and Feature Engineering

After cleaning, the 12 input features were on very different scales. For example, Final Temperature (FT) values can be in the hundreds (300–900 °C), while Particle Size (PS) values are usually small decimals (0.1–10 mm). The whole set of features was normalized so that the XGBoost algorithm wouldn't give more weight to features just because their numerical scale was bigger.

We used the `StandardScaler` from the Scikit-learn library for this project. There were two important reasons why this was a key step:

- **Uniform Contribution:** Scaling changes each feature so that its mean is 0 and its standard deviation is 1. This puts all features on a common ground, making sure that the model learns the true predictive power of each one, not just its magnitude.
- **Preventing Data Leakage:** To stop data leakage, the `StandardScaler` was fit only using the training data, which is the right way to do it. The scaler then used this learned distribution to change both the training and test datasets. This stops any information from the test set from getting leaked into the training process, which makes sure that the model's performance is evaluated fairly.

We thought about using more advanced feature engineering methods, like making interaction terms (like temperature \times heating rate) or using logarithmic transformations to make the data less skewed, but we chose not to do so. Tree-based models like XGBoost are naturally good at capturing complicated, non-linear interactions between features because of their deep, branching structure. Making these terms explicit is often unnecessary and can even introduce noise. We decided to use the raw (but scaled) features and the XGBoost algorithm's natural power to find these complicated relationships. This approach proved highly effective.

3.1.5. Exploratory Data Analysis (EDA)

We did an exploratory data analysis (EDA) on our cleaned 396-sample dataset before training the model to make sure it was accurate. The main goal was to look for multicollinearity, which is when input features are very similar to each other and can sometimes make it hard for a model to figure out which features are most important.

The Pearson Correlation Coefficient (PCC) was the main tool used for this analysis. It looks at how closely two variables, X and Y , are related in a straight line. It is calculated using the formula:

$$r_{xy} = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}} \quad (1)$$

For all 12 input features, a correlation heatmap (see Figure 5) was generated. The analysis showed that there was no serious multicollinearity between the key process and biomass variables (i.e., no PCC values > 0.8 or < -0.8). There were some moderate, expected correlations, like

between the amounts of C, H, and O, which is normal for organic matter. However, none were strong enough to justify taking away a feature.

We also considered the Variance Inflation Factor (VIF), which is a more formal way to test for multicollinearity. It is calculated as:

$$\text{VIF} = \frac{1}{1 - R_i^2} \quad (2)$$

A common thumb rule is to investigate features that have a VIF of more than 10. But this step wasn't as important for our model. Multiple tree-based models, such as the XGBoost algorithm we used, are naturally resistant to multicollinearity. They don't get "confused" by features that are related to each other like linear regression does; instead, they just choose the feature that gives the best split point at any given node. So, the PCC heatmap was enough to show that our feature set was suitable for modeling.

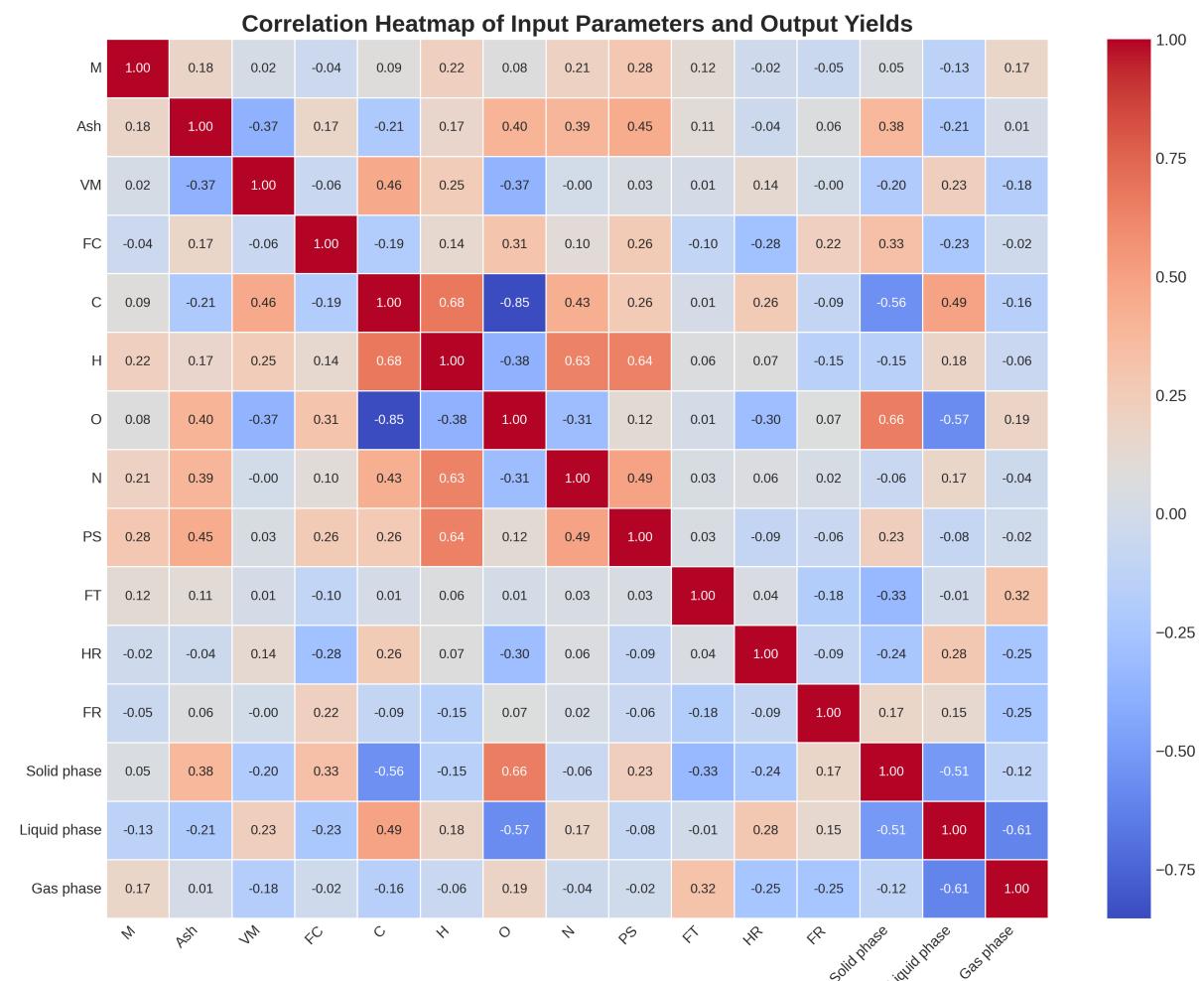


Figure 5: Correlation heatmap of input parameters and output yields.

3.1.6. Data Partitioning and Cross Validation

We divided the last cleaned dataset of 396 samples into a test and a training set to enable us to construct a safe model and truthfully test how effective it was on data it had never encountered before. We employed the use of the `train_test_split` function from the Scikit-learn library to achieve this. This is a standard and safe method of testing a model's accuracy.

The `train_and_load_model` function in our model employs an extremely precise 80/20 split. This split reserved 80% of the data (316 samples) for the training set and the remaining 20% (80 samples) as a test set which was not utilized for training. We also create a `random_state` of 42 while splitting to ensure our outcomes are reproducible and can be replicated by other researchers. This ensures that each time the code is executed, it will generate the same 316 training samples and 80 test samples.

Whereas k-fold cross-validation is an excellent method of hyperparameter tuning with smaller data sets, within the scope of this project, we trusted in the established, strong default parameters of XGBoost. The primary objective of this framework is to identify how well the ultimate model performs on new, real-world data. An obvious 80/20 train-test split is the optimal means of doing this. We used the training set for everything related to making the model, including fitting the `StandardScaler`. We never utilized the test set. The performance statistics in Section 4.1 constitute the final, most honest way to judge how well the model worked.

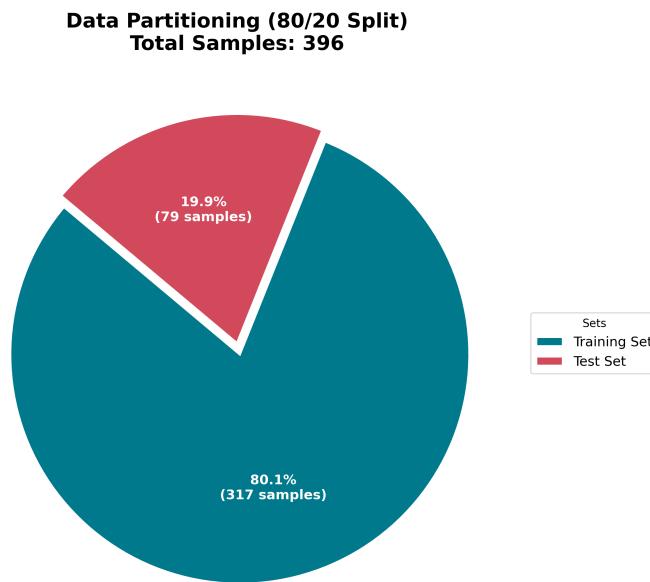


Figure 6: Data Split Pie Chart.

3.2. Deployment of the AI-Driven Pyrolysis Optimization Framework

3.2.1. AI/ML Models Used

This framework is built on the eXtreme Gradient Boosting (XGBoost) algorithm, which is a powerful, open-source library made for high performance. We chose XGBRegressor as our model because it is a “boosting” type of ensemble algorithm. This means that it builds hundreds of individual decision trees one after the other, and each new tree is smartly made to fix the mistakes and weaknesses of the previous ones.

This sequential, error-correcting method makes XGBoost exceptionally accurate at modeling the type of problem found in biomass pyrolysis—a process defined by highly complex, non-linear, and interactive relationships between many different variables (like temperature, moisture, carbon content, and heating rate). A basic linear model cannot do this, but XGBoost does a great job.

A fundamental methodological choice in the architecture of our model is the implementation of a specialized, multi-model framework. Our system trains three separate and distinct XGBRegressor models instead of one overly complex model to predict all three product yields at once. One model is trained to be an expert on the “Solid phase” (biochar) yield, a second on the “Liquid phase” (bio-oil) yield, and a third on the “Gas phase” yield.

This specialized method has a big advantage when it comes to accuracy and understanding. It lets each model find the best combination of features for its own target. The “Liquid phase” model, for instance, can become very sensitive to Heating Rate, and the “Solid phase” model can learn to focus more heavily on Final Temperature and Carbon content, just like a human chemical engineer would. By separating these concerns, we can make models for each product that are more accurate, reliable, and easy to understand. For this project, the models were trained with 100 estimators and a maximum depth of 5. This was the best way to get good predictions without overfitting.

3.2.2. Web-Based User Interface (UI) for Industrial Application

We made a separate web app to turn the powerful predictive and optimization features of our XGBoost models into a tool that is easy to use and useful in the real industrial world. This web interface is the main way that end users, like process engineers or researchers, can interact with the system. It lets them use the complex backend models without having to know how to program.

The UI (as shown in Figure 7) is divided into three easy-to-understand sections that are meant to look like a normal experimental or operational workflow:

- **Biomass Properties:** This part allows users to enter the exact details about their biomass feedstock. Users can enter values for the following: Ash, Moisture (M), Volatile Matter (VM), Fixed Carbon (FC), Carbon (C), Hydrogen (H), Oxygen (O), and Nitrogen (N). These

eight features are sent straight to the backend XGBoost models, which use them to change predictions based on the specific biomass being analyzed.

- **Process Parameters & Goal:** Here, users define the four primary process parameters that the model will optimize: Particle Size (PS), Final Temperature (FT), Heating Rate (HR), and Flow Rate (FR). This is also where the user selects their Optimization Goal, i.e., “Maximize Liquid phase (Bio-oil),” “Maximize Solid phase (Biochar),” or “Maximize Gas phase.”

Figure 7: Web-Page UI Interface.

- **Operational Constraints:** This section improves the program’s practicality by allowing users to specify realistic lower and upper limits for the four process parameters which can be optimized. For example, a user can state that Final Temperature must be between 300 °C and 900 °C, or that Heating Rate cannot exceed 50 °C/min due to equipment limitations. The Differential Evolution optimizer applies such constraints in order to ensure that the proposed best conditions are always feasible in the user’s real-world situation.

Connection to the Backend Model: When the “Run Optimization” button is clicked, the UI collects all the data that the user has entered, including the eight biomass properties, the optimization goal that was chosen, and the minimum and maximum operational constraints. The backend, which is powered by a FastAPI server, then gets this full request. The server, in turn, manages the whole process:

1. It loads the three XGBoost models that have already been trained.
2. It sets up the Differential Evolution optimizer with the goal and limits that were given.
3. It uses the properties of the biomass to set the input features that stay the same during the optimization run.
4. Finally, it runs the Differential Evolution algorithm, which uses the XGBoost models to look at possible solutions to the objective function.

The backend then sends the optimized process parameters (and the predicted maximum yield) back to the user, who sees them on the webpage. This smooth integration makes it possible to use complex AI/ML models in industrial settings. The easy-to-use design and strong backend connection make this a powerful tool for speeding up research, process development, and yield optimization in biomass pyrolysis.

3.3. Optimization Techniques

One of the main goals of this project was to make a framework that goes beyond simple prediction (a “what-if” tool) and allows for real prescriptive optimization (a “what’s-best” tool). To make a real decision-support system, we combine our trained XGBoost models with an advanced optimization algorithm.

The **Differential Evolution (DE)** algorithm, a powerful function from the `scipy.optimize` library, is used in our model, which is defined in `app.py`. There were a few important reasons why this algorithm was chosen. First, it is a stochastic, population-based method, which means it looks at many possible solutions at the same time. Second, it is very good at finding a global optimum in search spaces that are complicated, non-linear, and “bumpy,” like the one for pyrolysis yields. This is better than a simpler algorithm, which might get stuck in a sub-optimal “local” solution. This is very important for our process because we know that the relationship between inputs and yields is very interactive.

A custom-built **objective function** connects the machine learning model and the optimizer. This function is like the “brain” of the whole optimization process:

- **The Digital Twin:** Our three validated XGBoost models act as a fast and accurate “digital twin” of the real pyrolysis reactor. They can predict the yield of all three product phases in milliseconds.
- **The Goal:** When someone asks for an optimization, the Differential Evolution algorithm starts to intelligently test thousands of different combinations of the four process parameters that can be changed: Particle Size (PS), Final Temperature (FT), Heating Rate (HR), and Flow Rate (FR). The DE algorithm calls our `objective_function` for each new combination it tests. This function sends the parameters to our trained XGBoost model, gets the predicted yield for the user’s goal, and then gives back the opposite of that yield.
- **The Solution:** The only thing the Differential Evolution algorithm does is make this function return the smallest value possible. By determining the set of parameters that yield the lowest negative yield, one effectively identifies the parameters that yield the highest positive yield.

The web application we made, as seen in Figures 8 and 9, is the best way to show how useful this whole system is.

In this real-world example (Figure 8), a user has put the feedstock properties from a lab test (6.33% Moisture, 29.51% Ash, 22.49% Carbon, etc.). They have also entered their current (not optimized) operating settings, which include a Final Temp of 900 °C, a Heating Rate of

15 °C/min,etc. They want to “Maximize Target Yield: Solid phase (Bio-char)” as their goal.

The backend system runs the Differential Evolution algorithm after the user clicks “Run Optimization.” The “Optimization Results” table (Figure 9) makes the result very clear.

Industrial Pyrolysis Optimizer

A high-accuracy tool to find the highest possible yield within your equipment's limits.

1. Biomass Properties

Moisture (M, %)	6.33
Ash (%), db	29.51
Volatile Matter (VM, %)	51.64
Fixed Carbon (FC, %)	12.53
Carbon (C, %)	22.49
Hydrogen (H, %)	3.3
Oxygen (O, %)	73.68
Nitrogen (N, %)	0.51

2. Process Parameters & Goal

Particle Size (mm)	1.25
Final Temp (°C)	900
Heating Rate (°C/min)	15
Flow Rate (mL/min)	50

Optimization Goal:
Maximize Target Yield:

3. Operational Constraints

Set the feasible min/max range for your reactor.

Particle Size (mm)	0.1	-	1.35
Final Temp (°C)	300	-	900
Heating Rate (°C/min)	0	-	50
Flow Rate (mL/min)	0	-	500

Figure 8: Input Provided to Model.

Optimization Results

Metric	Current Value	Current Yield	Optimized Value	Optimized Yield
Solid phase (%)	-	51.01	-	66.35
Liquid phase (%)	-	6.93	-	11.23
Gas phase (%)	-	42.06	-	22.43
Particle Size (mm)	1.25	-	1.04	-
Final Temp (°C)	900.00	-	325.94	-
Heating Rate (°C/min)	15.00	-	10.32	-
Flow Rate (mL/min)	50.00	-	61.62	-

Figure 9: Output Obtained from Model.

The model first figures out that the user's current settings would give them a Solid phase yield of 51.01%.

After that, the DE optimizer runs and looks for the best settings that work within the user's limits. It finds a new, best set of conditions. Following these new rules has led to a big improvement in the yield for the Solid phase (the target), which is now 66.35%. The table is very important because it gives the exact, usable settings needed to do this. It tells the user to:

- Lower the Final Temp from 900.00 °C to 325.94 °C.
- Change the size of the particles from 1.25 mm to 1.04 mm.
- Lower the heating rate from 15.00 to 10.32 °C/min.
- Raise the flow rate from 50.00 to 61.62 mL/min.

This example shows how useful the tool is in industry: it takes a real-world situation and gives you a clear, data-driven set of optimized parameters that lead to a significant and measurable increase in the yield of the target product.

4. Results and Discussion

4.1. Performance Metrics

The first and most important step in validating our framework is to examine how well the trained models can predict outcomes. A model is only useful for optimization if it can be proven to be a true and accurate “digital twin” of the real-world process. To accomplish this, the models were trained using 80% of the dataset (316 samples) and subsequently assessed with the 20% held-out test set (80 samples) that they had not previously encountered.

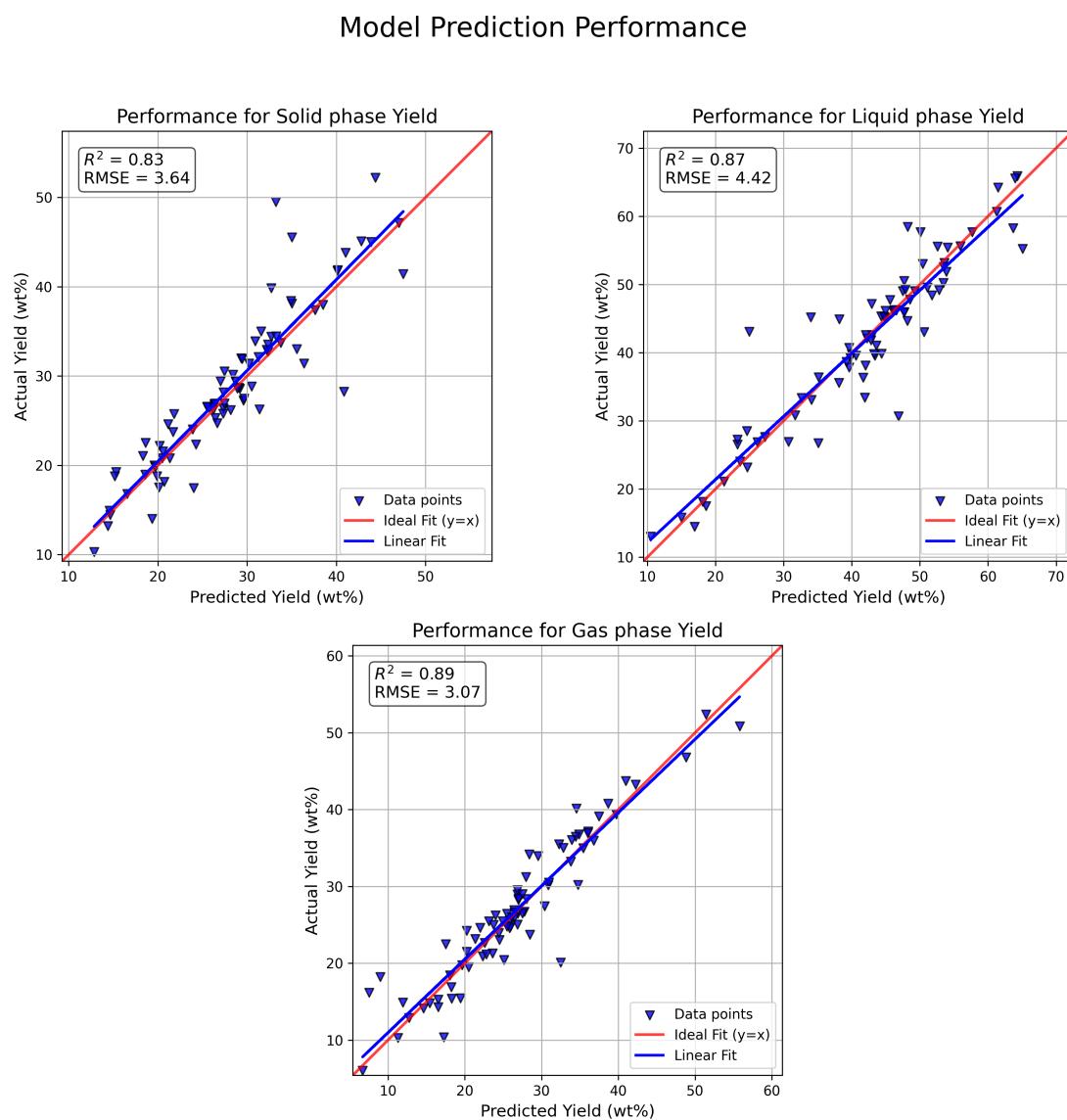


Figure 10: Model Prediction Performance for Yields in Solid, Liquid, and Gas Phases

Figure 10 shows the main results of this evaluation. It presents the *Actual vs. Predicted* scatter plots for each of the three product phases.

The 20% of the test set that was not used for training was employed to calculate the R^2 and RMSE values. These plots provide the most direct proof of the models' predictive performance. The x-axis represents the yield predicted by the XGBoost model, while the y-axis shows the actual yield from the test data. If the model were perfect, all data points would lie exactly on the red “Ideal Fit ($y = x$)” line.

The data points for all three models cluster closely around this ideal line, indicating a strong positive correlation between the model's predictions and actual experimental values. The blue “Linear Fit” line, representing the model's actual predictive trend, is also very close to the red ideal line, confirming that the predictions are unbiased.

We use two standard statistical measures, displayed on the graphs, to quantify model performance:

- **Coefficient of Determination (R^2):** This metric indicates how much of the variance in the actual yield can be explained by the input features. A value of 1.0 represents a perfect prediction, while 0.0 represents no predictive power beyond the mean.
- **Root Mean Squared Error (RMSE):** This represents the average magnitude of prediction errors. Since the yield (wt%) is in the same units as RMSE, a smaller RMSE value indicates better model performance.

Our XGBoost models demonstrated high predictive accuracy across all three product phases:

- **Gas phase:** $R^2 = 0.89$, RMSE = 3.07 wt%.
- **Liquid phase (Bio-oil):** $R^2 = 0.87$, RMSE = 4.42 wt%.
- **Solid phase (Bio-char):** $R^2 = 0.83$, RMSE = 3.64 wt%.

The fact that all three models achieved R^2 values well above 0.80 (and two above 0.85, often considered the threshold for “good” performance) on unseen test data strongly indicates model robustness. This high fidelity confirms that the models successfully captured the complex, non-linear interactions inherent in the pyrolysis process, rather than simply memorizing training data.

This validated accuracy provides the foundation for confidently applying these models in scientific analysis and industrial-scale optimization, as detailed in the following sections.

4.2. Model Interpretation and Feature Analysis

Since we proved in Section 4.1 that our XGBoost models are very accurate, we can now use them as a trusted analytical tool. This part “opens the black box” so we can see what the models learnt about pyrolysis and why they make the predictions they do. This analysis is important because it helps us make sure that our models are not only statistically correct, but also scientifically and chemically sound.

4.2.1. Feature Importance

The initial thing to do is to determine which among the 12 input features is most effective in making predictions. This allows us to determine which operational parameters and biomass qualities the model learned had the greatest influence on the final product yields. The XGBoost model computed the feature importance scores for each of the three specialized models, which you can find in Figure 11. The scores indicate the degree to which each feature impacts the accuracy of the model. They also provide us with a great deal of significant information that confirms to widely recognized chemical principles.

Solid Phase (Bio-char): The model learned that Oxygen (O), Ash, and Carbon (C) quantities are the most significant aspects in the solid phase (Bio-char). That is in accordance with principles of chemistry because bio-char is the non-volatile, carbon-rich residue from pyrolysis. Ash constitutes the inorganic and non-combustible component of biomass, and thus it is understandable that it directly and significantly influences the final solid yield. Similarly, biomass high in oxygen initially tends to have more thermally stable organic frameworks that are less prone to convert to gas, thereby contributing more mass to the final char. Lastly, because carbon is the predominant building block of bio-char, the carbon content in the feedstock at the beginning is a significant determinant of how much char can be produced.

Liquid Phase (Bio-oil): The model indicated that Oxygen (O), Nitrogen (N), and Volatile Matter (VM) were the most significant factors for the liquid phase (bio-oil). This is also correct from a chemical perspective. During pyrolysis, when aerosols and vapours (volatiles) are emitted, they rapidly condense and cool to form bio-oil. Therefore, having a high Volatile Matter (VM) content is the most obvious prerequisite for obtaining much bio-oil. It is understandable that oxygen (O) and nitrogen (N) are essential because they are components of the complex oxygenated and nitrogenated organic molecules (such as phenols, furans, and pyridines) that constitute the chemical composition of bio-oil. A feedstock containing a great deal of these components will, of course, produce a bio-oil with a different yield and composition.

Gas Phase: The model indicated that for the gas phase, the most significant factors are Moisture (M), Ash, Carbon (C), and Oxygen (O). This rightly identifies that initial Moisture (M) greatly affects gas production, as water converts to steam and participates in secondary water-gas and steam-reforming reactions, leading to more CO, CO₂, and H₂. Ash content is negative because it indicates that there is less organic material that can be converted to gases that burn.

Feature Importance for Each Product Yield

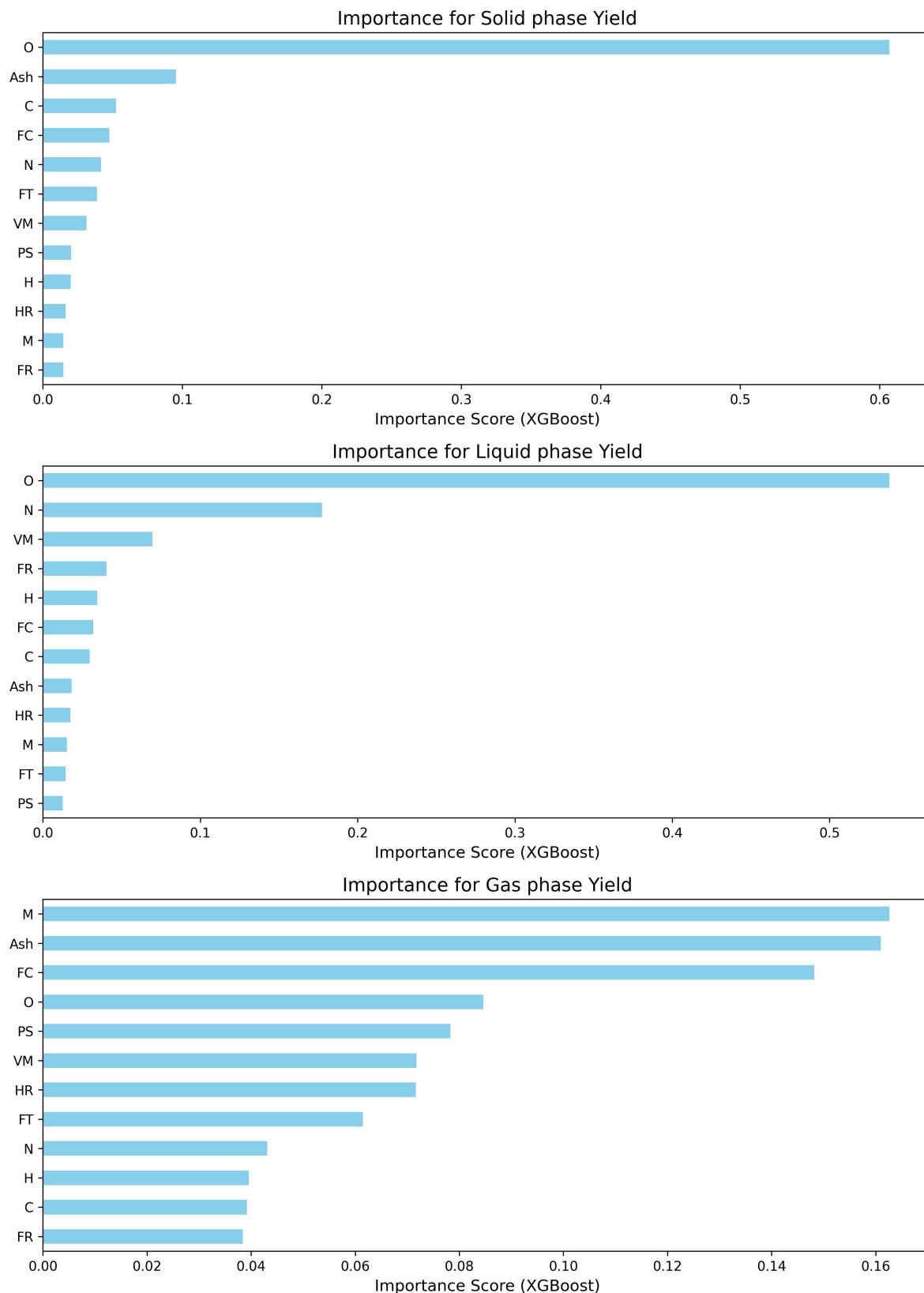


Figure 11: Feature Importance for models of solids (top), liquids (middle), and gases (bottom). The XGBoost algorithm calculates scores that show how much each feature helps the model's accuracy.

4.2.2. Univariate Partial Dependence (Single-Feature Effects)

Feature importance lets us know what is important, but it doesn't tell us how that feature affects the yield. We use Univariate Partial Dependence Plots (PDPs) to answer this question, as shown in Figure 12. These plots show what happens when you change just one feature (like Oxygen) across its whole range, while keeping the other 11 features at their average values.

These graphs show that the model has learnt how to model real-world, non-linear relationships:

- **Solid phase:** The plots show that the yield goes up a lot when the Oxygen (O) and Ash content goes up. This is a direct validation because these parts are the main non-volatile parts of biomass that make up the final char.

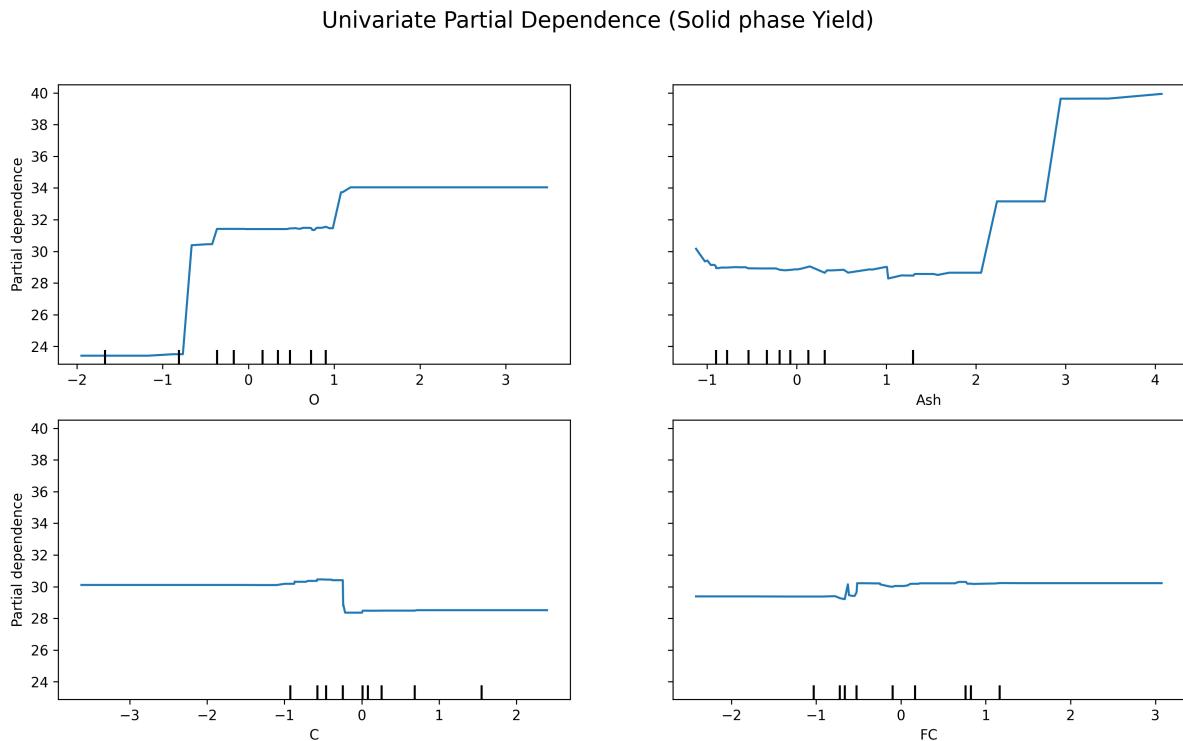


Figure 12: Univariate Partial Dependence Plot for the Solid phase model.

- **Liquid phase:** The model found a strong negative link between Oxygen (O) and the Liquid phase. The plot shows that when the amount of oxygen goes over a certain level, the amount of bio-oil produced drops quickly. This is a known chemical phenomenon: when the feedstock has more oxygen, it often makes more water and CO/CO₂, which lowers the final organic liquid yield.
- **Gas phase:** The model accurately determined that in the gas phase, there is a strong positive correlation with moisture (M) (as moisture turns into steam and causes water-gas shift reactions) and a negative correlation with ash (as ash is inert and does not contribute to gas production).

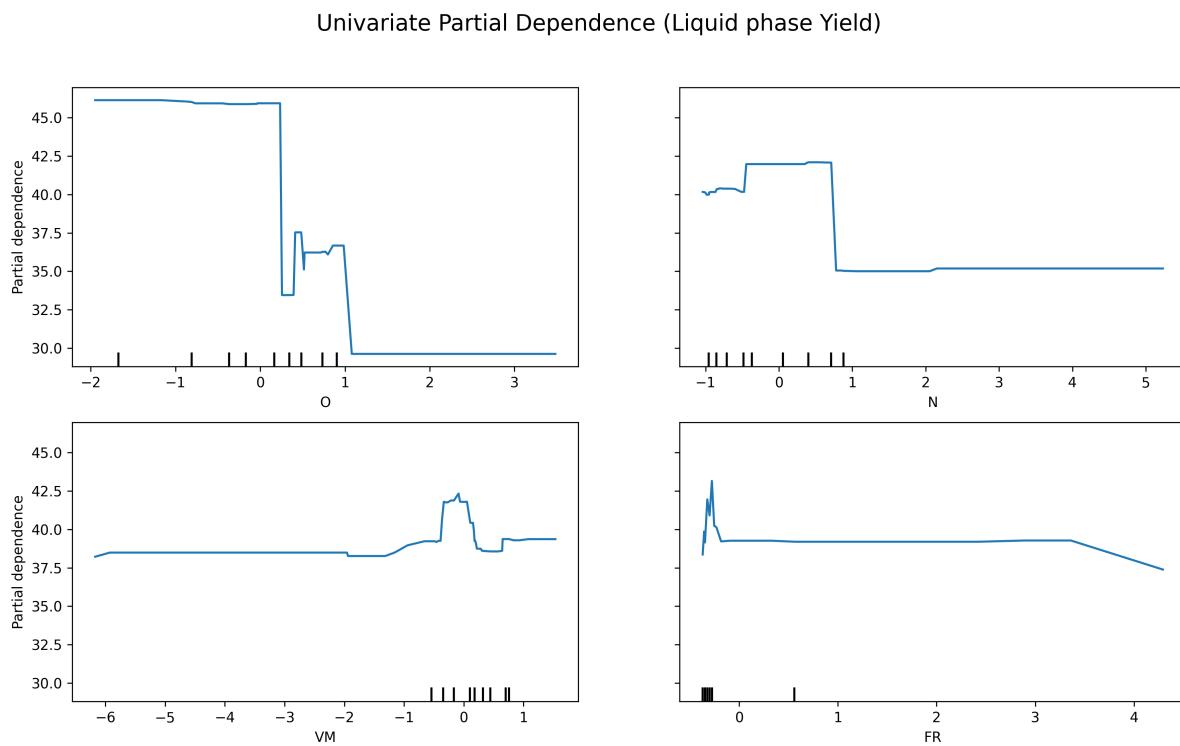


Figure 13: Univariate Partial Dependence Plot for the Liquid phase model.

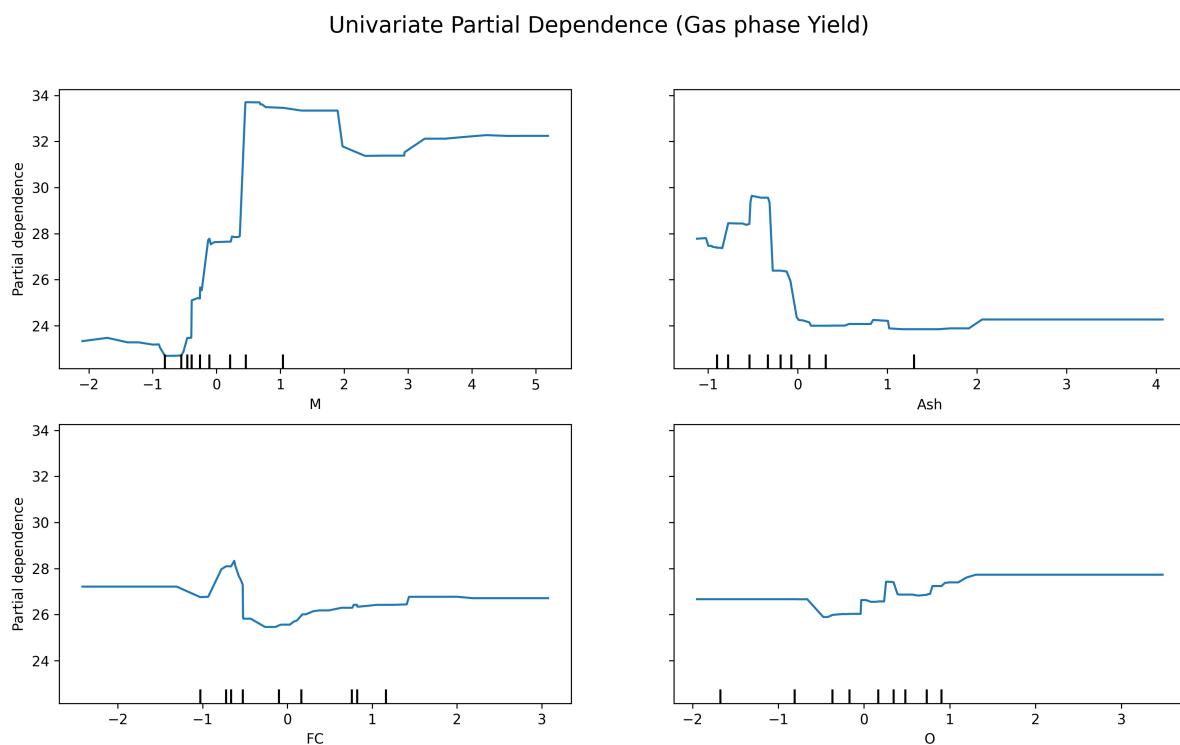


Figure 14: Univariate Partial Dependence Plot for the Gas phase model.

4.2.3. Analysis of Bivariate Interaction (Effects of Two Features)

The Bivariate PDPs give us the most advanced insight from our model. For true optimisation, these plots are very important because they show how two features work together to have a synergistic or interactive effect.

This analysis gives a clear, visual guide for making processes better:

- **Solid Phase (Top):** The O vs C plot shows that the best bio-char yields (the yellow “hot zone,” >45 wt%) come from a certain mix of high Oxygen (>2) and low Carbon (<-2), not just high Oxygen.

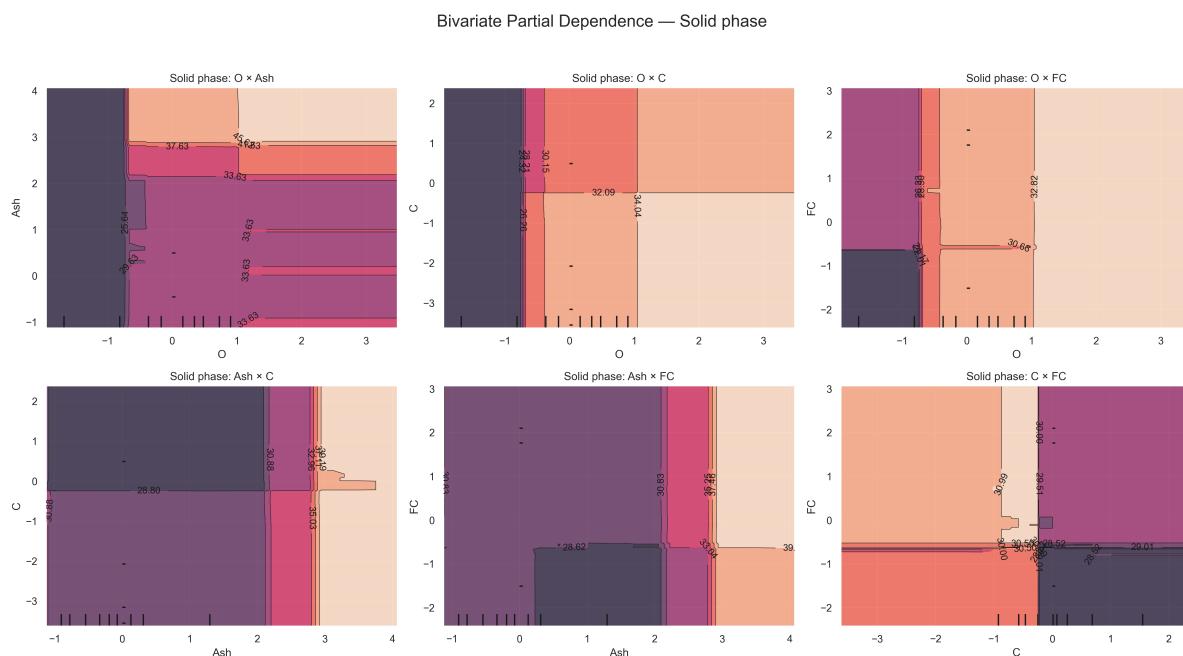


Figure 15: Bivariate Partial Dependence Plot for the Solid phase model.

- **Liquid Phase (Middle):** The O vs. N plot shows that the best bio-oil yields (>45 wt%) can only be found in a very specific operational corner, where the Oxygen level is low (<-1) and the Nitrogen level is high (>1). This shows that it's not enough to optimize one variable on its own.
 - **Gas Phase (Bottom):** The M vs Ash plot makes it clear that the best gas yields (>32 wt%) are achieved when the moisture is high (>0) and the ash is low (0).

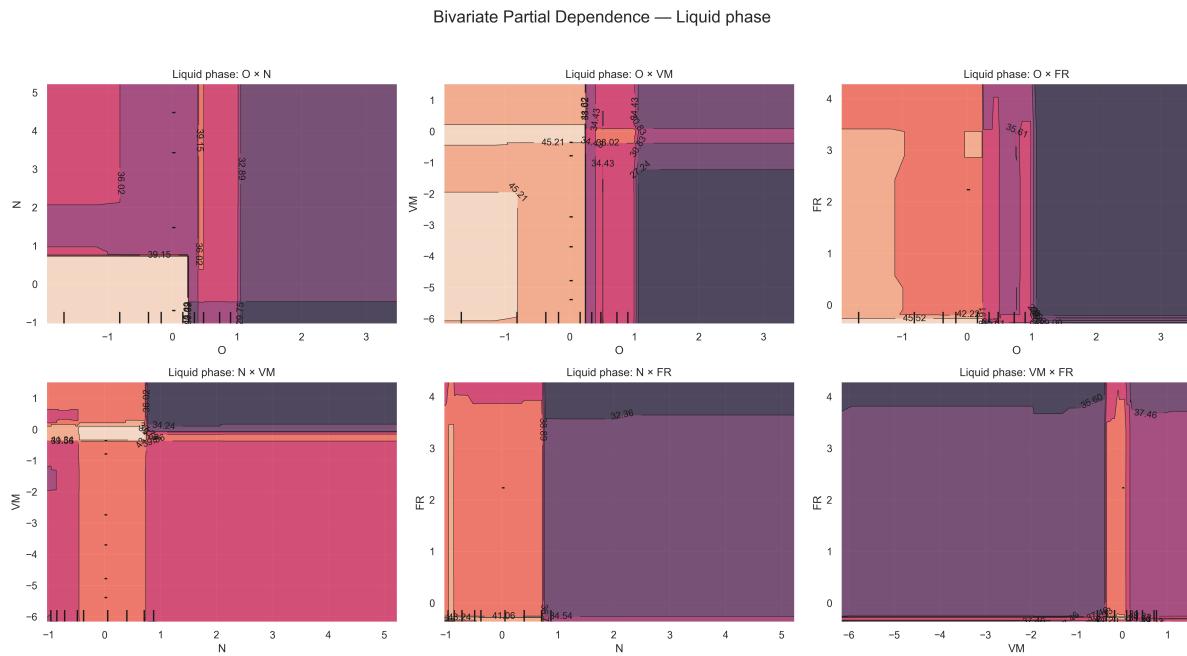


Figure 16: Bivariate Partial Dependence Plot for the Liquid phase model.

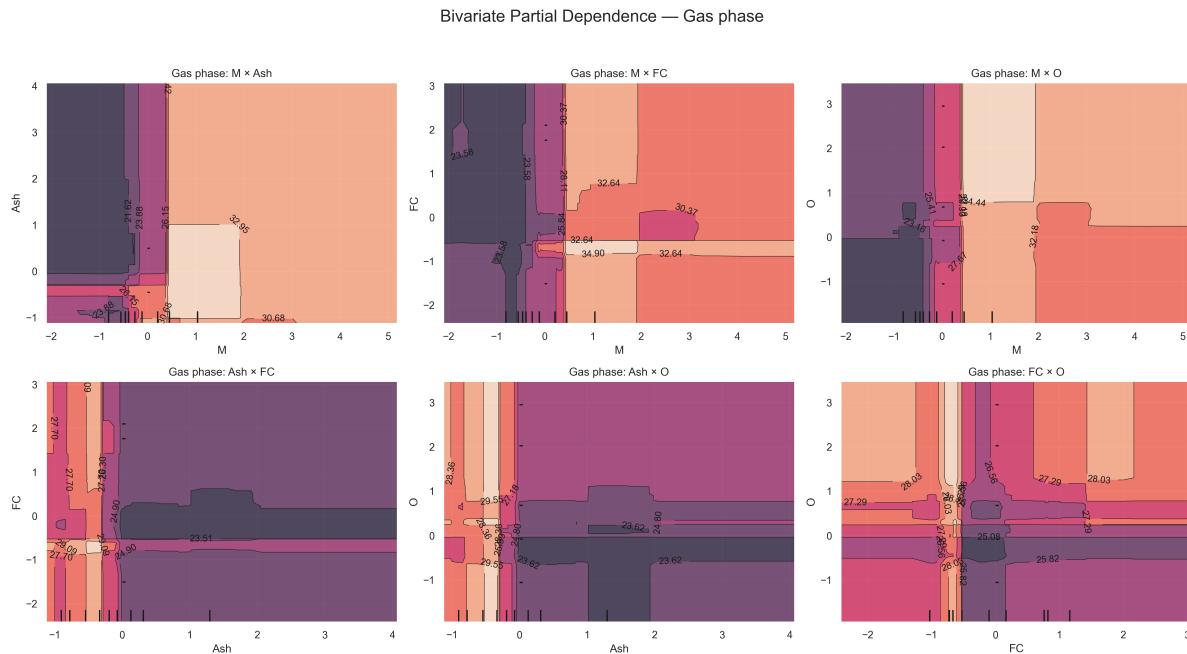


Figure 17: Bivariate Partial Dependence Plot for the Gas phase model.

In short, this in-depth study shows that our models are not just a “black box.” They have successfully learnt and replicated the complex, multi-variable, and non-linear relationships of biomass pyrolysis, proving that they can be used as a reliable tool for scientific analysis and industrial optimization.

5. Applications

5.1. Industrial Impact

This **AI-powered optimisation framework** has a big impact on industry that goes beyond just making predictions. It adds real value in three main areas: **accelerating research and development, increasing profits, and enabling operational flexibility.**

1. Drastic Reduction in R&D Costs and Time:

A big drop in the time and money spent on research and development. Finding the best process conditions for a new biomass feedstock in a traditional industrial setting is a long, costly, and material-intensive process. It takes a “design of experiments” (DoE) approach, which means engineers have to run dozens of “experimental sweeps” in a pilot-scale reactor to physically map out the yield response. These tests are expensive and take a long time. Our tool makes this whole process digital. From the web interface, an engineer can now run thousands of virtual tests in just a few seconds. They can find out how good a new feedstock is before they buy a whole batch by entering its lab analysis. This shortens the research and development (R&D) and process-tuning cycle from weeks or months to just a few minutes.

2. Direct Yield and Revenue Maximization:

The tool’s ability to find non-intuitive optimal settings that maximize the yield of high-value products has the most direct and important effect. Figures 8 and 9 show a great example of how to do this.

In this case, the plant’s current (unoptimized) settings were set to a very high 900°C, which made sense because more heat should mean more char production. The “Current Yield” calculation in our model showed that this would make a good 51.01% bio-char. But this assumption was not the best.

The **Differential Evolution algorithm**, which used the **XGBoost model** as a guide, found a much better, less obvious set of parameters after running the optimization with the goal of “Maximize Solid phase.” The optimizer said to drop the temperature by a lot to 325.94°C, as well as make the particles a little smaller and the heating rate a little slower. It was thought that the “Optimized Yield” for these new settings would be 66.35%.

This isn’t just a 15% increase; it’s a **30% relative improvement in product yield** that was made possible by changing the reactor’s setpoints without adding any new equipment. For an industrial plant, a 30% increase in the output of a high-value product like bio-char, along with a 20% drop in the output of lower-value gas (from 42% to 22%), means a huge and immediate boost in profitability and process efficiency.

3. Critical Feedstock Flexibility and Process Stability:

Finally, this tool solves one of the biggest problems in the biorefinery business: the fact that feedstocks can change. Industrial plants don’t have the luxury of always having a lab-grade feedstock. Depending on the season, location, and market prices, their supply of biomass can

change every week. For example, they might get pine wood one week and corn stover the next. A traditional physical or kinetic model is often “brittle” and only works with one type of feedstock. This means that it doesn’t work when the supply changes.

Our tool is naturally adaptable. An operator can easily enter the new lab analysis for the incoming batch of biomass because the 8 biomass properties are inputs to the model. The optimiser will then give them a new set of best running conditions that are specific to that feedstock. This ability to quickly adjust to changes in the real-world supply chain is a big advantage for businesses. It lets a plant stay as efficient and profitable as possible by making feedstock variability a manageable variable instead of a big operational problem. This makes the operation much more resilient and competitive in the market, with less waste and more consistent product quality.

6. Conclusion

This project successfully created, tested, and put into use a full machine learning framework for predicting and optimising biomass pyrolysis yields. The main goal—to go beyond basic academic prediction and make a useful, industrial-strength decision-support tool—was fully met.

Three specialised eXtreme Gradient Boosting (XGBoost) models make up the core of this system. They were very good at modelling the complicated, non-linear dynamics of the pyrolysis process. The models showed high fidelity on unseen test data after being trained on a cleaned and validated dataset of 396 experimental samples. They had strong predictive performance across all three product phases (Solid phase R^2 : 0.83, Liquid phase R^2 : 0.87, and Gas phase R^2 : 0.89).

A thorough analysis of these models also showed that they were scientifically sound. The rankings of feature importance and the partial dependence plots showed that the models had learnt and copied basic principles of chemical engineering on their own. They accurately identified the primary influences of Final Temperature (FT) and feedstock composition (O, C, Ash) on biochar yield, and substantiated the industrial notion of “fast pyrolysis” by correlating elevated Heating Rates (HR) with optimal bio-oil production.

The main new thing about this work was that it was able to combine this “digital twin” with a Differential Evolution (DE) optimiser. The web application shows that this system is not only predictive, but also prescriptive. It lets an operator enter real-world feedstock data and operational limits and get a new set of parameters that can be used to get the best yield. Our case study showed that the system was useful because it could find non-intuitive settings that increased the bio-char yield from 51.01% to 66.35%. This was a big improvement in process efficiency and value.

In the end, this project gives us a strong and successful model for adding modern AI tools to process engineering. We built the whole system, from cleaning the data and training the model to optimising and deploying it with a FastAPI backend. This tool solves the important industrial problems of feedstock variability and expensive experimental “sweeps.” This framework makes biorefinery operations smarter, more flexible, and more profitable, which speeds up the shift to sustainable energy and materials.

7. References

1. Jerzak, W.; Acha, E.; Li, B. Comprehensive Review of Biomass Pyrolysis: Conventional and Advanced Technologies, Reactor Designs, Product Compositions and Yields, and Techno-Economic Analysis. *Energies* 2024, 17(20), 5082.
2. Parvari, E.; Mahajan, D.; Hewitt, E.L. A Review of Biomass Pyrolysis for Production of Fuels: Chemistry, Processing, and Techno-Economic Analysis. *Biomass* 2025, 5(3), 54.
3. Nandabalan, Y.; Poornima Devi, T.; Sivashanmugam, P.; Kavitha, S. Lignocellulosic Biomass-based Pyrolysis: A Comprehensive Review. *Chemosphere* 2021, 286 (Pt 2), 131824.
4. Ranzi, E.; Cuoci, A.; Faravelli, T.; Frassoldati, A.; Migliavacca, G.; Pierucci, S.; Sommariva, S. Chemical Kinetics of Biomass Pyrolysis. *Energy & Fuels* 2008, 22(6), 4292–4300.
5. Dhyani, V.; Bhaskar, T. A Comprehensive Review on the Pyrolysis of Lignocellulosic Biomass. *Renewable Energy* 2018, 129, 695–716.
6. https://www.mdpi.com/suschem/suschem-06-00013/article_deploy/html/images/suschem-06-00013-g001.png
7. <https://arhse.com/wp-content/uploads/2020/08/Steps-in-Pyrolysis.jpg>
8. <https://www.kaggle.com/datasets/mustafakeser4/biomass-pyrolysis-data?select=pyrolysis.csv>



AI Detector

- Paraphraser
- Grammar Checker
- AI Detector
- Plagiarism Checker
- AI Humanizer
- More

English French Spanish German All ▾

1 Prepared By:

Scarface

Sandesh Wabale : U24CH016

Pratham Saini : U24CH012

Ayush Chaudhari : U24CH089

Zohair Nalawala : U24CH037

Mentors :

Om Prajapati

Krishna Karnani A Eureka Report on :

"AI-ML Driven Enhancement of Biomass

Pyrolysis Yield " Advanced Process Engineering and

System DesignAI-Enhanced Biomass Pyrolysis Report

Advanced Process Engineering and System Design Eureka

Report Submission-2 Problem Statement: Developing a

Machine Learning Framework using Artificial Intelligence
for Predicting and Optimizing Biomass Pyrolysis Yields.

8,811 Words



Analysis complete

Model Version: v5.7.1

Share Download

13%

of text is likely AI ⓘ

QuillBot



AI-generated ⓘ

Human-written & AI-refined ⓘ

Human-written ⓘ

13%

2%

85%

Main AI Contributors ⓘ

30% OFF

No Words Limit Plagiarism API User Seats No Ads Accurate Result

 Starting from \$9.80

Tools

Plagiarism Checker

[Make it Unique](#)

[Check Grammar](#)

[Detect AI](#)

[Summarizer](#)

[AI Humanizer](#)

OMIT SETTINGS (1)

citation

Result

Word Statistics

Prepared By:
Scarface
Sandesh Wabale : U24CH016
Pratham Saini : U24CH012
Ayush Chaudhari : U24CH089
Zohair Nalawala : U24CH037

Mentors:
Om Prajapati
Krishna Karnani A Eureka Report on:
"AI-ML Driven Enhancement of Biomass Pyrolysis Yield" Advanced Process Engineering and System DesignAI-Enhanced Biomass Pyrolysis Report
Advanced Process Engineering and System Design Eureka Report Submission-2 Problem Statement:
Developing a Machine Learning Framework using Artificial Intelligence for Predicting and Optimizing Biomass Pyrolysis Yields. Team Name: Scarface November 1, 2025 1 Advanced Process Engineering and System DesignAI-Enhanced Biomass Pyrolysis Report Abstract The depletion of fossil fuels and growing environmental concerns have led to increased interest in biomass pyrolysis as a renewable and carbon-neutral way to produce useful energy products like bio-oil, biochar, and pyrolytic gas. However, the efficiency and quality of pyrolysis outputs depend largely on the physical and chemical properties of biomass as well as the operating conditions, making process optimization quite complex when using

 Plagiarism Results for 1000 Words:

0% Plagiarism  





Congratulation!
No Plagiarism Found