Historical Data Mining for Predicting Air Quality in Victoria, Canada

SENG 474 Data Mining Project Report

Alvin Guo - V00987315
Daming Wang - V00960801
Christopher Xu - V01007912

Table of Contents

1.0 Introduction

1.1 Background

Air quality has emerged as a crucial factor influencing the environment and the health and well-being of populations worldwide. Like many urban environments, Victoria, Canada, has been grappling with complex air quality challenges in recent years. These multifaceted challenges include climate change, industry emissions, vehicular pollutants, and rapidly evolving urban development patterns. All these elements interact in complex ways, altering the constituents of the air and, consequently, the quality.

Numerous scholarly investigations, such as those by Huang(2018) and Liu (2022), have demonstrated a potent correlation between suboptimal air quality and negative health effects such as respiratory complications, cardiovascular diseases, and premature mortality. Also, the deleterious impacts of pollution on the environment, including biodiversity loss, weather trends, and human-induced climatic changes, have been well-documented(Huang, 2018; Liu, 2022). Consequently, comprehending and forecasting air quality is paramount for planning more sustainable and healthier urban environments(Huang, 2018; Liu, 2022).

Given this backdrop, the present research explores a novel approach to predicting air quality - using data mining techniques on historical data.

1.2. Objective

The primary objective of this research is to develop a data-driven model that accurately predicts future air quality in Victoria, Canada, based on historical data. To meet this objective, the study is guided by the hypothesis that discernible patterns and trends in past air quality data, combined with relevant location and date factors, can provide reliable predictive capabilities.

This ultimate goal is to contribute to the scientific understanding of air quality dynamics and provide actionable insights to guide policy development, inform public health strategies, support environmental conservation efforts in Victoria, and potentially extendable to other regions.

2.0 Related Work

2.1 Prediction of PM2.5 Concentrations using Random Forest Models

A significant body of research has examined data mining models, such as the random forest model, for predicting PM2.5 concentrations. Huang(2018) conducted a comprehensive analysis of the prediction of high-resolution PM2.5 concentrations using the random forest model in the North China Plain. Traditional methods, including statistical models, time series analysis, and regression techniques, were also acknowledged; however, their limitations in capturing the complex spatial and temporal dynamics of PM2.5 concentrations led to the exploration of the random forest model. Huang underscored the random forest model’s capabilities in handling large-scale datasets, nonlinear relationships, and interactions among predictors.

Nevertheless, the study also emphasized the need for further research to address data sparsity and enhance model interpretability(Huang, 2018).

2.2 Extreme Gradient Boosting (XGBoost) Models in Estimating NO2 Concentrations

Recent advancements in remote sensing have facilitated the use of satellite data in environmental research. Liu(2022) employed Extreme Gradient Boosting (XGBoost), a popular machine learning and data mining algorithm. In conjunction with MODIS satellite retrievals, generate 250 m-resolution regional NO2 concentration products. The XGBoost model's ability to capture complex relationships and handle large datasets was emphasized, contributing to improved estimation accuracy.

Liu (2022) concludes the research by delineating the model's future applications in air quality monitoring, urban planning, and policy formulation. Furthermore, the study indicates the necessity for future investigations to incorporate supplemental spatial and temporal predictors and amplify the XGBoost model's interpretability.

2.3 Contribution of the Current Study

Building on these studies, our research aims to develop a data-driven model for accurately predicting future air quality in Victoria, Canada, leveraging historical data. When combined with location and date factors, we hypothesize that patterns and trends in past air quality data can offer reliable predictive capabilities.

Informed by an elegant yet practical vision, our model utilizes a quintet of strategies encompassing Linear Regression, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting techniques.

Predicting air quality indicators, such as PM2.5 and NO2, is an important area of focus in environmental research. Various techniques, both traditional and advanced, have been employed to model and forecast these atmospheric variables accurately. We explore pertinent studies on these predictive methodologies, focusing on data mining applications.

3.0 Data Preprocessing

3.1. Raw Database:

3.1.1 For Windows Operating System:

a. Open File Explorer.
b. Click on the folder icon located on the taskbar.
Alternatively, press the Windows key + E on your keyboard.
Enter the URL for FTP:
c. Locate the File Explorer address bar at the top of the window.
Type the desired URL into the address bar. For example,
"ftp://ftp.env.gov.bc.ca/pub/outgoing/AIR/AnnualSummary/" is the URL for our project.

3.1.2 For UNIX/LINUX Operating System:

a. Open the terminal.
b. Type the desired URL into the address bar. For example,
"ftp://ftp.env.gov.bc.ca/pub/outgoing/AIR/AnnualSummary/" is the URL for our project.

3.2. Initiate the Connection:

a. Press the Enter key or click the Go button next to the address bar.

b. File Explorer will attempt to connect with the provided FTP server.

3.3. Verify the Connection:

a. Upon successful connection, the contents of the FTP server will be displayed in the File Explorer window.

b. Users can now navigate the directories, open files, and perform various file operations.

3.4. Data Combine

The way we used to combine multiple CSV files containing PM2.5
monitoring data from 2011 to 2021 into a single file named "PM25_2010_to_2020.csv".:

1. Importing Required Libraries
- The code begins by importing the necessary libraries: `pandas` and `google.colab.drive'.
- `pandas` is used for data manipulation and analysis.
- `google.colab.drive` mounts Google Drive to access the data files.

2. Mounting Google Drive:
- The line `drive.mount('/content/drive')` mounts the Google Drive to the Colab notebook. This allows access to the files stored in Google Drive

3. Setting the Directory:
- The line `os.chdir("/content/drive/MyDrive/Seng474Project/PM25")` changes the current working directory to the specified path. Make sure to provide the correct path where PM2.5 data files are located.

4. Defining File Extension:
- The line `extension = 'csv'` sets the file extension variable to 'csv'. This specifies that we want to work with CSV files.

5. Getting File Names:
- The line `all_filenames = [i for i in glob.glob('*.{}'.format(extension))]` uses the `glob.glob` function to retrieve a list of all file names in the current directory that match the specified file extension ('csv' in this case). It searches for all CSV files in the directory and stores their names in the `all_filenames` list.

6. Combining CSV Files:

- The line `combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])` reads each CSV file from the `all_filenames` list using list comprehension and concatenates them into a single data frame using `pd.concat()`. Each CSV file is read as a separate DataFrame using `pd.read_csv()`, and all the DataFrames are combined into one using `pd.concat()`.

7. Saving Combined Data to CSV:
- The line `combined_csv.to_csv( "PM25_2010_to_2020.csv", index=False)` saves the combined DataFrame as a new CSV file named "PM25_2010_to_2020.csv" in the current directory. The `index=False` parameter specifies not to include the index column in the output file.

- Adjust the directory path and file extension based on a specific setup. Once executed, this code will combine all the PM2.5 monitoring data files from 2011 to 2021 into a CSV file named "PM25_2010_to_2020.csv" in the specified directory.

- The code demonstrates a data management process for PM2.5 values from 2010 to 2020, as well as for the year 2021. It utilizes Python's `os` and `gdown` libraries for file management and downloading from Google Drive.

3.5 Data Process Broken Down

The data process can be broken down into the following steps:

1. Checking File Existence:
The code begins by defining a function named `file_exists()` that checks if a file exists locally. This function is later used to determine whether the required data files are already present or need to be downloaded.

2. Retrieving PM2.5 Data from 2010 to 2020:
The code utilizes a Google Drive sharing link to access the PM2.5 data from 2010 to 2020. It extracts the file ID from the sharing link and defines the output file path for storing the downloaded data. It then checks if the local file exists using the `file_exists()` function. If the file is not found, it downloads the CSV file containing the PM2.5 data using the `gdown.download()` function. Otherwise, it simply prints a message indicating the use of the local file.

3. Retrieving PM2.5 Data for 2021:
Similarly, the code defines a Google Drive sharing link for the PM2.5 data in 2021 and extracts the file ID. It defines the output file path for storing the data and checks if the local file exists. If the file is not found, it downloads the CSV file using the `gdown.download()` function. Otherwise, it prints a message indicating the use of the local file.
This code segment focuses on retrieving and managing PM2.5 data for the specified years. It ensures that the necessary data files are available by downloading them from Google Drive or using local copies if they exist. This data preparation step is essential for subsequent analysis and modeling tasks, such as PM2.5 value prediction and forecasting.

4. Importing Libraries:
The code starts by importing necessary libraries for data analysis and modeling, including `pandas`, `sklearn` modules, `matplotlib.pyplot`, `numpy`, `statsmodels.tsa.arima.model`, and `datetime`.

5. Loading PM2.5 Training Data:
The code uses the `pd.read_csv()` function from the `pandas` library to read the PM2.5 training data from the file '/content/pm25_2010_to_2020.csv' into a pandas DataFrame named `PM25_train`. This dataset contains historical PM2.5 values from 2010 to 2020. The `head()` function is then called to display the first few rows of the dataset.

6. The `train_and_predict()` Function:
The `train_and_predict` function takes four parameters:
- `model`: The machine learning model to be trained and used for prediction.
- `X_train`: The input features of the training data.
- `y_train`: The training data's target variable (PM2.5 values).
- `X_test`: The input features of the test data.

Within the function:
- The `model.fit(X_train, y_train)` line trains the specified model using the training data, where `X_train` represents the input features and `y_train` represents the corresponding target variable.
- The trained model is then used to make predictions on the test data using the `model.predict(X_test)` line.
- The predicted values are returned as the output of the function.

This function encapsulates training a given model with the provided training data and uses it to predict the test data. It abstracts away the specific training and prediction logic, making it convenient to use different models interchangeably for prediction tasks.

7. `evaluate_model()` Function:
The `evaluate_model` function takes two parameters:
- `y_test`: The actual target variable values from the test data.
- `y_pred`: The predicted target variable values

Within the function:
- The `mean_squared_error(y_test, y_pred)` function from the `sklearn.metrics` module calculates the MSE between the actual and predicted values.
- The `mean_absolute_error(y_test, y_pred)` function from the same module calculates the MAE between the actual and predicted values.
- The calculated MSE and MAE values are then returned as a tuple `(mse, mae)`

8. Calculating Mean PM2.5 Value:
- `PM25_train['RAW_VALUE']` represents the 'RAW_VALUE' column in the `PM25_train` DataFrame, which contains the PM2.5 values.
- The `.mean()` function is applied to this column to calculate the average or mean value of the PM2.5 readings from the training data.

This calculation provides the mean PM2.5 value from the available training data. It can be used to gain insights into the average level of PM2.5 pollution during the specified period (2010 to 2020).

9. `preprocess_data()` Function:
The `preprocess_data` function takes a DataFrame (`df`) as a parameter and performs the following preprocessing steps:
- Converting Date and Time.
  - The 'DATE' column in the DataFrame is converted to a DateTime format using `pd.to_datetime(df['DATE'])`.

- The 'YEAR', 'MONTH', 'DAY', and 'HOUR' columns are created by extracting the corresponding values from the 'DATE' and 'TIME' columns using the `dt.year`, `dt.month`, `dt.day`, and `apply(lambda x: int(x.split(':')[0]))` operations, respectively.
- Dropping Columns:
  - The 'DATE', 'DATE_PST', 'TIME', and 'ROUNDED_VALUE' columns are dropped using `df.drop(['DATE', 'DATE_PST', 'TIME','ROUNDED_VALUE'], axis=1)`.

- Encoding Categorical Columns:
  - The 'REGION' and 'STATION_NAME' columns are encoded using label encoding. The `LabelEncoder()` object is created as `le`, and then `le.fit_transform()` is applied to these columns (`df['REGION']` and `df['STATION_NAME']`).
- Dropping Additional Columns:
  - Several additional columns such as 'STATION_NAME_FULL', 'NAPS_ID', 'UNIT', 'INSTRUMENT', 'OWNER', 'EMS_ID', and 'PARAMETER' are dropped using `df.drop(['STATION_NAME_FULL', 'NAPS_ID', 'UNIT', 'INSTRUMENT', 'OWNER', 'EMS_ID', 'PARAMETER'], axis=1)`.
- Handling Missing Values:
  - The missing values in the DataFrame are filled with the mean value of each column using `df.fillna(df.mean())`.
- Returning the Preprocessed DataFrame:
  - The preprocessed DataFrame is returned as the function's output.

This function prepares the data for further analysis and modeling by converting date and time columns, encoding categorical variables, dropping unnecessary columns, and filling missing values with the mean. These preprocessing steps help to ensure that the data is in a suitable format for modeling purposes.

10. Linear Regression:
- A `LinearRegression` model (`lr`) is instantiated.
- The `train_and_predict()` function is called with the Linear Regression model, training data (`X_train`, `y_train`), and test data (`X_test`) to make predictions (`y_pred_lr`).
- The MSE and MAE between the actual test labels (`y_test`) and the predicted labels (`y_pred_lr`) are calculated and stored in `mse_lr` and `mae_lr`, respectively

11. Logistic Regression:
- Binary classification is performed by converting the target variable `y` into a binary variable based on a threshold value (`threshold`). In this case, whether the 'RAW_VALUE' is above the mean value of the training data.
- The Logistic Regression model (`logr`) is instantiated.
- The Logistic Regression model is trained using the training data (`X_train_log`, `y_train_log`).
- The model makes predictions (`y_pred_logr`) on the test data (`X_test_log`).
- The MSE and MAE between the actual test labels (`y_test`) and the predicted labels (`y_pred_logr`) are calculated and stored in `mse_logr` and `mae_logr`, respectively.
- The accuracy of the Logistic Regression model is calculated using `accuracy_score` between the actual
binary labels (`y_test_log`) and the predicted binary labels (`y_pred_logr`).

12. Decision Tree:
- A `DecisionTreeRegressor` model (`dt`) is instantiated.
- The `train_and_predict()` function is called with the Decision Tree model, training data (`X_train`, `y_train`), and test data (`X_test`) to make predictions (`y_pred_dt`).

- The MSE and MAE between the actual test labels (`y_test`) and the predicted labels (`y_pred_dt`) are calculated and stored in `mse_dt` and `mae_dt`, respectively.

13. Random Forest:
- A `RandomForestRegressor` model (`rf`) is initiated with specified hyperparameters.
- The `train_and_predict()` function is called with the Random Forest model, training data (`X_train`, `y_train`), and test data (`X_test`) to make predictions (`y_pred_rf`).
- The MSE and MAE between the actual test labels (`y_test`) and the predicted labels (`y_pred_rf`) are calculated and stored in `mse_rf` and `mae_rf`, respectively.

14. Gradient Boosting:
- A `GradientBoostingRegressor` model (`gb`) is instantiated with specified hyperparameters.
- The `train_and_predict()` function is called with the Gradient Boosting model, training data (`X_train`, `y_train`), and test data (`X_test`) to make predictions (`y_pred_gb`).
- The MSE and MAE between the actual test labels (`y_test`) and the predicted labels (`y_pred_gb`) are calculated and stored in `mse_gb` and `mae_gb`, respectively.

15. Printing the Errors:
- Each model's calculated MSE and MAE values are printed to evaluate their performance.

4.0 Data Mining

4.1 Observation (PM 2.5)

After applying the model to the processed data frame, we can get information on the relationship between the PM 2.5 value and the feature attributes.

4.1.1 Linear Regression

As shown in Figure 1, the linear regression cannot show the actual outstanding raw values, but it can show the primary trend of our prediction. With the Linear Regression Coefficients of values: [0.00444238 0.14915865 0.19843008 0.17044219 -0.02064721 0.04612791], which stands for STATION_NAME, REGION, YEAR, MONTH, DAY, HOUR's slope, as we can see, the raw_value is going up with all positive slopes. The **YEAR** attribute affects the PM 2.5 value the most among all these features, with the most significant absolute value of ~0.2. This evidence shows that PM 25 increases as the year increases.

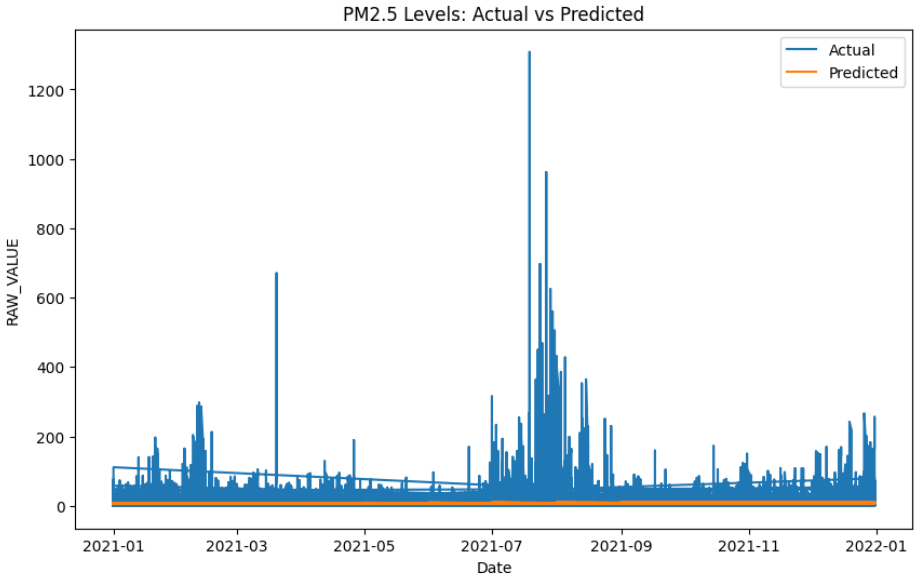

Figure 1: Actual and Predicted PM 2.5 Values for Linear Regression

4.1.2 Decision Tree

As shown in Figure 2, the decision tree shows more notable changes in PM 2.5 values than linear regression, which makes it a better way to predict future stats. However, the situation may vary greatly, as shown in this figure. For example, from July to September 2021, it shows extreme curves of high PM 2.5 pollution values, where the current model predicts that this will happen around September 2021. This may happen due to other factors like political changes and climate effects. Therefore, this would only be a reference where we only considered location and time factors.

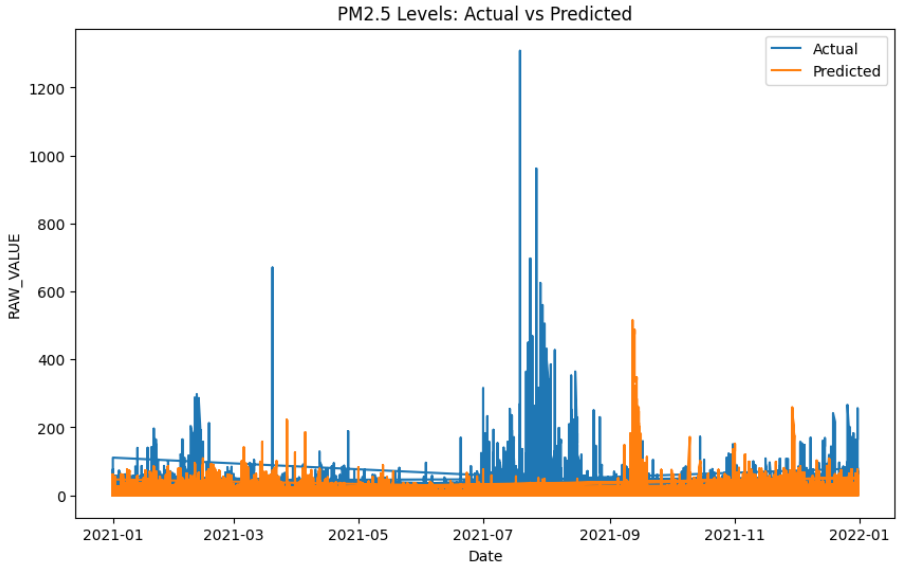

Figure 2. Actual and Predicted PM 2.5 Values for Decision Tree

Figure 3 is a part of our prediction, and it shows only for region 0 of our 2021 prediction. We can see that it shows the correct curves of changes based on date. Moreover, it can correctly predict the high raw value in September and November. Similarly, Figure 4 shows the prediction of Station 0. However, the

situation in station 0 is just like the overall prediction mentioned previously, and it does not predict the high raw value in February and August correctly. Instead, it says they will happen in April and September.

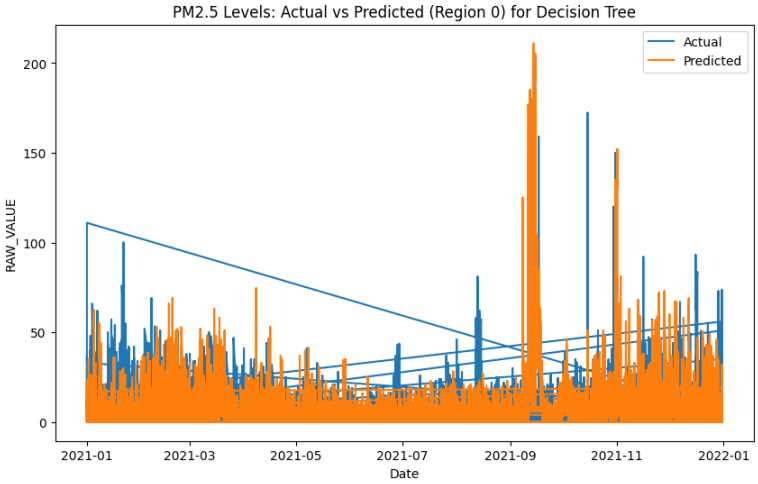

Figure 3. PM2.5 Levels: Actual vs Predicted (Region 0) for Decision Tree

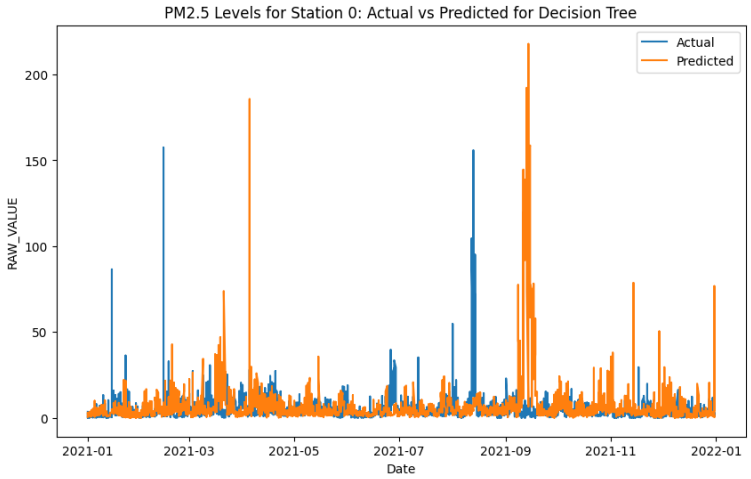

Figure 4. PM2.5 Levels: Actual vs Predicted (Station 0) for Decision Tree

4.1.3 Other Method

We also used Logistic Regression, Random Forest, Gradient Boosting, and Ensemble methods to make the prediction (Detailed figures can be found in the code section). We concluded that the Random Forest performs like the Decision tree, and the Gradient Boosting performs like Linear Regression. As for the Logistic Regression, it can only partly predict the correctness of raw_value over mean raw value or not with 69% of accuracy. We found that linear regression is the best way to show the trend with the lowest MSE, and the Decision Tree (also Random Forest) shows the best graphical trend but with a low MSE.

4.2 Observation (NO2)

4.2.1 Linear Regression

Using Linear Regression prediction, we can get an overall decreasing slope over time. With a list of coefficients: [0.03421592 -0.51486128 -0.05093445 -0.04161687 -0.00686398 0.06135829], we can

now that the region is the most significant feature attribute that affects the NO2 value with an absolute value of 0.5. This means NO2 is unrelated to the time factor but will change when it is in a different location.

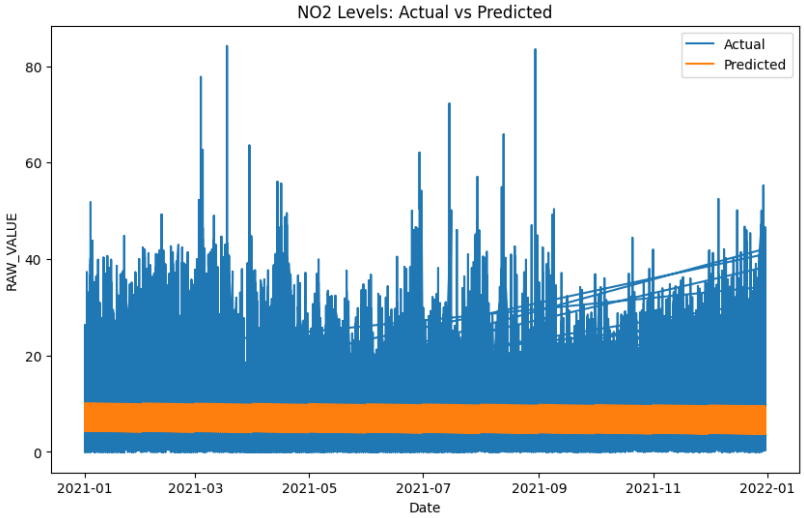

Figure 5. NO2 Levels: Actual vs Predicted using Linear Regression

4.2.2. Decision Tree

Since the raw_value range of NO2 is relatively small, with only 0 - 90, we can see that the decision tree works well with it graphically of a matching trend (Figure 6 and Figure 7). However, there are many errors on days. Some may predict several days before or after. This is out of our control since we only have a few factors, and the realistic situation constantly changes.

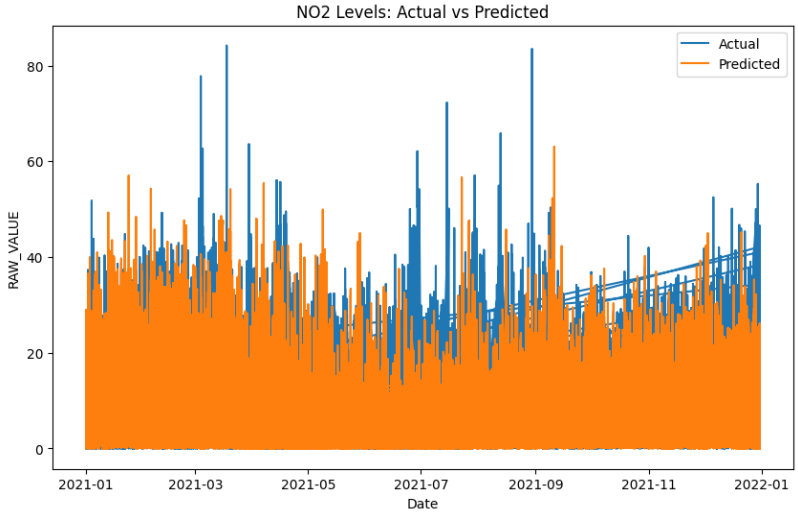

Figure 6. NO2 Levels: Actual vs Predicted using Decision Tree

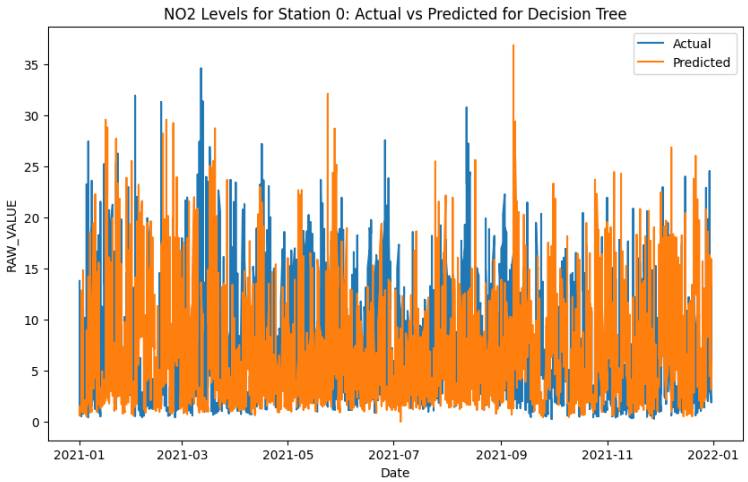

Figure 7. NO2 Levels for Station 0: Actual vs Predicted for Decision Tree

4.2.3 Other Method

When using Ensemble Prediction of combing linear regression, decision tree, and gradient boosting method, it shows a primary linear trend but with some notable changes (Figure 8). It has a relative MSE value with linear regression and gradient boosting. However, it has more changes on the graph, making it the best method to predict NO2, considering both digital and graphical errors. The other methods' figures can be found in the code sections.

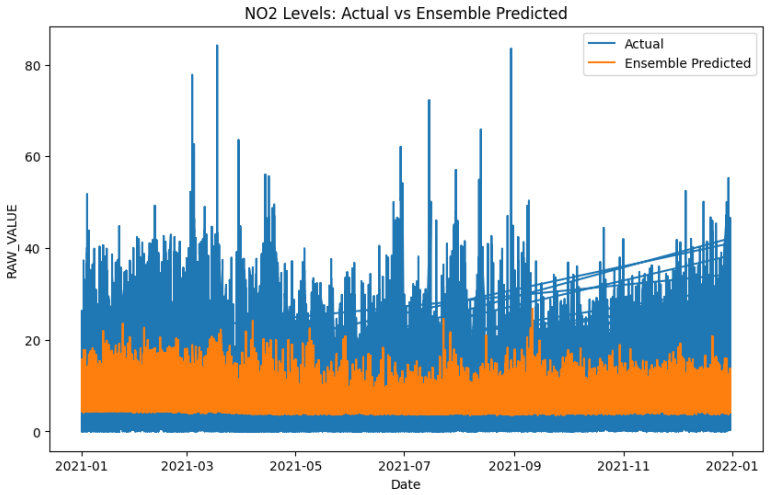

Figure 8. NO2 Levels: Actual vs Ensemble Predicted

4.3 Observation (O3)

4.3.1 Linear Regression

In Figure 9 shows a clear decreasing trend of the O3 raw value over time. The coefficient of the slope shows the most considerable effectiveness on month with a value of 0.72. It means that the pollution index will vary based on the month of each month.

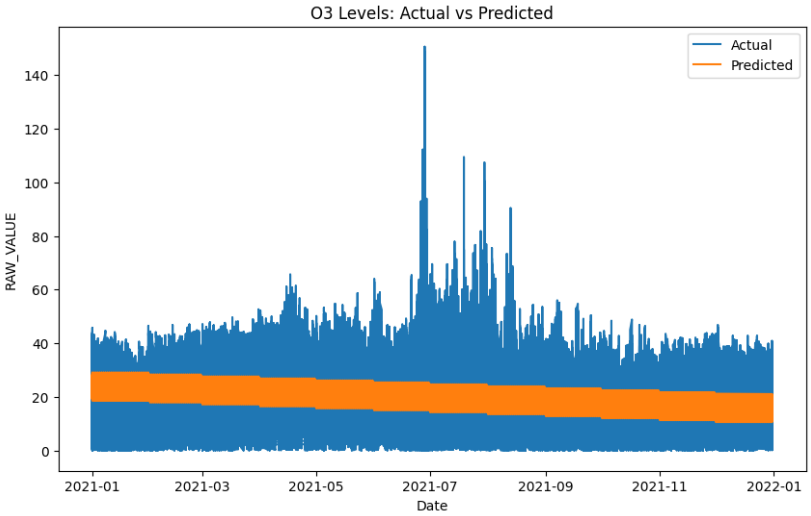

Figure 9. O3 Levels: Actual vs Predicted for Linear Regression

4.3.2 Other Method

The decision tree can predict the value correctly most of the time, as illustrated in Figure 10. However, it works poorly when it comes to the time around summer. Similarly, with other model predictions previously, they perform almost the same. Furthermore, the Ensemble method works best with the lowest MSE value again.

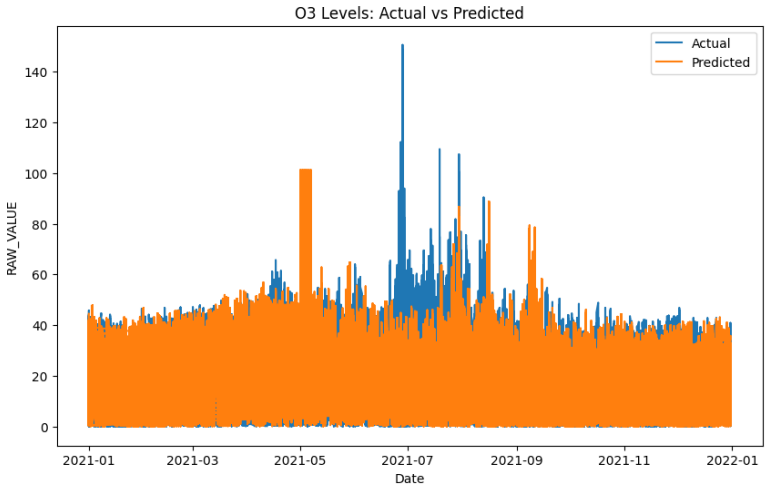

Figure 10. O3 Levels: Actual vs Predicted for Decision Tree

4.4 Other Observations

As shown in previous figures, we can notice that the pollution is significantly high during July and August of 2021, whereas in previous years, it happened in August and September. Besides, for PM 2.5, NO2, and O3, these three values are pretty related, it shows almost the same trending curves in 2021 and previous years. Upon research, Random forest and gradient boosting are based on decision trees and linear regression. However, they can perform better with these data frames, making them better options for predicting air pollution values, just like the related work mentioned using random forest and gradient boosting.

5.0 Evaluation

Model evaluation ensures that our predictive models are reliable and robust. We mainly use MSE and MAE values to evaluate the accuracy. Most of the evaluation figures and results are printed in the code sections. Below are some overall evaluations of those figures and calculated results.

For the PM 2.5 model, the best MSE is 188.7 of the Gradient Boosting method, which performs like Linear regression. Given that we only have time and location factors and the range of raw_value of PM 2.5 is quite large, it is an acceptable result. The year will affect the value the most. For NO2 and O3, their best models are the Ensemble method of combing linear regression, decision tree, and gradient boosting. The MSE values are 38.6 and 119, respectively. The location will affect NO2 the most, and the month factor will influence the O3 value the most. Moreover, the most significant air pollution often happens around summer for all three targets, where all three values are very high. Air pollution is also high in winter, but preceded only by summer.

Finally, our models show that linear predictions are mostly the best to predict all three targets of air pollution. However, decision tree-based predictions make more sense when it comes to graphical with much more changes up and down.

6.0 Implications and Future Work

6.1 Implications

Our study has many potential implications. Such predictive capabilities also hold potential in various sectors, ranging from public health, environmental conservation to urban planning. An accurate air quality prediction can guide public health initiatives by informing the timing and nature of interventions for preventing and managing health conditions exacerbated by poor air quality, such as asthma or cardiovascular diseases.

In environmental conservation, understanding future air quality trends can assist in formulating strategies to mitigate pollutants' negative effects on biodiversity and ecosystems. In urban planning, insights from accurate air quality predictions can contribute to designing cities that minimize air pollution, promoting healthier and more sustainable urban environments. For example, data on predicted air pollution levels can influence the location and design of public institutions like parks, hospitals, roads, and residential buildings to mitigate pollution exposure.

6.2 Future Work

Our future research should aim to enhance the model by incorporating a broader set of predictive variables. We have PM2.5, NO2, and O3 levels now. We will include more detailed industrial emission data, like sulfur dioxide(SO2), carbon monoxide(CO), and localized meteorological data, like atmospheric pressure, wind speed, and direction.

Moreover, advanced machine learning techniques such as deep learning and ensemble methods could be integrated into our model. These techniques are adept at handling high-dimensional, complex datasets and could enhance the model's ability to identify and learn intricate patterns within the air quality data.

7.0 Conclusion

Maintaining good air quality and its impact on public health and the environment must be balanced. Recognizing this, we embarked on a comprehensive research project to develop a predictive model for forecasting air quality in Victoria, British Columbia, explicitly emphasizing three pollutants - PM2.5, O3, and NO2. These pollutants, owing to their severe health and environmental implications, were the primary focus of our study.

After meticulous data gathering, preprocessing, and rigorous exploration of machine learning techniques, we anticipate our research will yield a model capable of providing accurate air quality forecasts. Our model, based on linear regression, logistic regression, decision tree, and gradient boosting, aims to predict future PM2.5, O3, and NO2 levels, utilizing historical and current data and considering many factors. We predict the successful detection of trends and relationships between these factors and the levels of pollutants in the atmosphere.

In conclusion, our evaluation of the predictive models for PM 2.5, NO2, and O3 air pollutants reveals several key findings. The PM 2.5 model performs well using the Gradient Boosting method, achieving a reasonably low MSE value of 188.7. Considering the limited input factors of time and location and the large range of PM 2.5 values, this result is acceptable. The most influential factor for PM 2.5 is found to be the year, indicating a temporal trend in air pollution levels.

For NO2 and O3, the best models are ensemble methods combining linear regression, decision trees, and gradient boosting. These models yield MSE values of 38.6 for NO2 and 119 for O3. The location factor significantly impacts NO2 levels, while the month factor influences O3 concentrations the most.

Notably, all three air pollutants exhibit higher pollution levels during the summer months, with exceptionally high values observed during this season. Winter also shows elevated pollution levels, though not as severe as during summer. This finding suggests a seasonal pattern in air pollution, with summer being the peak period.

Looking beyond the immediate application of our model, we see vast potential for future research. Whether it is extending our model to include additional variables or exploring its applicability to other geographical locations, our research paves the way for a deeper exploration into the field of air quality prediction.

In conclusion, our research is not merely an academic endeavor but has wide-ranging implications for the well-being of our society and the environment. The potential of our predictive model extends beyond its immediate application and lays the groundwork for future research, technological advancements, and proactive interventions to safeguard our environment and public health. This project thus stands as a testament to the significance of data mining and predictive modeling in addressing pressing environmental challenges. As we continue refining our model and uncovering deeper insights, we are confident that our work will significantly contribute to the air quality management and environmental conservation discourse in Victoria, British Columbia, and beyond.

8.0 Bibliography

Data Source:

https://catalogue.data.gov.bc.ca/dataset/air-quality-monitoring-verified-hourly-data

Our Open-Source Project Repository:
https://github.com/EdNovas/seng474-project

Huang, K. (2018). Predicting monthly high-resolution PM2.5 concentrations with a random forest model in the North China Plain. Science Direct. https://doi.org/10.1016/j.envpol.2018.07.016

Liu, J. (2022). Generating 250 m-resolution regional NO2 concentration products first from MODIS retrievals using extreme gradient boosting.Springer Link. https://doi.org/10.1007/s11869-022-01285-x