University for the Creative Arts

BERLIN SCHOOL OF BUSINESS & INNOVATION

Essay / Assignment Title: Machine Learning and Visualization

Programme title: Environmental Impact Analysis of Air Quality Data

Name: Jay Dilipbhai Moradiya

Year: 2024-25

CONTENTS

Statement of compliance with academic ethics and the avoidance of plagiarism

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the postgraduate program).

Name and Surname (Capital letters):

JAY DILIPBHAI MORADIYA

Date: 30/01/2025

INTRODUCTION

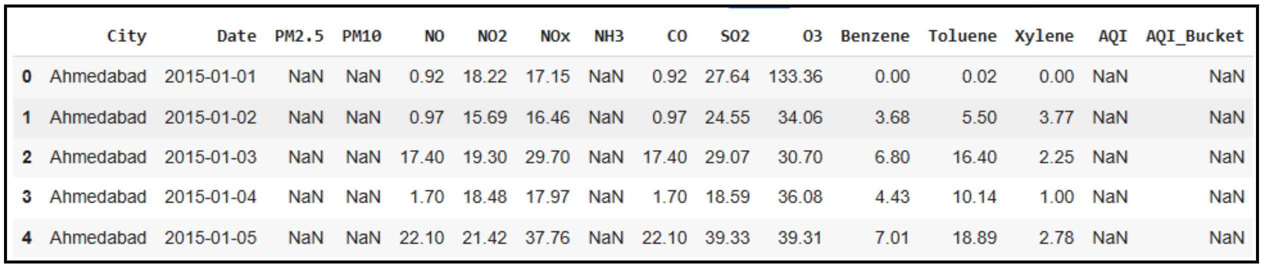
Air quality is the critical environmental factor that directly manages the impact of ecosystems, and climate patterns. This study focuses on evaluating air quality data to identify key patterns, and pollution hotspots, and predict future air quality levels based on historical data. Using machine learning approaches in combination with Tableau's provides advanced visualization tools which generate useful information about the scale of air pollution along with it is environmental outcomes. The analysis focuses on data exploration steps while detailing machine learning applications and assessing model performance results. The machine learning tools produce powerful capabilities to select the feature, pattern recognition, predictive modeling which enable more accurate forecasting of air quality trends. In order to, tableau provides a platform for visualizing complex data, ensuring findings are accessible to stakeholders.

CHAPTER ONE – PROBLEM AND DATA DESCRIPTION

Problem Description

This study focuses on analyzing air quality data through advanced analysis to determine pollution evolution patterns and dominant pollution sources and evaluate environmental conditions to produce meaningful guidance for pollution reduction strategies. Air quality deterioration is a pressing global issue, particularly in urban areas where industrial activities, vehicle emissions, and population density significantly contribute to pollution levels. "Air Quality Index" (AQI) that exceeded 400, far above the "good" range of 0-50 (Ouma *et al*., 2024). The pollution levels of PM2.5 and PM10 along with nitrogen dioxide (NO2) and sulfur dioxide (SO2) generate significant health problems which increased risk of respiratory disease while damaging hearts.

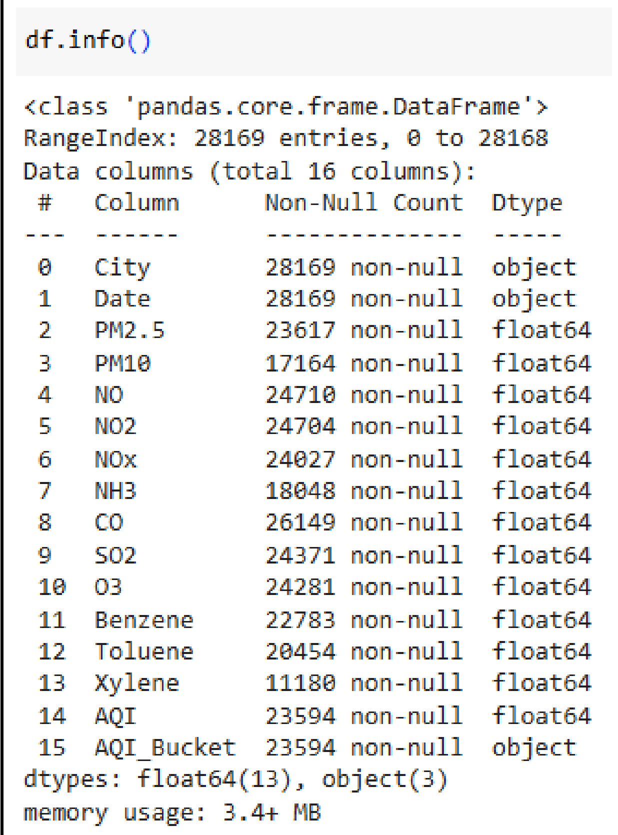
Data Description



	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN

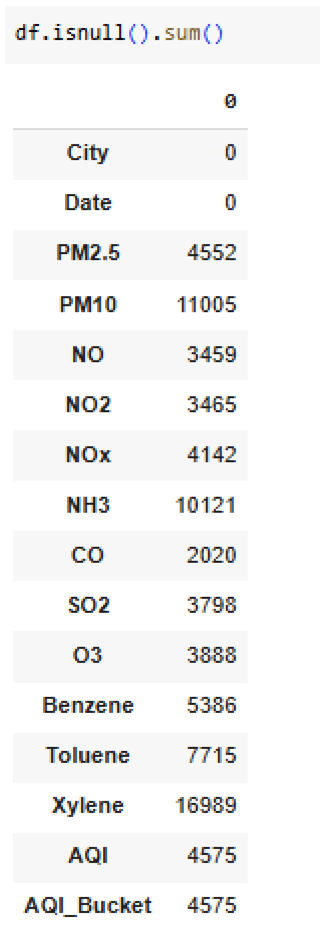
Figure 1: Data under observation

The dataset contains air quality data from Ahmedabad, revealing various measurements of *PM2.5, PM10, NO, NO2 , NOx , NH3 ,CO, SO2* and *O3*, *Benzene* , *Toluene* and *Xylene*. This dataset provides time series data from multiple dates starting January 1st 2015 which demonstrates how pollutants change throughout the period. The available data presents daily AQI measurements alongside respective ratings.



```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28169 entries, 0 to 28168
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   City        28169 non-null  object 
 1   Date        28169 non-null  object 
 2   PM2.5       23617 non-null  float64
 3   PM10        17164 non-null  float64
 4   NO          24710 non-null  float64
 5   NO2         24704 non-null  float64
 6   NOx         24027 non-null  float64
 7   NH3         18048 non-null  float64
 8   CO          26149 non-null  float64
 9   SO2         24371 non-null  float64
 10  O3          24281 non-null  float64
 11  Benzene     22783 non-null  float64
 12  Toluene     20454 non-null  float64
 13  Xylene      11180 non-null  float64
 14  AQI         23594 non-null  float64
 15  AQI_Bucket  23594 non-null  object 
dtypes: float64(13), object(3)
memory usage: 3.4+ MB
```



```
df.isnull().sum()
```

	0
City	0
Date	0
PM2.5	4552
PM10	11005
NO	3459
NO2	3465
NOx	4142
NH3	10121
CO	2020
SO2	3798
O3	3888
Benzene	5386
Toluene	7715
Xylene	16989
AQI	4575
AQI_Bucket	4575

Figure 2: Inspecting missing data and data types

The dataset contains 28,169 entries across 16 columns. The columns include city, date, various pollutant measurements (PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2), for benzene, toluene, xylene, and air quality metrics such as AQI and AQI Bucket. The *df.info()* shows float64 data type and three object columns. The *df.isnull().sum()* function in pandas is used to check for missing or null values in each column of a DataFrame *(df)*. The insights highlight key variables, requiring preprocessing.

CHAPTER TWO – METHODS

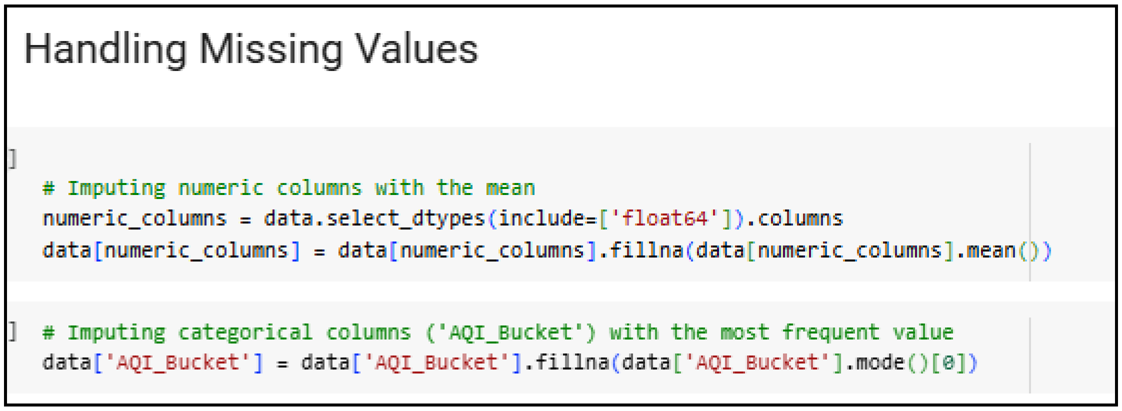
The methods applied during analysis and modelling has covered a range of tasks, from data preparation and exploration to model development and evaluation. The project follows specific procedures for environmental air quality data processing and analysis and modeling to assess its effects. The system involves multiple operations starting with data processing and visual exploration and continues through modeling and completed with model assessment and validation. The project uses pandas read_csv() to read CSV format data available as "*city_day.csv*". Using the *read_csv()* function in *pandas* the data transforms into a DataFrame structure that enables manipulation of the dataset. The following step involves analyzing the dataset after data loading to understand its content structure. A snapshot of dataset information becomes accessible through the head() function during the data review process. By using isnull().sum() users can evaluate missing value frequencies within the dataset which helps determine necessary cleaning steps for the data.

The project used the mean value from each numerical column to replace missing data through the pandas "fillna()" function. This approach which maintains data integrity enables the resolution of empty data fields. Replace missing values in categorical data the most commonly occurring mode value is used because it works well for categorical data types. In order to select imputation methods for data modeling it is essential to recognize they have significant effects on final performance. The analysis requires detecting and handling outliers because extreme values in the dataset need to be eliminated to avoid result distortion. The Interquartile Range (IQR) method serves as an identifier for numerical column outliers. By using the interquartile range method the calculator determines IQR values from 25th percentile to 75th percentile to identify values that extend past 1.5 times IQR limits. The analysis protects itself from outlier disturbances by replacing these values with the median of the column. Feature engineering is another crucial step in preparing the data for machine learning models (Liu *et al*., 2024). This involves creating new features that may improve the model's predictive power. Once the feature engineering is completed, the dataset is divided into the features (X) and target (y) variables. The features are the independent variables used to predict the target variable, which represents the air quality measurements or other key environmental metrics.

Tableau creates visualizations which simplify the interpretation of data and modeling outcomes. Tableau helps stakeholders to analyze data and detect patterns which enable an understanding of how environmental air quality has changed throughout time. Decision-making receives critical insights from line graphs and bar charts and heatmaps through visualization tools. The project implements data preprocessing along with advanced machine learning models along with interactive visualization to report environmental impacts in air quality assessment. The systematic workflow implements detailed actions to generate precise data sets and maintain stable predictive models while providing users with Tableau visual representations of results.

CHAPTER THREE – DATA PREPROCESSING AND EXPLORATION

Data Preprocessing Using Python



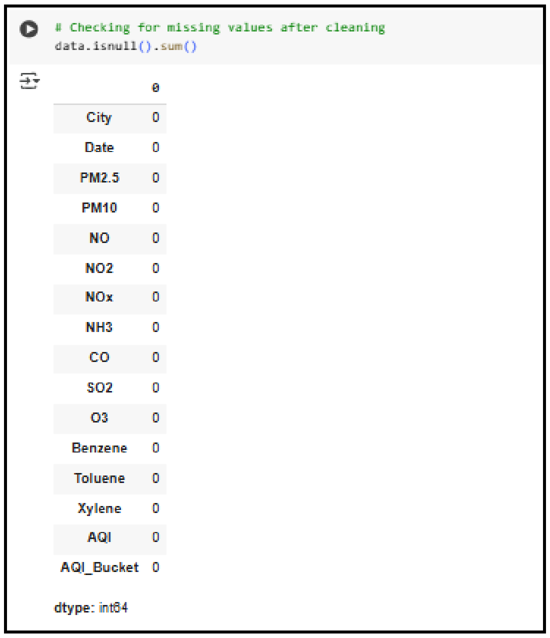
Handling Missing Values

```python
# Imputing numeric columns with the mean
numeric_columns = data.select_dtypes(include=['float64']).columns
data[numeric_columns] = data[numeric_columns].fillna(data[numeric_columns].mean())
```

```python
# Imputing categorical columns ('AQI_Bucket') with the most frequent value
data['AQI_Bucket'] = data['AQI_Bucket'].fillna(data['AQI_Bucket'].mode()[0])
```

Figure 3: Importing Numeric columns

The process of imputing missing values includes using means from each numeric column modulus replacing missing data. A combination of statistical analysis identified mode as the most frequent value to carry forward for AQI_Bucket missing values because this method creates a complete dataset for accurate analysis during model training.

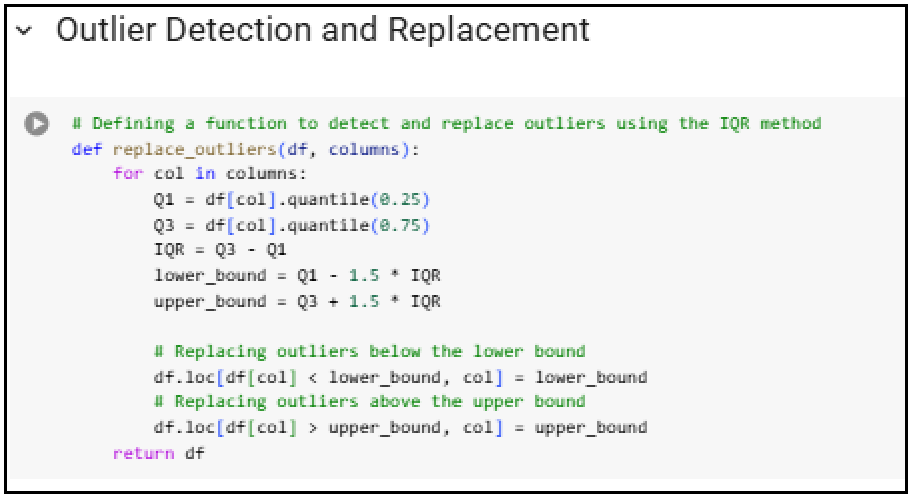


```python
# Checking for missing values after cleaning
data.isnull().sum()
```

```
              0
City          0
Date          0
PM2.5         0
PM10          0
NO            0
NO2           0
NOx           0
NH3           0
CO            0
SO2           0
O3            0
Benzene       0
Toluene       0
Xylene        0
AQI           0
AQI_Bucket    0

dtype: int64
```

Figure 4: Checking if there are any null values left or not

The code uses *data.isnull().sum()* to verify the missing values. A complete analysis of the output reports zero null values exist in all dataset columns including City, Date, PM2.5, PM10 and others. The data cleaning process for these variables was successful which means the data can be used for analytics purposes.

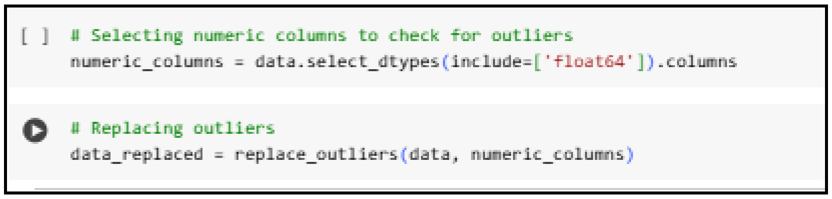


Outlier Detection and Replacement

```
# Defining a function to detect and replace outliers using the IQR method
def replace_outliers(df, columns):
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        # Replacing outliers below the lower bound
        df.loc[df[col] < lower_bound, col] = lower_bound
        # Replacing outliers above the upper bound
        df.loc[df[col] > upper_bound, col] = upper_bound
    return df
```

Figure 5: Function to detect and replace outliers

Interquartile Range (IQR) to detect and transform outliers across selected columns. First quartile value (Q1) and third quartile value (Q3) calculation precedes IQR determination followed by boundary determination. Outside values trigger boundary value assignment to reduce their effect on the analysis.

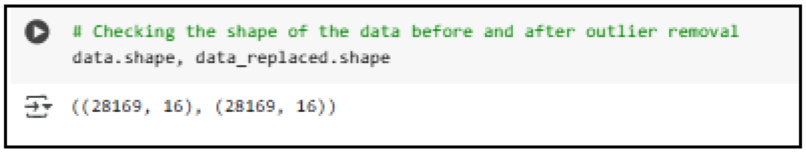


```
# Selecting numeric columns to check for outliers
numeric_columns = data.select_dtypes(include=['float64']).columns
```

```
# Replacing outliers
data_replaced = replace_outliers(data, numeric_columns)
```

Figure 6: Outlier Replacing outliers

Numeric columns in the dataset are selected for outlier detection. The chosen columns containing numeric data were selected specifically for performing outlier detection analysis. The *replace_outliers* function utilises an IQR method to detect and substitute outliers present in identified columns. The updated dataset containing outliers replacement goes into *data_replaced* for further analysis.

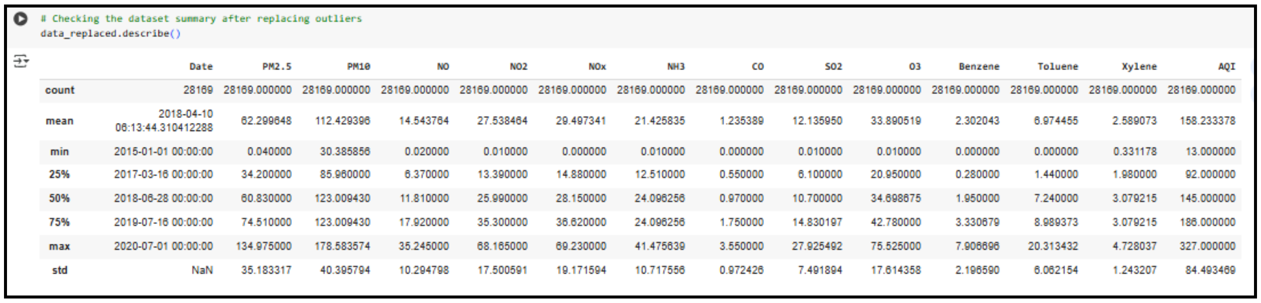


```
# Checking the shape of the data before and after outlier removal
data.shape, data_replaced.shape
```

```
((28169, 16), (28169, 16))
```

Figure 7:Before and after outlier remove

Dataset is checked before and after outlier removal to ensure that no rows or columns The output (('28169', '16'), ('28169', '16')) shows no alterations to the dataset dimensions which verifies that outlier replacement preserved the original organization of the data.



```
# Checking the dataset summary after replacing outliers
data_replaced.describe()
```

	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
count	28169	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000	28169.000000
mean	2018-04-10 06:13:44.310412288	62.299648	112.429396	14.543764	27.538464	29.497341	21.425835	1.235389	12.135950	33.890519	2.302043	6.974455	2.589073	158.233378
min	2015-01-01 00:00:00	0.040000	30.385856	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000	0.000000	0.000000	0.331178	13.000000
25%	2017-03-16 00:00:00	34.200000	85.960000	6.370000	13.390000	14.880000	12.510000	0.550000	6.100000	20.950000	0.280000	1.440000	1.980000	92.000000
50%	2018-06-28 00:00:00	60.830000	123.009430	11.810000	25.990000	28.150000	24.096256	0.970000	10.700000	34.698675	1.950000	7.240000	3.079215	145.000000
75%	2019-07-16 00:00:00	74.510000	123.009430	17.920000	35.300000	36.620000	24.096256	1.750000	14.830197	42.780000	3.330679	8.989373	3.079215	186.000000
max	2020-07-01 00:00:00	134.975000	178.583574	35.245000	68.165000	69.230000	41.475639	3.550000	27.925492	75.525000	7.906696	20.313432	4.728037	327.000000
std	NaN	35.183317	40.395794	10.294798	17.500591	19.171594	10.717556	0.972426	7.491894	17.614358	2.196590	6.062154	1.243207	84.493469

Figure 8:Checking the dataset

The dataset summary shows descriptive statistics for air quality parameters after replacing outliers. It includes count, mean, min, max, standard deviation, and quartile values for variables such as PM2.5, PM10, NO, NO2, NH3, and others over a period from 2015 to 2020.

Data Exploration Using Tableau

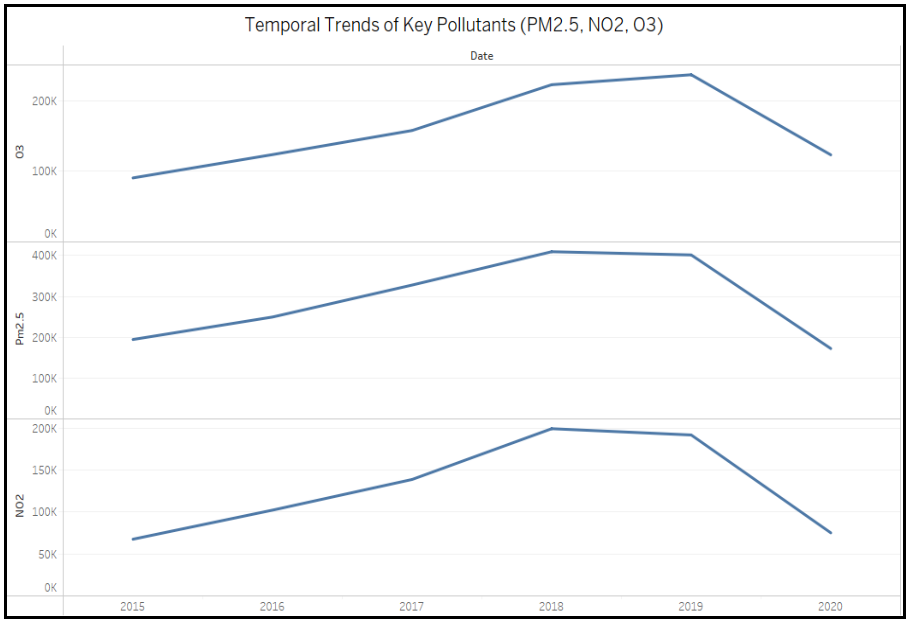


Figure 9 :Temporal Trends of Key Pollutants

Key pollutants PM2.5, NO2, O3 have evolved during 2015 to 2020. The data visualization presents changing pollutant level patterns through different time periods. The illustrated data allows viewers to track air quality shifts along with environmental regulations' effects over time.

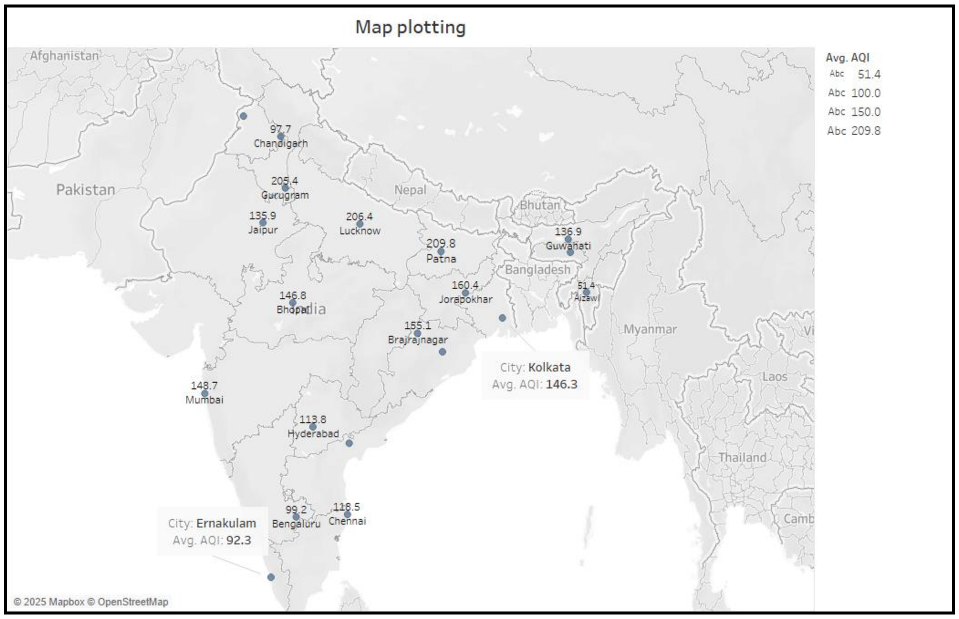


Figure 10: Map Plotting

Representing the Average Air Quality Index (AQI) throughout different Indian cities. Every location on the map displays the AQI measurement that reveals distinct air standard levels across cities. The map shows Delhi with high AQI marks while Kolkata maintains a moderate AQI through a visible legend that shows AQI ranges.

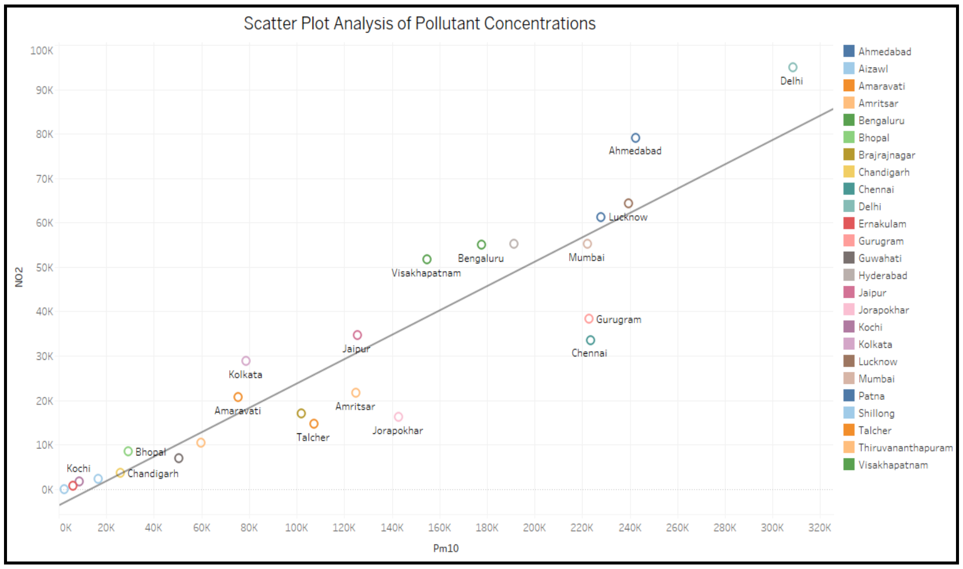


Figure 11: Scatter Plot Analysis of Pollutant Concentrations

Scatter plot analysis reveals data patterns surrounding pollutant relationships by showing their distribution configurations. Through data point visualizations researchers document patterns along with relationship formations and clustering effects to examine pollutant engagement mechanisms throughout space and time. By plotting data points, identify patterns, correlations, or clusters, indicating how pollutants interact or vary across locations or time. Air quality assessment through high concentration readings reveals problematic locations which helps environmental professionals direct their improvement initiatives to specific regions. From the scatter plot analysis, City with the highest pollutant concentration is Delhi and City with the lowest pollutant concentration is Kochi.

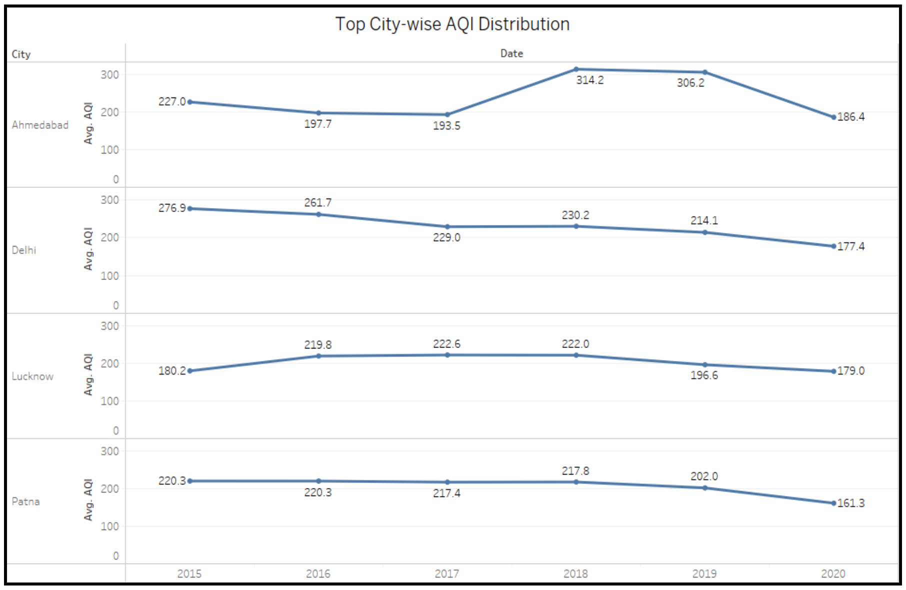


Figure 12:Top City-wise AQI Distribution

The graphic visualizes shifts in air quality measurements for cities between 2015 and 2020. This dataset demonstrates that cities present different levels of air quality that either remain stable or rise and fall. The analysis reveals cities facing immediate pollution control demands alongside cities where air quality already remains decent.

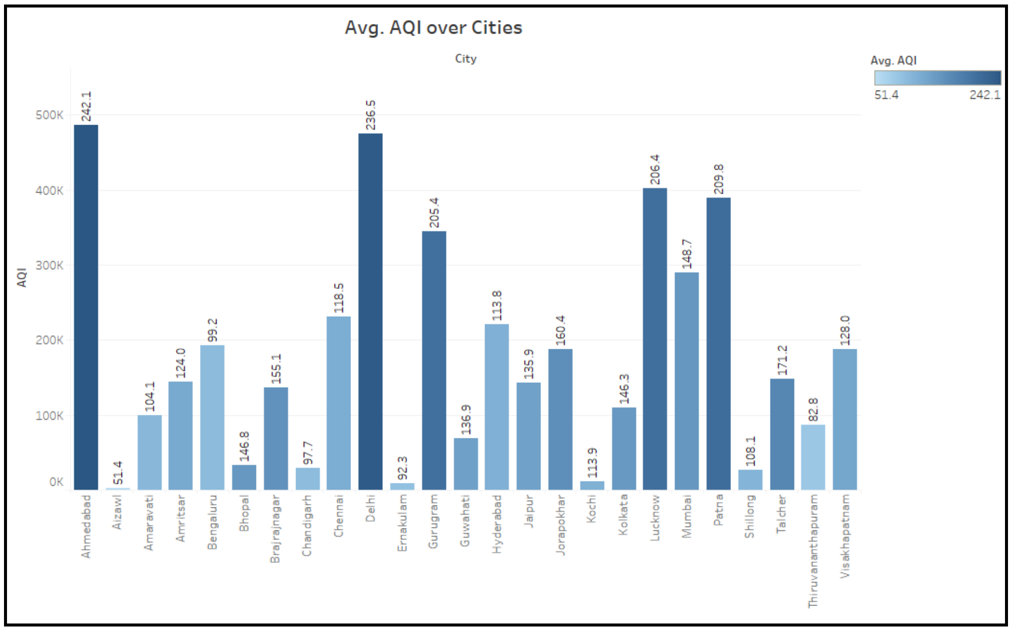


Figure 13: Avg. AQI over Cities

Average Air Quality Index measurements demonstrate the data points from various cities which appear on the graph. The most severe pollution occurs in the city where a high AQI reading appears whereas the lowest rate shows clean air quality. The highest average AQI over cities is 242.1 in Ahmedabad, and the lowest is 51.4 in Aizawl.

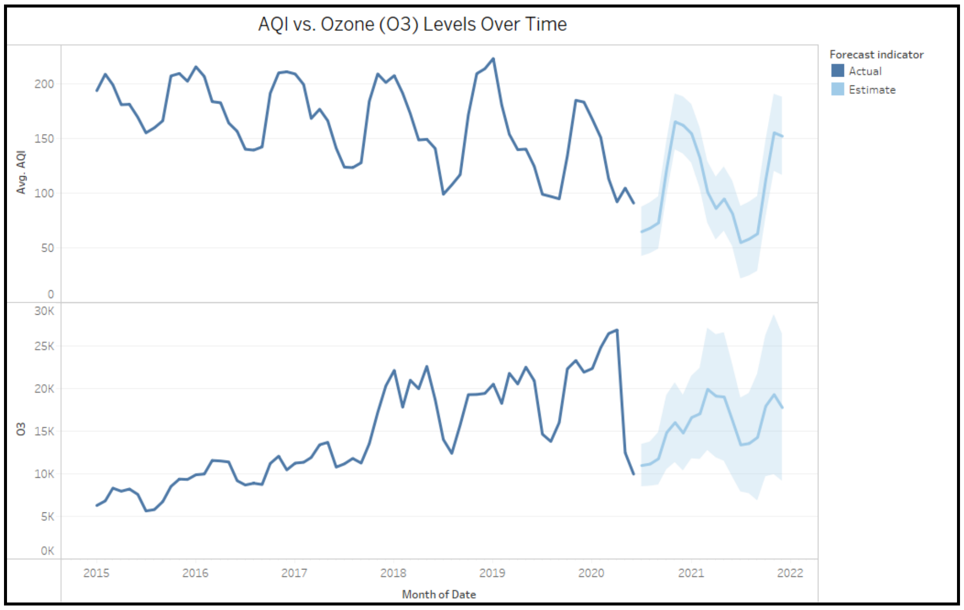


Figure 14: AQI vs. Ozone Levels Over Time

The graphical representation shows how AQI interacts with O3 levels from 2015 through 2022. The data establishes the connection between ozone concentrations and air quality measurements while demonstrating how seasonal trends and pollution reduction strategies perform regarding ozone pollution.

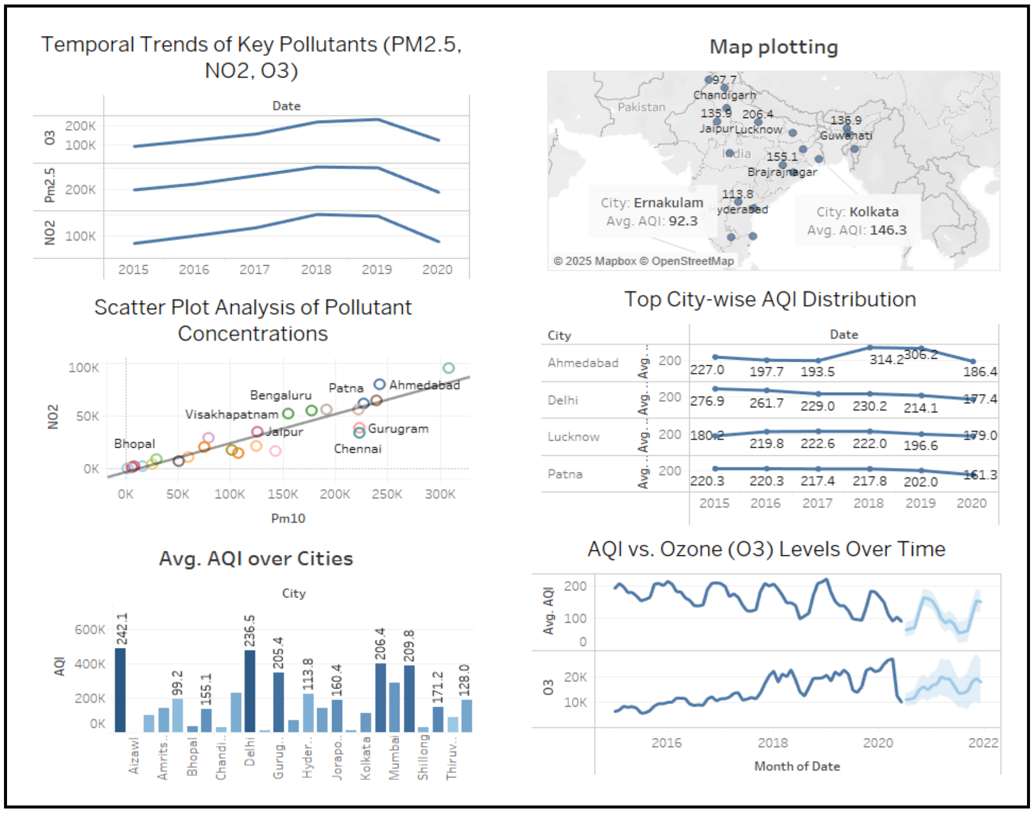
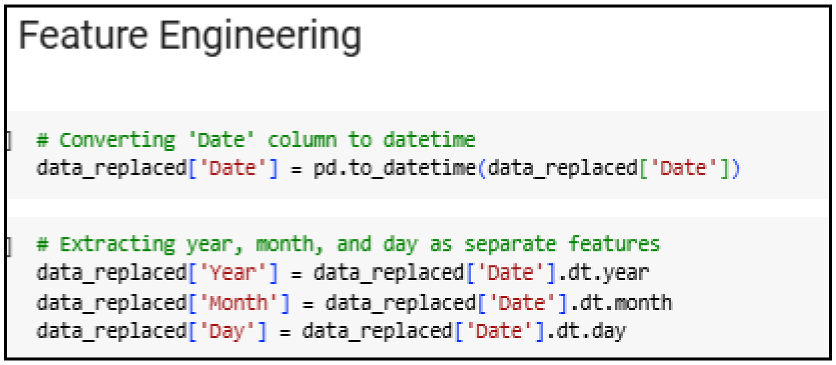


Figure 15: Dashboard

The dashboard displays air quality data from Indian cities throughout its platform. The interface presents data through multiple visual displays that demonstrate temporal pollution trends alongside scatter plot pollutant examinations and city-based average air quality index mapping with additional time-series comparisons of AQI to ozone measurements. The Air Quality Index reaches its peak level at 242.1 in Ahmedabad and maintains its lowest point at 15.4 in Anantnag. Through graphical illustrations users can locate air quality problem areas and examine interconnections between separate pollutants.

CHAPTER FOUR – MACHINE LEARNING APPLICATION

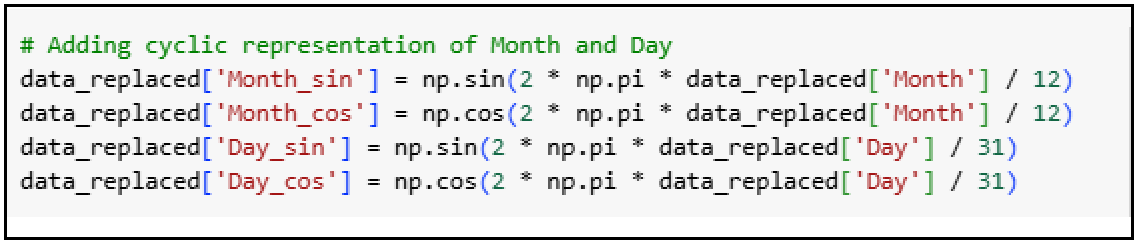
Feature Selection and Engineering


Feature Engineering

```
# Converting 'Date' column to datetime
data_replaced['Date'] = pd.to_datetime(data_replaced['Date'])

# Extracting year, month, and day as separate features
data_replaced['Year'] = data_replaced['Date'].dt.year
data_replaced['Month'] = data_replaced['Date'].dt.month
data_replaced['Day'] = data_replaced['Date'].dt.day
```

Figure 9: Converting date column

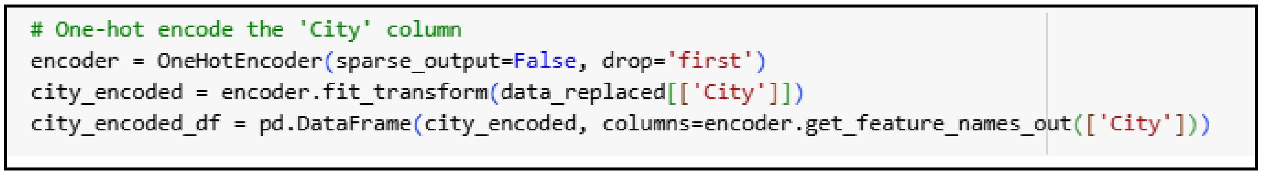
The "Date" column received a datetime format transformation that makes it possible to extract relevant time-based features more effectively. The *dt* creates distinct 'Year', 'Month' and 'Day' columns from the data. Through this transformation process machine learning models can analyze temporal patterns more effectively while performing feature engineering.



```
# Adding cyclic representation of Month and Day
data_replaced['Month_sin'] = np.sin(2 * np.pi * data_replaced['Month'] / 12)
data_replaced['Month_cos'] = np.cos(2 * np.pi * data_replaced['Month'] / 12)
data_replaced['Day_sin'] = np.sin(2 * np.pi * data_replaced['Day'] / 31)
data_replaced['Day_cos'] = np.cos(2 * np.pi * data_replaced['Day'] / 31)
```

Figure 10: Adding cyclic Representation

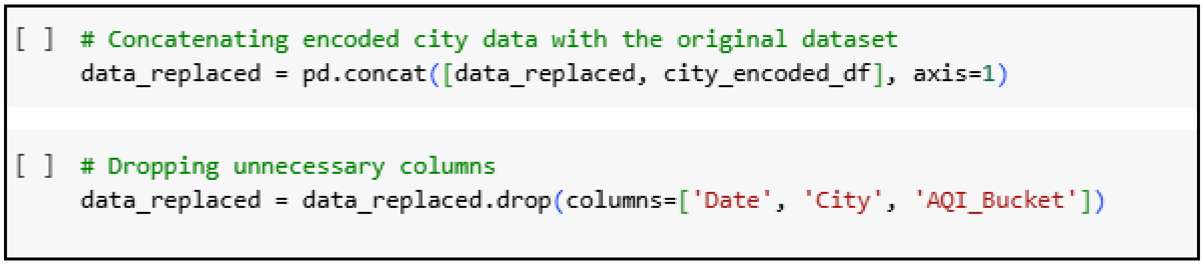
Cyclic representations of features 'Month' and 'Day' are generated through the application of sin and cos transformation at this step. The transformation methods spread month and day values along circular dimensions which help reveal their recurring patterns. By creating four new columns called 'Month_sin', 'Month_cos', 'Day_sin', and 'Day_cos' the model gains better capability in recognizing seasonal patterns and recurring cyclic events.



```
# One-hot encode the 'City' column
encoder = OneHotEncoder(sparse_output=False, drop='first')
city_encoded = encoder.fit_transform(data_replaced[['City']])
city_encoded_df = pd.DataFrame(city_encoded, columns=encoder.get_feature_names_out(['City']))
```

Figure 11: One-hot encodes the city column

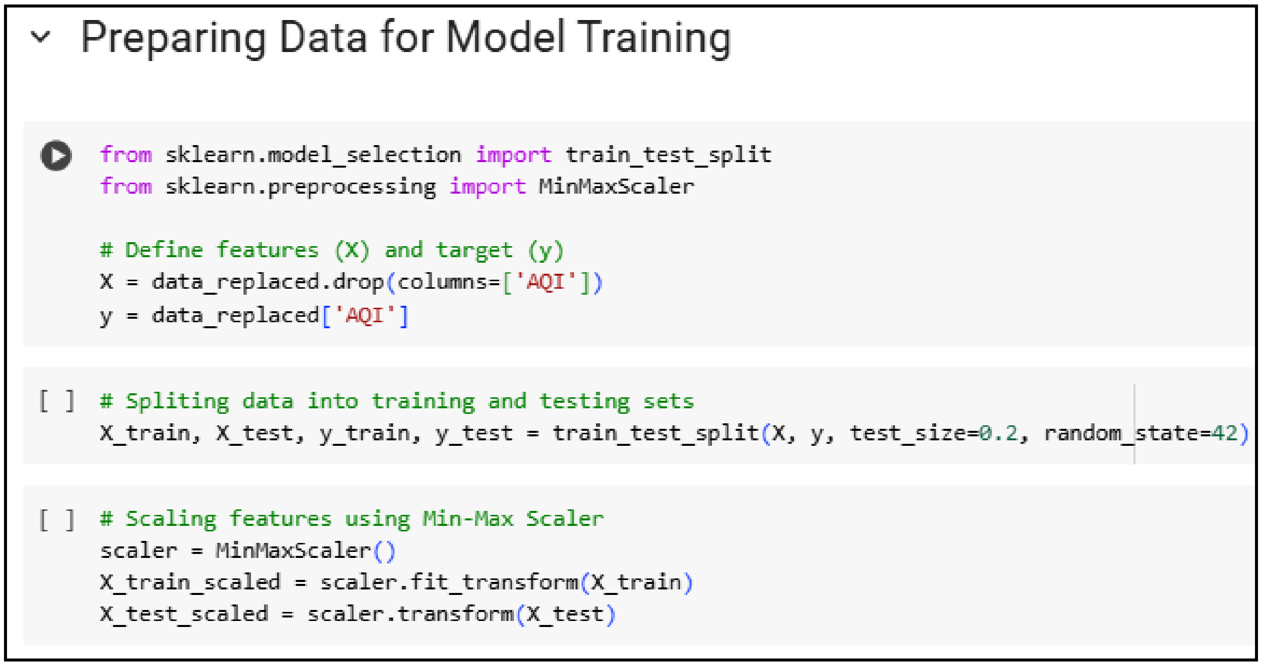
City-encoded columns exist in city_encoded_df. Information on seasonal patterns in the data will be captured by adding cyclic representations of time-based features including month and day.



```
[ ] # Concatenating encoded city data with the original dataset
    data_replaced = pd.concat([data_replaced, city_encoded_df], axis=1)

[ ] # Dropping unnecessary columns
    data_replaced = data_replaced.drop(columns=['Date', 'City', 'AQI_Bucket'])
```

Figure 12: Dropping unnecessary columns

Encoded city data is concatenated with the original dataset to integrate one-hot encoded features. Next step involves dropping the extraneous columns 'Date' 'City' and 'AQI_Bucket'. Fitting data analysis purposes requires the execution of this step. Including cyclic representations helps machine learning models detect periodic patterns which improves their ability to understand features.



Preparing Data for Model Training

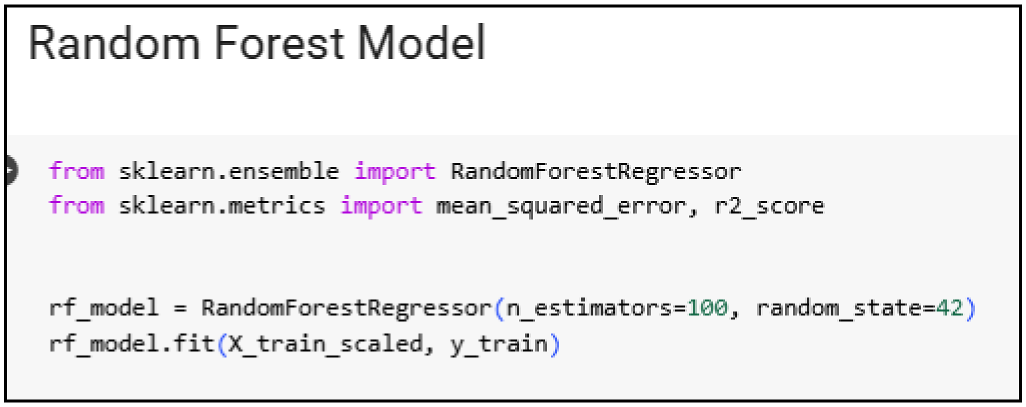
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

# Define features (X) and target (y)
X = data_replaced.drop(columns=['AQI'])
y = data_replaced['AQI']

[ ] # Spliting data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Scaling features using Min-Max Scaler
    scaler = MinMaxScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)
```

Figure 13: Data splitting and scaling features

Data preparation process determines both x(features) and y (the target outcome) while breaking the information into training and testing portions and applying Min-Max Scaling to normalize features. Model training and evaluation receive two normalized feature datasets through X_train_scaled and X_test_scaled.

Model Training

Random Forest



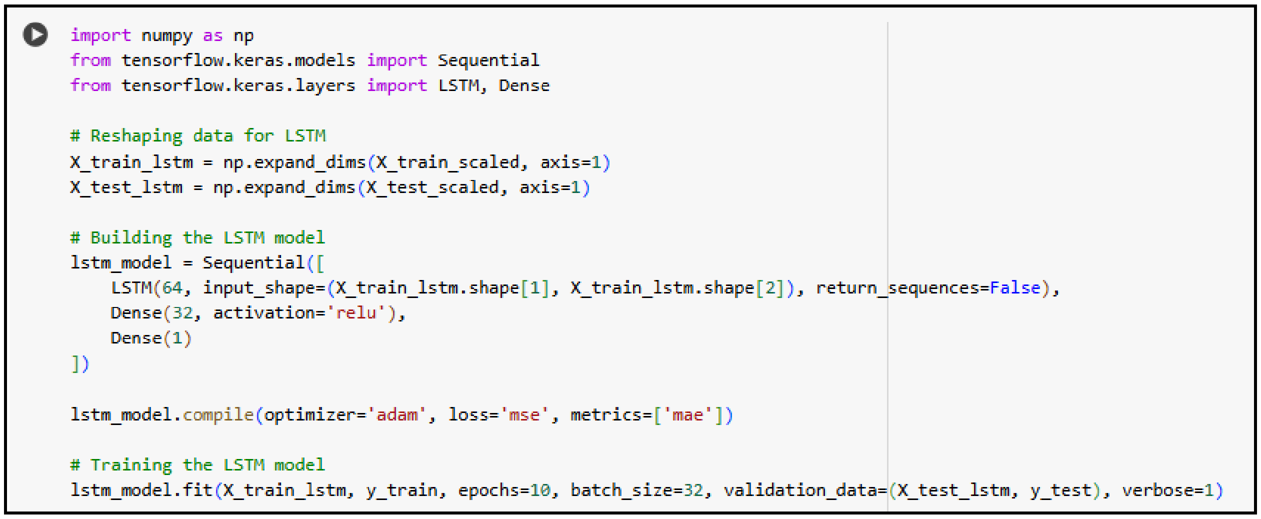
Random Forest Model

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score


rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train_scaled, y_train)
```

Figure 14: Random Forest Model training

Random Forest Regressor model requires 100 estimators and a specified random state to support reproducibility in the experiments. The trained model operates on the processing data set consisting of X_train_scaled and y_train. The scaled training data enables the Random Forest model to become a trained predictor and analytical tool for target variable analysis.

LSTM

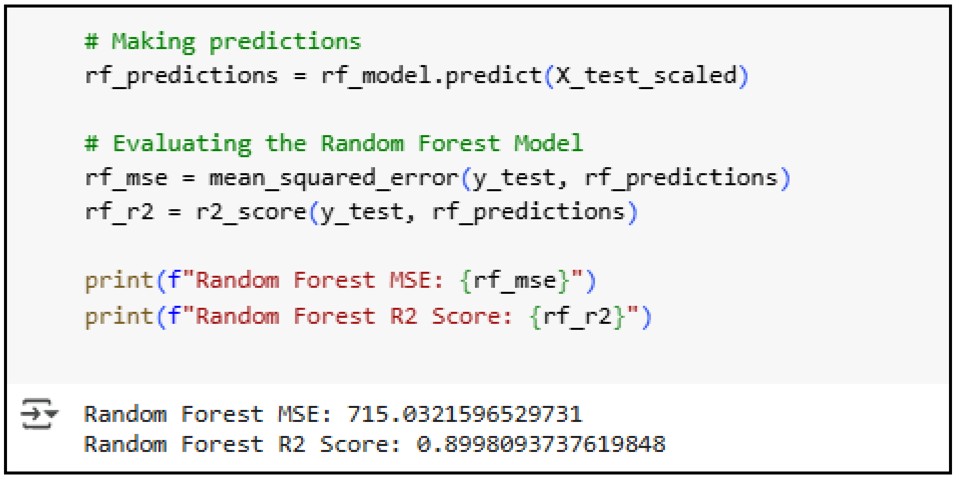


```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Reshaping data for LSTM
X_train_lstm = np.expand_dims(X_train_scaled, axis=1)
X_test_lstm = np.expand_dims(X_test_scaled, axis=1)

# Building the LSTM model
lstm_model = Sequential([
    LSTM(64, input_shape=(X_train_lstm.shape[1], X_train_lstm.shape[2]), return_sequences=False),
    Dense(32, activation='relu'),
    Dense(1)
])

lstm_model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Training the LSTM model
lstm_model.fit(X_train_lstm, y_train, epochs=10, batch_size=32, validation_data=(X_test_lstm, y_test), verbose=1)
```

Figure 15: LSTM Model Training

The method incorporates three stages which include reshaping normalized data for LSTM compatibility followed by model construction of an LSTM sequence learning and dense prediction system using the Adam optimizer and MSE loss. The model trains through 10 epochs to generate air quality predictions while MAE stands as one measurement metric.

Model Performance Evaluation

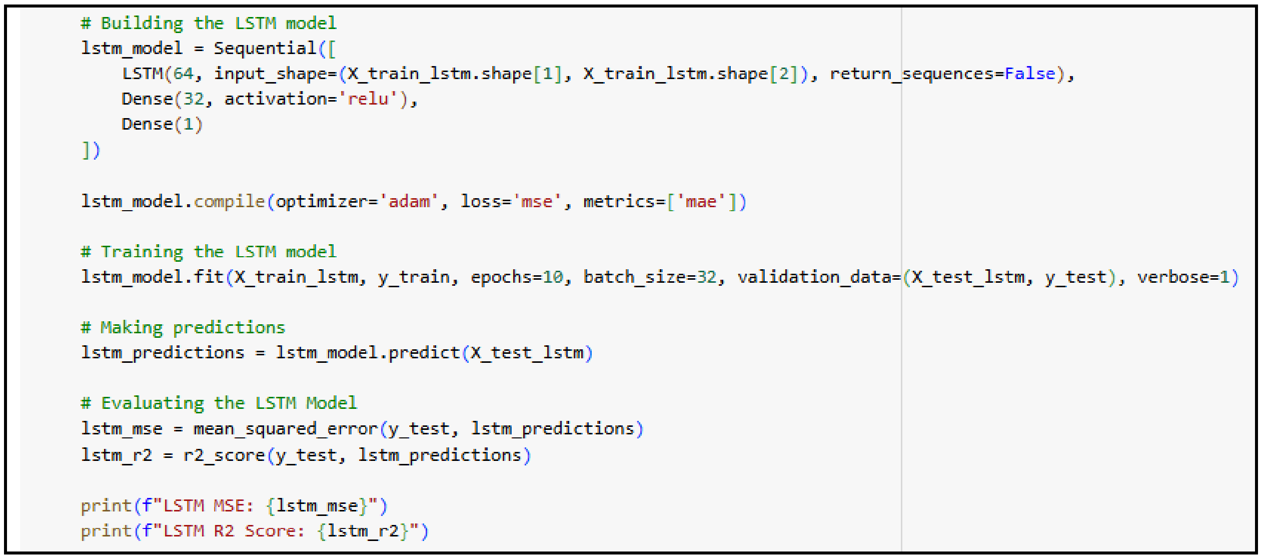


```
# Making predictions
rf_predictions = rf_model.predict(X_test_scaled)

# Evaluating the Random Forest Model
rf_mse = mean_squared_error(y_test, rf_predictions)
rf_r2 = r2_score(y_test, rf_predictions)

print(f"Random Forest MSE: {rf_mse}")
print(f"Random Forest R2 Score: {rf_r2}")
```

```
Random Forest MSE: 715.0321596529731
Random Forest R2 Score: 0.8998093737619848
```

Figure 16: Random Forest Model Evaluation

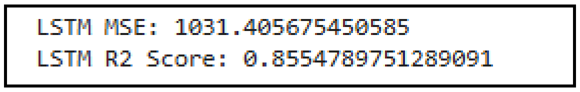
The Random Forest model predicts outcomes on scaled test data. Its performance is evaluated using Mean Squared Error (MSE) and R-squared (R^2) score. The results show an MSE of 715.03, indicating prediction errors, and an R^2 score of 0.8998, demonstrating strong predictive accuracy and model reliability



```
# Building the LSTM model
lstm_model = Sequential([
    LSTM(64, input_shape=(X_train_lstm.shape[1], X_train_lstm.shape[2]), return_sequences=False),
    Dense(32, activation='relu'),
    Dense(1)
])

lstm_model.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Training the LSTM model
lstm_model.fit(X_train_lstm, y_train, epochs=10, batch_size=32, validation_data=(X_test_lstm, y_test), verbose=1)

# Making predictions
lstm_predictions = lstm_model.predict(X_test_lstm)

# Evaluating the LSTM Model
lstm_mse = mean_squared_error(y_test, lstm_predictions)
lstm_r2 = r2_score(y_test, lstm_predictions)

print(f"LSTM MSE: {lstm_mse}")
print(f"LSTM R2 Score: {lstm_r2}")
```

Figure 19: LSTM Model Evaluation

Machine learning model trains with Adam optimizer and mean squared error loss for the training data. The model produces predictions for test samples where Mean Squared Error (MSE) and R-squared (R2) metrics verify the performance by measuring accuracy alongside fit quality.



```
LSTM MSE: 1031.405675450585
LSTM R2 Score: 0.8554789751289091
```

Figure 20: Model Accuracy Score

The LSTM model demonstrated prediction accuracy through an MSE of 1031.40 together with an R^2 score of 0.8554 which indicates low prediction mistakes while showing exceptional efficiency in temporal pattern detection and outcome forecasting.

CHAPTER FIVE – EVALUATION

The analyzed data presents multiple visualizations that assess air quality patterns along with pollutant distribution throughout Indian cities from 2015 to 2022. Key pollutants namely PM2.5 along with NO2 and O3 show shifting patterns during the period allowing us to understand how environmental rules and weather conditions influence air quality patterns. Maps show that Delhi acts as a hotspot of pollution while Kolkata performs as an example of average air quality conditions. Scatter plots have identified Delhi as the highest pollutant city and Kochi as the lowest while showing pollutant relationships and clusters. The city-wise analysis demonstrates Ahmedabad's position as the location with highest average AQI (242.1) while Aizawl maintains the lowest average of 51.4 among the examined cities. The graph which connects Ozone levels to air quality measurements provides critical information about seasonal pollution patterns and program effectiveness at air pollution control (Horn and Dasgupta, 2024). The central dashboard combines these analytic findings to show users exactly what areas need attention and how pollutants relate to each other. Visual data reveals targeted pollution mitigation efforts must focus on cities such as Delhi and Ahmedabad although Aizawl and Kochi demonstrate relatively cleaner conditions. Developed findings provide essential resources for people making decisions and for the effective management of air quality regulations.

Analyze performance and reliability in air quality predictions machine learning models such as Random Forest and LSTM need thorough evaluation (Meena *et al.*, 2024). The applied dataset received comprehensive preprocessing through strategies which addressed missing values and performed outlier detections and implemented feature engineering techniques before training. The analysis used Mean Squared Error (MSE) in combination with R-squared score (R^2) to evaluate the predictive accuracy and model predictive power weights of model performance (Garcia Garcia *et al.*, 2024).

Random Forest Model Evaluation

Robust ensemble method called Random Forest received training through 100 estimators with defined random states for reproducibility purposes. The evaluation took place on an optimized test dataset. The evaluation showed that the average squared difference between prediction and actual measurement yielded an MSE value of 715.03. The model performance exists when MSE values are low which demonstrates that predictions from the model match actual measurement values reasonably well. The independent variables used for prediction revealed an R^2 score value of 0.8998 for the dependent variable variance. The Random Forest model successfully explains 90% of air quality data variations by achieving a strong predictive R^2 value.

The Random Forest model achieves successful performance by its ability to identify complex connections hidden in the dataset.

LSTM Model Evaluation

The LSTM (Long Short-Term Memory) model analyzes sequential data patterns. LSTM models contain LSTM layers as well as dense layers. The training of the model utilized Adam optimizer combined with mean squared error loss across ten iterations. During evaluation the LSTM predictions received assessment through both MSE and R^2 score tests. Results from the LSTM model indicate prediction errors reached 1031.41 MSE compared to the Random Forest model's results suggesting larger prediction errors. Alexandrov's model revealed a strong capacity to forecast air quality through an R^2 score reaching 0.8554 which explains 85.5% of data variance. The LSTM model demonstrates a strong ability to detect time-dependent patterns and sequence relationships within air quality datasets which makes it a powerful forecasting instrument.

CONCLUDING REMARKS

The study successfully analyzed air quality data across Indian cities from 2015 to 2022, identifying key pollution trends, hotspots, and predictive insights using machine learning and Tableau visualizations. The study reveals the crucial need for specific pollution reduction programs in contaminated areas together with the essential role of analytics systems for air risk management. Performance metrics from Random Forest surpassed LSTM predictions by achieving stronger values of MSE at 715.03 with R^2 score at 0.8998 while the LSTM model recorded MSE at 1031.41 and R^2 score at 0.8554 The research delivers concrete pollution reduction guidelines using machine learning methods combined with data visualization tools to enable stakeholders and policymakers in making well-informed decisions.

REFERENCES AND BIBLIOGRAPHY

de la Rosa, R., Le, A., Holm, S., Ye, M., Bush, N.R., Hessler, D., Koita, K., Bucci, M., Long, D. and Thakur, N., 2024. Associations between early-life adversity, ambient air pollution, and telomere length in children. *Psychosomatic Medicine*, pp.10-1097.

Feng, T., Sun, Y., Shi, Y., Ma, J., Feng, C. and Chen, Z., 2024. Air pollution control policies and impacts: A review. *Renewable and Sustainable Energy Reviews*, *191*, p.114071.

Garcia Garcia, C., Salmeron Gomez, R. and Garcia Perez, J., 2024. A review of ridge parameter selection: minimization of the mean squared error vs. mitigation of multicollinearity. *Communications in Statistics-Simulation and Computation*, *53*(8), pp.3686-3698.

Horn, S.A. and Dasgupta, P.K., 2024. The Air Quality Index (AQI) in historical and analytical perspective a tutorial review. *Talanta*, *267*, p.125260.

Liu, Q., Cui, B. and Liu, Z., 2024. Air Quality Class Prediction Using Machine Learning Methods Based on Monitoring Data and Secondary Modeling. *Atmosphere*, *15*(5), p.553.

Meena, K.K., Bairwa, D. and Agarwal, A., 2024. A machine learning approach for unraveling the influence of air quality awareness on travel behavior. *Decision Analytics Journal*, *11*, p.100459.

Meo, S.A., Salih, M.A., Al-Hussain, F., Alkhalifah, J.M., Meo, A.S. and Akram, A., 2024. Environmental pollutants PM2. 5, PM10, carbon monoxide (CO), nitrogen dioxide (NO. *European Review for Medical and Pharmacological Sciences*, *28*, pp.789-796.

Ouma, Y.O., Keitsile, A., Lottering, L., Nkwae, B. and Odirile, P., 2024. Spatiotemporal empirical analysis of particulate matter PM2. 5 pollution and air quality index (AQI) trends in Africa using MERRA-2 reanalysis datasets (1980–2021). *Science of The Total Environment*, *912*, p.169027.

Rusmayadi, G., Zulfikhar, R., Angrianto, R., Sutiharni, S. and Tuhumena, V.L., 2024. Analysis of Mangrove Ecosystems and Number of Plants on Air Pollution Reduction by Mangrove Plants. *West Science Agro*, *2*(01), pp.1-9.

Shao, M., Lv, S., Wei, Y. and Zhu, J., 2024. The various synergistic impacts of precursor emission reduction on PM2. 5 and O3 in a typical industrial city with complex distributions of emissions. *Science of The Total Environment*, p.173497.

Shi, K., Mei, X., Chen, C.R. and Liu, C., 2025. Impact of atmospheric O3 and NO2 on the secondary sulfates in real atmosphere. *Journal of Environmental Sciences*, *150*, pp.277-287.

Wang, L., Zhao, Y., Liu, X. and Shi, J., 2024. Enhancement of atmospheric oxidation capacity induced co-pollution of the O3 and PM2. 5 in Lanzhou, northwest China. *Environmental Pollution*, *341*, p.122951.

Yi, J., Kim, S.H., Lee, H., Chin, H.J., Park, J.Y., Jung, J., Song, J., Kwak, N., Ryu, J. and Kim, S., 2024. Air quality and kidney health: Assessing the effects of PM10, PM2. 5, CO, and NO2 on renal function in primary glomerulonephritis. *Ecotoxicology and Environmental Safety*, *281*, p.116593.

Git Hub Link

https://github.com/jayymoradiya/Machine-Learning-and-Visualization---Project-Outputs-.git