Evaluating Biochar for the Removal of Pharmaceutical Pollutants from Wastewater

Oohasripriya Nandipati
Harika Venagala
Tolulope Okeowo

DSA 5900

4 Credit Hours

Faculty Supervisor
Jude Okolie

Fall Semester, 2024

Table of content

List of Figures

List of Tables

Glossary

Adsorption Capacity: The ability of biochar to remove pharmaceutical pollutants from wastewater. It is measured in micrograms per gram.
Biochar: Carbon-rich material produced by the pyrolysis of organic matter, which is used for environmental applications like pollutant removal.
Boruta: A feature selection algorithm used to identify the most important features for predicting adsorption capacity.
Cross-Validation: In this project, we employed k-fold cross-validation, a robust technique for dividing data into subsets to evaluate and validate model performance effectively.
Ensemble Methods: Machine learning techniques that combine multiple models to improve predictive performance, including Bagging (Bootstrap Aggregating): this creates multiple models using random subsets of data, Boosting: this sequentially builds models focusing on previous models' errors.
Feature Engineering: this is the process of transforming raw data into more meaningful features. Some examples are log transformations, label encoding, and feature importance analysis.
GridSearchCV: A hyperparameter tuning technique that systematically searches for optimal model parameters.
Hyperparameter Tuning: This is the process of optimizing the model parameters to improve predictive performance.
MLflow: A platform used for model management and tracking during the project.
Performance metrics: RMSE (Root Mean Squared Error): this measures the average magnitude of prediction errors, MAPE (Mean Absolute Percentage Error): this calculates the average percentage difference between predicted and actual values.
Supervised Learning: This is a machine learning approach where models are trained on labeled data to make predictions.
Streamlit: A framework used to deploy the machine learning model as an interactive web application.

1 Acknowledgement

The authors extend their heartfelt gratitude to our project advisor, Jude Okolie, for his guidance and support that made the successful completion of this project possible. His invaluable expertise and mentorship have provided us with a profound knowledge and experience that we will carry forward. Additionally, we would like to acknowledge and appreciate ourselves for the mutual support and collaboration that were crucial throughout the production of this project.
Oohasripriya Nandipati
Harika Venagala
Tolulope okeowo

2 Introduction

Pharmaceutical pollutants in wastewater have become a growing concern due to their potential risks to both human health and aquatic ecosystems. These contaminants, including antibiotics, hormones, and other pharmaceuticals, often persist in the environment because traditional wastewater treatment methods are not designed to detect or remove them effectively. The microscopic nature of these pollutants further complicates their detection, allowing them to accumulate over time and posing serious long-term risks to biodiversity, water quality, and public health. As such, addressing the issue of pharmaceutical pollutants in wastewater is essential for safeguarding the environment and ensuring the health and well-being of communities.

One promising solution for the removal of pharmaceutical pollutants is the use of biochar. Biochar, a carbon-rich material produced by the pyrolysis of biomass under low-oxygen conditions, has garnered attention for its high adsorption capacity. Its porous structure(PS), large surface area(BET), and ability to adsorb a wide range of contaminants make it an effective adsorbent for wastewater treatment. Compared to other methods, such as chemical coagulation or membrane filtration, biochar offers a sustainable and cost-effective approach, utilizing organic waste materials while simultaneously reducing environmental impact. Additionally, biochar can be applied to soil, where it contributes to soil health, improves crop yields, and acts as a carbon sink, further enhancing its environmental benefits.

This project aims to evaluate the effectiveness of biochar as an adsorbent for pharmaceutical micro pollutants. By exploring the properties of biochar, such as its surface area and porosity, we aim to optimize the conditions (e.g., temperature, pH, and contact time) that maximize its adsorption capacity. Machine learning techniques will be applied to analyze and model the adsorption process, helping to identify the most efficient configurations for biochar use in wastewater treatment. The long-term goal of this project is to develop a scalable and sustainable method for removing pharmaceutical pollutants, ultimately contributing to cleaner water sources and healthier ecosystems.

The future benefits of this research extend beyond just wastewater treatment. By incorporating biochar into agricultural practices, we can enhance soil quality, sequester carbon, and reduce greenhouse gas emissions, contributing to the fight against climate change. Moreover, the use of biochar for pollutant removal will help mitigate the harmful effects of pharmaceutical contaminants on

aquatic life, protect human health, and ensure the sustainability of freshwater resources for future generations.

3 Objectives

This project aims to evaluate the effectiveness of biochar as an adsorbent for the removal of pharmaceutical micro pollutants from wastewater. The specific problem addressed in this project is the persistent presence of pharmaceutical pollutants in wastewater, which poses significant risks to human health and aquatic ecosystems. By harnessing the unique properties of biochar, this project seeks to develop a comprehensive understanding of its capabilities in wastewater treatment.

To accomplish this objective, the following specific objectives have been established:

- Optimize Conditions: The objective is to identify the optimal conditions for maximizing biochar's adsorption capacity. This includes evaluating variables such as temperature, pH, and contact time to ascertain their effects on the adsorption process. By optimizing these conditions, the project aims to enhance the practical applicability of biochar in real-world wastewater treatment scenarios.

- Comparing the Machine Learning Models: In this project, various machine learning algorithms will be employed to predict the maximum adsorption capacity (Qm in mg/g) of biochar for pharmaceutical pollutants. The algorithms include Random forest, Decision tree, XG Boost, Gradient boost. Each model will be evaluated based on performance metrics such as Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) in order to identify the most effective approach for this task. The comparison will involve hyper-parameter tuning to optimize model performance.

- Deployment: The final stage of the project is having our best machine learning model as an inference endpoint. This will be achieved using ML flow, which allows for easy model management and tracking. For deployment Stream-lit framework will be utilized to create an interactive web application, enabling users to input parameters and receive predictions on the effectiveness of bio char in removing pharmaceutical pollutants. This streamlined process ensures that the model is readily available for practical applications and user engagement.

In conclusion, this project aims to address a critical environmental challenge by harnessing the potential of biochar in wastewater treatment. By evaluating its effectiveness, optimizing conditions, understanding property-performance relationships, and developing predictive software, the project seeks to contribute to sustainable practices in managing pharmaceutical pollutants and improving water quality for the benefit of human health and the environment.

4 Data

The dataset provided by the faculty supervisor, stored in Excel format, contains more than 34 raw materials used to produce biomass. It consists of 12 features (11 predictors and one target variable) and 86 observations. Given the limited number of observations relative to the number of features, there is a potential risk of under-fitting when training the model. Ensemble methods and adding complexity to the model will be useful to reduce under-fitting. The features consists of Biomass(Raw material used to produce biochar), TP(Type of Pollutant absorbed in the experiment), Temp(Pylorysis Temperature), Time(Pylorysis Time in minutes), PS(Pore Size of Biochar), BET(Surface Area of Biochar), PV(Pore Volume of Biochar), C(Carbon), H(Hydrogen), N(Nitrogen), O(Oxygen). Finally, the target variable is Qm(Absorption Capacity in milligrams per gram ,mg/m).These properties are influenced by the nature of the feedstock(biomass) and the preparation method, such as pyrolysis. High-temperature pyrolysis typically produces hydrophobic biochars with higher surface area and pore volume, making them ideal for organic pollutant adsorption. Conversely, low-temperature pyrolysis yields biochars with smaller pore sizes, lower surface area, and higher oxygen-containing functional groups, better suited for inorganic contaminants. The enhanced characteristics of biochar, such as its stable structure, rich carbon content, and adsorption capacity, make it an efficient material for removing heavy metals, dyes, and even pathogens from wastewater. A few rows of raw data can be seen here in Figure 1.

	Biomass	TP	TemP	Time (min)	PS	BET	PV	C	H	N	O	Qm (mg/g)
0	Waste sludge (paper)	Citalopram	315	150.0	9.82	3.43	0.02	30.84	2.14	0.43	20.32	4.4
1	Waste sludge (paper)	Citalopram	600	10.0	1.37	94.39	0.06	30.69	0.96	0.32	20.41	3.8
2	Waste sludge (paper)	Citalopram	800	10.0	1.37	120.86	0.08	28.81	0.47	0.33	19.29	8.5
3	Waste sludge (Biological)	Citalopram	800	150.0	1.41	209.12	0.13	27.05	0.82	0.33	9.73	19.6
4	Waste sludge (Biological)	Citalopram	800	10.0	3.69	10.82	0.02	35.35	0.72	2.47	2.01	4.3

Figure 1: First 5 rows of Data

4.1 Exploration and Cleaning

In the dataset, there are columns that initially contained different data types. To streamline our analysis and facilitate numerical computations, we converted all column data types to `float`, except for Biomass and TP. Dataset also had missing values in several columns: **Biomass**, **PS**, **PV**, **C**, **H**, **N**, and **O**. To address missing values in this project, different imputation strategies were applied based on the nature of the features. For the numerical columns, **linear interpolation** was used with a forward direction. This was chosen because it estimates missing values based on the trend of the surrounding data points. In contrast, mean and mode imputations might distort the data distribution. Linear interpolation is particularly effective when the data exhibits a continuous trend over time or other ordered variables. The missing value count is shown in figure 2.

Column names	Missing count
PS, PV	1
C, O	3
H	4
N	7

Figure 2: Missing Count

For categorical features, such as '**Biomass**', missing values were handled using the forward-fill (fill) method with the previous value. In our dataset, which is relatively small, analysis of the missing values in 'Biomass' revealed that the same biomass is shared by two rows, possibly due to a data entry error. This observation justifies the use of the forward-fill method. For larger datasets, mode imputation would be a more suitable approach for handling missing values in categorical columns.

4.2 Outlier Detection

To understand the dataset's distribution, box plots were created for each column. Box plots provide insights into central tendency (median), spread (inter quartile range), and outliers (anomalous data points). These plots help identify outliers and reveal patterns within the data. The box plot for all the variables is plotted in figure 3.

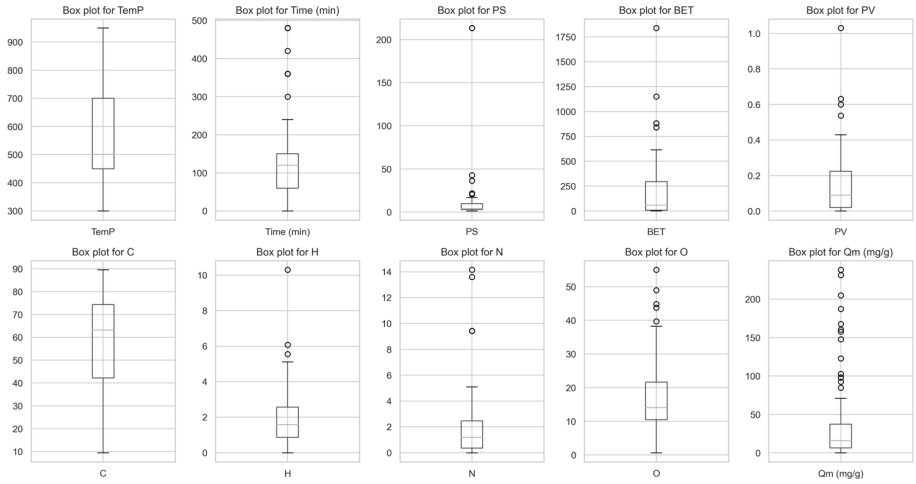

Figure 3: Box Plot

We observed potential outliers in the boxplot, but these are not true outliers, as confirmed by applying Grubbs' test with an alpha of 0.05. Grubbs' test is applied to multivariate data by first converting it into univariate data using the Mahalanobis distance. This distance measures how far each data point is from the mean of the distribution, accounting for correlations between variables. After calculating the Mahalanobis distance for each data point, Grubbs' test computes a statistic that quantifies the extremity of each point's distance relative to the mean and standard deviation of the distances. A critical value is then determined based on the sample size and significance level based on the t-distribution. If a data point's statistic exceeds the critical value, it is flagged as an outlier. However, in our case, no data points exceeded this threshold, confirming that there are no true outliers.

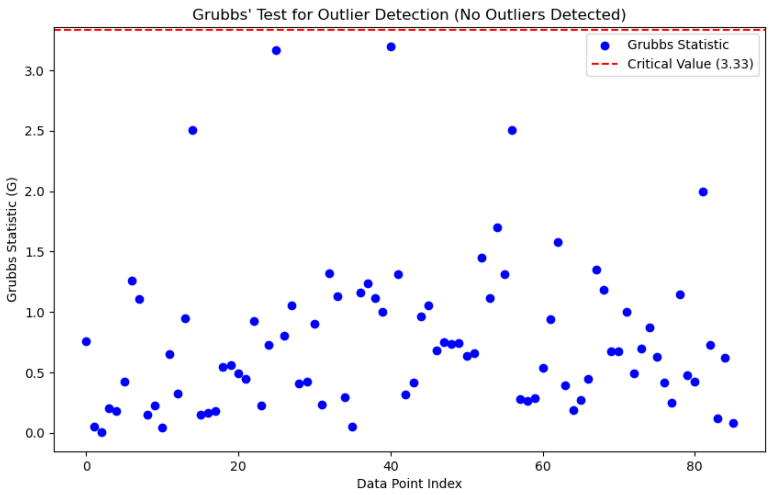

Figure 4: Grubb's Test for Outlier Detection

In Figure 4, all the data points are below the threshold line, which means there are no outliers at an alpha level of 0.05 and with 95% confidence.

4.3 Data Distribution

Histograms were plotted to identify the skewness of all variables in figure 5.

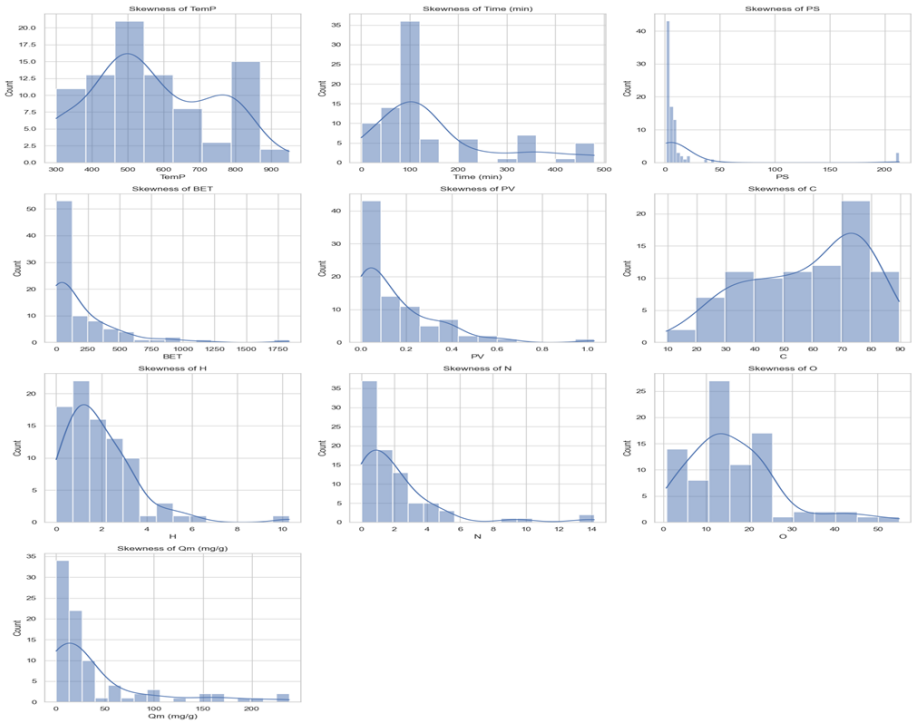

Figure 5: Histograms(before log transformation)

Features exhibiting right skewness (Time, PS, BET, PV) underwent log transformation to reduce skewness and achieve a more normal distribution. This step helps improve the performance of machine learning models. Histograms were used to confirm the normalization effect of the log transformation.

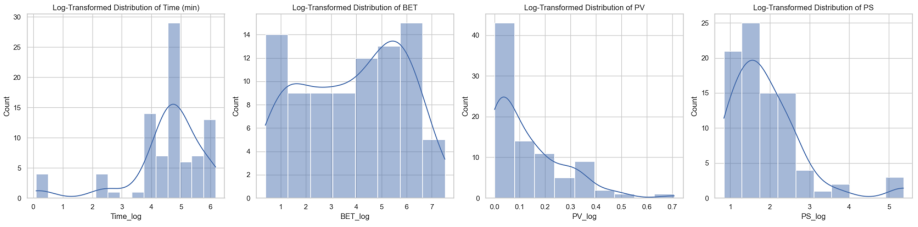

Figure 6: Histograms(after Log Transformation)

4.4 Correlation Matrix

Columns in the dataset were updated with the log transformation and a correlated heat map was applied on the data in figure 6. A correlation heat map was used to analyze the linear relationships between variables. Key Finding: **BET** and **PV** showed a positive correlation of **0.68**. Features with positive correlations to the target variable **Qm (mg/g)**: **PV_log**, **BET_log**, **N**, and **O**. Features with negative correlation: C, Time_log, and PS_log. **Low correlation does not necessarily mean these features are irrelevant. They may still contribute to the model through nonlinear relationships or interactions with other features.**

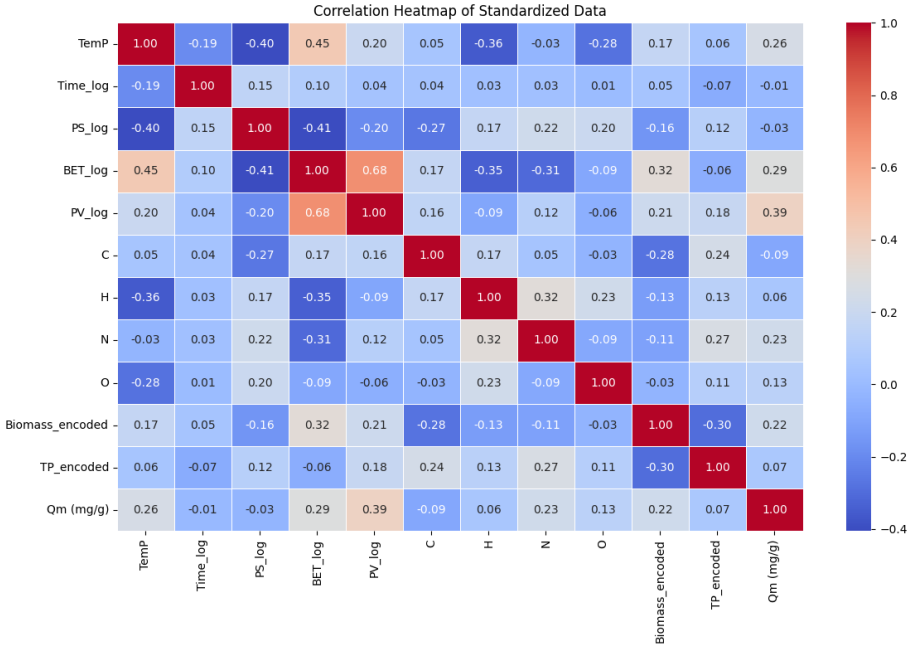

Figure 7: Heat Map

4.5 Feature Selection

Boruta is a feature selection algorithm designed to identify the most relevant features by iteratively removing features deemed less important than randomized versions of the same data (shadow features). In this analysis, the BorutaPy implementation was used with a base model of RandomForestRegressor, which leverages the power of ensemble learning to evaluate feature importance. Boruta automatically optimizes the number of estimators (n estimators='auto') to ensure robust feature selection. Features are ranked based on their importance, with a ranking of 1 indicating the most significant features.

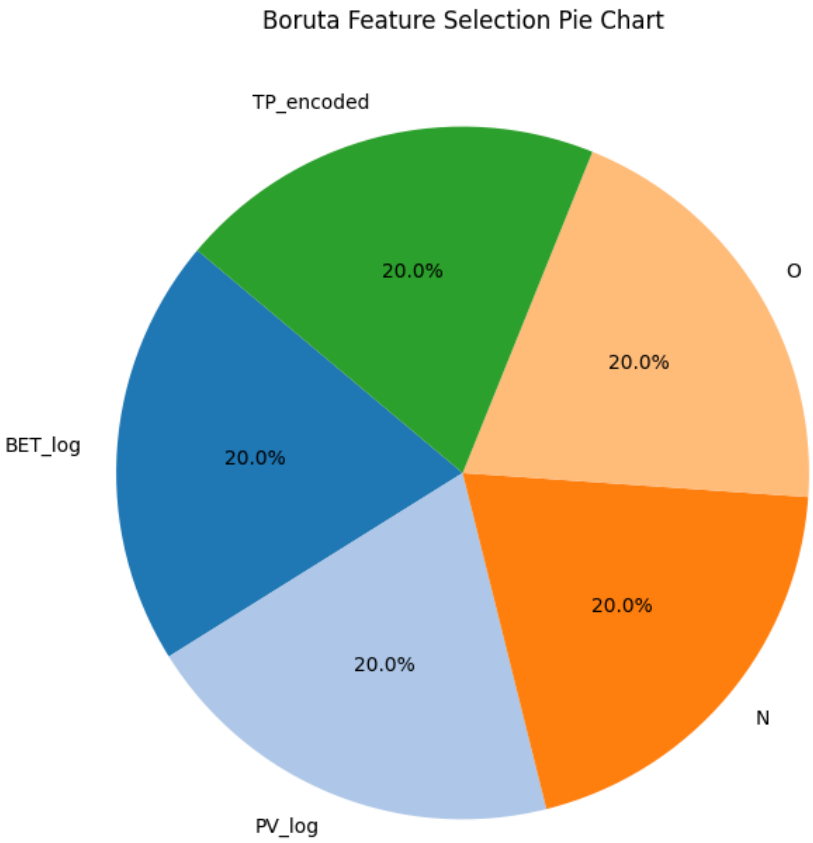

Figure 8: Boruta Pie chart

From the feature importance analysis using Boruta, the features PV log, BET log, TP encoded, N, and O were identified as the most influential. These features were selected for building machine learning models, as they contribute significantly to predictive performance.

5 Methodology

5.1 Techniques

This project aims to predict the adsorption capacity (mg/g) of biochar, making it as a regression problem. For this task, supervised machine learning models were used since the target variable is continuous. Before sending the data into the models, categorical columns such as Biomass and Type of Pollutant were transformed into numerical representations using label encoding. Instead of relying on the traditional train-test split method, k-fold cross-validation was used due to the limited number of observations. Cross-validation provides a more reliable estimation of model performance by addressing issues like overfitting and underfitting while utilizing all data subsets for both training and testing.

Several ensemble-based regression algorithms, including Decision Trees, Random Forest Regressor, Gradient Boosting, and XGBoost, were utilized for prediction. These algorithms were chosen for their proven ability to handle complex datasets and capture non-linear relationships in the data. Ensemble methods, such as Bagging (Bootstrap Aggregating) and Boosting, were particularly advantageous as they improve prediction accuracy by combining multiple weak learners into a strong learner. Ensemble Methods like **Bagging (Bootstrap Aggregating)** involves sampling the training data with replacement to create multiple bootstrap samples, each as large as the original dataset, where approximately 63% of the original data is included in each sample, with some instances repeated and others omitted. A separate regressor is trained on each bootstrap sample, and the predictions are aggregated, typically by averaging, to improve accuracy. This method reduces the variance of the base regressors and is particularly effective when the base regressor is unstable, mitigating errors caused by random fluctuations in the training data. In contrast, **Boosting** sequentially builds models by focusing on errors made by previous models to iteratively improve performance. By assigning higher weights to data points not predicted in earlier iterations, boosting ensures that subsequent models focus more on difficult cases. Techniques such as Gradient Boosting and XGBoost are highly effective in capturing complex patterns and enhancing overall prediction accuracy.

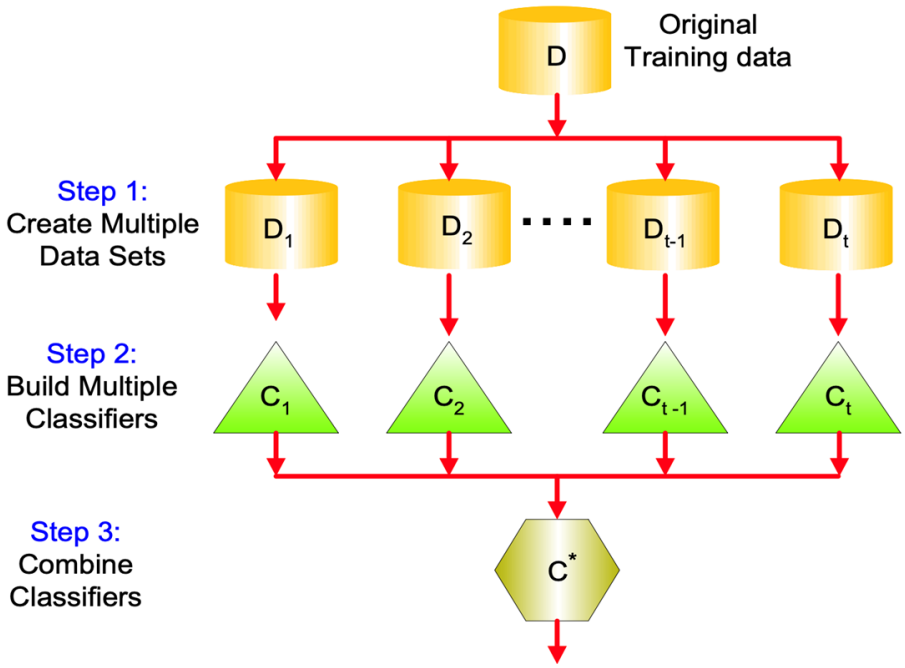

Figure 9: General Approach on how Ensemble Works

Hyperparameter tuning was performed to optimize the model's performance by fine-tuning key hyperparameters, including n estimators (the number of trees in the ensemble), learning rate (the step size for updating predictions), max depth (the maximum depth of individual decision trees), min samples split (the minimum number of samples required to split an internal node), and min samples leaf (the minimum number of samples required at a leaf node). This process enhances the model's accuracy and generalization ability. GridSearchCV was employed for hyperparameter optimization, systematically searching through a predefined grid of hyperparameter values to identify the best combination. By performing an exhaustive search over the specified parameter space with cross-validation, GridSearchCV ensured that the selected hyperparameters provided optimal performance on the validation data, leading to improved predictive accuracy.The k-fold cross-validation technique was utilized to evaluate the models, ensuring a robust assessment of their performance. This method involves dividing the data into k subsets (folds) and iteratively training the model on k-1 folds while testing it on the remaining fold. This process is repeated for all folds, and the average performance metric is computed. By allowing all data points to be used for both training and validation, k-fold cross-validation provides a comprehensive evaluation of model performance. Through the integration of ensemble methods, hyperparameter tuning, and cross-validation, the models were effectively optimized to predict the adsorption capacity of biochar with high accuracy and generalization ability.

5.2 Procedure

This project aimed to predict the adsorption capacity of biochar using supervised machine learning techniques, leveraging skills gained from core courses in Data Science and Analytics.

The dataset, provided by the professor, was in Excel format and contained a limited number of observations. Due to the small dataset size, there was a risk of the model underfitting. While adding synthetic data could address this, we decided against it, as it might introduce noise and lead to overfitting. Instead, we selected models capable of handling complex data and non-linear relationships. Outliers were ruled out using Grubbs' test with a 95% confidence interval. Data distribution was examined through histograms, revealing that some columns were right-skewed. Log transformations were applied to normalize these columns. Feature selection was performed using Boruta with a Random Forest model, identifying columns such as PV log, BET log, N, O, TP as highly important. These selected columns were then used as inputs for the model. Since decision trees are condition-based and do not depend on feature scaling, standardization was deemed unnecessary.

Supervised learning algorithms, including Decision Trees, Random Forest Regressor, and XGBoost, were implemented using Scikit-learn. Scikit-learn uses the CART algorithm for decision trees, which builds binary trees by selecting the best split at each node. For classification tasks, it uses measures like Gini impurity to evaluate the quality of a split, while for regression tasks, it minimizes the Mean Squared Error (MSE) at each split. The Gini index assesses the quality of a split and can handle both numerical and categorical features during the tree-building process. For regression, the tree minimizes the MSE at each split, which is calculated as: The tree grows by selecting the feature that results in the greatest reduction in impurity. This process continues until a stopping condition is met, such as reaching the maximum depth or the minimum number of samples per leaf.

The Random Forest Regressor is an ensemble learning technique that utilizes bagging (Bootstrap Aggregating) and feature sampling. It trains multiple decision trees on different random subsets of the training data and combines their outputs to provide more accurate and robust predictions. This method is particularly effective for non-linear regression problems, as it models complex relationships by averaging the predictions of numerous decision trees. This approach reduces variance, prevents overfitting, and ensures that at each node, only a random subset of features is considered for splitting. The final prediction is calculated as the average of the predictions made by all the trees in the ensemble.

Gradient Boosting builds an ensemble of weak learners, usually decision trees, in a sequential manner. It works by fitting each new tree to the residual errors (the difference between the predicted and actual values) of the previous trees, thereby "boosting" the model's performance in areas where it previously

made errors. The goal is to minimize the loss function by iteratively adding models that correct the mistakes of prior models, with each tree trained to predict the residuals of the previous ensemble. Gradient Boosting is highly effective for both regression and classification tasks, especially when the data involves complex, non-linear relationships, as it can capture subtle patterns through successive corrections. The final prediction is obtained by summing the outputs of all the individual models.

XGBoost (Extreme Gradient Boosting) is an optimized implementation of gradient boosting that enhances the performance and efficiency of traditional gradient boosting models. It works by building an ensemble of decision trees sequentially, where each new tree corrects the errors made by the previous trees. XGBoost incorporates advanced features such as regularization (L1 and L2) to prevent overfitting, and it uses a more efficient tree-building algorithm that speeds up training and reduces computation time. Additionally, XGBoost handles missing data automatically and supports parallel processing. The final prediction is made by summing the predictions from all individual trees in the ensemble, with the model's performance optimized through techniques like shrinkage (learning rate).

Cross-validation was applied during training to prevent overfitting and underfitting. The project involved iterative experimentation, including hyperparameter tuning using GridSearchCV, to optimize model performance. Evaluation metrics such as RMSE and MAPE were used to assess the model's accuracy.

For model management and tracking, MLflow was employed to monitor the training process and deploy the best-performing model as an inference endpoint. This endpoint was used for making predictions. After identifying the most important features influencing biochar adsorption, the final model was deployed on a website using the Streamlit framework. The website allows users to input features and receive predictions for biochar adsorption capacity.

Throughout the project, skills from courses such as Machine Learning and Intelligent Data Analytics were applied, including data preprocessing, exploration, feature engineering, and model evaluation. The methods followed a systematic approach, combining data preparation, model training, and evaluation with cutting-edge tools and techniques. The results and findings of this project will be detailed in another section of the report.

6 Results and Analysis

The table 1 presents the hyperparameters and their respective options for four machine learning models: Decision Tree, Random Forest, XgBoost, and Gradient Boosting. For each model, different hyperparameters such as "Max depth," "Min sample split," "Min sample leaf," "Max features," and "Criterion" for the Decision Tree model, or "n estimators," "max depth," "learning rate," and "min samples leaf" for other models, are listed with their possible values or ranges. The "Selection" column provides the chosen hyperparameter values for each model based on tuning. These hyperparameters significantly impact the model's performance.

Model	Hyperparameter	Option	Selection
Decision Tree	Max depth	None, 6, 8, 10	8
	Min sample split	2, 4, 6	6
	Min sample leaf	1, 2, 4	2
	Max features	auto, sqrt, log2, None	None
	Criterion	squared error, absolute error	absolute error
RandomForest	n estimators	15, 25, 50, 100, 150	25
	max depth	None, 6, 8	None
	min samples split	2, 4	2
	min samples leaf	1, 2, 4	2
XgBoost	n estimators	15, 25, 50, 100	15
	max depth	3, 6, 8	3
	learning rate	0.01, 0.05, 0.1	0.05
	min child weight	1, 3, 5	3
Gradient Boosting	n estimators	50, 100, 150	50
	learning rate	0.01, 0.1, 0.2	0.01
	max depth	3, 5, 7	3
	min samples split	2, 5	2
	min samples leaf	1, 2	2

Table 1: Hyperparameter Tunning

Hyperparameters for each algorithm were fine-tuned using the grid search technique to optimize model performance. After training, model predictions were evaluated through cross-validation to ensure that performance metrics, such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), accurately reflected the models' ability to generalize unseen data. RMSE (Root Mean Squared Error) is a commonly used metric to measure the average magnitude of errors between predicted and actual values. It is calculated by taking the square root of the average of the squared differences between predictions and actual values, with lower values indicating better predictive

accuracy.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE is sensitive to large errors, which makes it particularly useful when large deviations in predictions are undesirable. MAPE (Mean Absolute Percentage Error), on the other hand, measures the average percentage difference between predicted and actual values, providing a relative measure of accuracy.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

It is especially useful for understanding how well a model performs across different scales and is easy to interpret. Both RMSE and MAPE are crucial for evaluating model performance as they provide insights into the magnitude and relative error of predictions.

6.1 Model Results

Table 2: Untuned Results

	Scoring	**random forest**	**decision tree**	**Xgboost**	**Gradient Boost**
0	RMSE	56.26	67.72	55.02	65.98
1	MAPE	5.49	5.56	3.69	5.69

Table 3: Tuned Results

	Scoring	**random forest**	**decision tree**	**Xgboost**	**Gradient Boost**
0	RMSE	53.80	65.75	55.17	62.52
1	MAPE	4.81	5.24	6.17	6.53

Hyperparameter tuning significantly improves the model's performance by finding the best combination of hyperparameters. In the case of the **untuned models**, such as the ones shown in the table 2, the performance metrics (RMSE and MAPE) are relatively higher, indicating that the models are not fully optimized and are possibly underperforming. For example, the Random Forest model has an RMSE of 56.27 and MAPE of 5.49 before tuning, whereas after **tuning**, the RMSE drops to 53.81 and MAPE to 4.81. This reduction in error metrics demonstrates the impact of hyperparameter optimization. Similarly, for other models, such as XgBoost and Gradient Boosting, tuning leads to lower RMSE and MAPE values, showcasing the improvement in model accuracy and prediction reliability. The tuning process adjusts parameters like the number of estimators, tree depth, and learning rate, ensuring that the model generalizes better on unseen data. Therefore, tuning not only enhances predictive accuracy but also contributes to the model's robustness and efficiency, as reflected in the significant improvements in the tuned results compared to the untuned ones.

6.2 K fold Results

Table 4: K Fold Results

	Scoring	random forest	decision tree	Xgboost	Gradient Boost
0	RMSE	48.786340	52.314689	47.778119	53.265480
1	MAPE	4.525112	4.142943	5.229774	6.433223

The results of k-fold cross-validation provide valuable insights into the model's performance across multiple folds of the data, helping to ensure that the evaluation metrics are robust and not overly dependent on a single train-test split. In this analysis, XGBoost was chosen as the best model, primarily based on its RMSE value of 47.78, which is the lowest among all the models. The decision to prioritize RMSE over MAPE as the metric for selecting the best model stems from the specific nature of the dataset and the problem context. RMSE gives more weight to larger errors, which is particularly important when the dataset contains a wide range of target values, as it penalizes significant deviations more heavily. This makes RMSE a more appropriate metric when accuracy in predicting absolute values is critical. While MAPE is a useful metric for understanding the percentage deviation of predictions from true values, it can be misleading when the target values include zeros or are close to zero, as it results in undefined or disproportionately large values. The data itself plays a role in the model's RMSE and MAPE performance; the relatively low RMSE of XGBoost reflects the model's ability to learn effectively from the dataset.

7 Deliverables

The feature importance analysis from the XGBoost model highlights the relative contribution of each feature to predicting biochar's effectiveness in removing pharmaceutical pollutants. The most influential feature is PV log (0.426), indicating that the log transformation effectively captures its relationship with adsorption capacity. Biomass encoded (0.242) ranks second, reflecting biomass's significant role in pollutant removal. O (0.144), representing oxygen, is also important, suggesting its involvement in the adsorption process. BET log (0.119), the log-transformed BET surface area, is moderately influential, consistent with its known impact on adsorption. N (0.070), representing nitrogen content, has lower importance but remains relevant. In contrast, TP encoded (0.0) contributes no predictive value, indicating its limited role in the process or an ineffective encoding method.

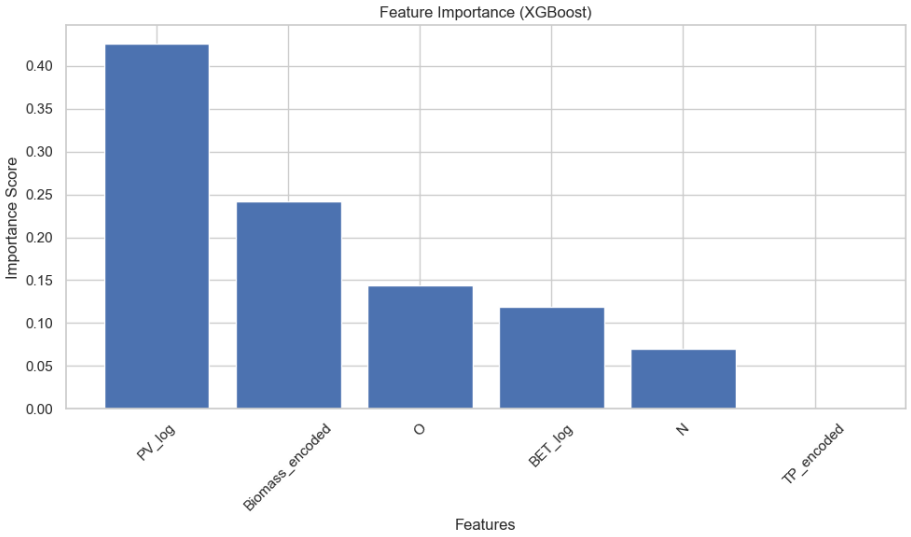

Figure 10: Feature Importance

Figure 10 answers one of the objectives of this project. The feature importance analysis highlights the relative contribution of each feature to predicting biochar's effectiveness in removing pharmaceutical pollutants. The most influential feature is PV log (0.426), indicating that the log transformation effectively captures its relationship with adsorption capacity. Biomass encoded (0.242) ranks second, reflecting biomass's significant role in pollutant removal. O (0.144), likely representing oxygen, is also important, suggesting its involvement in the adsorption process. BET log (0.119), the log-transformed BET surface area, is moderately influential, consistent with its known impact on adsorption. N (0.070), representing nitrogen content, has lower importance but remains relevant. In contrast, TP encoded (0.0) contributes no predictive value, indicating its limited role in the process or an ineffective encoding method. The outcomes of this project significantly impact both research and business objectives by providing actionable insights into optimizing biochar for environmental applications. From a research perspective, the findings highlight key features such as surface area, porosity, and biomass composition, which influence biochar's effectiveness in adsorbing pharmaceutical pollutants. These insights can guide future studies in material engineering and environmental science to develop more efficient biochars.

7.1 Deployment

The Streamlit framework is utilized for deploying an interactive web application that allows users to input parameters and receive real-time predictions on biochar's effectiveness in removing pharmaceutical pollutants. Streamlit simplifies the deployment of machine learning models by converting Python scripts

into user-friendly web apps with minimal configuration. The interface enables users to seamlessly input data and obtain predictions from the model, with the framework managing the backend processes. The application, accessible at **https://biocharapp.streamlit.app/**, supports future research and production strategies by providing insights to optimize biochar characteristics for enhanced performance in diverse environmental applications.

8 References

Jagadeesh, N., & Sundaram, B. (2022). Adsorption of pollutants from wastewater by biochar: A review. Research Scholar, Dept. of Civil Engineering, National Institute of Technology Andhra Pradesh. Received 19 May 2022, Revised 30 November 2022, Accepted 22 December 2022, Available online 23 December 2022, Version of Record 29 December 2022.

Dashti, A., Raji, M., Harami, H. R., Zhou, J. L., & Asghari, M. (Year). Biochar performance evaluation for heavy metals removal from industrial wastewater based on machine learning: Application for environmental protection. Separation and Purification Technology.

Jagadeesh, N., & Sundaram, B. (Year). Adsorption of pollutants from wastewater by biochar: A review. *Journal of Hazardous Materials Advances.* National Institute of Technology Andhra Pradesh, India.

CommentsViewingCalisto, V., Ferreira, C.I.A., Santos, S.M., Gil, M.V., Otero, M., Esteves, V.I. 2014. Production of adsorbents by pyrolysis of paper mill sludge and application on the removal of citalopram from water. Bioresource Technology, 166, 335-344.

9 Self Assessment

Our learning objectives are:

- Understanding the dataset
- importance of data exploring and analysis
- Data cleaning using different techniques
- understanding how a model works
- what model suits for our data

The successful completion of this project required the application of various Data Science and Analysis (DSA) skills. The project began with identifying a real-world problem—predicting the adsorption capacity of biochar—to address challenges in wastewater treatment and pharmaceutical pollutant removal. This

required transforming the problem into a predictive modeling task, which involved understanding the dataset thoroughly through techniques such as data cleaning, preprocessing, and analysis. Feature selection was employed to identify the most impactful variables, enhancing the predictive accuracy of the model.

Furthermore, data visualization and exploratory data analysis were crucial in uncovering patterns and relationships within the data, while model development and evaluation required knowledge of machine learning algorithms. Throughout the project, advanced skills in Python, feature engineering, and model tuning were essential to improve performance.

This practicum, supervised by Dr. Jude Okolie and worth 4 credit hours, played a key role in developing these essential skills science skills and providing hands-on experience in solving a practical, impactful problem.