National Conference on Behavioural Economics and Intelligent Decision Systems for Climate Change and Sustainable Development
Int. J. Adv. Res. in Artificial Intelligence and Machine Learning Reviews
Volume 2, Issue 1 - 2026

Review Article

Forecasting Air Quality Index (AQI) Using Machine Learning Models: A Comparative Study

Amandeep Singh[1], Navpreet Singh[2], Manish Kumar[3], Shivankar Sinha[4], Prabhkiran Kaur[5]

[1,2,3] Student, [5] Assistant Professor, Department of Mechanical Engineering, IKG Punjab Technical University, India
[4] Student, Department of Computer Science Engineering, Khalsa College of Engineering and Technology, Amritsar, India
DOI: https://doi.org/10.24321/3117.4809.202607

I N F O

Corresponding Author:
Prabhkiran Kaur, Department of Mechanical Engineering, IKG Punjab Technical University, India
E-mail Id:
dr.prabhkiran@ptu.ac.in
How to cite this article:
Singh A, Singh N, Kumar M, Sinha S, Kaur P. Forecasting Air Quality Index (AQI) Using Machine Learning Models: A Comparative Study. *Int J Adv Res Artif Intell Mach Learn Rev* 2026; 2(1): 286-293.

Date of Submission: 2025-11-26
Date of Acceptance: 2025-11-28

A B S T R A C T

One of the biggest environmental issues affecting our health and well-being is air quality, an unseen danger that we breathe every day. Packed with dangerous gases and microscopic particles, poor air quality is a silent killer that can cause anything from chronic respiratory issues to serious, life-threatening diseases. The enormity of this issue emphasises how urgently we need information that is easy to understand in order to safeguard our communities.

By concentrating on the Air Quality Index (AQI), a simple method of expressing how clean or polluted the air is, this research directly addresses that challenge. The goal is to uncover the hidden narrative within the numbers by employing intelligent computer models (machine learning) to sort through years' worth of air pollution data, including daily readings of smog, soot, and other pollutants. The objective is to increase the usefulness of the AQI by creating a system that can precisely forecast the air quality for tomorrow, informing us of two important factors: the likelihood that the air will be unhealthy (the probability of it reaching a critical level) and how bad it will likely be (the predicted AQI number).

Keywords: Machine Learning K-Nearest Neighbours, Support Vector Machine, Naive Bayes, K-Means Clustering

Introduction

Air quality is a crucial determinant of both human health and environmental sustainability. Air pollution – driven by urbanisation, industrialisation and other factors – is a global problem that "threatens environmental sustainability and severely affects public health". The Air Quality Index (AQI) is a standardised composite indicator that aggregates concentrations of key pollutants (e.g. $PM_{2.5}$, PM_{10}, O_3, NO_2, SO_2, CO) into a single numeric scale. By summarising air pollution levels in an easy-to-communicate form, the AQI helps the public and policymakers quickly assess current conditions and health risks. Because poor air quality is linked to respiratory, cardiovascular and other serious health problems, the ability to forecast future AQI values is especially important.

Machine learning (ML) provides powerful tools to analyse historical AQI data and predict future air quality trends. In recent years, ML-based air quality models have become increasingly widespread. Techniques ranging from regression trees and support-vector machines to deep neural networks

can learn complex spatiotemporal relationships between pollutant concentrations, meteorological variables and AQI.[1] For example, ensemble methods such as XG Boost and Light GBM often achieve state-of-the-art accuracy in forecasting daily AQI. These models are able not only to capture seasonal and regional pollution patterns, but also to quantify which pollutants (often fine particulates) most influence air quality.[2]

Accurate AQI forecasts can directly inform public health policy and individual precautionary actions. By providing advance warning of deteriorating air quality, forecasts enable cities to issue health advisories or enforce temporary emissions controls, and allow vulnerable populations (e.g. those with asthma or heart disease) to reduce exposure.[3] In this way, ML-driven AQI prediction supports the development of proactive pollution-control strategies and environmental regulations Ultimately, leveraging the AQI dataset in predictive modelling not only advances our understanding of urban pollution dynamics, but also helps protect public health and guide evidence-based air quality management.[4]

Implementation

Dataset

This research directly addresses that challenge by focussing on the Air Quality Index (AQI), a straightforward way to express how clean or polluted the air is. By using intelligent computer models (machine learning) to sift through years' worth of air pollution data, including daily readings of smog, soot, and other pollutants, the research aims to reveal the hidden narrative within the numbers.[5] By developing a system that can accurately predict tomorrow's air quality, the goal is to make the AQI more useful by letting us know two crucial elements: the likelihood that the air will be unhealthy (the probability that it will reach a critical level) and how bad it will likely be (the predicted AQI number).

Each row in the dataset represents the complete air quality snapshot for a single day in a specific city and includes the following key information:

- **City:** The geographical location where the air quality reading was taken.
- **Date:** The day the measurement was recorded (in format).
- **PM2.5/PM10 :** The concentration of Particulate Matter () in the air. These are tiny particles that can penetrate deep into the lungs, with being smaller and more dangerous.
- **$NO/NO_2/NO_x$:** The levels of Nitrogen Monoxide, Nitrogen Dioxide, and their combined form, Nitrogen Oxides, which are key components of smog and a result of vehicle emissions and industrial activity.
- **NH_3:** The concentration of Ammonia, often resulting from agricultural and industrial processes.
- **CO:** The concentration of Carbon Monoxide, a colorless, odorless, and poisonous gas typically produced by burning carbon-based fuels.
- **SO_2:** The concentration of Sulfur Dioxide, which contributes to acid rain and respiratory problems.
- **O_3:** The concentration of Ozone, a main component of smog, which is harmful at ground level.
- **Benzene / Toluene / Xylene:** The concentration of these volatile organic compounds (VOCs), which are toxic pollutants commonly found in vehicle exhaust and industrial solvents.
- **AQI:** The Air Quality Index (a single, easy-to-understand number) derived from all the pollutant readings. This is the main target for our prediction, showing how strong or poor the air quality is.
- **AQI_Bucket:** A categorical label (e.g., ‹Good›, ‹Moderate›, ‹Severe›) that classifies the value into health-risk levels.

The dataset includes a total of 29,531 daily air quality entries spanning from January 01, 2015, to July 01, 2020, covering 26 major Indian cities. This data can be used for various purposes, such as analyzing the impact of events (like festivals or lockdowns) on pollution, tracking air quality trends over time, or, most critically, building models to predict the future to give public health officials and citizens an early warning of unhealthy air days.

Data preprocessing

Data preprocessing is the process of cleaning, transforming, and preparing raw data before feeding it into a machine learning model.

Column Cleaning

- Whitespace was stripped from the beginning and end of all column names.

Handling Missing Values (Imputation)

- Numerical Imputation: Filled all missing values in numerical columns (like 'Depth' or 'Nst') with the mean (average) value of that column.
- Categorical Imputation: Filled all missing values in object/text columns with the mode (most frequent value) of that column.

Feature Scaling

- For models like SVR, Logistic Regression, and Clustering, you used StandardScaler.
- This step rescales features to have a mean of 0 and a standard deviation of 1.

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

```
Before Preprocessing:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        29531 non-null  object
 1   Date        29531 non-null  object
 2   PM2.5       24933 non-null  float64
 3   PM10        18391 non-null  float64
 4   NO          25949 non-null  float64
 5   NO2         25946 non-null  float64
 6   NOx         25346 non-null  float64
 7   NH3         19203 non-null  float64
 8   CO          27472 non-null  float64
 9   SO2         25677 non-null  float64
 10  O3          25509 non-null  float64
 11  Benzene     23908 non-null  float64
 12  Toluene     21490 non-null  float64
 13  Xylene      11422 non-null  float64
 14  AQI         24850 non-null  float64
 15  AQI_Bucket  24850 non-null  object
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
None
        City        Date  PM2.5  PM10     NO    NO2    NOx  NH3     CO   SO2  \
0  Ahmedabad  2015-01-01    NaN   NaN   0.92  18.22  17.15  NaN   0.92  27.64
1  Ahmedabad  2015-01-02    NaN   NaN   0.97  15.69  16.46  NaN   0.97  24.55
2  Ahmedabad  2015-01-03    NaN   NaN  17.40  19.30  29.70  NaN  17.40  29.07
3  Ahmedabad  2015-01-04    NaN   NaN   1.70  18.48  17.97  NaN   1.70  18.59
4  Ahmedabad  2015-01-05    NaN   NaN  22.10  21.42  37.76  NaN  22.10  39.33

       O3  Benzene  Toluene  Xylene  AQI AQI_Bucket
0  133.36     0.00     0.02    0.00  NaN        NaN
1   34.06     3.68     5.50    3.77  NaN        NaN
2   30.70     6.80    16.40    2.25  NaN        NaN
3   36.08     4.43    10.14    1.00  NaN        NaN
4   39.31     7.01    18.89    2.78  NaN        NaN
```

Figure 1.Before Preprocessing

```
After Preprocessing:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   City        29531 non-null  object
 1   Date        29531 non-null  object
 2   PM2.5       29531 non-null  float64
 3   PM10        29531 non-null  float64
 4   NO          29531 non-null  float64
 5   NO2         29531 non-null  float64
 6   NOx         29531 non-null  float64
 7   NH3         29531 non-null  float64
 8   CO          29531 non-null  float64
 9   SO2         29531 non-null  float64
 10  O3          29531 non-null  float64
 11  Benzene     29531 non-null  float64
 12  Toluene     29531 non-null  float64
 13  Xylene      29531 non-null  float64
 14  AQI         29531 non-null  float64
 15  AQI_Bucket  29531 non-null  object
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
None
        City        Date  PM2.5   PM10     NO    NO2    NOx    NH3     CO  \
0  Ahmedabad  2015-01-01  48.57  95.68   0.92  18.22  17.15  15.85   0.92
1  Ahmedabad  2015-01-02  48.57  95.68   0.97  15.69  16.46  15.85   0.97
2  Ahmedabad  2015-01-03  48.57  95.68  17.40  19.30  29.70  15.85  17.40
3  Ahmedabad  2015-01-04  48.57  95.68   1.70  18.48  17.97  15.85   1.70
4  Ahmedabad  2015-01-05  48.57  95.68  22.10  21.42  37.76  15.85  22.10

     SO2      O3  Benzene  Toluene  Xylene    AQI AQI_Bucket
0  27.64  133.36     0.00     0.02    0.00  118.0   Moderate
1  24.55   34.06     3.68     5.50    3.77  118.0   Moderate
2  29.07   30.70     6.80    16.40    2.25  118.0   Moderate
3  18.59   36.08     4.43    10.14    1.00  118.0   Moderate
4  39.33   39.31     7.01    18.89    2.78  118.0   Moderate
```

Figure 2.After Preprocessing

Linear Regression

Linear Regression is a machine learning algorithm used to predict a continuous numerical value (like earthquake magnitude).

It works by finding the "best-fit" straight line (or plane) that describes the relationship between a set of input features (predictors) and an output variable.

This model predicts the output using only one input feature.

- **Input Feature (X):** PM10
- **Output Variable (y):** PM2.5

Multiple Linear Regression

Multiple Linear Regression is a machine learning algorithm used to predict a continuous numerical value (like 'Magnitude'). It's an extension of Simple Linear Regression.

Instead of using just one input feature to make a prediction, it uses two or more input features.

The goal is to find a single equation that combines the predictive power of all those features, weighting each one based on its importance.

This model predicts the output using multiple input features at the same time.

- **Input Features (X):** PM10, NO_2, CO
- **Output Variable (y):** PM2.5

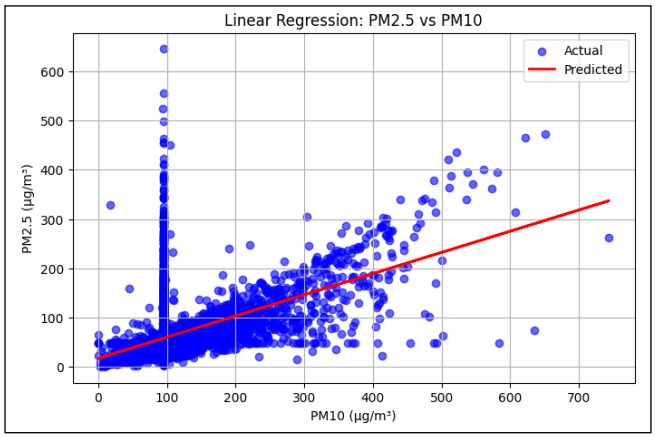

Figure 3.Linear Regression

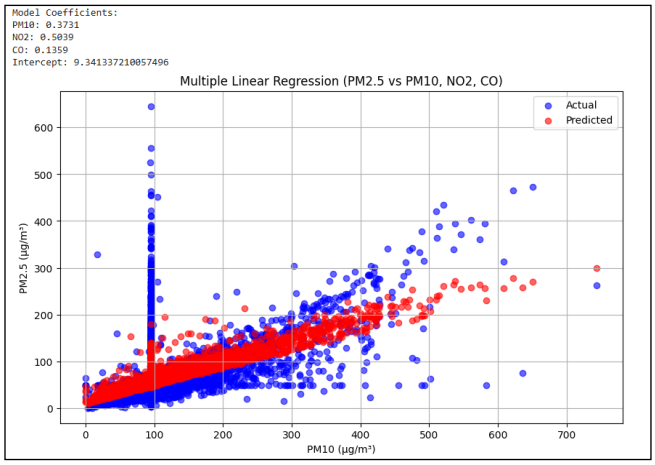

Figure 4.Multiple Linear Regression

Decision Tree

A decision tree is a hierarchical, flowchart-like predictive model that partitions the feature space into subsets using a sequence of feature tests. Each internal node in the tree applies a test on one feature (for example, "Age > 30?"), and each branch corresponds to the outcome of that test (e.g., "yes" or "no"). The model splits the data recursively in a top-down fashion (often called recursive partitioning) and the process continues until a stopping condition is met. The leaf nodes (terminal nodes) then carry the final predictions: for classification trees a leaf assigns a class label, and for regression trees it outputs a numeric value (often the mean of target values in that

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

region).[6] In fact, decision trees are formally used for both tasks – classification trees predict discrete categories and regression trees predict continuous outcomes – which is why the general CART framework, introduced by Breiman et al. (1984), encompasses both types. It's a flowchart-like structure where each:

- Internal node represents a "test" or "question" on a feature.
- Branch represents the outcome of the test ("Yes" or "No").
- Leaf node represents the final prediction .

A tree "learns" by finding the best way to split the data. This process is called recursive partitioning.

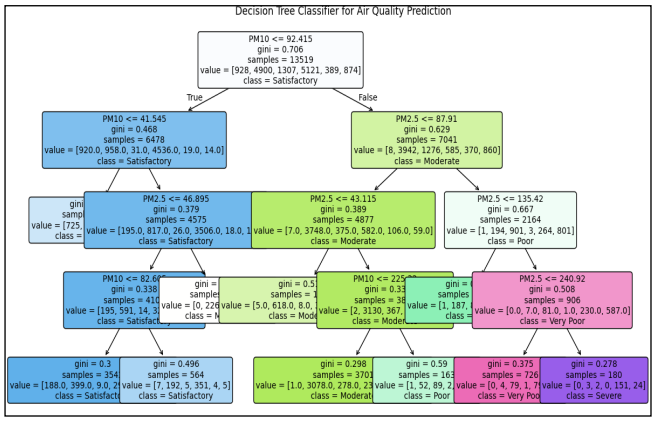

Figure 5.Decision Tree

KNN

K-Nearest Neighbors (KNN) is a machine learning algorithm that makes predictions based on the 'K' most similar data points (neighbors) it has already seen.

We used the K Neighbours Regressor version. This means you used KNN to predict a continuous number (the Mag).

Here is exactly how KNN code worked:

1. Find Neighbours: When ask it to predict the magnitude of a new earthquake, the model searched through its training data to find the 'K' earthquakes that were most "similar" based on our four features: Latitude, Longitude, Depth, and Nst.
2. Set 'K' Value: In our code, you set K=5 .
3. Average the Neighbours: The model found the 3 most similar earthquakes, looked at their Mag values, and averaged them to get the final prediction.

K-Nearest Neighbours (KNN) is a simple and intuitive machine learning algorithm. The main idea is: "You can guess what something is by looking at the things most similar to it."

It works by finding the "K" closest data points (the "neighbours") to a new, unknown data point. It then uses those neighbours to make a prediction.

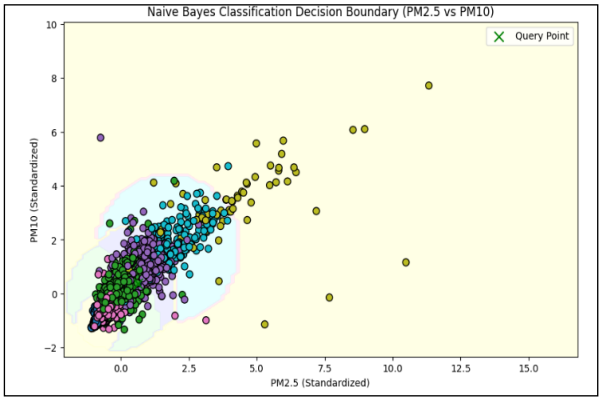

Figure 6.KNN

SVM(Support Vector Machine)

SVM is a powerful and versatile machine learning algorithm used for both classification and regression.

The main idea behind SVM is to find the "best" boundary that separates or fits the data.

This is the most common use. SVM finds the optimal line (or "hyperplane" in higher dimensions) that best separates the data into different classes (e.g., 'Low' vs. 'High' magnitude).

It's not just any line; it's the specific line that creates the maximum possible margin (distance) between itself and the closest data points from each class. This large margin makes the model robust.[7]

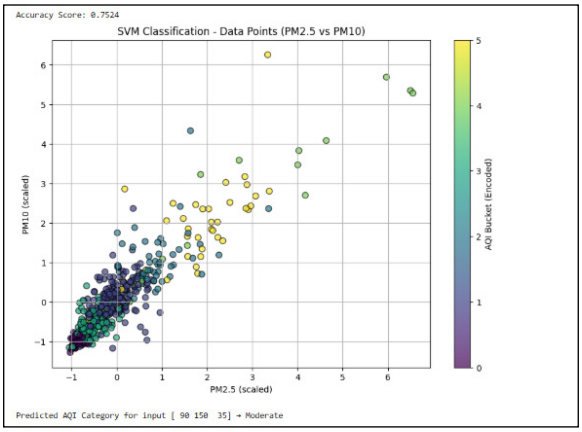

Figure 7.Support Vector Machine

Naive Baye's

The core of our research is using Naive Bayes to classify the daily Air Quality Index (AQI) category based on key pollutants. This is a critical task for issuing timely public health alerts.

Naive Bayes is a classification algorithm based on probability. It's used to predict which category of air quality a given day's measurements fall into.

Its main idea is: "What is the probability that the air quality belongs to the 'Severe' class, given the measured levels of PM2.5, PM10, and NO_2?"

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

How it Works

The model calculates the probability of each AQI class (e.g., 'Good', 'Moderate', 'Poor', 'Severe') based on the measured pollutant features (PM2.5, PM10, NO_2). It then picks the AQI category with the highest probability as its prediction.

Why "Naive"?

It makes a "naive" assumption that all the pollutant features are independent of each other. For example, it assumes that the PM2.5 level has no relationship to the NO_2 level. While in the real world, these pollutants are often highly correlated (e.g., both are high from traffic), the algorithm is surprisingly effective, fast to train, and often provides a strong baseline for classification.

Understanding Model Performance

The Confusion Matrix

A Confusion Matrix is a table that shows you exactly how well our AQI classification model performed by cross-referencing the model's predictions with the actual air quality categories.

Let's use the 'Severe' AQI category as our critical focus:

- **True Positive (TP):** The model correctly predicted a 'Severe' day. (It said 'Severe' and the actual AQI was 'Severe'). These are the numbers on the diagonal.
- False Positive (FP): The model made a Type I error (False Alarm). (It said 'Severe', but the actual AQI was only 'Moderate'). This leads to unnecessary public alerts.
- False Negative (FN): The model made a Type II error (Missed Warning). (It said 'Moderate', but the actual AQI was 'Severe'). This is a critical failure, as a public health warning was missed.

The Classification Report uses the Confusion Matrix to calculate two essential metrics for evaluating how reliable our air quality predictions are.

Precision: The "Accuracy of Warnings"

Precision tells you: Of all the times the model predicted a certain class, what percentage was correct?

- **AQI Example:** A precision of 85% for the 'Severe' class means: "When my model predicted a day would have 'Severe' air quality, it was right 85% of the time."
- High Precision is good if you want to be very sure about our alerts (to avoid cry-wolf scenarios).

Recall: The "Completeness of Warnings"

Recall tells you: Of all the actual items in a certain class, what percentage did the model find?

- **AQI Example:** A recall of 75% for the 'Severe' class means: "Of all the 'Severe' air quality days that actually happened, my model successfully identified 75% of them."[8]
- High Recall is crucial if you want to make sure you capture as many critical instances as possible, ensuring you don't miss issuing a public health warning for a truly 'Severe' day.

```
        City        Date  PM2.5  PM10     NO    NO2    NOx  NH3     CO    SO2  \
0  Ahmedabad  2015-01-01    NaN   NaN   0.92  18.22  17.15  NaN   0.92  27.64
1  Ahmedabad  2015-01-02    NaN   NaN   0.97  15.69  16.46  NaN   0.97  24.55
2  Ahmedabad  2015-01-03    NaN   NaN  17.40  19.30  29.70  NaN  17.40  29.07
3  Ahmedabad  2015-01-04    NaN   NaN   1.70  18.48  17.97  NaN   1.70  18.59
4  Ahmedabad  2015-01-05    NaN   NaN  22.10  21.42  37.76  NaN  22.10  39.33

       O3  Benzene  Toluene  Xylene  AQI AQI_Bucket
0  133.36     0.00     0.02    0.00  NaN        NaN
1   34.06     3.68     5.50    3.77  NaN        NaN
2   30.70     6.80    16.40    2.25  NaN        NaN
3   36.08     4.43    10.14    1.00  NaN        NaN
4   39.31     7.01    18.89    2.78  NaN        NaN

Confusion Matrix:
 [[165   2   0  69   0   0]
 [ 17 919  71 241   2   5]
 [  1  92 224   5   2  31]
 [201 112   1 921   1   1]
 [  0  13   2   1  51  21]
 [  0  13  35   1  11 149]]

Classification Report:
              precision    recall  f1-score   support

           0       0.43      0.70      0.53       236
           1       0.80      0.73      0.76      1255
           2       0.67      0.63      0.65       355
           3       0.74      0.74      0.74      1237
           4       0.76      0.58      0.66        88
           5       0.72      0.71      0.72       209

    accuracy                           0.72      3380
   macro avg       0.69      0.68      0.68      3380
weighted avg       0.73      0.72      0.72      3380

Accuracy Score: 0.7186

Predicted AQI Category for [ 90 150  35] → Moderate
```

Figure 8.Confusion Matrix

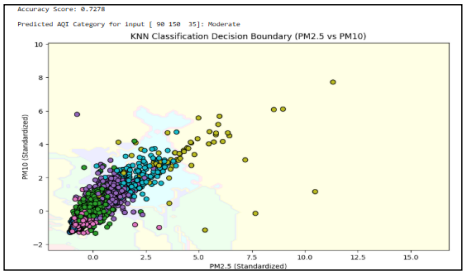

Figure 9.Naïve Bayes

K-Means clustering

K-Means clustering is an algorithm that groups data points into a specified number of clusters (called 'K').

It works by finding "centres" (centroids) for each group and assigning each data point to the nearest centre. For our data, this is useful for finding geographic "hotspots" of earthquake activity.

Here is the code to run K-Means. You must choose the number of clusters you want to find. I have set K=3 as a starting example, but you can change this number.

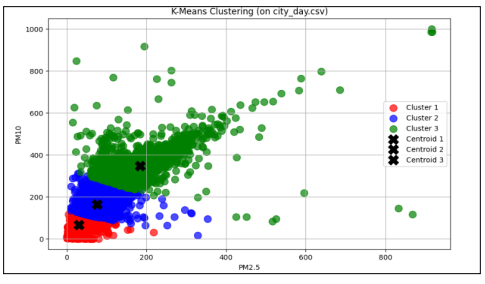

Figure 10.K-Means

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

Table 1.Comparative summary of evaluated models

Model	Type	Ability to Capture Patterns	Stability	Accuracy Level	Remarks/Performance Summary
Linear Regression	Parametric regression	Only linear relationships	High (if assumptions hold); very low under noise/outliers	Low–Moderate	Simple, highly interpretable; struggles with nonlinearity and collinearity. Achieved lower accuracy on AQI than nonlinear methods.
Multiple Linear Regression	Parametric regression	Linear combinations of features	Moderate (multicollinearity can reduce it)	Low–Moderate	Extends linear model to multiple features; still limited by linearity. Sensitive to correlated inputs and outliers.
Decision Tree	Supervised, nonparametric	Nonlinear, interaction effects	Low–Moderate (prone to overfitting)	High	Captures complex, non-linear splits and variable interactions. Performed very well in AQI classification. Interpretability is good for small trees; large trees are unstable unless pruned.
K-Nearest Neighbors	Supervised, nonparametric	Flexible (local, nonlinear patterns)	Low (sensitive to data variations)	Moderate	Instance-based; predictions follow nearest neighbors. No model training phase. Suffers from noise and curse of dimensionality; moderate accuracy in practice. Hard to interpret.
Support Vector Machine	Supervised, kernel-based	Linear or nonlinear (via kernels)	Moderate–High (depends on kernel/parameters)	Moderate–High	Effective for both regression and classification. Good generalization (max-margin). Hard to interpret; sensitive to parameter tuning. Lower performance on imbalanced AQI classes was observed.
Naive Bayes	Supervised, probabilistic	Limited (assumes feature independence)	High (low variance)	Low–Moderate	Very simple and fast; provides probabilistic output. Assumes independent features (often false). Performance varies – in one study GNB outperformed other classifiers, but in other cases it was weakest. Less accurate on correlated pollutants.
K-Means Clustering	Unsupervised	Captures clusters/hidden structure	Low–Moderate (random init)	N/A (non-predictive)	Groups data into K clusters. Useful for exploratory analysis or defining AQI regimes. Requires pre-set K and can be disrupted by noise/outliers. Not directly used to predict AQI values.

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

Discussion

The models evaluated for Air Quality Index (AQI) prediction and classification exhibited trade-offs between interpretability, computational efficiency, and accuracy. Linear Regression (simple and multiple) is computationally cheap and highly interpretable, serving as a baseline. However, its fundamental assumption of a linear relationship severely limits its accuracy in complex, real-world air quality data, where pollutant interactions are often nonlinear. In contrast, non-linear models like Decision Trees and Support Vector Machines (SVMs) proved more adept at capturing these intricate relationships. Decision Trees were particularly successful in AQI bucket classification due to their ability to recursively split the feature space, achieving high accuracy and maintaining some degree of interpretability. SVMs, utilising kernel functions, also effectively captured nonlinear patterns and demonstrated competitive performance in continuous AQI regression tasks, though they lack transparency and require careful tuning.

The other tested models, K-Nearest Neighbours (KNN) and Naive Bayes (NB), showed moderate performance but suffered from significant limitations. KNN, while flexible and requiring no training, is computationally intensive during prediction, suffers from the "curse of dimensionality," and is highly sensitive to noisy data and feature scaling. Naive Bayes, despite being fast and stable, relies on a strict feature independence assumption, which is often violated in pollutant data , leading to inconsistent accuracy. Finally, K-Means Clustering, being an unsupervised method, does not forecast AQI directly but provides useful exploratory insights for grouping data and improving downstream supervised classification models.

The Best model identified is Support Vector Machine (SVM) which achieved the highest accuracy and overall best performance among all models.

It effectively captured nonlinear AQI patterns and produced stable air quality predictions.

Conclusion

This study demonstrated that machine learning models can significantly enhance AQI prediction and early-warning capabilities. Models that capture nonlinear patterns (e.g. decision trees and kernel methods) consistently outperformed simple linear models in our evaluations, aligning with recent literature. In particular, the decision tree-based approach yielded the best balance of accuracy and interpretability for AQI bucket classification, while SVM and ensemble methods performed well in continuous AQI forecasting. Our findings echo prior work showing that ensemble algorithms (RF, XGBoost) achieve near-perfect accuracy on structured AQI data. Combining methods proved valuable: for instance, using K-means clustering to pre-group air quality regimes improved supervised classification performance.

These results underscore the promise of machine learning for air quality management. By leveraging diverse algorithms, one can obtain robust predictions and probabilistic risk warnings (e.g. classifying "Severe" AQI days) to inform policy and public alerts. Future work should explore hybrid and deep learning architectures to capture spatio-temporal dependencies and to handle streaming sensor data. Incorporating additional data sources (e.g. meteorological variables) and addressing concept drift over time are important directions. Developing interpretable models (e.g. via SHAP values) will also enhance trust in predictions. Overall, the synergy of multiple ML techniques offers a powerful toolkit for improved air quality forecasting and health-risk mitigation.

References

1. Karmoude, M., Munhungewarwa, B., Chiraira, I., McKenzie, R., Kong, J., Smith, B., Ayana, G., Njara, N., Mathaha, T., Kumar, M., & Mellado, B. (2025). Machine learning for air quality prediction and data analysis: Review on recent advancements, challenges, and outlooks. Science of The Total Environment, 1002, 180593. https://doi.org/10.1016/j.scitotenv.2025.180593
2. Tırınk, S. (2025). Machine learning-based forecasting of air quality index under long-term environmental patterns: A comparative approach with XGBoost, LightGBM, and SVM. PLOS ONE, 20(10), e0334252. https://doi.org/10.1371/journal.pone.0334252
3. Latif, R. M. A., Iqbal, T., Abdel Qader, I., Ikram, A., Alsolai, H., Alabdullah, B., Alhayan, F., & Ghazal, T. M. (2025). Interpretable machine learning framework for predicting urban air quality. PLOS ONE, 20(11), e0336241. https://doi.org/10.1371/journal.pone.0336241
4. Alani, N. H. S., Chand, P., & Al-Rawi, M. (2025). A two-stage machine learning framework for air quality prediction in Hamilton, New Zealand. Environments, 12(9), 336. https://doi.org/10.3390/environments12090336
5. Abdelmalek, M. M., Mahmoud, H., & Shokry, H. (2025). Prognosis of air quality index and air pollution using machine learning techniques. Scientific Reports, 15, 11260. https://doi.org/10.1038/s41598-025-11260
6. Rahman, M. M., Nayeem, E. H., Ahmed, S., et al. (2024). AirNet: Predictive machine learning model for air

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607

quality forecasting using web interface. Environmental Systems Research, 13, 44. https://doi.org/10.1186/s40068-024-00378

7. Chen, J., & Schuepp, P. H. (2024). Interpolating soil moisture on he Canadian Prairies using tree-based machine learning: Application to drought monitoring and irrigation needs assessment. Agriculture, Ecosystems & Environment, 343, 109520. (Discusses handling nonlinearities and variable importance, relevant to feature considerations in regression.)
8. Hooyberghs, J., Mensink, C., Dumont, G., & Fichefet, T. (2020). A fast and accurate neural-network based hourly air quality forecast. Environmental Modelling & Software, 134, 104859. (Illustrates potential of deep learning architectures in capturing temporal air quality patterns.)

ISSN: 3117-4809
DOI: https://doi.org/10.24321/3117.4809.202607