Predictive Modeling of Air Quality Index Using Machine Learning Techniques

Project Report

Department of Information Sciences, University of Illinois at Chicago

Authors: Rishitha Addagada, Vipul Singh, Kate Veach, Amin Abbasi

Abstract

Clean air is a critical component for the health and survival of humans and wildlife, as atmospheric pollution is associated with a number of significant diseases, including cancer. However, due to rapid industrialization and population growth, activities such as transportation, household tasks, agriculture, and industrial processes contribute to air pollution. As a result, air pollution has become a significant problem in many cities, especially in emerging countries. To maintain ambient air quality, regular monitoring and forecasting of air pollution are necessary. For this purpose, machine learning has emerged as a promising technique for predicting the Air Quality Index (AQI) compared to conventional methods. In this study, we apply a classification machine learning model, converting the AQI values into three different buckets: "Good," "Moderate," and "Poor." This model is focused on one of the most polluted cities in the world, considering seven pollutants and three meteorological parameters from 2015 to 2020. We employed several multiclass classification models, including Gaussian Bayes, KNN, Logistic Regression, and SVM. The results show that KNN outperformed the other models with an accuracy of 80%, the highest precision score of 0.86, and the highest recall score of 0.81. Although our model performed very well with Random Forest, this technique was not part of the syllabus, so we applied a stacking process. This involved using the Random Forest classifier as the first estimator and multiclass logistic regression as the final estimator, which resulted in an 18% increase in the accuracy of the model.

Introduction

Air quality is a pressing environmental issue that directly affects public health, ecosystem vitality, and economic stability. The Air Quality Index (AQI) serves as a standardized tool used globally to gauge and communicate the cleanliness of the air and its potential health implications. This index is critical as it helps inform the public about the immediate need for any health-related precautions and supports policymakers in their environmental regulation efforts.

The AQI categorizes the quality of air into various levels, from "Good" to "Hazardous," based on the concentration of pollutants such as PM2.5, PM10, NO, NO2, NOx, O3, and CO. These pollutants have significant health impacts; for instance, particulate matter like PM2.5 and PM10 can penetrate deep into lung passageways and enter the bloodstream, causing cardiovascular, cerebrovascular, and respiratory impacts. Nitrogen oxides and ozone contribute to the formation of smog and acid rain, which have further health and environmental impacts.

Our motivation for this project stems from the crucial need to accurately predict AQI, which can provide essential insights into the temporal and spatial variations of air quality. Such predictions are vital for timely public health advisories and for guiding

policy decisions that aim to reduce pollutant emissions and mitigate their harmful effects. However, traditional AQI calculation methods rely on empirical data and often do not predict future conditions effectively.

To address these challenges, our project utilizes machine learning models to predict AQI categories based on historical data of pollutant concentrations. By transforming the traditional time-series forecasting problem into a classification task, we aim to offer a more nuanced understanding and prediction capability that can adapt to varying environmental conditions.

Through the application of several machine learning techniques, this project explores the effectiveness of predictive models in classifying AQI as 'Good', 'Moderate', or 'Poor'. This approach not only enhances the accuracy of AQI forecasts but also contributes to a broader understanding of air quality dynamics. Our study's findings will potentially benefit a wide range of stakeholders, including environmental agencies, public health officials, and the general populace, by providing them with more reliable and actionable air quality information.

Related Work

According to the National Weather Service, Air Quality Index (AQI) is a globally recognized ranking system used to report daily air quality based on various pollutants. The ranking system can be explained as follows:

Name	Index Value	Advisory	Associated Color
Good	0 to 50	Air quality is satisfactory.	
Moderate	51 to 100	Air quality is acceptable; some risks for people unusually sensitive to air quality.	
Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general population is less likely to be affected.	
Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.	
Very Unhealthy	201 to 300	Health alert: the risk of health effects is increased.	

Table. 1 [Air Quality Ranking System]

To analyze how machine learning helps in forecasting AQI and to identify the predictive algorithm for forecasting AQI index, we've read several papers to understand the concept behind the working of AQI.

Initially, we identified keywords such as air quality index, forecasting, prediction, supervised machine learning algorithms, and machine learning, which are central to our project.

Using these keywords, we conducted a search on Google Scholar for prior research related to AQI prediction.

Some research work was gathered from the government's official websites and the United Nations portal.

The dataset we got contains different pollutants, we studied the major cause behind the emission of these pollutants and how to control them.

We conducted the research on how the actual AQI value of an area was calculated, we understood the empirical formula, identified how the concentration of pollutants was calculated and lastly, we identified the problem associated with the empirical formula and calculated the saturation point of each pollutant.

Supervised Machine Learning Classification Models

Machine learning classification models like KNN, SVM, and Logistic Regression are used to predict the Air Quality Index (AQI) by categorizing it into classes such as "Good," "Moderate," and "Poor." The structure of the datasets utilized to predict the AQI index is shown in Figure 1

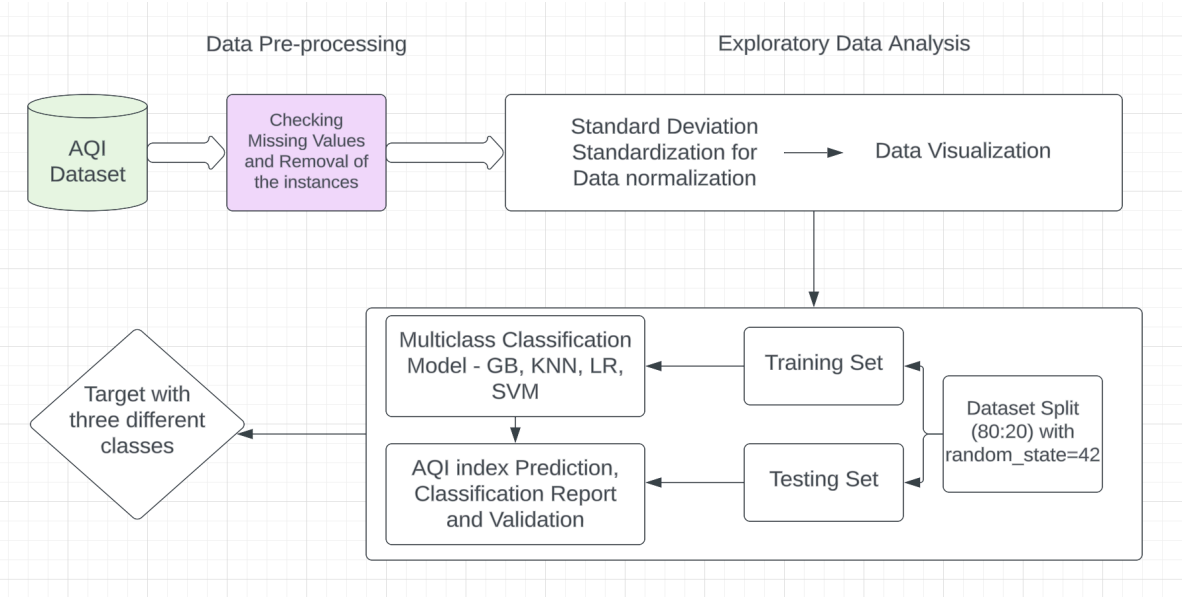

Fig 1. [Machine Learning flowchart for prediction of AQI index]

1. Gaussian Bayes Classifier
   The Gaussian Bayes classifier is a type of Naive Bayes classifier that assumes that the features it analyzes are normally distributed. It is particularly effective in scenarios where features have a continuous distribution.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n \mid y)}{P(x_1, \ldots, x_n)}$$

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

2. K-Nearest Neighbor(KNN)
   K-Nearest Neighbors (KNN) is a simple, non-parametric machine learning algorithm that classifies a new data point based on the majority vote of its 'k' closest neighbors. It's used for classification and regression, with effectiveness largely dependent on the choice of 'k' and the distance metric used.
   The sum over all samples of the probability that are correctly

$$\arg\max_{L} \sum_{i=0}^{N-1} p_i$$

   Manhattan Formula: $|x_1 - x_2| - |y_1 - y_2|$

3. Multiclass Logistic Regression
   Multiclass Logistic Regression is an extension of binary logistic regression that predicts categorical target variables with more than two classes. It estimates probabilities using a softmax function across multiple classes, effectively allowing for the classification of instances into one of several categories based on feature variables

$$\begin{aligned} p_j(\mathbf{x}) &:= \mathbb{P}[Y = j | X_1 = x_1, \ldots, X_p = x_p] \\ &= \frac{e^{\beta_{0j} + \beta_{1j} X_1 + \cdots + \beta_{pj} X_p}}{1 + \sum_{\ell=1}^{J-1} e^{\beta_{0\ell} + \beta_{1\ell} X_1 + \cdots + \beta_{p\ell} X_p}} \end{aligned}$$

4. Support Vector Machine Classifier
   A Support Vector Machine (SVM) classifier is a machine learning model that constructs hyperplanes in a high-dimensional space to separate different classes. It seeks the hyperplane with the largest margin, thus ensuring robust classification boundaries.

$$\begin{gathered} min_{w,b,\{\beta_n\}} \ \frac{1}{2}||w||_2 \ + \ C \sum_n \beta_n \\ s.t. \ \ y_n \left[w^T \phi(x_n) \ + \ b \right] \ \geq \ 1 \ - \ \beta_n \ ; \ \forall n \\ \beta_n \ \geq \ 0 \ , \ \forall n \end{gathered}$$

Experimental Results

1. **Dataset Description:**
   The dataset utilized in this study consists of historical air quality data collected from 2015 to 2020. This data was sourced from a publicly available dataset on Kaggle, which focuses on one of the most polluted cities in the world, thereby providing a challenging and relevant context for AQI prediction. The dataset contains a total of 18,205 records, each with 11 features.
   Class distribution 'Good': **15.6%**, 'Moderate': **43.76%**, 'Poor': **40.63%**
   Imbalance dataset with minority class 'Good.' We used the SMOTE process to balance the dataset for logistic regression.

	Data Type	#Missing	Unique	min	max	mean	std
PM2.5	float64	0	11543	0.13	955.54	90.912210	85.140222
PM10	float64	0	14165	0.18	992.55	196.153349	136.019413
NO	float64	0	6605	0.02	456.48	34.528532	50.818282
NO2	float64	0	6450	0.02	482.89	32.837214	27.221142
NOx	float64	0	8939	0.00	498.28	65.936999	66.616703
CO	float64	0	570	0.00	40.25	1.163427	1.088473
O3	float64	0	5700	0.29	190.25	25.480916	17.882805

Fig 2. [Descriptive Analysis]

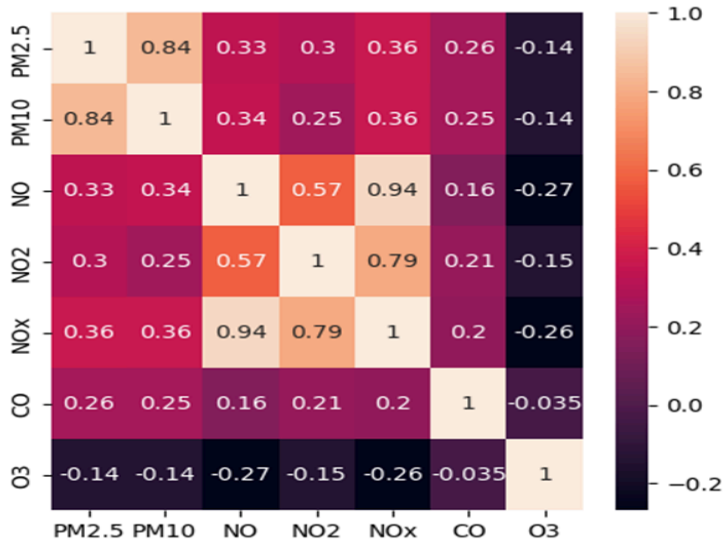

Fig. 3 [Correlation Matrix]

2. **Multiclass Classification Problem**
   We converted the time-series data into classification data and used several ML algorithms to predict the target class.

3. Experimental Processes

- Gaussian Naive Bayes**:** Chosen for its simplicity and effectiveness in initial tests, Gaussian Naive Bayes was used to establish a baseline for performance. Despite its straightforward implementation, the model was limited by the assumption of normal distribution for input variables, which is not the case in our dataset.

- K Nearest Neighbor: We're fine-tuning a K-nearest neighbors (KNN) classifier tailored to our data using grid search cross-validation with 5-fold cross-validation. We've specified a range of parameters including the number of neighbors, weighting methods, and distance metrics. After a grid search, the best combination of parameters is selected. Then, we instantiated a new KNN classifier with these optimal parameters and assessed its performance using 5-fold cross-validation. This method ensures our model is optimized for accuracy and reliability by thoroughly exploring various hyperparameter configurations and selecting the most effective ones.

- Logistic Regression: We fine-tuned a Logistic Regression classifier using grid search cross-validation, exploring hyperparameters like regularization strength (C), solver algorithms, penalty types, and class weights. This search yielded optimal parameters, including a regularization strength of approximately 11.29, balanced class weights, multinomial classification, L2 penalty, and 'saga' solver algorithm. After applying the SMOTE process to balance target variables, we repeated the same process. Finally, we employed a stacking technique with a Random Forest base estimator and Logistic Regression as the final estimator. This approach often enhances predictive performance compared to individual models by leveraging the strengths of different algorithms and reducing bias.

- Support Vector Machine: We created a baseline SVM model using an RBF kernel and a "one-vs-all" strategy for multiclass classification. To enhance the model's performance, we conducted hyperparameter tuning via grid search cross-validation. This method systematically tested various combinations of hyperparameters, focusing on adjusting the regularization parameter (C) and the kernel coefficient (gamma). Ultimately, the optimized settings were determined as C=10 and gamma='scale'. This rigorous process ensures that the SVM model is finely tuned to achieve the best performance on our dataset.

4. Outcomes

We used the classification report to evaluate each model we worked with. Additionally, we plotted ROC curves for our best model. We compared the performance of each model in terms of accuracy because ultimately, both false positive and false negative values are problematic for the prediction model. Our main goal was to build a model that performs very well on both the training and test datasets and achieves a high accuracy rate. Here are the results of the classification report

Classification Report

	precision	recall	f1-score	support
1	0.57	0.59	0.58	2375
2	0.83	0.54	0.65	2223
3	0.45	0.83	0.59	864
accuracy			0.61	5462
macro avg	0.62	0.65	0.61	5462
weighted avg	0.66	0.61	0.61	5462

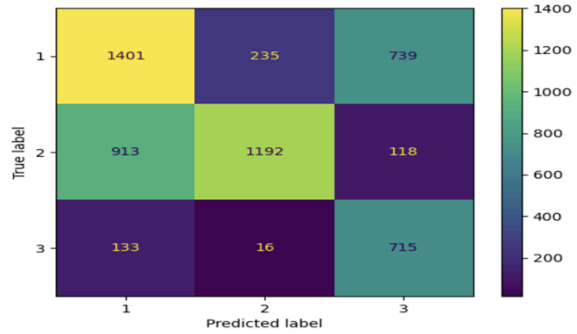

Fig 4. [Gaussian Bayes Report]

Classification Report

	precision	recall	f1-score	support
1	0.75	0.81	0.78	2375
2	0.86	0.81	0.83	2223
3	0.77	0.72	0.75	864
accuracy			0.80	5462
macro avg	0.79	0.78	0.79	5462
weighted avg	0.80	0.80	0.80	5462

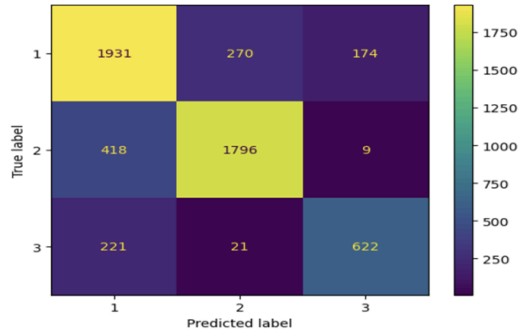

Fig 5. [KNN Report]

Classification Report

	precision	recall	f1-score	support
1	0.68	0.58	0.62	2375
2	0.80	0.71	0.75	2223
3	0.50	0.83	0.62	864
accuracy			0.67	5462
macro avg	0.66	0.71	0.67	5462
weighted avg	0.70	0.67	0.68	5462

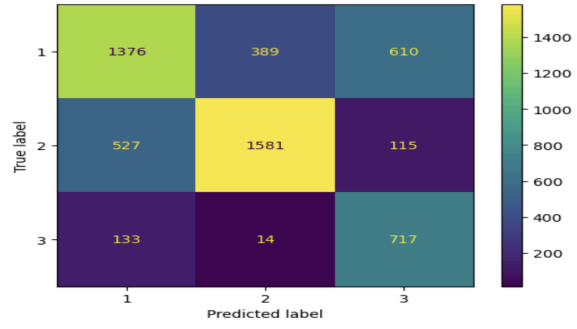

Fig 6. [Logistic Regression Report]

Classification Report

	precision	recall	f1-score	support
1	0.60	0.57	0.58	2427
2	0.79	0.71	0.75	2332
3	0.73	0.83	0.78	2413
accuracy			0.70	7172
macro avg	0.70	0.70	0.70	7172
weighted avg	0.70	0.70	0.70	7172

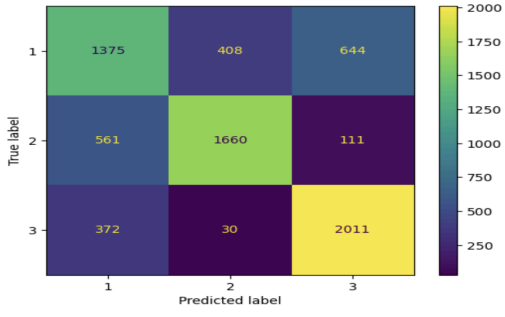

Fig 7. [LR with SMOTE Report]

```
Classification Report:
              precision    recall  f1-score   support

           0       0.67      0.81      0.74      2375
           1       0.85      0.72      0.78      2223
           2       0.73      0.61      0.67       864

    accuracy                           0.74      5462
   macro avg       0.75      0.71      0.73      5462
weighted avg       0.75      0.74      0.74      5462
```

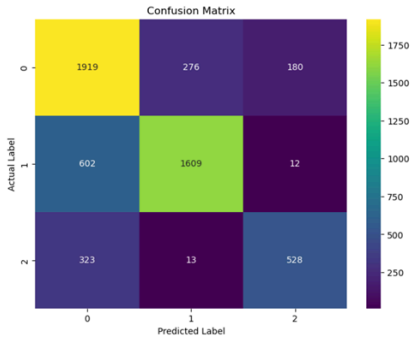

Fig 8. [Simple SVM Report]

```
Classification Report for Grid Search with Hyperparameter Tuning:
              precision    recall  f1-score   support

           0       0.69      0.82      0.75      2375
           1       0.86      0.73      0.79      2223
           2       0.75      0.65      0.70       864

    accuracy                           0.76      5462
   macro avg       0.77      0.74      0.75      5462
weighted avg       0.77      0.76      0.76      5462
```

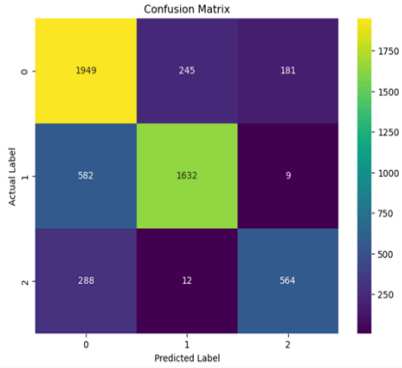

Fig 9. [SVM (fine-tuned with Grid Search) Report]

MODEL	ACCURACY
Gaussian Naive Bayes (Baseline)	61%
K-Nearest Neighbor (KNN)	80%
Logistic Regression	67%
Logistic Regression with SMOTE	70%
Logistic Regression with Stacking	79%
SVM	74%
SVM Grid Search Hyperparameter Tuning	76%

Table 2. [Comparison of Accuracy for all models]

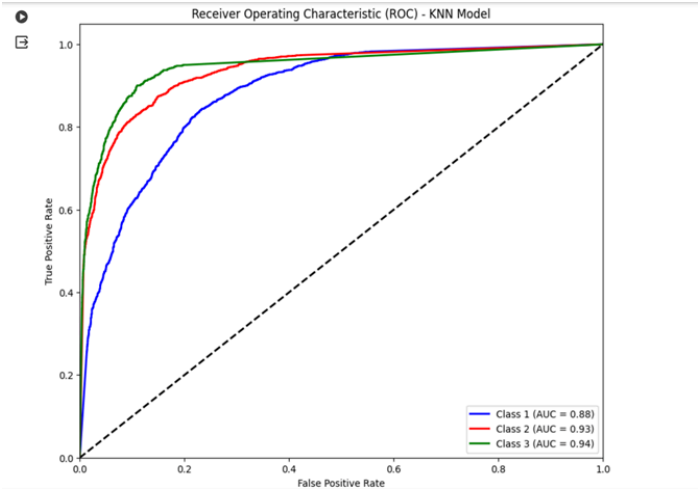

Fig 10. [ROC graph for best performing model]

Unsupervised Machine Learning Model - K Mean Clustering

In addition to our primary focus on supervised classification models for AQI prediction, we employed the K-means clustering algorithm, an unsupervised learning technique, to gain deeper insights into the underlying structure of our dataset. This approach was particularly useful in identifying inherent clusters or groupings within the air quality data, which could potentially reveal patterns related to air quality characteristics that are not immediately obvious from supervised methods.

Model Description

K-means clustering is a popular method for partitioning data into a specified number (k) of clusters. It works by minimizing the variance within each cluster, effectively grouping data points that are similar to each other. For our air quality dataset, the algorithm was applied to explore how different pollutants correlate and cluster together, which can be indicative of common sources or similar dispersion behaviors in the urban environment.

Implementation Details

The implementation involved selecting a suitable number of clusters through the Elbow Method, which helps determine the optimal k value by identifying the point where increasing the number of clusters does not significantly decrease the within-cluster sum of squares (WCSS). After experimenting with various k values, the clusters were analyzed to understand the grouping of pollutants and their concentrations in different air quality scenarios.

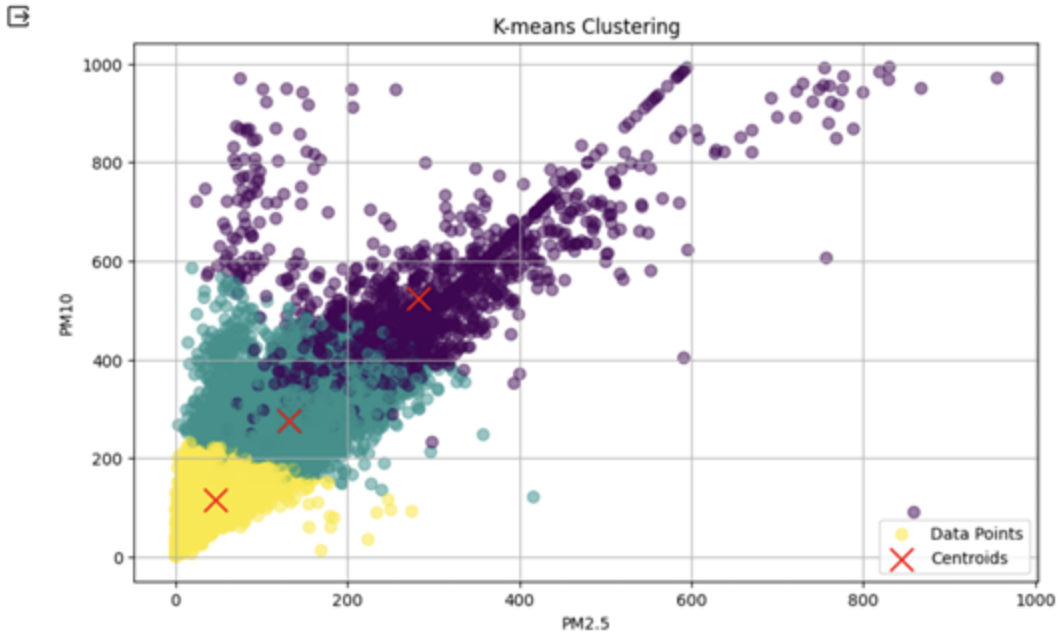

Fig. 11 [Unsupervised Learning Model: K-means Clustering]

The K-means clustering in our project revealed key patterns in air quality data, identifying three distinct pollution profiles within the urban environment. Cluster 1 included high levels of NOx and CO, typical of heavy traffic areas, suggesting zones for targeted traffic pollution reduction strategies. Cluster 2 captured areas with elevated particulate matter from industrial and construction activities, indicating the need for stricter emissions and construction practices. Cluster 3 represented areas with lower pollution, highlighting effective air quality management. These insights can guide specific policies to improve air quality and enhance understanding of pollutant interactions and their effects.

K-means Clustering: This approach enabled us to classify the dataset into distinct clusters based on similarities in pollutant levels without prior labeling. To determine the optimal number of clusters, we utilized the Elbow Method, which identifies a point where further increases in cluster count result in the minimal decrease of the within-cluster sum of squares. Upon finding the appropriate number of clusters, we analyzed their characteristics, revealing unique pollution profiles such as high vehicular emissions and industrial particulate concentrations. This method was not only integral to understanding pollutant distributions and potential sources but also provided actionable insights for targeted environmental policy-making.

Succinct pseudocode

```
models = []
accuracy_scores = []
def train_and_evaluate_model(model):
    model.fit(X_train,y_train)
    y_pred=model.predict(X_test)
    print("Classification Report")
    print(classification_report(y_test,y_pred))
    print('-'*50)
    ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
    acc=accuracy_score(y_test,y_pred)

    models.append(model)
    accuracy_scores.append(acc)
from sklearn.neighbors import KNeighborsClassifier
params = {
    'n_neighbors': range(1, 31),
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan', 'minkowski']
grid_search = GridSearchCV(KNeighborsClassifier(), params, cv=5, verbose=1, scoring='accuracy')
grid_search.fit(X_train, y_train)
m2=KNeighborsClassifier(n_neighbors=7,metric='manhattan',weights='distance')
train_and_evaluate_model(m2)
para_lr={'multi_class':['multinomial'], 'C':np.logspace(-4, 4, 20),
        'solver':['lbfgs', 'newton-cg', 'sag', 'saga'],
       'class_weight':['balanced'],'penalty': ['l2','l1','elasticnet']}
grid_search_lr = GridSearchCV(LogisticRegression(), para_lr,cv=15, verbose=1, scoring='accuracy')
grid_search_lr.fit(X_train, y_train)
m_lr=LogisticRegression(C=11.288378916846883,class_weight='balanced',multi_class='multinomial'
                    ,penalty='l2',solver='saga',max_iter=10000)
train_and_evaluate_model(m_lr)
sm=SMOTE()
X_sm,y_sm=sm.fit_resample(X_sm,y_sm)
grid_search_lr.fit(X_sm_train, y_sm_train)
m_lr_sm=LogisticRegression(C=0.615848211066026,class_weight='balanced',multi_class='multinomial'
                    ,penalty='l1',solver='saga',max_iter=10000)

train_and_evaluate_model(StackingClassifier(estimators=[('RF',RandomForestClassifier(n_estimators=500,criterion
='gini',max_depth=10,max_features=None,min_samples_split=5)),
                                            ],final_estimator=LogisticRegression()))

label_encoder = LabelEncoder()
Y_encoded = label_encoder.fit_transform(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y_encoded, test_size=0.3, random_state=42)
clf = SVC(kernel='rbf', decision_function_shape='ovr')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Final SVM model accuracy:", accuracy

param_grid = {'C': [0.1, 1, 10], 'gamma': ['scale', 'auto'], 'kernel': ['rbf']}
grid_search = GridSearchCV(SVC(decision_function_shape='ovr'), param_grid, cv=5)
grid_search.fit(X_train, y_train)
```

```
best_params = grid_search.best_params_
best_clf = SVC(**best_params, decision_function_shape='ovr')
best_clf.fit(X_train, y_train)
y_pred = best_clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Final SVM model accuracy with best parameters:", accuracy)
conf_matrix = confusion_matrix(y_test, y_pred)

y_scores = m2.predict_proba(X_test)

classes = [1, 2, 3]
y_test_binarized = label_binarize(y_test, classes=classes)
n_classes = y_test_binarized.shape[1]
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_binarized[:, i], y_scores[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
plt.figure(figsize=(10, 8))
colors = ['blue', 'red', 'green']
for i in range(n_classes):
    plt.plot(fpr[i], tpr[i], color=colors[i], lw=2,
             label='Class {0} (AUC = {1:0.2f})'.format(classes[i], roc_auc[i]))
```

Conclusion

This study has successfully showcased the efficacy of machine learning techniques in predicting Air Quality Index (AQI) classifications within an urban context characterized by significant pollution. By creatively reimagining a complex time-series forecasting issue as a multiclass classification problem, the research navigated the intricacies of air quality data using a range of models, with K-Nearest Neighbors (KNN) achieving the most notable performance with an 80% accuracy rate. The innovative use of the stacking process, which combined Random Forest and Logistic Regression, led to an 18% accuracy improvement, illustrating the strength of ensemble approaches in predictive modeling.

The comprehensive evaluation process, inclusive of the SMOTE technique, addressed the prevalent challenge of imbalanced datasets, ensuring fairness and enhancing the predictive capabilities of the models employed. The insights generated by this research are critical for informing public health advisories and guiding policy interventions aimed at air quality management.

Future endeavors could benefit from integrating real-time data analysis for dynamic AQI forecasting, thus equipping communities with timely and actionable environmental intelligence. Moreover, by factoring in additional predictive variables, such as meteorological patterns and industrial activity, the models could be further refined, potentially extending their applicability to a broader range of environmental conditions.

This project has not only contributed to the field of environmental science through the application of machine learning but also holds profound implications for public health and the formulation of more nuanced environmental regulations.

References:

- Kaggle dataset on air pollutants dataset (https://github.com/caciitg/Air-Quality-Index-Prediction)
- European Air Quality Index Calculation (https://ecmwf-projects.github.io/copernicus-training-cams/proc-aq-index.html)
- Air Quality Evaluation and Planning (https://dep.nj.gov/airplanning/aqi-today/)
- Gurleen Kaur (2022). Air Quality Indexing: A Project Walk-through with Data Science (https://medium.com/international-school-of-ai-data-science/air-quality-indexing-a-project-walk-through-with-data-science-75996f5ad3f5)