Dhaka Hourly Air Quality Prediction

MD. TARIQUL ISLAM
Dept. of CSE
Brac University
ID: 20301044
md.tariqul.islam3@g.bracu.ac.bd

MD. YASIN ARAFAT TAMIM
Dept. of CSE
Brac University
ID: 20301029
yasin.arafat.tamim@g.bracu.ac.bd

Protiva Das
Dept. of CSE
Brac University
ID: 18101382
protiva.das@g.bracu.ac.bd

Aseer Iqtider Chowdhury
Dept. of CSE
Brac University
ID: 18301182
aseer.iqtider.chowdhury@g.bracu.ac.bd

***Abstract*—Air quality monitoring is a critical concern in densely populated urban areas like Dhaka, Bangladesh, where air pollution poses significant health and environmental challenges. This study examines an air quality dataset through data preparation and a variety of machine learning models for prediction. Preprocessing operations are carried out on the dataset, including managing missing values, encoding category characteristics, and standardizing numerical features. We train and test three distinct regression models: Decision Tree, K-Nearest Neighbors (KNN), and Linear Regression. Performance of the models is contrasted between before and after hyperparameter adjustment. Insights regarding feature scaling, model training, testing, comparison, and selection are provided in the study.**

***Index Terms*—Artificial Intellegence, Air Quality, KNN, Decision Tree, Linear Regression, Accuracy.**

I. INTRODUCTION

In the pursuit of advancing our understanding and prediction capabilities in the realm of air quality, this paper employs cutting-edge data science and machine learning techniques. We first make sure the dataset is ready for use by preprocessing, which includes managing missing values, encoding features, and normalization, using a dataset that includes hourly data, air quality indices, and categorical AQI classifications. We concentrate on three regression models: Linear Regression, K-Nearest Neighbors (KNN), and Decision Tree. These models go through extensive testing and training, are assessed using crucial metrics like MAE, MSE, and RMSE, and are cross-validated for stability. For the Decision Tree model in particular, hyperparameter adjustment is done to improve model performance. Visual comparisons help to highlight the effects of these tuning efforts and point us in the direction of the most precise model for estimating air pollutant concentrations, which is essential for environmental monitoring and the protection of public health.

II. DATASET DESCRIPTION

The dataset employed in our research, "Dhaka, Bangladesh Hourly Air Quality (2016-2022)," serves as a pivotal source of empirical data and can be referenced through the Kaggle platform (accessible via the following link: Dhaka Hourly Air Quality Dataset)It is a comprehensive collection of records relating to air quality monitoring operations carried out in Dhaka, Bangladesh, across a wide temporal range from 2016 to 2022. Over 55,000 data instances spread across eight distinct columns make up the dataset, which is a sizable collection[15].

We have to classify the current research problem as a regression problem due to its nature. The main justification for this classification comes from our attempt to forecast the Air Quality Index (AQI), a continuous variable by nature, at various temporal intersections. We find eight prominent features in the dataset, each with its own set of attributes. They are "Date (LT)," "Hour," "NowCast Conversion," "Raw Conversion," "Conc. Unit," "AQI," "AQI Category," and "QC Name." Notably, the remaining attributes are mainly categorical or textual in nature, whereas "NowCast Conc.," "Raw Conc.," and "AQI" are quantitative features[16].

A correlation matrix was created to capture the links between every dataset component in order to further thorough data analysis. This measure is a crucial tool for highlighting how different traits interact with one another and offers insightful information about how features are dependent on one another. Additionally, evaluating the class distribution inside the 'AQI Category' feature was given particular focus in the effort to determine the dataset's equilibrium. The goal of this investigation was to find any hidden class disparities and preserve the accuracy of the data.

III. Data Pre-processing

In the early stages of our study, careful data preprocessing was necessary to remove any inherent irregularities from the dataset and make it suitable for in-depth analysis and machine learning modeling. The management of missing data entries was the main issue among the persistent problems. Resolution was attained by selectively removing rows with null or missing values. At the same time, instances of data that had unnecessary information, such as punctuation, emoticons, or inconsistent, incomplete information, were carefully removed. These steps added up to a dataset that was enhanced and contained over 53,000 high-quality data points[13].

Additionally, to ensure that the model was ready, categorical feature encoding was carefully carried out. It involved transforming text-based categorical features, specifically ”Conc. Unit” and ”AQI Category,” into a numerical format that could be ingested by machine learning models[14]. Each preprocessing difficulty was methodically handled, starting with the problem’s identification and ending with the deployment of specific solutions. The resulting clean dataset—now free of noise and aberrations—emerged as the best foundation for the subsequent construction of machine learning models.

IV. Dataset Splitting

The dataset underwent the essential partitioning step that facilitated the upcoming stages of model creation, training, and assessment. The revised dataset was divided into a training set, which contained 80 percent of the data, and a testing set, which kept the remaining 20 percent as a result of this separation, also known as dataset splitting. To avoid any unintentional bias in the selection of data samples for training and testing, this split followed a random allocation technique[17]. This division supports the broader goal of objectively evaluating the models’ generalizability, providing a neutral assessment of predictive models for Dhaka’s air quality based on historical data sets.

V. Feature Scalling

Data preparation is necessary before model training to make sure the dataset is acceptable for machine learning methods. Handling missing values and encoding categorical characteristics are phases in the preprocessing process. The column mean is used to impute missing values, assuring the accuracy of the data. Label encoding converts categorical information into numerical values. Additionally, the StandardScaler is used to standardize numerical properties and scale them across the board. This stage stops the modeling process from being dominated by characteristics with bigger scales. To evaluate model performance, the preprocessed dataset is subsequently divided into training and testing sets.

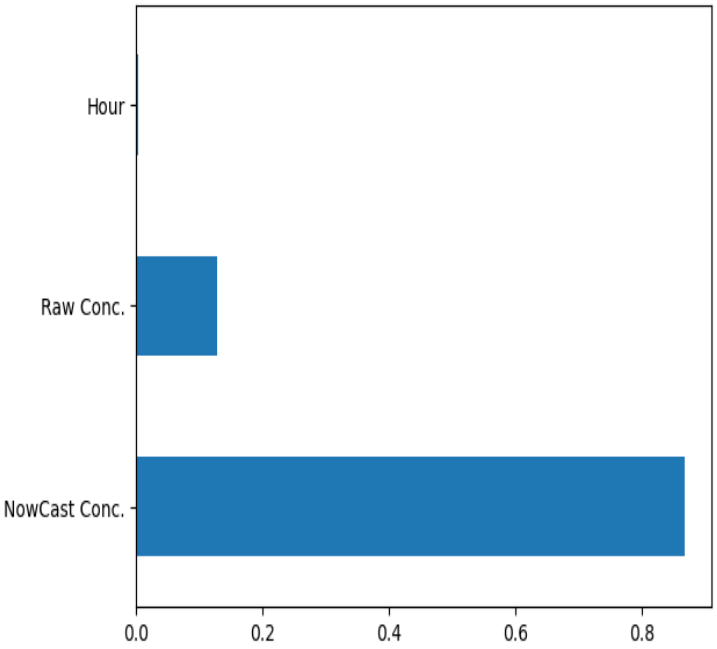

Figure 1.1: Feature importance.

Here in the graph we can see that in case of feature selection we got maximum value in NowCast Conc. That’s why we can decide that it will be our best choice in case of feature selection. Now a question can arise like what would happen if we chose another one. If we chose other features for these datasets then maybe it will not give us a better decision tree which will leave a legacy in our accuracy of any model.

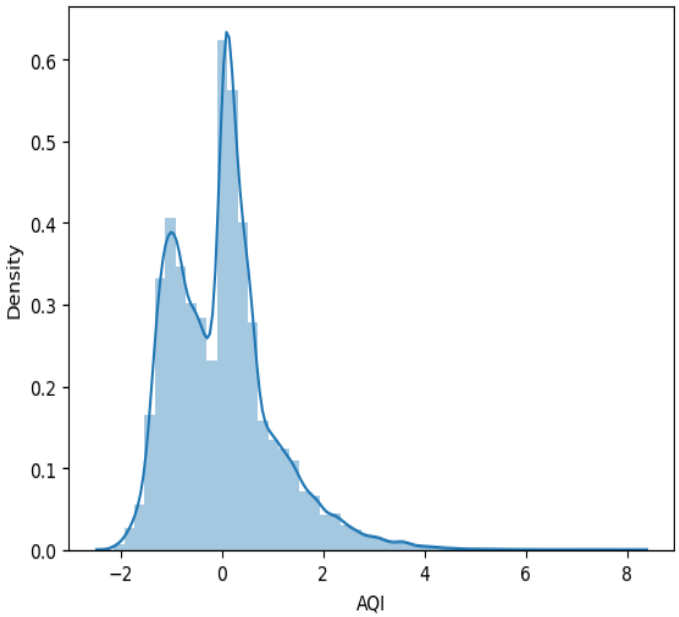

Figure 1.2: Density Vs AQI

In this graph we plot AQI vs Density of our datasets. From this graph we can see that our AQI values are not changing much. We can see the same type values are coming again and again for AQI in case of different density.

VI. Model Training and Testing

A. *Decision Tree*

A supervised learning approach known as a decision tree model may be applied to both classification and regression issues. It is a tree-structured model that bases judgments on incoming data on a flowchart-like structure. The model is divided into branches, and each branch stands for a potential course of action. The decisions' results are represented by the leaf nodes of the tree.

According to Saini et. al [1] the process of creating a decision tree model involves repeatedly dividing the data into smaller and smaller sections. Each subset is split again and again until it contains only data points from the same class, or until it is homogenous. A decision rule, which is a mathematical formula used to choose which feature to split on, directs the splitting procedure. The selection of the decision rule aims to increase the homogeneity of the subgroups.

Although there are several other decision rules that may be applied, some of the more popular ones are as follows:

(i) Gini Impurity: Gini impurity measures how evenly distributed the classes are inside a subset. The subgroup is more homogenous the smaller the Gini impurity.

(ii) Entropy: Entropy is a measurement of a data point's degree of uncertainty regarding its class within a subset. The class of a data point is more certain the lower the entropy.

(iii)Information Gain(IG): This is the amount of knowledge acquired through segmenting a subset based on a certain characteristic. The benefit of splitting the subgroup on that attribute increases with the information gain[4].

(a) Usage of decision tree models

According to Song et. al [2] regression trees and classification trees are the two primary categories of decision tree models.To determine the class of a data point, classification trees are utilized. For instance, a classification tree might be used to foretell whether or not a buyer would purchase a product.A continuous number, like the cost of a house, may be predicted using regression trees.

(b) Advantages and disadvantages of decision tree models

Advantage

Decision tree models provide a number of benefits, such as:
(i) They are simple to interpret and comprehend.
(ii) Problems involving classification and regression may both be solved using them.
(iii) It is not too difficult to train them.
(iv) Missing data can be handled using them.

Disadvantage

Decision tree models, however, can have several drawbacks, such as:
(i) Overfitting might make them sensitive.
(ii) The cost of training them computationally can be high.
(iii) As a result, even little changes in the data may result in significant changes in the model[3].

(c) Applications of decision tree models

Applications for decision tree models are numerous and include:

(i)Fraud detection: To spot fraudulent transactions, decision tree models can be utilized. At present different financial sectors like- Bank, offices use decision tree to detect fraud credit card for their security.
(ii)Disease diagnosis: Diseases may be identified using decision tree models. It is one of common sectors nowadays, where decision trees are mainly used.
(iii)Risk evaluation: Decision tree models may be used to gauge the likelihood of a certain occurrence, such a loan default.
(iv)Customer segmentation: Based on their features, decision tree models may be used to divide consumers into several categories.
(v)Product recommendation: The material that is presented to visitors on a website can be personalized using decision tree models.

Here we can see the AQI vs Density for Decision tree model. In the decision tree model our maximum AQI values are around 0. And our Density values are from 0 to 120. If we look at the accuracy of the Decision tree model then we can see that In case of Model training we got better accuracy but when we test the model with different values then it shows us some overfitting problem and also it taking some noise which are not our predictable values. That's why our accuracy is falling slowly but the amount of accuracy falls is not high.

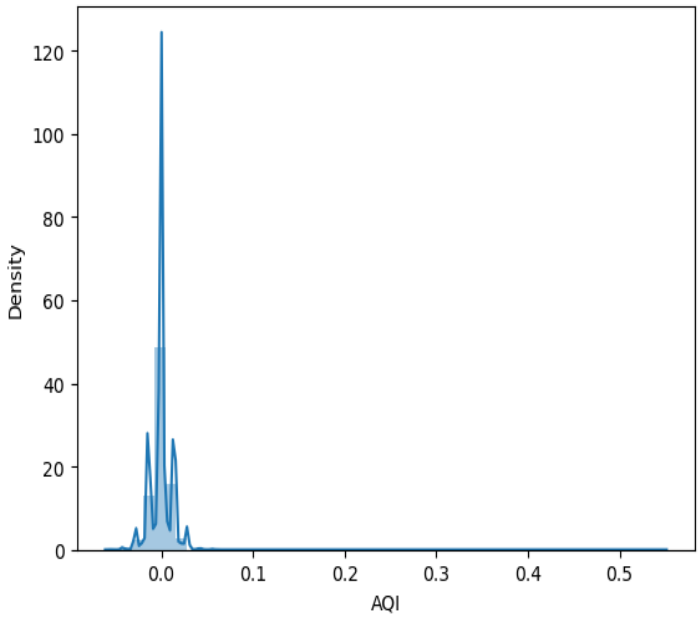

Figure 2.1: Density Vs AQI (Decision Tree)

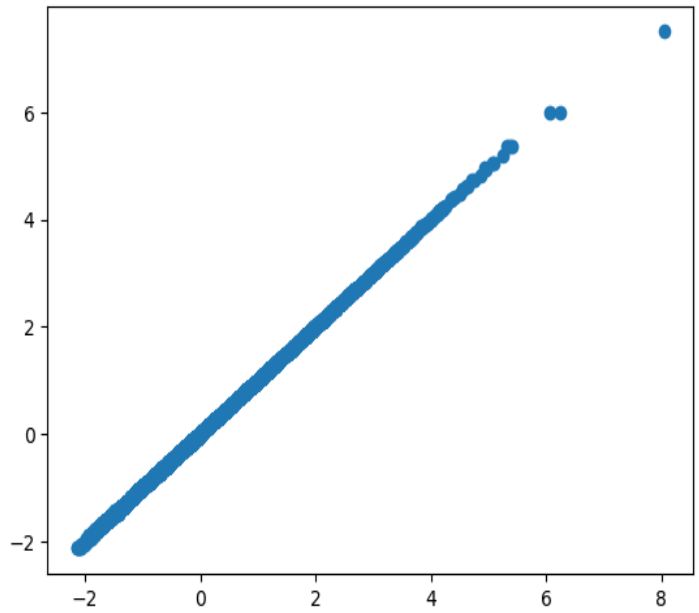

Figure 2.2: Decision Tree matplot

Firstly we got 99 percent accuracy in model training time but in testing model we see some accuracy drop. And we use hyperparameter tuning in this model. After hyper parameter tuning we see some overfitting problems in this model which give us some error and for this our accuracy becomes lower. But now the drop of accuracy is lower but if we will give more new datasets for testing then maybe it will be a bigger value of accuracy drop.

B. KNN (K-Nearest Neighbors)

The K-Nearest Neighbors (KNN) technique is a supervised machine learning algorithm that is used for classification and regression applications. KNN classifies or assigns values to data points based on the majority class or the average of the k-nearest neighbors data points in the feature space. It's a basic yet successful method that doesn't require a training phase; instead, it predicts using the full dataset. The number of neighbors chosen by KNN and the distance metric employed are important characteristics that determine its performance. In our paper, we have used KNN as our second model where we get an accuracy of 99 percent for the detection of air quality[5].

(a) Utilizing KNN

(i)Data Preparation: The first stage is gathering and putting together the training data, which consists of examples that have been labeled with input features and target labels.

(ii)Selecting a K Value: The number of neighbors to take into account while making predictions is indicated by the parameter "k" in the k-NN model. This value is crucial and may be decided using a variety of techniques, including cross-validation.

(iii)Distance measurement: A distance metric is used to gauge how similar two data points are in order to discover the k nearest neighbors for a new data point. Euclidean distance, Manhattan distance, and cosine similarity are examples of common distance measures.

(iv)Finding Neighbors: The algorithm determines the distance between each new data point and each training data point for a given new data point. Then, the k data points with the shortest distances are chosen.

According to Harrison [6], in classification tasks, the method counts the occurrences of each class among the k neighbors and designates the class that appears most frequently as the predicted class for the new data point. In regression tasks, the system averages the predicted classes for all the data points. In regression problems, the algorithm predicts by averaging the target values of the k neighbors.

(b) Benefits of KNN

1. Simple and simple to comprehend.
2. makes weak assumptions about the distribution of the underlying data.
3. may be applied to jobs requiring both classification and regression.
4. without having to retrain the model, adapt effectively to fresh training data[7].

(c) K-NN has certain limitations

1. sensitive to noisy data and irrelevant characteristics.
2. It involves computing distances for each data point, which may be computationally costly for huge datasets.
3.The findings can be considerably impacted by the choice of k.
4. Does not perform well in high-dimensional feature spaces[8].

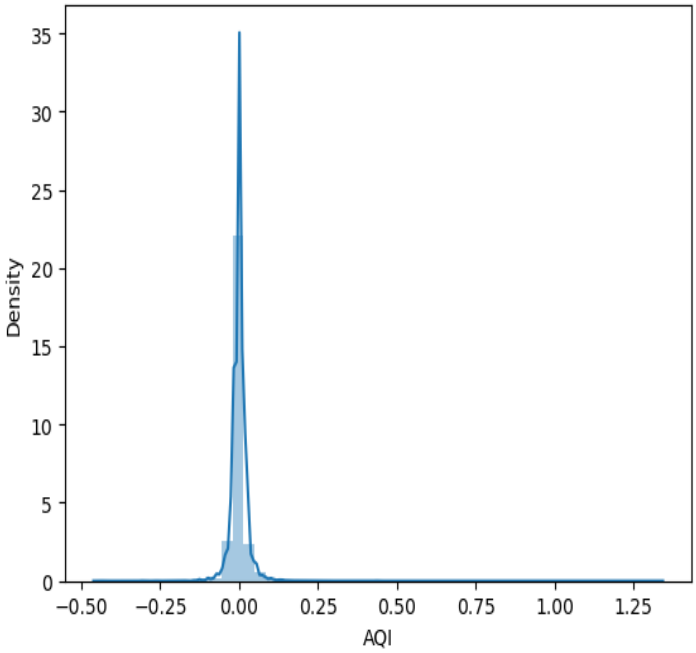

Figure 3.1: Density Vs AQI (KNN).

Here we can see the AQI vs Density for KNN model.In the decision tree model our maximum AQI values are around 0. And our Density values are from 0 to 120.

If we look at accuracy of KNN model then we can see that In case of Model training we got better accuracy but when we test the model with different values then it shows us some overfitting problem and also it taking some noise which are not our predictable values. That's why our accuracy is falling slowly but the amount of accuracy falls is not high.

Firstly we got 99 percent accuracy in model training time but in testing model we see some accuracy drop. And we use hyperparameter tuning in this model. After hyper parameter tuning we see some overfitting problems in this model which give us some error and for this our accuracy becomes lower. But now the drop of accuracy is lower but if we will give more new datasets for testing then maybe it will be a bigger value of accuracy drop.

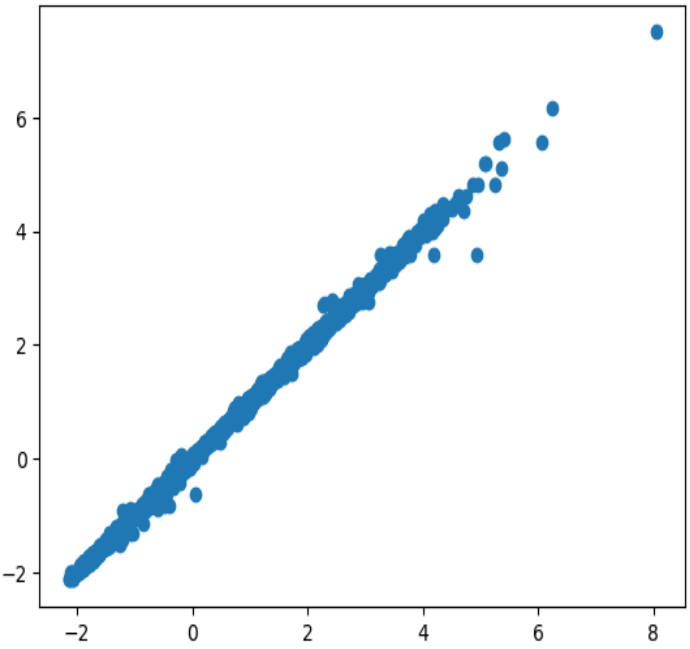

Figure 3.2: KNN matplot.

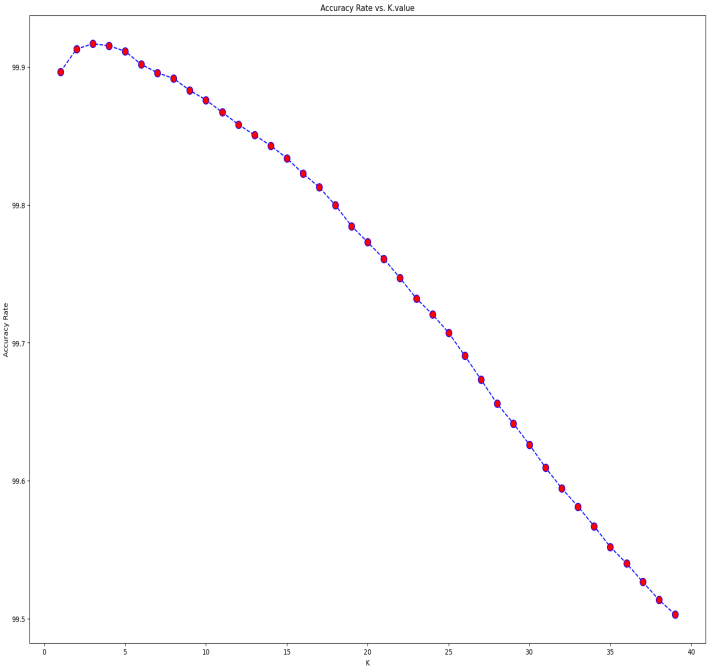

Figure 3.3: KNN Accuracy Vs k$_v akue$.

C. *Linear Regression*

By fitting a linear equation to the observed data, linear regression is a sort of regression analysis that simulates the connection between a dependent variable (the target) and one or more independent variables (features). The objective is to identify the line that minimizes the sum of squared discrepancies between the expected and actual values (a hyperplane in higher dimensions)[9].

(a) Working of Linear Regression

(i)Data Preparation: Collect and preprocess the training data, which consists of input features and corresponding target values.

(ii)Hypothesis Function: Linear Regression assumes a linear relationship between the input features and the target variable. The hypothesis function is defined as:

h(x) = + x + x + ... + x

Here, h(x) is the predicted target value, is the intercept, to are the coefficients for each feature, and x to x are the input feature values.

(iii)Cost Function (Mean Squared Error): The goal is to find the optimal values for the coefficients to that minimize the difference between predicted and actual values. This is achieved by minimizing the Mean Squared Error (MSE), given by:

$$MSE = (1/n) * (y - h(x))^2$$

where n is the number of data points, y is the actual target value, and h(x) is the predicted value.

(iv)Gradient Descent: To minimize the cost function, gradient descent is used. It iteratively updates the coefficients by taking steps in the direction of the steepest decrease in the cost function.

(v)Model Evaluation: After training, the model's performance is evaluated on new, unseen data using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared[11].

(b) Benefits of Linear Regression

Here are the few benefits of linear regression:

1. Interpretability and simplicity.
2. Both basic and complicated connections are suitable.
3. Reveals information on the importance of each feature's effect on the desired variable.
4. Unlike some more complex machine learning models, linear regression models are transparent and easy to explain to stakeholders, making it a preferred choice in many business and research settings.
5. Despite its simplicity, linear regression can provide reasonably accurate predictions when the underlying relationship between variables is approximately linear[10].

(c) Drawbacks of Linear Regression

Regardless of its numerious benefits, linear regression have some drawbacks. Few of the drawbacks are mentioned below:

1. A linear connection is assumed, which might not be true for all datasets.
2. Attentive to extremes.
3. May have trouble managing multicollinearity (strong feature correlation).

Density Vs AQI of Linear Regression

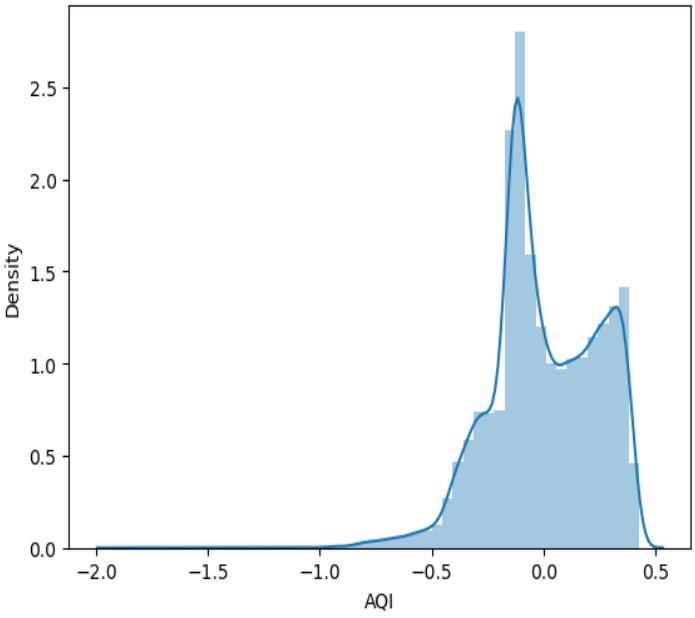

Figure 4.1: Density Vs AQI (Linear Regression).

Here we can see the AQI vs Density for Linear regression model. In the Linear regression model our maximum AQI values are around 0. And our Density values are from 0 to 2.5.

(d) Matplot of Linear Regression

If we look at the accuracy of the Linear regression model then we can see that in training it gives 94 percent accuracy and testing we got the same type of accuracy. In the Linear regression model we didn't use hyper parameter tuning. The process of identifying the ideal collection of hyperparameters for a machine learning model to obtain the greatest performance on a given dataset is known as hyperparameter tuning. It is true that linear regression models frequently have fewer hyperparameters to modify than more

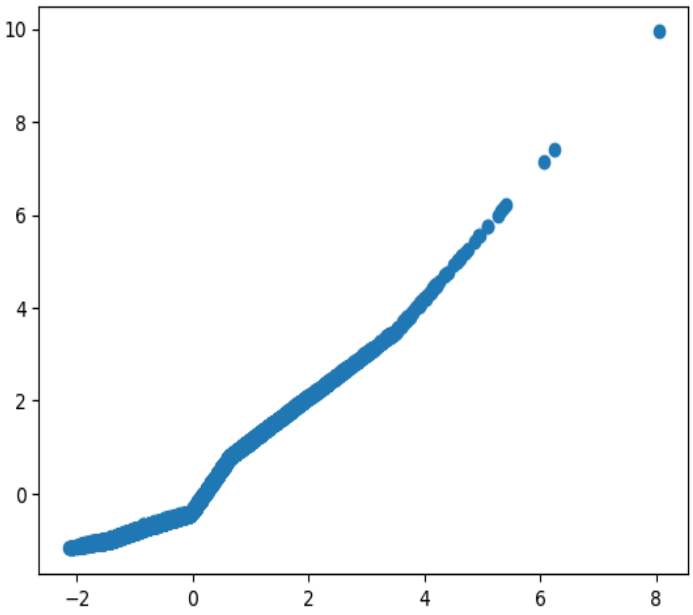

Figure 4.2: MatPlot(Linear Regression).

complicated models like decision trees, random forests, support vector machines, and neural networks. The following explains why hyperparameter tweaking for linear regression models is less important.

(e) Why Linear Regression Is a Better Model?

Linear regression is a straightforward approach that uses a linear equation to represent the connection between the input data and the target variable. The algorithm seeks for the line that fits the data the best and minimizes residuals, or the discrepancies between projected and actual values. In comparison to more complicated models, linear regression contains fewer hyperparameters because of its simplicity[12].

(i) Lack of Complex Structures: Complex structures like decision trees or neural networks, which have several hyperparameters controlling their depth, width, activation functions, etc., are absent from linear regression models. To prevent either overfitting or underfitting, these complicated structures frequently require adjustment. Given that it just fits a straight line through the data points, linear regression is less likely to overfit.

(ii) Regularization as Main Hyperparameter: The main hyperparameter to adjust is the regularization strength (alpha/lambda) if you're using regularized linear regression variations like Ridge (L2 regularization) or Lasso (L1 regularization). By penalizing high coefficients, regularization aids in maintaining model complexity and minimizes overfitting. Even in this case, there are still just a few hyperparameters.

(iii) No non-linearity or interaction: A linear connection between the input characteristics and the target variable is assumed in linear regression. Complex non-linear correlations or interactions between features are not captured. As a result, there is no need to adjust hyperparameters that deal with interactions or non-linearities.

(iv) Feature Scaling and normalizing: Although feature scaling and normalizing are preprocessing processes rather than hyperparameter adjustment, they can improve the performance of linear regression. While it can aid convergence during optimization, scaling features is not as crucial as it is in some other methods.

(v) Stability and Intrepretability: The stability and interpretability of linear regression models are excellent. Particularly when contrasted to more complicated models where modest parameter changes might result in large changes in predictions, small changes in hyperparameters could not have a substantial effect on their behavior.

Although basic linear regression models may not require as much hyperparameter tweaking, it is still a good idea to test your hypothesis and experiment with alternative settings. Additionally, to achieve the ideal balance between bias and variance when employing more sophisticated regularized linear regression variations, tweaking the regularization intensity might still be crucial. Never forget that the nature of the problem at hand, the complexity of the model, and the precise goals of your study will determine if hyperparameter adjustment is necessary.

VII. Model Selection and Comparison Analysis

Hyperparameter tuning using GridSearchCV is employed to fine-tune the Decision Tree model, with the aim of improving its performance. This process identifies the best hyperparameters that result in the model's optimal performance. Here,the Decision Tree model is refined using GridSearchCV, which involves hyperparameter adjustment to improve overall performance. This strategy enables us to discover the best hyperparameters for achieving the best model performance. We compare the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) of the three models—Decision Tree, KNN, and Linear Regression—before and after hyperparameter change. This comparison demonstrates how changing hyperparameters affects each model's prediction skills.

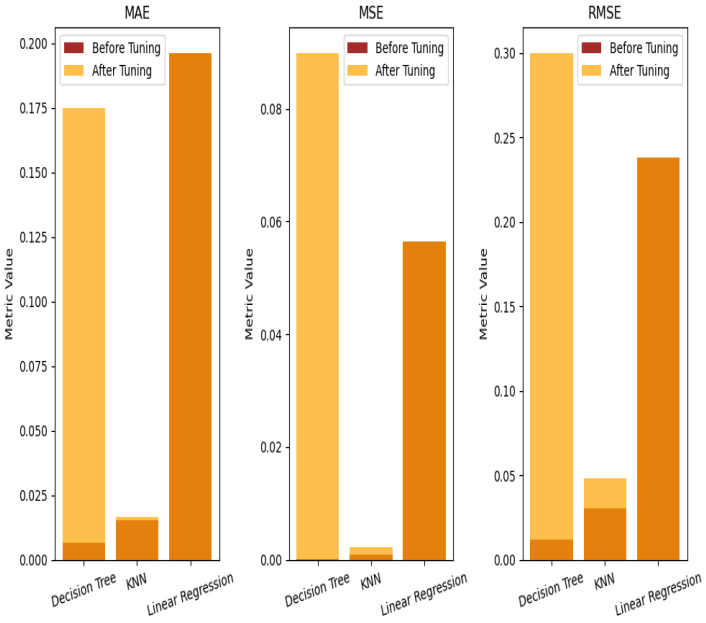

Figure 5: Model Comparision.

Comparision between our models

1. MAE: Mean Absolute Error

(i) All three models perform best (have the lowest MAE) before tweaking for the 'MAE' metric. KNN has the lowest MAE before tuning of the models.

(ii) The 'Decision Tree' and 'KNN' models exhibit a large rise in MAE upon tweaking, indicating a decline in performance. The MAE for "Linear Regression" likewise slightly rises after tweaking.

(iii) In terms of MAE, 'Decision Tree' and 'KNN' models are more negatively impacted by tuning than 'Linear Regression'.

2. MSE: Mean Squared Error

(i) In line with MAE, all three models have the lowest MSE prior to tuning, with 'KNN' exhibiting the lowest MSE among them.

(ii) Both the "Decision Tree" and the "KNN" models show an increase in MSE after tweaking, indicating decreased performance. A modest rise in MSE is also shown in "Linear Regression."

(iii) Once more, compared to 'Linear Regression', 'Decision Tree' and 'KNN' models exhibit a more notable increase in MSE following tweaking.

3. RMSE: Root Mean Squared Error

(i) The RMSE pattern exhibits the same general pattern as the other measures. Before tweaking, all models had the lowest RMSE values, with the 'KNN' model having the lowest RMSE.

(ii) Both the "Decision Tree" and the "KNN" models show greater RMSE values after tweaking, indicating a performance decline. After adjusting, "Linear Regression" likewise exhibits a modest rise in RMSE.

(iii) 'Decision Tree' and 'KNN' models are more adversely affected by tuning than 'Linear Regression', much like in the other metrics.

General Analysis

'KNN' consistently outperforms the other two models across all three measures (MAE, MSE, RMSE) before tweaking. This implies that the initial 'KNN' setup worked well for the dataset and situation at hand. 'Decision Tree' and 'KNN' models perform worse after tweaking, as seen by higher values for all measures. This can mean that the tuning settings used weren't the best ones for these models. After tuning, "Linear Regression" displays a modest rise in performance measures, indicating that tuning produced marginal gains.

Key Learnings

With various techniques and datasets, model tuning's efficacy varies. It's crucial to thoroughly examine how tuning effects model performance to make sure that outcomes are improved rather than worsened. The precise tweaks made during tuning, such as changing a parameter, can have a big impact on how a model behaves.

VIII. Conclusion

This investigation demonstrates the value of precise machine learning-based air quality prediction. We learn more about the viability of forecasting air pollution concentrations by preprocessing the dataset, training regression models, and assessing their performance. The Decision Tree model performs better after hyperparameter adjustment, highlighting the value of parameter optimization. A steady performance with ideal neighbor selection is also provided by the KNN model. As a benchmark, linear regression illustrates the room for additional model development. The intended trade-offs between prediction accuracy and model complexity determine the best model to use. This work emphasizes the importance of preprocessing, model choice, and hyperparameter tuning in creating accurate air quality prediction models for environmental monitoring and public health evaluation.

IX. REFERENCES

1.Saini, A. (2023). Decision Tree Algorithm – A Complete Guide. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

2.Song, Y. (2015). Decision tree methods: applications for classification and prediction. PubMed Central (PMC). https://doi.org/10.11919/j.issn.1002-0829.215044

3.Decision Trees. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/tree.html

4.Wikipedia contributors. (2023). Decision tree learning. Wikipedia. https://en.wikipedia.org/wiki/Decision$_{t}ree_{l}earning$

5.1.6. Nearest neighbors. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/neighbors.html

6.Harrison, O. (2019, July 14). Machine Learning Basics with the K-Nearest Neighbors Algorithm. Medium. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

7.Brownlee, J. (2020). Develop K-Nearest Neighbors in Python from scratch. MachineLearningMastery.com. https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/

8.GeeksforGeeks. (2023). K Nearest neighbor KNN algorithm. GeeksforGeeks. https://www.geeksforgeeks.org/k-nearest-neighbours/

9.1.1. Linear models. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/linear$_{m}odel.html$

10.Swaminathan, S. (2019, January 18). Linear regression — detailed view - towards data science. Medium. https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86

11.Brownlee, J. (2020b). Linear regression for machine learning. MachineLearningMastery.com. https://machinelearningmastery.com/linear-regression-for-machine-learning/

12.3.2. Tuning the hyper-parameters of an estimator. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/grid$_{s}earch.html$

13.Vivone, G., Dalla Mura, M., Garzelli, A., Pacifici, F. (2021). A benchmarking protocol for pansharpening: Dataset, preprocessing, and quality assessment. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14, 6102-6118.

14.Bilalli, B., Abelló, A., Aluja-Banet, T., Wrembel, R. (2018). Intelligent assistance for data pre-processing. Computer Standards Interfaces, 57, 101-109.

15.Chowdhury, A. S., Uddin, M. S., Tanjim, M. R., Noor, F., Rahman, R. M. (2020, August). Application of data mining techniques on air pollution of Dhaka city. In 2020 IEEE 10th International Conference on Intelligent Systems (IS) (pp. 562-567). IEEE.

16.Rahman, M. M., Mahamud, S., Thurston, G. D. (2019). Recent spatial gradients and time trends in Dhaka, Bangladesh, air pollution and their human health implications. Journal of the Air Waste Management Association, 69(4), 478-501.

17.Meng, Z., McCreadie, R., Macdonald, C., Ounis, I. (2020, September). Exploring data splitting strategies for the evaluation of recommendation models. In Proceedings of the 14th ACM Conference on Recommender Systems (pp. 681-686).