Final Project: Air Pollution and Respiratory Disease: Modeling Asthma and COPD Prevalence Using U.S. County-Level Data

Enoghayin Imasuen, Jan Tobias Boehnke, Megha Sharma

STAT 109: Introduction to Statistical Modeling

Bharatendra Rai

May 2024

1 Research Question and Motivation

Air pollution is a well-documented environmental risk factor for numerous respiratory and cardiovascular diseases (U.S. Environmental Protection Agency [EPA], 2021). Exposure to air pollutants such as fine particulate matter (PM2.5), ozone (O_3), volatile organic compounds (formaldehyde, acetaldehyde), and diesel particulate matter (DPM) has been associated with both the development and exacerbation of asthma and chronic obstructive pulmonary disease (COPD), due to their roles in triggering airway inflammation, oxidative stress, and impaired pulmonary function (EPA, 2019). Public health surveillance data show considerable variation in disease prevalence and incidence across U.S. counties, raising the question of whether this variability can be partially explained by differences in local air quality (Centers for Disease Control and Prevention [CDC], 2021).

The purpose of this study is to examine whether the concentration of specific air pollutants is associated with the prevalence of asthma and the incidence of chronic obstructive pulmonary disease (COPD) of each county level in the United States in 2021. This analysis utilizes datasets from the National Ambient Air Quality Standards (NAAQS) and the Centers for Disease Control and Prevention (CDC, 2021; EPA, 2021) that report annual air pollution levels and age-adjusted disease statistics for adult populations. While numerous toxicological studies have demonstrated pollutant effects using rat models or respiratory disease correlation to pollutants in other countries, there remains a gap in large-scale, human-centered analyses at an expansive level within the United States. Therefore, this study helps fulfill the need for comprehensive, population-based studies that evaluate the real-world associations between air quality and respiratory disease prevalence across the United States' diverse geographic regions. To uncover our research, we applied exploratory data analysis and machine learning models to assess which pollutants are most strongly associated with increased disease prevalence.

Understanding these associations has practical implications for environmental health policy and resource allocation. If air pollutants are shown to be significant predictors of disease prevalence, interventions targeting air quality improvement could help mitigate the burden of respiratory illnesses in high-risk communities (CDC, 2021a; CDC, 2021b; EPA, 2021). These associations further prove the necessity to be explored with increasing urbanization and climate-driven air quality changes. Increasing wildfires in particular, have significantly contributed to air pollution through its release of PM2.5. For instance, one study reported that "wildfires in the western US contributed 23% of surface PM2.5 during the 2020 wildfire season," (Li et al., 2023). This finding illustrates the significant role of environmental events in shaping air quality trends and highlights the importance of quantifying their impact on public health through empirical analysis.

2 Hypothesis

We evaluate the relationship between air pollution and two health outcomes using binary classification based on epidemiologically informed thresholds.

Asthma:

Null Hypothesis (H_0): There is no association between volatile air pollutant concentrations (PM2.5, O_3, formaldehyde, acetaldehyde, DPM) and high asthma prevalence among adults.

Alternative Hypothesis (H_1): Higher volatile air pollutant concentrations are significantly associated with high asthma prevalence among adults.

COPD:

Null Hypothesis (H_0): There is no association between volatile air pollutant concentrations (PM2.5, O_3, formaldehyde, acetaldehyde, DPM) and high COPD prevalence amongst adults.

Alternative Hypothesis (H_1): Higher volatile air pollutant concentrations are significantly associated with high COPD prevalence amongst adults.

3 Data & Pre-processing

3 Data

We began by importing and cleaning four core datasets: one containing county-level Air Quality Index (AQI) values across pollutants, one with CDC-reported prevalence data for asthma, another for COPD, and a fourth CDC dataset that reported the concentrations of various air toxins by county. For our final dataframe, we needed both the disease prevalence and the concentration or AQI values for each pollutant by county, which required merging these datasets.

A major challenge was that the asthma and COPD datasets (df_a and df_c) and the pollutant concentration dataset (df_pol) used the County FIPS code as a geographic identifier, while the AQI dataset used the CSBA code. Although both are geographic identifiers, they represent different regional groupings and cannot be directly merged. Moreover, the borders of these identifiers overlap inconsistently. To reconcile them, we used a mapping dataset provided by the National Bureau of Economic Research that links CSBA codes to FIPS county codes. After performing this mapping and merging on FIPS codes, we aggregated AQI readings by county and year.

It is important to note that the final AQI column in the merged dataframe includes both AQI values and pollutant concentrations. While the EPA provides a formula for converting concentrations to AQI, it requires the lower and upper bounds of the concentration breakpoints, which are only available for certain pollutants. Since the relationship is linear and we lacked complete conversion intervals, we chose to retain and work with the raw concentration values, which did not affect their relevance in our models.

We then classified pollutant levels based on EPA-provided thresholds above which concentrations are considered harmful. Using these thresholds, we categorized pollutant exposure into “High” and “Good” classes and compared the distribution of disease prevalence across these groups. While the distributions largely overlapped, some differences were visually observable.

For the classification model, we created binary variables (0 and 1) to indicate whether a county's prevalence of asthma or COPD was above or below the national median. This binary target allowed us to predict whether a given county had a high or normal/low prevalence based on environmental conditions. Our final classification targets labeled counties with above-average prevalence as high-risk. We used two primary datasets:

- Air Quality Data: County-level Air Quality Index (AQI) and pollutant concentration data for a range of pollutants, including PM2.5, ozone, formaldehyde, diesel particulate matter, and several volatile organic compounds (VOCs).

- Health Data: CDC's county-level estimates for age-adjusted asthma and COPD prevalence in 2021.

After merging datasets on State and County, we handled missing values using mean imputation and normalized prevalence values to a [0,1] scale for regression tasks. Categories for AQI (e.g., "Good", "High") were used to stratify prevalence distributions in the exploratory phase.

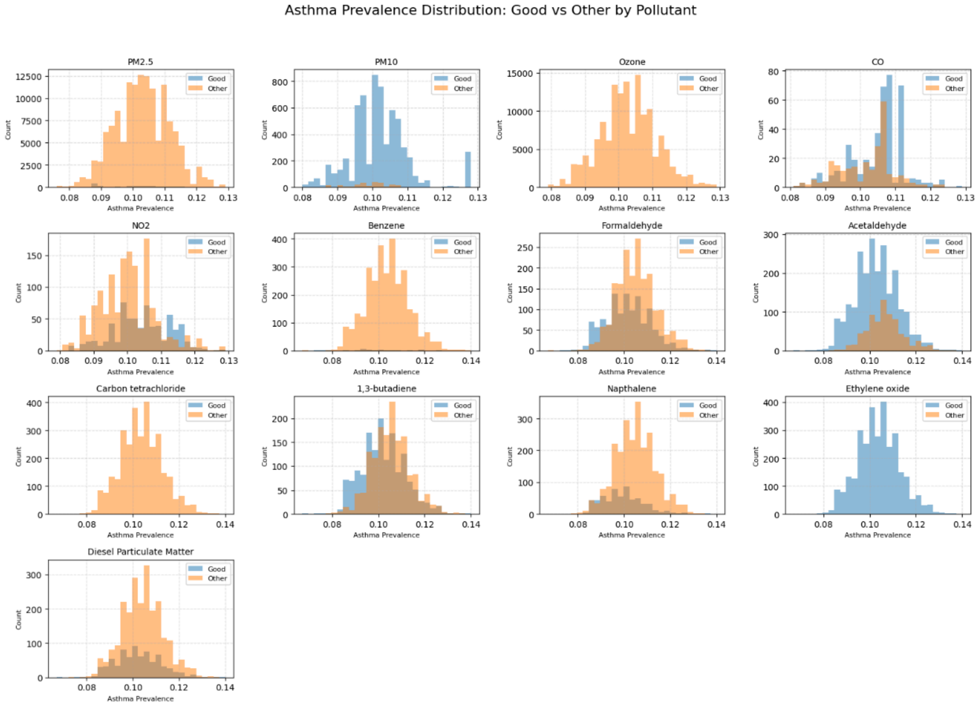


Figure 1. Histogram of asthma prevalence across counties grouped by "Good" vs "Other"

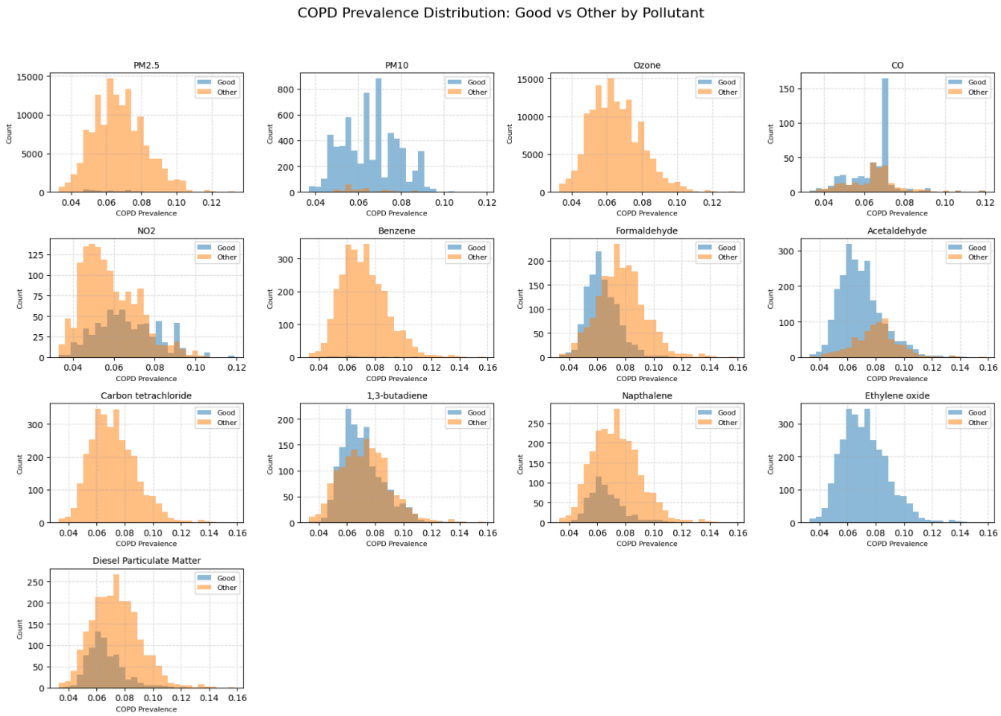


Figure 2. Histogram of COPD prevalence across counties grouped by "Good" vs "Other" AQI categories

Histograms were generated to compare asthma and COPD prevalence across AQI categories ("Good" vs "Other") for each pollutant. These visualizations revealed several key patterns. For many pollutants, including PM2.5, Ozone, Formaldehyde, and Acetaldehyde, counties with non-"Good" AQI levels showed noticeably higher disease prevalence. The differences were most visually pronounced for PM10, Acetaldehyde, and CO in the asthma plots and for Formaldehyde, Acetaldehyde, and CO in the COPD plots. These patterns were largely consistent across both asthma and COPD, suggesting shared environmental triggers.

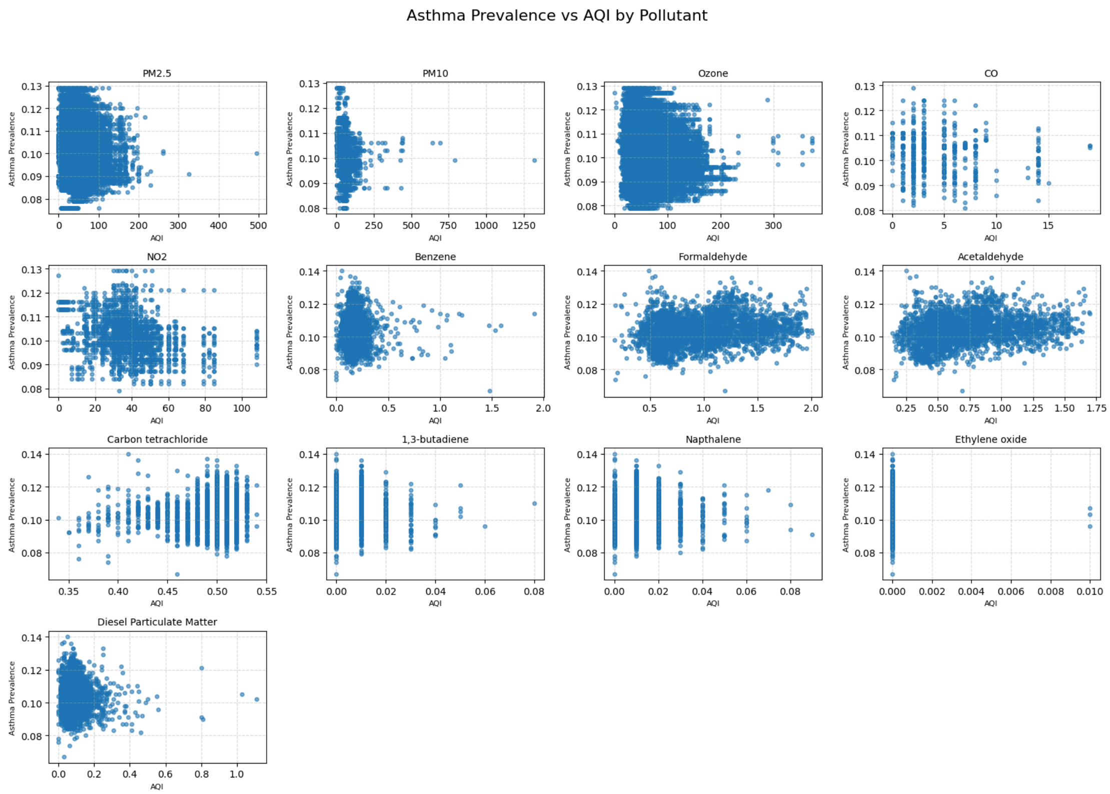


Figure 3. Scatter plots of asthma prevalence vs Air Quality Index (AQI) by pollutant

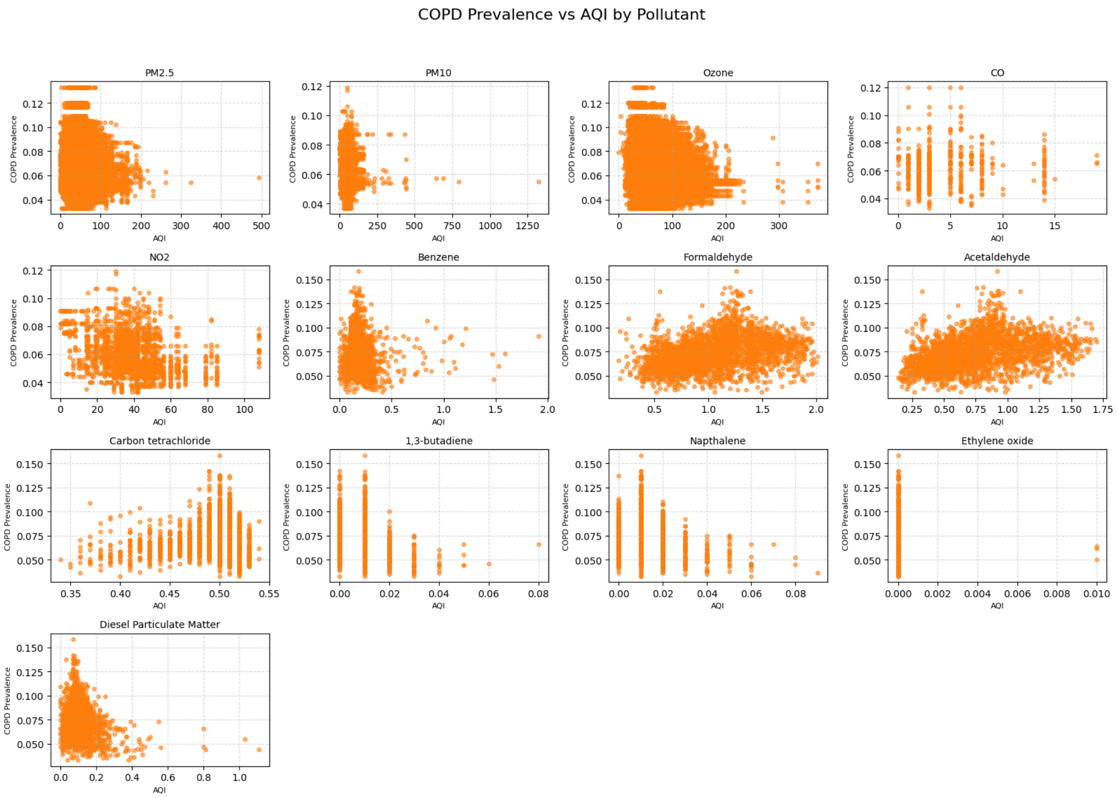


Figure 4. Scatter plots of COPD prevalence vs Air Quality Index (AQI) by pollutant

Scatter plots of asthma and COPD prevalence against AQI for each pollutant revealed several notable patterns (see Figures 3 and 4). Formaldehyde and acetaldehyde displayed clear positive associations with both asthma and COPD prevalence, supporting their importance in the classification model. For carbon tetrachloride, a similarly strong positive trend was observed, particularly in relation to COPD. PM2.5, PM10, and ozone showed weaker or non-linear relationships, with most counties exhibiting moderate prevalence regardless of AQI levels suggesting possible threshold effects or underlying confounding factors. In contrast, diesel particulate matter (DPM) demonstrated an unexpected inverse relationship with both conditions, potentially reflecting regional variation or demographic covariates. These visual patterns provide early evidence of pollutant-specific health risks.

4 Modeling

Classification

We applied a Random Forest Classifier using the formula sklearn.ensemble.RandomForestClassifier with class_weight='balanced' to handle class imbalance. Training and test splits were created using train_test_split, and performance was evaluated using precision, recall, F1-score, and ROC AUC metrics. Feature importance was extracted from the trained forest to identify top predictors for asthma and COPD. We trained a Random Forest Classifier to identify which pollutants most strongly predicted high vs low disease prevalence (split at the median). The top predictors for each condition were ranked by feature importance.

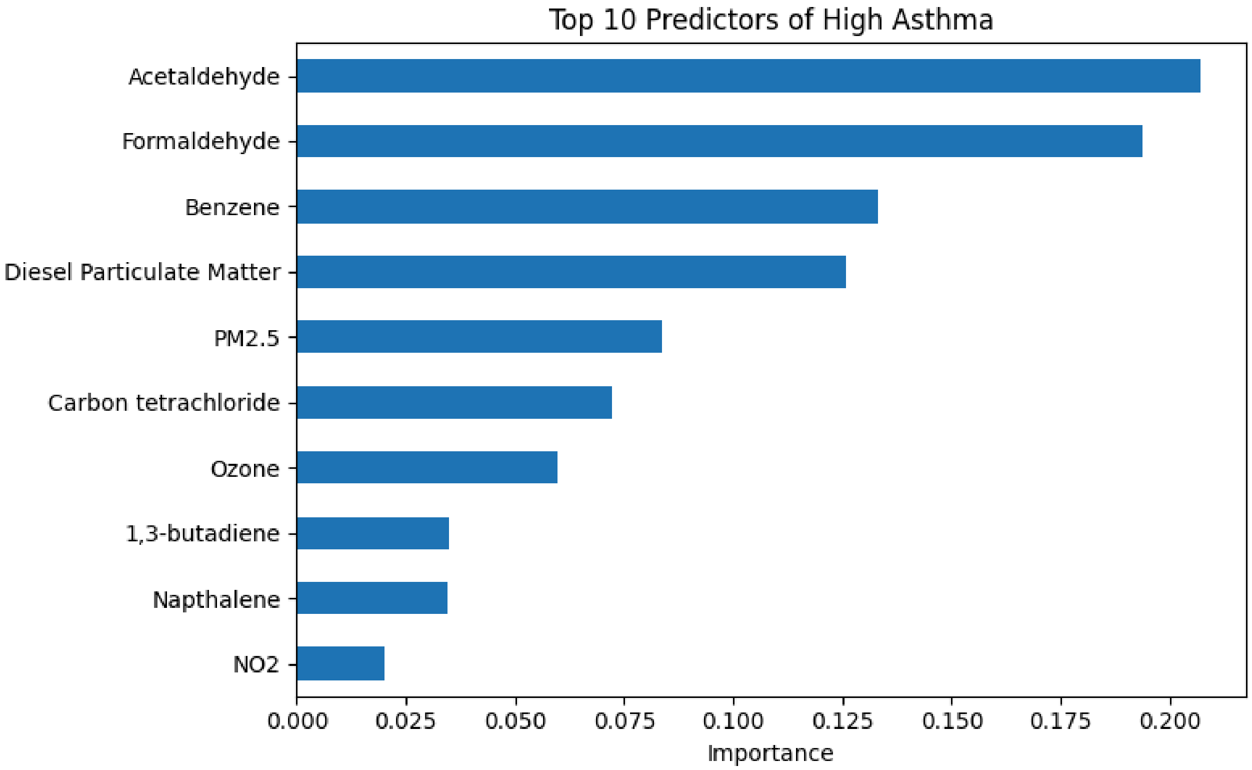


Figure 5. Feature importance plot from Random Forest classification model for asthma prevalence.

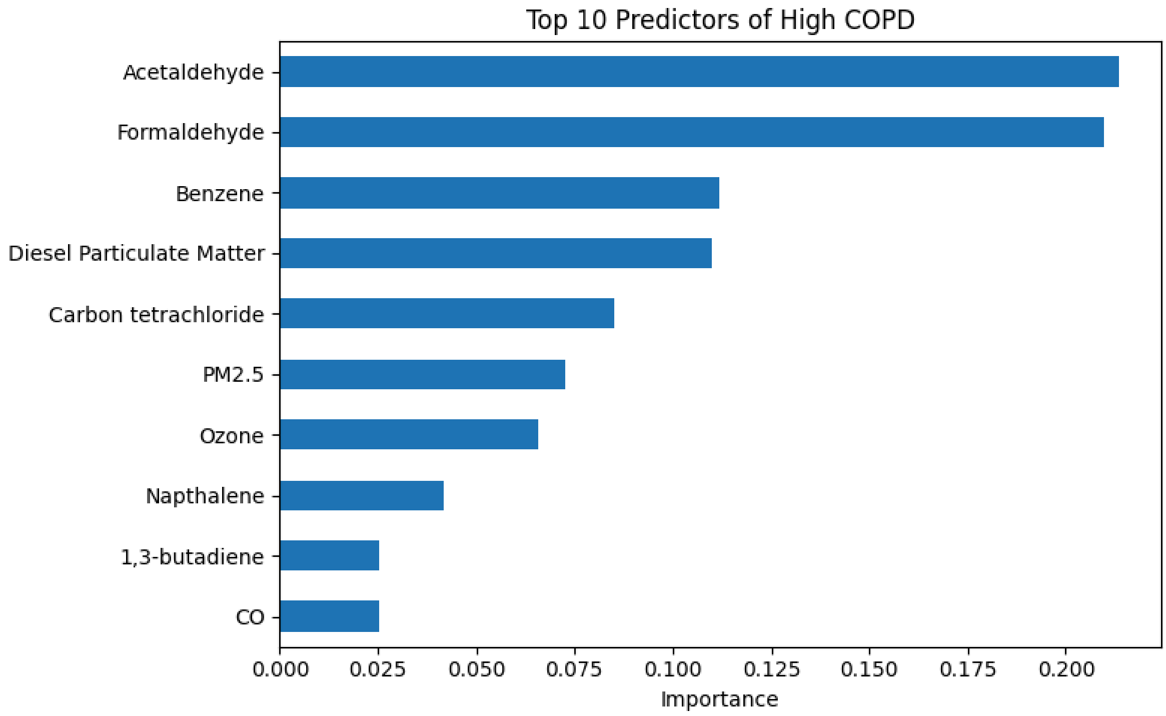


Figure 6. Feature importance plot from Random Forest classification model for COPD prevalence.

Classification Results

To identify counties with elevated asthma and COPD prevalence, we trained Random Forest classifiers using pollutant AQI and concentration values. The classifiers performed well for both conditions, with balanced precision and recall across low-risk and high-risk categories. The asthma classifier achieved an accuracy of 76%, with precision scores of 0.78 for low-risk and 0.75 for high-risk counties. Recall was slightly higher for high-risk areas (0.80), suggesting the model is slightly more sensitive to identifying high-prevalence regions. The top predictors based on feature importance (see Figure 5) were acetaldehyde, formaldehyde, benzene, and diesel particulate matter, all of which have established links to airway inflammation.

The COPD model demonstrated even stronger performance, with an overall accuracy of 78%. It achieved a high precision of 0.81 and recall of 0.81 for high-risk counties, making it well-suited for identifying areas with elevated COPD prevalence. Leading predictors (see Figure 6) included acetaldehyde, formaldehyde, benzene, and carbon tetrachloride. Notably, these predictors overlapped significantly with those identified in the asthma model, pointing to shared pollutant risk factors across respiratory conditions. Together, these classification models highlight the predictive utility of specific volatile organic compounds (VOCs) and fine particulates in assessing public health risks related to respiratory illness.

5 Results

Linear Regression

To estimate continuous prevalence values, we trained a linear regression model using sklearn.linear_model.LinearRegression, applying a sigmoid transformation to constrain predictions between 0 and 1. All visualizations and diagnostics were conducted using Seaborn and Matplotlib. Despite normalization and rescaling steps, both models demonstrated very poor predictive performance. The asthma regression model yielded a mean squared error of 0.0003 and an R² score of -2.5267. While the low MSE indicates tightly clustered predictions, the strongly negative R² shows that the model failed to capture variation in the data performing worse than a simple mean-based prediction. Predicted values were concentrated around the average and did not reflect the true range of asthma prevalence observed across counties. The COPD regression model performed even worse, with a mean squared error of 0.0013 and an R² score of -4.1079. This extremely poor fit confirms that linear regression is ill-equipped to model the complex and likely nonlinear relationships between air pollutant exposure and COPD prevalence.

These results suggest that linear regression is a poor modeling choice in this context. The relationships between environmental pollutants and chronic disease prevalence are likely nonlinear, may involve interaction effects, and may be influenced by latent confounding variables such as demographics or regional health disparities.

Logistic Regression Results (Asthma and COPD)

We additionally trained a logistic regression classifier to predict high asthma prevalence based on AQI & concentration threshold values for the top pollutants. The model achieved an accuracy of 69%, with a ROC AUC of 0.740, indicating moderate predictive power. The confusion matrix (Figure 7) shows balanced classification performance across low and high-risk counties, with slightly better precision for high-risk identification (precision = 0.72).

The most influential predictors in the model (Figure 8) were acetaldehyde, formaldehyde, and diesel particulate matter, which aligns with the feature rankings from the Random Forest model. Interestingly, while acetaldehyde had a strong positive coefficient, formaldehyde had a negative one, suggesting possible multicollinearity or contrasting regional effects. These coefficients provide a more interpretable view into how specific pollutants increase or decrease the odds of high asthma prevalence.

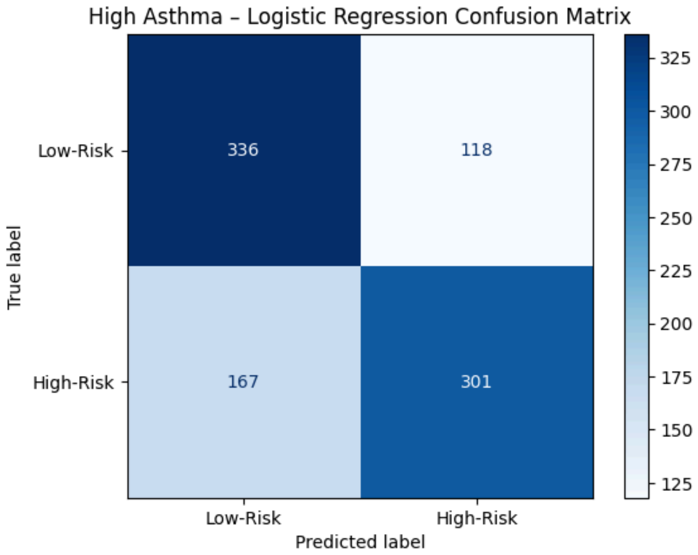


Figure 7. High asthma logistic regression confusion matrix.

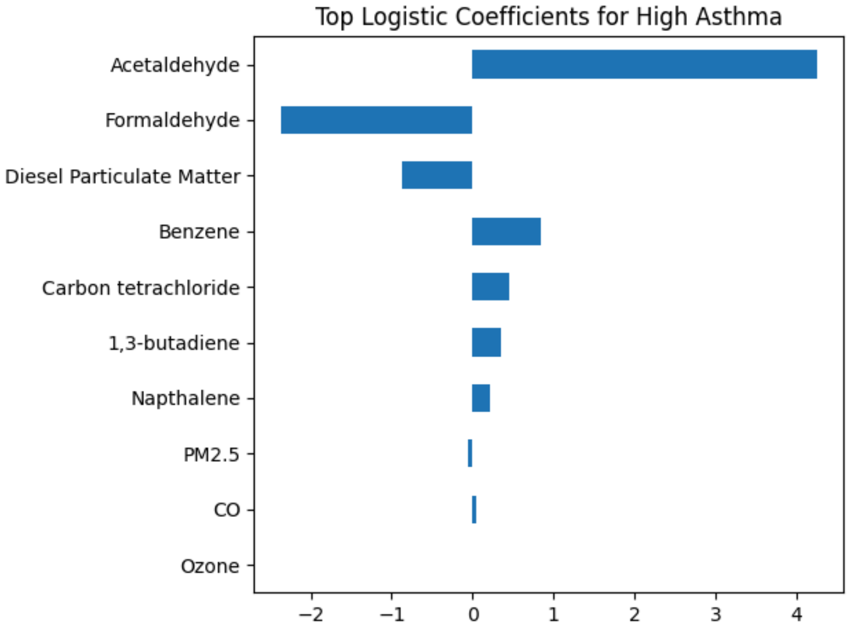


Figure 8. Top logistic coefficients for high asthma.

We also trained a logistic regression model to predict high COPD prevalence using AQI values for key pollutants. The model achieved an accuracy of 70% and a ROC AUC of 0.755, suggesting modest discriminative power. As shown in the confusion matrix (Figure 9), the model classified high-risk counties with reasonably high recall (0.72) and precision (0.76), demonstrating slightly better performance in identifying high-risk regions compared to low-risk ones. The most impactful predictors (Figure 10) included diesel particulate matter, acetaldehyde, and formaldehyde closely matching the top-ranked features in the Random Forest model. Diesel particulate matter had the strongest negative coefficient, possibly reflecting complex interactions between pollutant exposure and regional factors like smoking rates or occupational risk. Both acetaldehyde and formaldehyde exhibited positive coefficients, suggesting that higher concentrations are associated with greater odds of elevated COPD prevalence. These results reinforce the value of logistic regression for interpreting pollutant-specific contributions to disease risk.

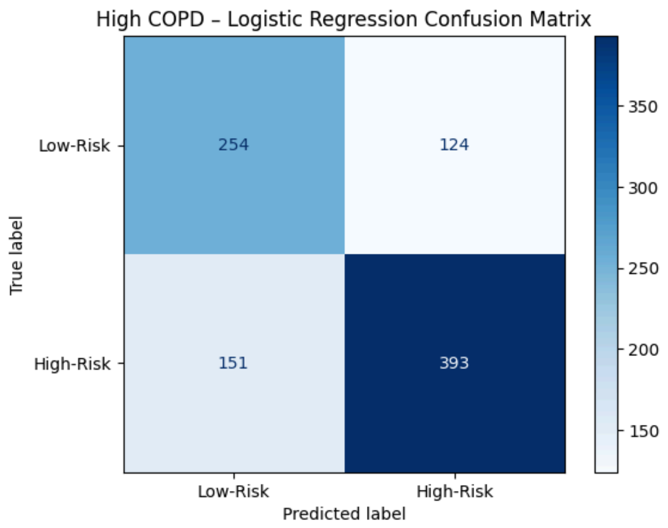


Figure 9. High COPD logistic regression confusion matrix.

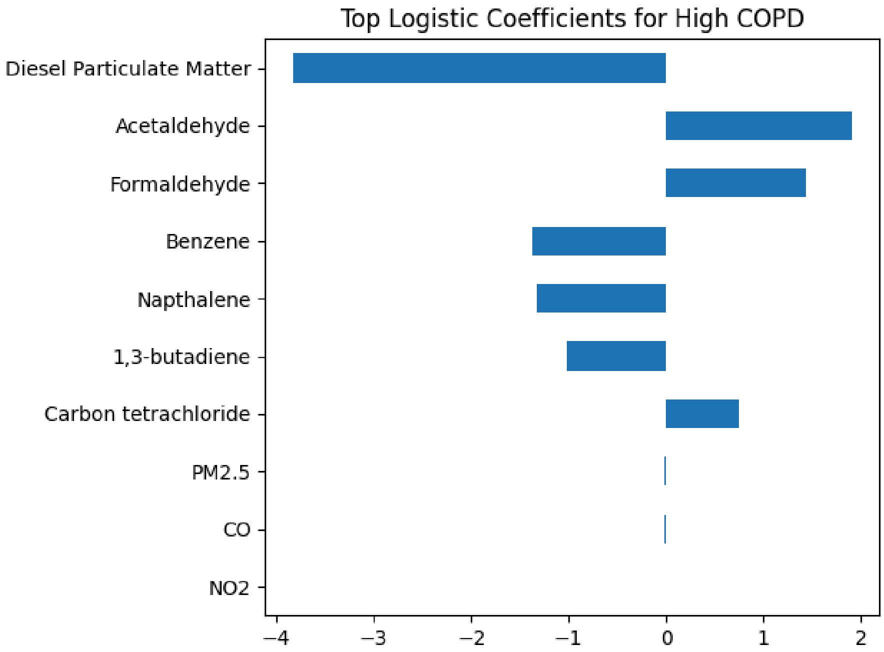


Figure 10. Top logistic coefficients for high COPD.

6 Discussion

This study investigated the association between ambient air pollution and the prevalence of asthma and COPD at the county level in the United States. Through a combination of exploratory visualizations and machine learning models, we found consistent evidence that specific pollutants, particularly volatile organic compounds (VOCs) like acetaldehyde and formaldehyde, are strongly associated with increased respiratory disease risk. Random Forest classification models provided the strongest results, achieving accuracies of 76% for asthma and 78% for COPD. Feature importance scores highlighted acetaldehyde, formaldehyde, diesel particulate matter, and benzene as key predictors for both conditions. These findings are consistent with toxicological research linking these pollutants to airway inflammation and pulmonary dysfunction, supporting their potential use in public health surveillance.

Logistic regression models, although slightly less accurate, offered the advantage of interpretability. For both asthma and COPD, the most influential coefficients matched those found in the Random Forest models. Diesel particulate matter showed both positive and negative associations depending on the outcome, suggesting complex interactions with regional factors such as smoking rates, occupation types, or demographic variables.

Linear regression models performed poorly. The asthma model had an R^2 of -2.53, and the COPD model had an even lower R^2 of -4.11. These negative values indicate that the models explained less variation than a simple mean-based prediction. Residual plots confirmed that the models tended to predict average values without capturing the true spread of prevalence across counties. This reinforces the idea that the relationship between air pollutants and disease prevalence is likely nonlinear and influenced by interacting or confounding variables.

Overall, this project demonstrates that machine learning classification techniques, particularly ensemble-based models, can successfully identify environmental predictors of respiratory health outcomes across geographic regions. However, the limitations of linear approaches and observed variations in pollutant effects suggest a need for more sophisticated models and richer data that include demographic and temporal context. It’s concerning that we can see a correlation with common pollutants and respiratory diseases across the country. This data can help us take steps forward in the right direction and pay attention to air quality.

7 Conclusion

This analysis confirms that ambient air pollutants, especially volatile organic compounds and fine particulate matter, are significant predictors of asthma and COPD prevalence across U.S. counties. Random Forest models provided strong performance in identifying high-risk regions, outperforming linear models in both accuracy and robustness. The overlap in key predictors for asthma and COPD suggests these chronic conditions may be influenced by similar environmental exposures. At the same time, the limited success of linear regression models highlights the importance of using models that can capture complex, nonlinear relationships in environmental health data. Logistic regression, while moderately accurate, offered insights into the direction and strength of individual pollutant effects. As air pollution patterns become more dynamic due to climate change and urban growth, machine learning tools can play a valuable role in supporting public health policy. The results of this study highlight the potential for predictive modeling to guide interventions, allocate resources more effectively, and help reduce the burden of respiratory illness in affected communities.

8 Challenges and Limitations

This project faced several limitations that may have influenced the results. One of the main challenges was aligning datasets that used different geographic identifiers, which required additional steps to map and merge county-level records accurately. The data also came in different dimensions and formats, making preprocessing more complex. Our analysis relied on annual average AQI and pollutant concentration data, which may obscure important temporal variation. Future studies should incorporate daily exceedance rates or seasonal trends to better capture short-term pollution spikes as this data was from 2021. Another limitation was the absence of key demographic and behavioral covariates, such as smoking rates, income levels, and healthcare access, all of which are known to influence respiratory health. Including these factors could improve model accuracy and offer a more holistic understanding of disease drivers.

9 Future Work

Future research should adopt a more dynamic modeling approach that integrates longitudinal data to assess changes in pollutant exposure and disease prevalence over time. Incorporating social determinants of health, such as racial disparities, urban density, and healthcare accessibility, would allow for more equity-focused predictions. Additionally, expanding the geographic and temporal scope of the dataset (e.g., multi-year or seasonal trends) could improve generalizability. Moreover, future scholars could also explore the use of neural networks to model potentially complex, non-linear relationships between pollutant exposure and respiratory health outcomes. While these models require careful tuning and are less interpretable than other methods, they may offer improved predictive performance in capturing intricate interactions among environmental variables. Finally, applying interpretable machine learning methods, such as SHAP (shapley additive explanations) values or penalized regression models, may provide greater clarity into how specific pollutants interact with contextual factors to influence respiratory health outcomes. With regard to the logistic regression model, we could work with methods like regularization (L1 or L2) to prevent overfitting.

References

Centers for Disease Control and Prevention. (2021). PLACES: County data 2021 release. U.S. Department of Health and Human Services. https://chronicdata.cdc.gov/500-Cities-Places/PLACES-County-Data-2021-release/yb82-3vrs

Centers for Disease Control and Prevention. (2021a). National Environmental Public Health Tracking Network. https://www.cdc.gov/ephtracking

Centers for Disease Control and Prevention. (2021b). PLACES: Local data for better health. https://www.cdc.gov/places/

Li, Y., Tong, D., Ma, S., Freitas, S. R., Ahmadov, R., Sofiev, M., Zhang, X., Kondragunta, S., Kahn, R., Tang, Y., Baker, B., Campbell, P., Saylor, R., Grell, G., & Li, F. (2023). Wildfire smoke contributed 23% of surface PM2.5 in the western United States during 2020. npj Climate and Atmospheric Science, 6(1), 1–9. https://www.nature.com/articles/s41612-023-00444-9

U.S. Environmental Protection Agency. (2019). Integrated science assessment (ISA) for particulate matter (EPA/600/R-19/188). https://cfpub.epa.gov/ncea/isa/recordisplay.cfm?deid=347534

U.S. Environmental Protection Agency. (2021). Our nation's air: Status and trends through 2021. https://gispub.epa.gov/air/trendsreport/2021/

U.S. Environmental Protection Agency. (2020). National ambient air quality standards (NAAQS). https://www.epa.gov/naaqs

In [89]:

```
```

In [90]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import (
    classification_report,
    roc_auc_score,
    confusion_matrix,
    ConfusionMatrixDisplay,
    mean_squared_error,
    r2_score
)
```

In [91]:

```
#loading disease (response variable datasets)
df_a = pd.read_csv('asthma.csv')
df_c = pd.read_csv('copd.csv')

#changing percent values in floats
df_a['Value'] = df_a['Value'].str.replace('%', '', regex=False).astype(float
df_c['Value'] = df_c['Value'].str.replace('%', '', regex=False).astype(float

#filtering out only the for us important columns
filter = ["State", "CountyFIPS", "County",  "Value"]


df_c = df_c[filter]
df_a = df_a[filter]
df_a["Value Asthma"] = df_a["Value"]
df_c["Value COPD"] = df_c["Value"]


#
df_aqi = pd.read_csv("daily_aqi_by_cbsa_2024.csv")
df_pol = pd.read_csv("data_121847.csv", encoding="ISO-8859-1")



#mapping
df_map = pd.read_csv("cbsa2fipsxw.csv")
df_map["fipsstatecode"]   = df_map["fipsstatecode"].astype(int)
df_map["fipscountycode"]  = df_map["fipscountycode"].astype(int)

df_map["StateFIPS"]   = df_map["fipsstatecode"].astype(str).str.zfill(2)
df_map["CountyFIPS"]  = df_map["fipscountycode"].astype(str).str.zfill(3)

df_map["CountyFIPS"] = df_map["StateFIPS"] + df_map["CountyFIPS"]
```

Here is a quick glimpse of the different datasets:

In [92]:

```
df_a
```

Out[92]:

	State	CountyFIPS	County	Value	Value Asthma
0	Alabama	1001	Autauga	0.102	0.102
1	Alabama	1003	Baldwin	0.099	0.099
2	Alabama	1005	Barbour	0.114	0.114
3	Alabama	1007	Bibb	0.104	0.104
4	Alabama	1009	Blount	0.101	0.101
...	...	...	...	...	...
3071	Wyoming	56037	Sweetwater	0.095	0.095
3072	Wyoming	56039	Teton	0.087	0.087
3073	Wyoming	56041	Uinta	0.095	0.095
3074	Wyoming	56043	Washakie	0.096	0.096
3075	Wyoming	56045	Weston	0.097	0.097

3076 rows × 5 columns

In [93]:

```
df_aqi.head()
```

Out[93]:

	CBSA	CBSA Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Aberdeen, SD	10100	2024-01-01	42	Good	PM2.5	46-013-0004	1
1	Aberdeen, SD	10100	2024-01-02	24	Good	PM2.5	46-013-0004	1
2	Aberdeen, SD	10100	2024-01-03	13	Good	PM2.5	46-013-0004	1
3	Aberdeen, SD	10100	2024-01-04	26	Good	PM2.5	46-013-0004	1
4	Aberdeen, SD	10100	2024-01-05	53	Moderate	PM2.5	46-013-0004	1

In [94]:

```
df_pol.head()
```

Out[94]:

	StateFIPS	State	CountyFIPS	County	Year	Value	Data Comment	Unnamed: 7	Polluta
0	1	Alabama	1001	Autauga	2019	0.15	NaN	NaN	Pollutar Benzer
1	1	Alabama	1001	Autauga	2019	1.84	NaN	NaN	Pollutar Formaldehyc
2	1	Alabama	1001	Autauga	2019	1.49	NaN	NaN	Pollutar Acetaldehyc
3	1	Alabama	1001	Autauga	2019	0.51	NaN	NaN	Pollutar Carbo tetrachloric
4	1	Alabama	1001	Autauga	2019	0.01	NaN	NaN	Pollutant: 1,: butadier

```
In [95]: df_map.head()
```

Out[95]:

	cbsacode	metropolitandivisioncode	csacode	cbsatitle	metropolitanmicropolitanstatis
0	33860	NaN	388.0	Montgomery, AL	Metropolitan Statistical Area
1	19300	NaN	380.0	Daphne-Fairhope-Foley, AL	Metropolitan Statistical Area
2	21640	NaN	NaN	Eufaula, AL-GA	Micropolitan Statistical Area
3	13820	NaN	142.0	Birmingham, AL	Metropolitan Statistical Area
4	13820	NaN	142.0	Birmingham, AL	Metropolitan Statistical Area

```
In [95]:
```

In [96]:

```
df_map["fipsstatecode"]  = df_map["fipsstatecode"].astype(int)
df_map["fipscountycode"] = df_map["fipscountycode"].astype(int)
df_map["StateFIPS"]      = df_map["fipsstatecode"].astype(str).str.zfill(2)
df_map["CountyFIPS"]     = df_map["fipscountycode"].astype(str).str.zfill(3)
df_map["CountyFIPS"]     = df_map["StateFIPS"] + df_map["CountyFIPS"]

#Renaming the CBSA columns
df_map = df_map.rename(columns={
    "cbsacode":  "CBSA Code",
    "cbsatitle": "CBSA Title"
})

#Converting CBSA Code in df_map to a zero-padded string
df_map["CBSA Code"] = (
    df_map["CBSA Code"]
    .astype(int)
    .astype(str)
    .str.zfill(5)
)


df_a["CountyFIPS"] = df_a["CountyFIPS"].astype(str).str.zfill(5)

# Merging to tag each county with its CBSA
df_cnty_cbsa = df_a.merge(
    df_map[["CountyFIPS", "CBSA Code", "CBSA Title"]],
    on="CountyFIPS", how="left"
)

# Now also normalize df_aqis CBSA Code the same way
df_aqi["CBSA Code"] = (
    df_aqi["CBSA Code"]
    .astype(int)
    .astype(str)
    .str.zfill(5)
)


df_final = df_cnty_cbsa.merge(
    df_aqi,
    on="CBSA Code",
    how="left",
    suffixes=("_county", "_aqi")
)

df_final.tail()
```

Out[96]:

	State	CountyFIPS	County	Value	Value Asthma	CBSA Code	CBSA Title	CBSA	Date
292345	Wyoming	56041	Uinta	0.095	0.095	21740	Evanston, WY-UT	Evanston, WY	2024-06-28
292346	Wyoming	56041	Uinta	0.095	0.095	21740	Evanston, WY-UT	Evanston, WY	2024-06-29
292347	Wyoming	56041	Uinta	0.095	0.095	21740	Evanston, WY-UT	Evanston, WY	2024-06-30
292348	Wyoming	56043	Washakie	0.096	0.096	NaN	NaN	NaN	NaN
292349	Wyoming	56045	Weston	0.097	0.097	NaN	NaN	NaN	NaN

In [97]:

```
import pandas as pd

#Renaming the “Value” columns to distinguish asthma vs COPD
df_a = df_a.rename(columns={'Value':'Value Asthma'})
df_c = df_c.rename(columns={'Value':'Value COPD'})

#Zero-paddding CountyFIPS in both
for df in (df_a, df_c):
    df['CountyFIPS'] = df['CountyFIPS'].astype(str).str.zfill(5)


df_map["fipsstatecode"]   = df_map["fipsstatecode"].astype(int)
df_map["fipscountycode"]  = df_map["fipscountycode"].astype(int)
df_map["StateFIPS"]       = df_map["fipsstatecode"].astype(str).str.zfill(2)
df_map["CountyFIPS"]      = df_map["fipscountycode"].astype(str).str.zfill(3
df_map["CountyFIPS"]     = df_map["StateFIPS"] + df_map["CountyFIPS"]
df_map = df_map.rename(columns={
    "cbsacode":"CBSA Code",
    "cbsatitle":"Defining Parameter"
})
df_map["CBSA Code"]       = df_map["CBSA Code"].astype(int).astype(str).str.

df_aqi["CBSA Code"]       = df_aqi["CBSA Code"].astype(int).astype(str).str.


df_cnty = (
    df_a
    .merge(df_c[['CountyFIPS','Value COPD']],
           on='CountyFIPS', how='left')
)


df_cnty_cbsa = (
    df_cnty
    .merge(df_map[['CountyFIPS','CBSA Code']],
           on='CountyFIPS', how='left')
)


df_final = (
    df_cnty_cbsa
    .merge(
       df_aqi[['CBSA Code','AQI','Category','Defining Parameter']],
       on='CBSA Code', how='left'
    )
)


df_final.head()
```

Out[97]:

	State	CountyFIPS	County	Value Asthma	Value Asthma	Value COPD	Value COPD	CBSA Code	AQI	Category
0	Alabama	01001	Autauga	0.102	0.102	0.068	0.068	33860	53.0	Moderate
1	Alabama	01001	Autauga	0.102	0.102	0.068	0.068	33860	57.0	Moderate
2	Alabama	01001	Autauga	0.102	0.102	0.068	0.068	33860	55.0	Moderate
3	Alabama	01001	Autauga	0.102	0.102	0.068	0.068	33860	53.0	Moderate
4	Alabama	01001	Autauga	0.102	0.102	0.068	0.068	33860	6.0	Good

In [97]:

```
```

In [86]:

```
df_pol
```

Out[86]:

	StateFIPS	State	CountyFIPS	County	Year	Value	Data Comment	Unnamed: 7	Po
0	1	Alabama	1001	Autauga	2019	0.15	NaN	NaN	Po B
1	1	Alabama	1001	Autauga	2019	1.84	NaN	NaN	Po Formal
2	1	Alabama	1001	Autauga	2019	1.49	NaN	NaN	Po Acetal
3	1	Alabama	1001	Autauga	2019	0.51	NaN	NaN	Po tetrac
4	1	Alabama	1001	Autauga	2019	0.01	NaN	NaN	Polluta bu
...	...	...	...	...	...	...	...	...	
25779	78	Virgin Islands of the US	78030	St. Thomas	2019	0.36	NaN	NaN	Po tetrac
25780	78	Virgin Islands of the US	78030	St. Thomas	2019	0.00	NaN	NaN	Polluta bu
25781	78	Virgin Islands of the US	78030	St. Thomas	2019	0.00	NaN	NaN	Po Nap
25782	78	Virgin Islands of the US	78030	St. Thomas	2019	0.00	NaN	NaN	Po E
25783	78	Virgin Islands of the US	78030	St. Thomas	2019	0.05	NaN	NaN	Po Par

25784 rows × 9 columns

```
In [87]: df_pol
```

Out[87]:

	StateFIPS	State_x	CountyFIPS	County_x	Year	Value	Data Comment	Unnamed: 7	P
0	1	Alabama	01001	Autauga	2019	0.15	NaN	NaN	
1	1	Alabama	01001	Autauga	2019	1.84	NaN	NaN	Forn
2	1	Alabama	01001	Autauga	2019	1.49	NaN	NaN	Ace
3	1	Alabama	01001	Autauga	2019	0.51	NaN	NaN	tet
4	1	Alabama	01001	Autauga	2019	0.01	NaN	NaN	1,3-
...	...	...	...	...	...	...	...	...	
24571	56	Wyoming	56045	Weston	2019	0.46	NaN	NaN	tet
24572	56	Wyoming	56045	Weston	2019	0.00	NaN	NaN	1,3-
24573	56	Wyoming	56045	Weston	2019	0.00	NaN	NaN	N
24574	56	Wyoming	56045	Weston	2019	0.00	NaN	NaN	
24575	56	Wyoming	56045	Weston	2019	0.03	NaN	NaN	F

24576 rows × 17 columns

In [102…

```
df_pol2 = (
    df_pol
    .assign(
        Pollutant=lambda d: d['Pollutant'].str.replace(r'^Pollutant:\s*', ''
        CountyFIPS=lambda d: d['CountyFIPS'].astype(int).astype(str).str.zfi
        AQI=lambda d: d['Value'],
        Category='tbd'
    )
    .rename(columns={'Pollutant': 'Defining Parameter'})
    [['CountyFIPS', 'AQI', 'Category', 'Defining Parameter']]
)


df_final = df_final.loc[:, ~df_final.columns.duplicated()]


health = (
    df_final
    .loc[:, ['CountyFIPS', 'Value Asthma', 'Value COPD']]
    .drop_duplicates(subset='CountyFIPS')
)


df_pol2 = df_pol2.merge(health, on='CountyFIPS', how='inner')


meta = (
    df_final
    .loc[:, ['CountyFIPS', 'State', 'County', 'CBSA Code']]
    .drop_duplicates(subset='CountyFIPS')
)
df_pol2 = df_pol2.merge(meta, on='CountyFIPS', how='left')


cols = [
    'State', 'CountyFIPS', 'County',
    'Value Asthma', 'Value COPD',
    'CBSA Code', 'AQI', 'Category', 'Defining Parameter'
]

for col in cols:
    if col not in df_final.columns:
        df_final[col] = pd.NA

df_combined = pd.concat([df_final[cols], df_pol2[cols]], ignore_index=True,

print(df_combined.shape)
df_combined.head(100)
```

```
(316926, 9)
```

Out[102]:

	State	CountyFIPS	County	Value Asthma	Value COPD	CBSA Code	AQI	Category	Defining Parameter
0	Alabama	01001	Autauga	0.102	0.068	33860	53.0	Moderate	PM2.5
1	Alabama	01001	Autauga	0.102	0.068	33860	57.0	Moderate	PM2.5
2	Alabama	01001	Autauga	0.102	0.068	33860	55.0	Moderate	PM2.5
3	Alabama	01001	Autauga	0.102	0.068	33860	53.0	Moderate	PM2.5
4	Alabama	01001	Autauga	0.102	0.068	33860	6.0	Good	PM10
...	...	...	...	...	...	...	...	...	...
95	Alabama	01001	Autauga	0.102	0.068	33860	48.0	Good	Ozone
96	Alabama	01001	Autauga	0.102	0.068	33860	50.0	Good	Ozone
97	Alabama	01001	Autauga	0.102	0.068	33860	52.0	Moderate	PM2.5
98	Alabama	01001	Autauga	0.102	0.068	33860	63.0	Moderate	PM2.5
99	Alabama	01001	Autauga	0.102	0.068	33860	51.0	Moderate	PM2.5

100 rows × 9 columns

In [ ]:

In [ ]:

In [ ]:

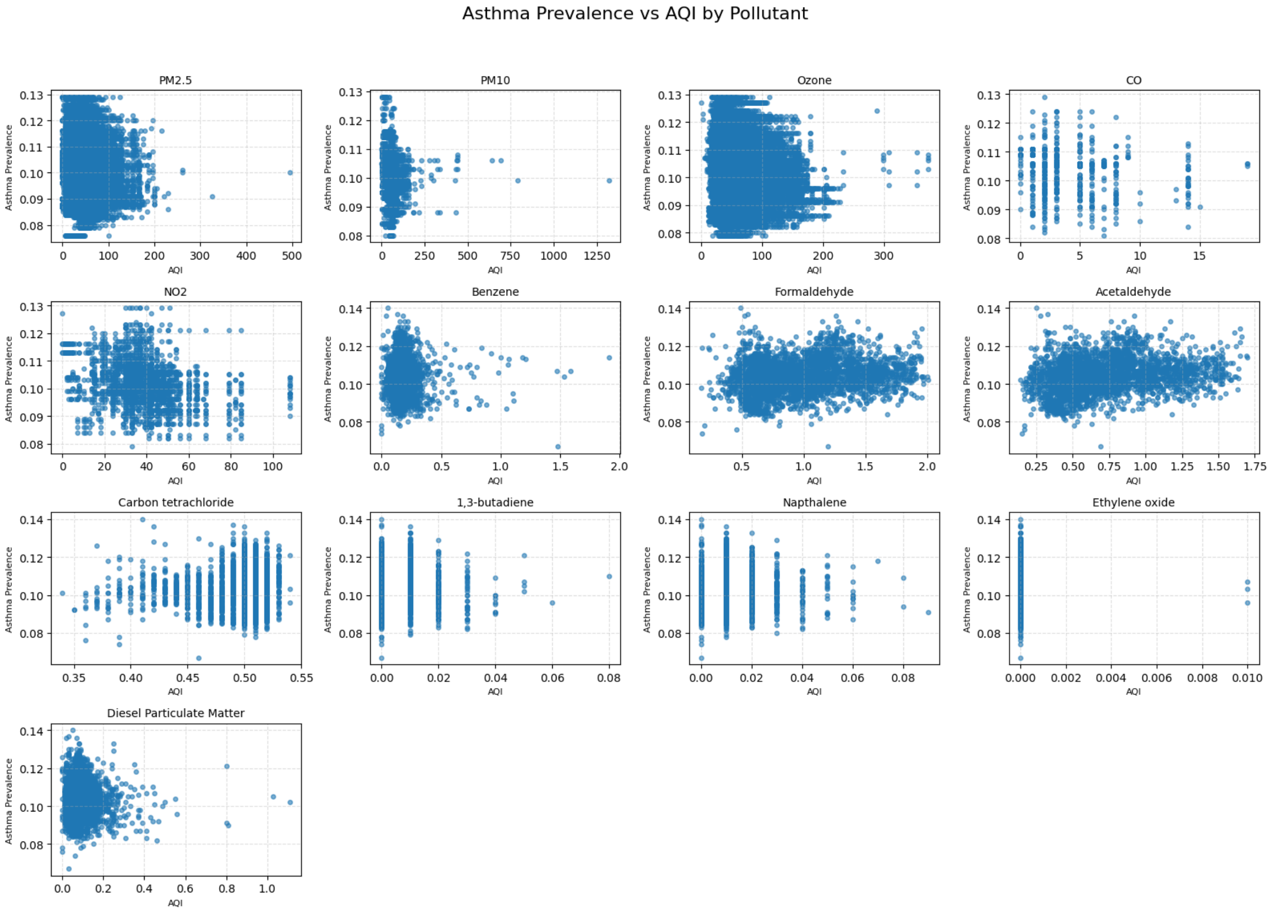
In [114…

In [100…

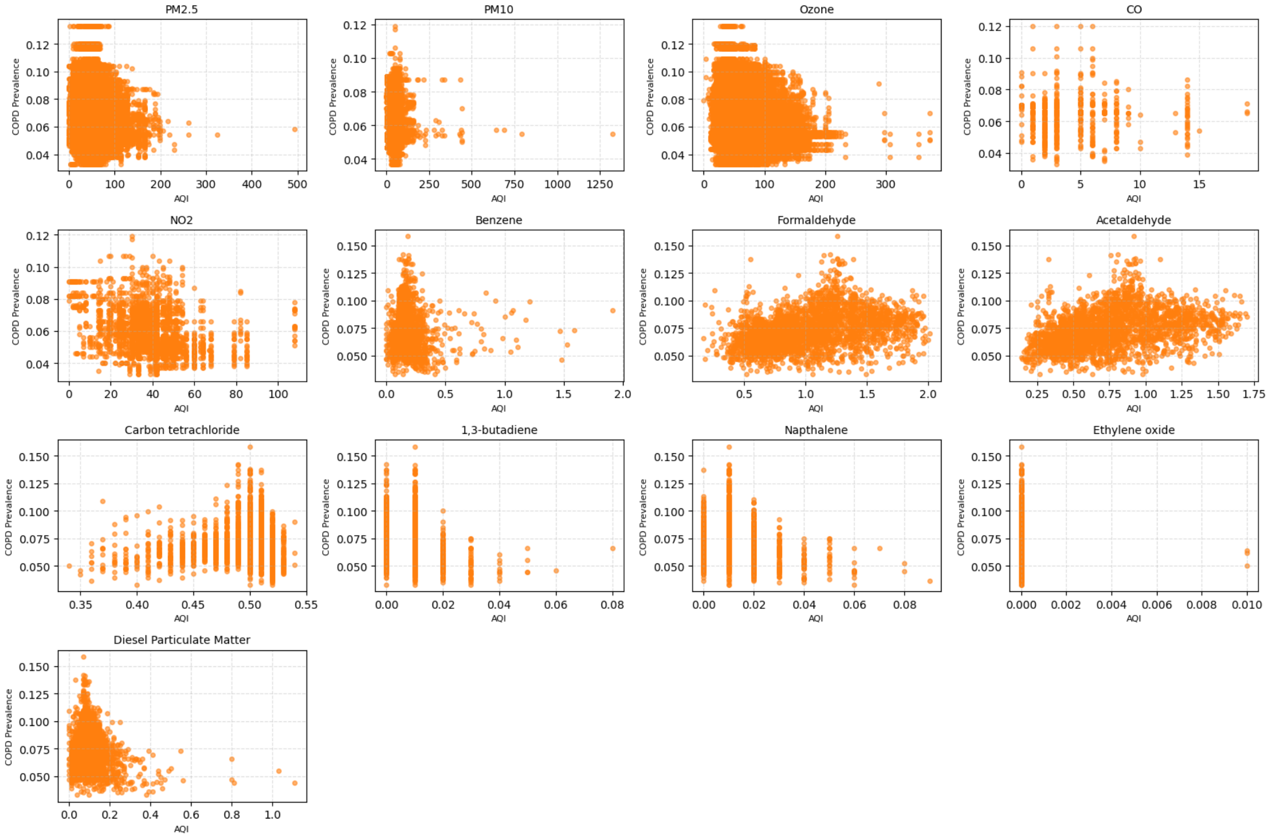
In [103…

```
filter_prep = ["State", "County", "Value Asthma", "Value COPD", "AQI", "Cate
df_prep = df_combined[filter_prep]



df_prep = df_prep.loc[:, ~df_prep.columns.duplicated()]


predictors = df_prep['Defining Parameter'].dropna().unique().tolist()


n_cols = 4
n_rows = int(np.ceil(len(predictors) / n_cols))
```

```python
fig1, axes1 = plt.subplots(n_rows, n_cols, figsize=(16, 12))

for ax, pred in zip(axes1.flat, predictors):
    sub = df_prep[df_prep['Defining Parameter'] == pred]
    x = sub['AQI']
    y = sub['Value Asthma']
    ax.scatter(x, y, s=15, alpha=0.6)
    ax.set_title(pred, fontsize=10)
    ax.set_xlabel('AQI', fontsize=8)
    ax.set_ylabel('Asthma Prevalence', fontsize=8)
    ax.grid(True, linestyle='--', alpha=0.4)


for ax in axes1.flat[len(predictors):]:
    ax.set_visible(False)

fig1.suptitle('Asthma Prevalence vs AQI by Pollutant', fontsize=16)
fig1.tight_layout(rect=[0, 0.03, 1, 0.95])



fig2, axes2 = plt.subplots(n_rows, n_cols, figsize=(16, 12))

for ax, pred in zip(axes2.flat, predictors):
    sub = df_prep[df_prep['Defining Parameter'] == pred]
    x = sub['AQI']
    y = sub['Value COPD']
    ax.scatter(x, y, s=15, alpha=0.6, color='tab:orange')
    ax.set_title(pred, fontsize=10)
    ax.set_xlabel('AQI', fontsize=8)
    ax.set_ylabel('COPD Prevalence', fontsize=8)
    ax.grid(True, linestyle='--', alpha=0.4)

for ax in axes2.flat[len(predictors):]:
    ax.set_visible(False)

fig2.suptitle('COPD Prevalence vs AQI by Pollutant', fontsize=16)
fig2.tight_layout(rect=[0, 0.03, 1, 0.95])

plt.show()
```


Asthma Prevalence vs AQI by Pollutant
PM2.5
PM10
Ozone
CO
NO2
Benzene
Formaldehyde
Acetaldehyde
Carbon tetrachloride
1,3-butadiene
Napthalene
Ethylene oxide
Diesel Particulate Matter
Asthma Prevalence
AQI
0.08
0.09
0.10
0.11
0.12
0.13
0.14
0
100
200
300
400
500
250
500
750
1000
1250
5
10
15
20
40
60
80
0.0
0.5
1.0
1.5
2.0
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0.35
0.40
0.45
0.50
0.55
0.00
0.02
0.04
0.06
0.08
0.000
0.002
0.004
0.006
0.008
0.010
0.2
0.4
0.6
0.8

COPD Prevalence vs AQI by Pollutant



In [104…

```
df_prep = df_prep.loc[:, ~df_prep.columns.duplicated()]

#We define thresholds that we got from epa website
thresholds = {

    'PM2.5': 8,         # µg/m3 annual
    'PM10': 100,        #  µg/m3 24-hr
    'Ozone': 0.02,      # ppm 8-hr
    'CO': 4,            # ppm 8-hr
    'NO2': 32,          # ppb  annual

    # EPA IRIS Reference concentrations (RfC) for air toxics (mg/m3)
    'Benzene':                0.03,      # RfC = 3×10−2 mg/m³
    'Formaldehyde':           0.8,     # RfC = 8×10−3 mg/m³
    'Acetaldehyde':           0.9,     # RfC = 9×10−3 mg/m³
    'Carbon tetrachloride':   0.1,       # RfC = 1×10−1 mg/m³
    '1,3-butadiene':          0.007,     # RfC ≈ 7×10−3 mg/m³
    'Napthalene':             0.005,     # RfC = 5×10−3 mg/m³
    'Ethylene oxide':         0.0001,    # RfC = 1×10−4 mg/m³

    'Diesel Particulate Matter': 0.05  # RfC = 5 µg/m³ 2002 National-Scale A
}


if 'Category' not in df_prep.columns:
    df_prep['Category'] = pd.NA

#Making binary variables (good or high)
for pollutant, thr in thresholds.items():
    mask = df_prep['Defining Parameter'] == pollutant
    df_prep.loc[mask & (df_prep['AQI'] <= thr), 'Category'] = 'Good'
    df_prep.loc[mask & (df_prep['AQI']  > thr), 'Category'] = 'High'

#we keep the tbd for everything else
df_prep['Category'] = df_prep['Category'].fillna('tbd')


print(df_prep[['Defining Parameter','AQI','Category']].drop_duplicates().sor
```

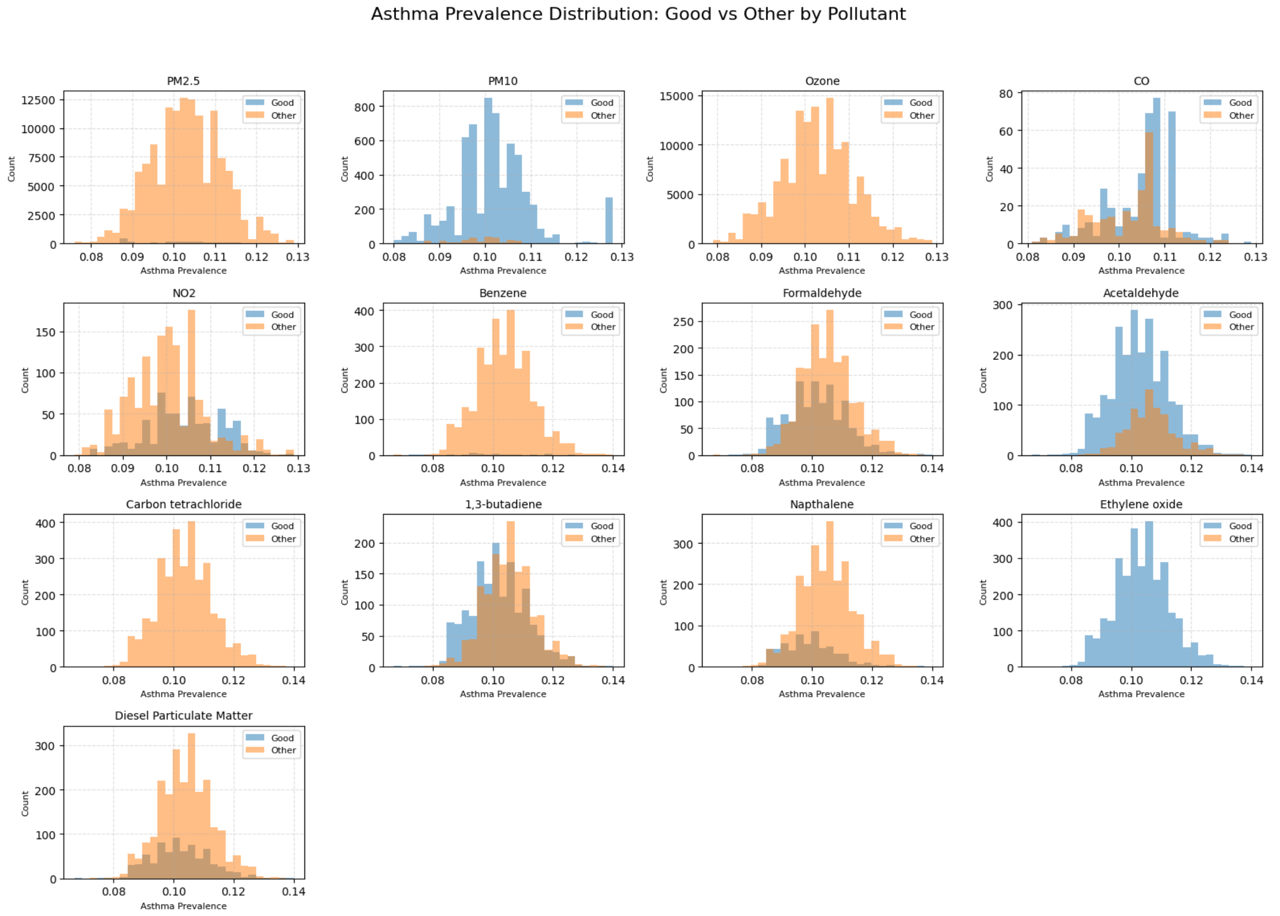
```
       Defining Parameter     AQI Category
294418      1,3-butadiene    0.06     High
294890      1,3-butadiene    0.05     High
292354      1,3-butadiene    0.01     High
292906      1,3-butadiene    0.02     High
315034      1,3-butadiene    0.08     High
...                   ...     ...      ...
5067                PM2.5  220.00     High
183401              PM2.5  118.00     High
5100                PM2.5  104.00     High
0                   PM2.5   53.00     High
415                   NaN     NaN      tbd

[1060 rows x 3 columns]
```

In [14]:

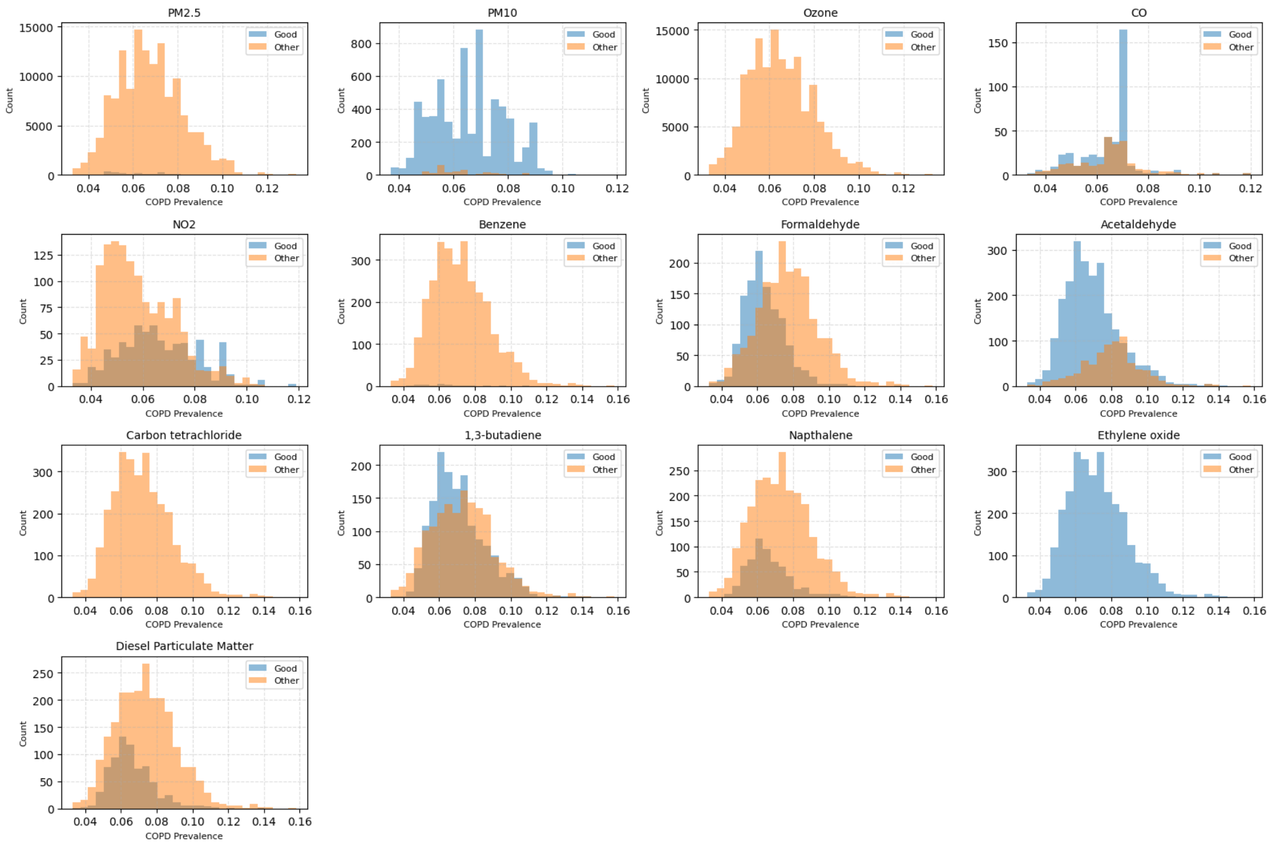
In [105…

```
df_prep = df_prep.loc[:, ~df_prep.columns.duplicated()]


predictors = df_prep['Defining Parameter'].dropna().unique().tolist()


n_cols = 4
n_rows = int(np.ceil(len(predictors) / n_cols))


fig_a, axes_a = plt.subplots(n_rows, n_cols, figsize=(16, 12), sharex=False,
for ax, pred in zip(axes_a.flat, predictors):
    sub = df_prep[df_prep['Defining Parameter']==pred].dropna(subset=['Value
    if sub.empty:
        ax.set_visible(False)
        continue
    good = sub[sub['Category']=='Good']['Value Asthma']
    other = sub[sub['Category']!='Good']['Value Asthma']
    bins = np.linspace(sub['Value Asthma'].min(), sub['Value Asthma'].max(),
    ax.hist(good,  bins=bins, histtype='stepfilled', alpha=0.5, label='Good'
    ax.hist(other, bins=bins, histtype='stepfilled', alpha=0.5, label='Other
    ax.set_title(pred, fontsize=10)
    ax.set_xlabel('Asthma Prevalence', fontsize=8)
    ax.set_ylabel('Count', fontsize=8)
    ax.legend(fontsize=8)
    ax.grid(True, linestyle='--', alpha=0.4)
for ax in axes_a.flat[len(predictors):]:
    ax.set_visible(False)
fig_a.suptitle('Asthma Prevalence Distribution: Good vs Other by Pollutant',
fig_a.tight_layout(rect=[0,0.03,1,0.95])


fig_c, axes_c = plt.subplots(n_rows, n_cols, figsize=(16, 12), sharex=False,
for ax, pred in zip(axes_c.flat, predictors):
    sub = df_prep[df_prep['Defining Parameter']==pred].dropna(subset=['Value
    if sub.empty:
        ax.set_visible(False)
        continue
    good = sub[sub['Category']=='Good']['Value COPD']
    other = sub[sub['Category']!='Good']['Value COPD']
    bins = np.linspace(sub['Value COPD'].min(), sub['Value COPD'].max(), 30)
    ax.hist(good,  bins=bins, histtype='stepfilled', alpha=0.5, label='Good'
    ax.hist(other, bins=bins, histtype='stepfilled', alpha=0.5, label='Other
    ax.set_title(pred, fontsize=10)
    ax.set_xlabel('COPD Prevalence', fontsize=8)
    ax.set_ylabel('Count', fontsize=8)
    ax.legend(fontsize=8)
    ax.grid(True, linestyle='--', alpha=0.4)
for ax in axes_c.flat[len(predictors):]:
    ax.set_visible(False)
```

```
fig_c.suptitle('COPD Prevalence Distribution: Good vs Other by Pollutant', f
fig_c.tight_layout(rect=[0,0.03,1,0.95])

plt.show()
```



COPD Prevalence Distribution: Good vs Other by Pollutant



In [106…

```
df = df_prep
```

In [106…

```
```

Using a Random Forest to identify most important predictors

In [106…

```
```

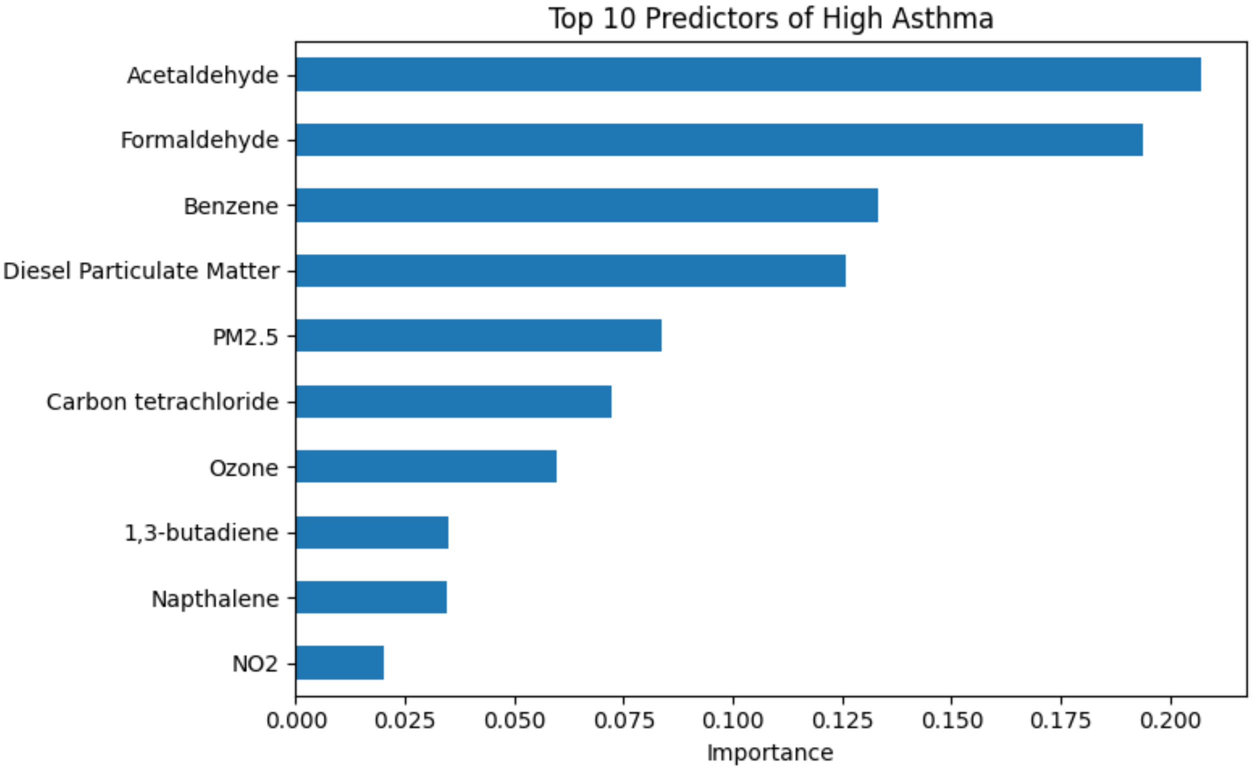
In [107…

```
#make binary
df['High_Asthma'] = (df['Value Asthma'] > df['Value Asthma'].median()).astyp
df['High_COPD']   = (df['Value COPD'] > df['Value COPD'].median()).astype(in
```

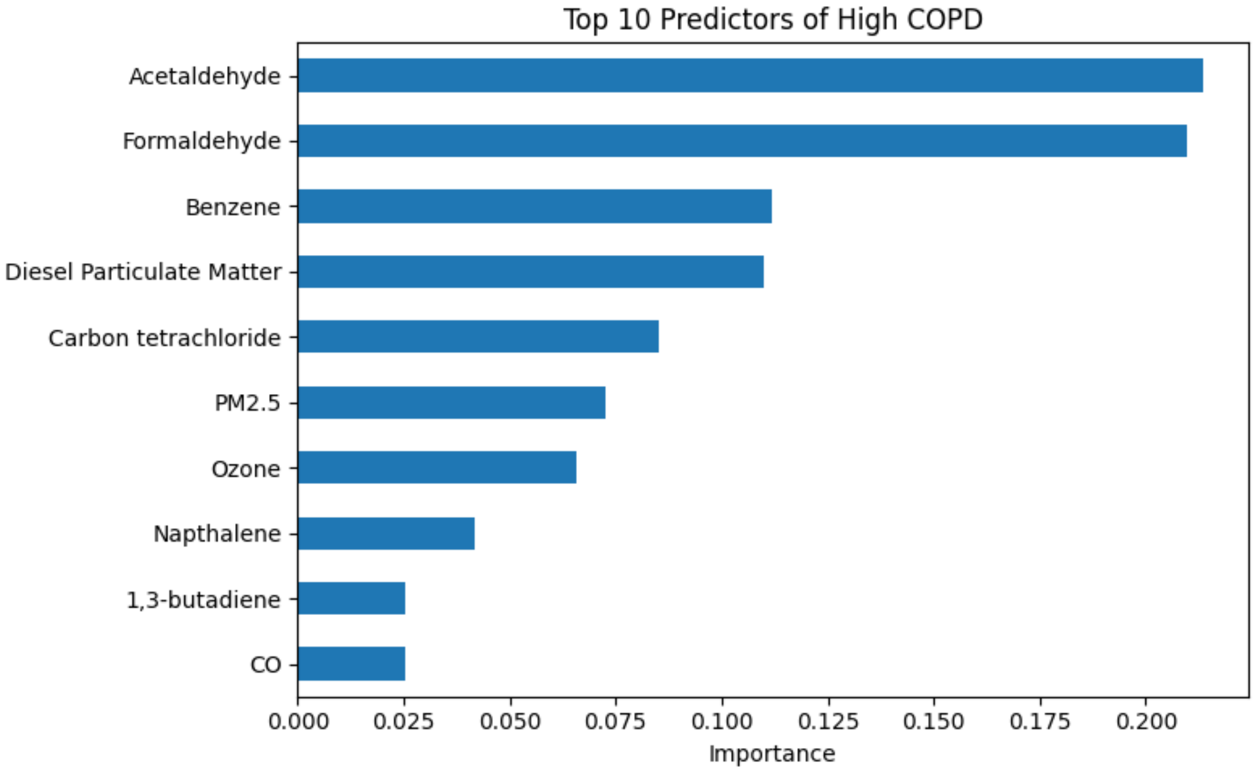
In [108…

```
df_prep = df_prep.loc[:, ~df_prep.columns.duplicated()]

df['High_Asthma'] = (df['Value Asthma'] > df['Value Asthma'].median()).astyp
df['High_COPD']   = (df['Value COPD'] > df['Value COPD'].median()).astype(in


df_wide = (
    df_prep
    .pivot_table(
        index=['State','County'],
        columns='Defining Parameter',
        values='AQI',
        aggfunc='mean'
    )
    .reset_index()
)


df_health = (
    df[['State','County','Value Asthma','Value COPD','High_Asthma','High_COP
    .drop_duplicates(subset=['State','County'])
)


df_ml = df_wide.merge(df_health, on=['State','County'], how='inner')


def run_rf(label_col, df):
    y = df[label_col]
    X = df.drop(columns=['State','County','Value Asthma','Value COPD','High_

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42, stratify=y
    )

    rf = RandomForestClassifier(n_estimators=200, class_weight='balanced', r
    rf.fit(X_train, y_train)

    print(f"\n=== {label_col} Model ===")
    y_pred = rf.predict(X_test)
    print(classification_report(y_test, y_pred, target_names=['Low-Risk','Hi


    imps = pd.Series(rf.feature_importances_, index=X.columns).sort_values(a
    top10 = imps.head(10)


    plt.figure(figsize=(8,5))
    top10.plot(kind='barh')
    plt.gca().invert_yaxis()
    plt.title(f"Top 10 Predictors of High {label_col.replace('High_', '')}")
    plt.xlabel("Importance")
    plt.tight_layout()
    plt.show()

    return imps

# 6) Run models
imps_asthma = run_rf('High_Asthma', df_ml)
imps_copd   = run_rf('High_COPD', df_ml)
```

```
=== High_Asthma Model ===
              precision    recall  f1-score   support

    Low-Risk       0.78      0.73      0.75       454
   High-Risk       0.75      0.80      0.78       468

    accuracy                           0.76       922
   macro avg       0.77      0.76      0.76       922
weighted avg       0.77      0.76      0.76       922
```



```
=== High_COPD Model ===
              precision    recall  f1-score   support

    Low-Risk       0.73      0.74      0.73       378
   High-Risk       0.81      0.81      0.81       544

    accuracy                           0.78       922
   macro avg       0.77      0.77      0.77       922
weighted avg       0.78      0.78      0.78       922
```


Top 10 Predictors of High COPD
Acetaldehyde
Formaldehyde
Benzene
Diesel Particulate Matter
Carbon tetrachloride
PM2.5
Ozone
Napthalene
1,3-butadiene
CO
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Importance

```
In [108…
```

```
In [108…
```

In [109…

```
asthma_min = df['Value Asthma'].min()
asthma_max = df['Value Asthma'].max()
copd_min = df['Value COPD'].min()
copd_max = df['Value COPD'].max()

# Normalize
df['Value Asthma Scaled'] = (df['Value Asthma'] - asthma_min) / (asthma_max
df['Value COPD Scaled'] = (df['Value COPD'] - copd_min) / (copd_max - copd_m


top_predictors = [
    'Formaldehyde', 'Acetaldehyde', 'Diesel Particulate Matter', 'Benzene',
    'Carbon tetrachloride', 'PM2.5', 'Ozone', 'Napthalene'
]

df_wide = df.pivot_table(
    index=['State', 'County'],
    columns='Defining Parameter',
    values='AQI',
    aggfunc='mean'
).reset_index()


target_df = df[['State', 'County', 'Value Asthma Scaled', 'Value COPD Scaled
df_wide = df_wide.merge(target_df, on=['State', 'County'], how='left')

X = df_wide[top_predictors]
y_asthma = df_wide['Value Asthma Scaled']
y_copd = df_wide['Value COPD Scaled']
```

In [110…

```
# Impute missing values with column mean
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X_imputed, y_asthma, tes

linreg = LinearRegression()
linreg.fit(X_train, y_train)

# Sigmoid and rescale
from scipy.special import expit
y_pred_scaled = expit(linreg.predict(X_test))
y_pred = y_pred_scaled * (asthma_max - asthma_min) + asthma_min
```

In [111…

```
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X_imputed, y_copd, test_

linreg_copd = LinearRegression()
linreg_copd.fit(X_train, y_train)


from scipy.special import expit
y_pred_scaled = expit(linreg_copd.predict(X_test))
y_pred = y_pred_scaled * (copd_max - copd_min) + copd_min
```

In [112…

```
# True values in original scale
y_true = y_test * (copd_max - copd_min) + copd_min

# 1. MSE and R²
mse = mean_squared_error(y_true, y_pred)
r2 = r2_score(y_true, y_pred)

print(f"Mean Squared Error: {mse:.4f}")
print(f"R² Score: {r2:.4f}")
```

```
Mean Squared Error: 0.0013
R² Score: -4.1079
```

In [113…

```
# True values in original scale
y_true = y_test * (asthma_max - asthma_min) + asthma_min

# 1. MSE and R²
mse = mean_squared_error(y_true, y_pred)
r2 = r2_score(y_true, y_pred)

print(f"Mean Squared Error: {mse:.4f}")
print(f"R² Score: {r2:.4f}")
```

```
Mean Squared Error: 0.0003
R² Score: -2.5267
```

In [113…

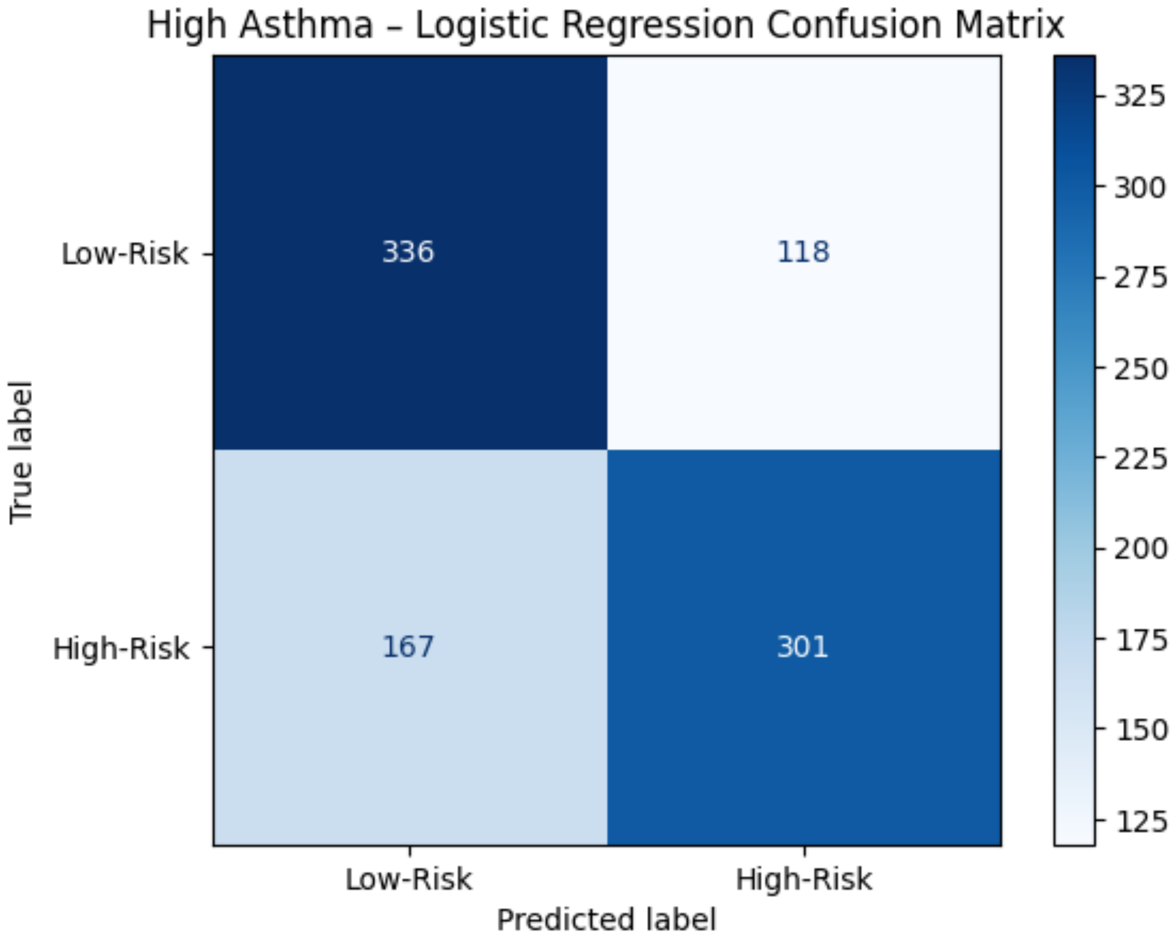
In [113…

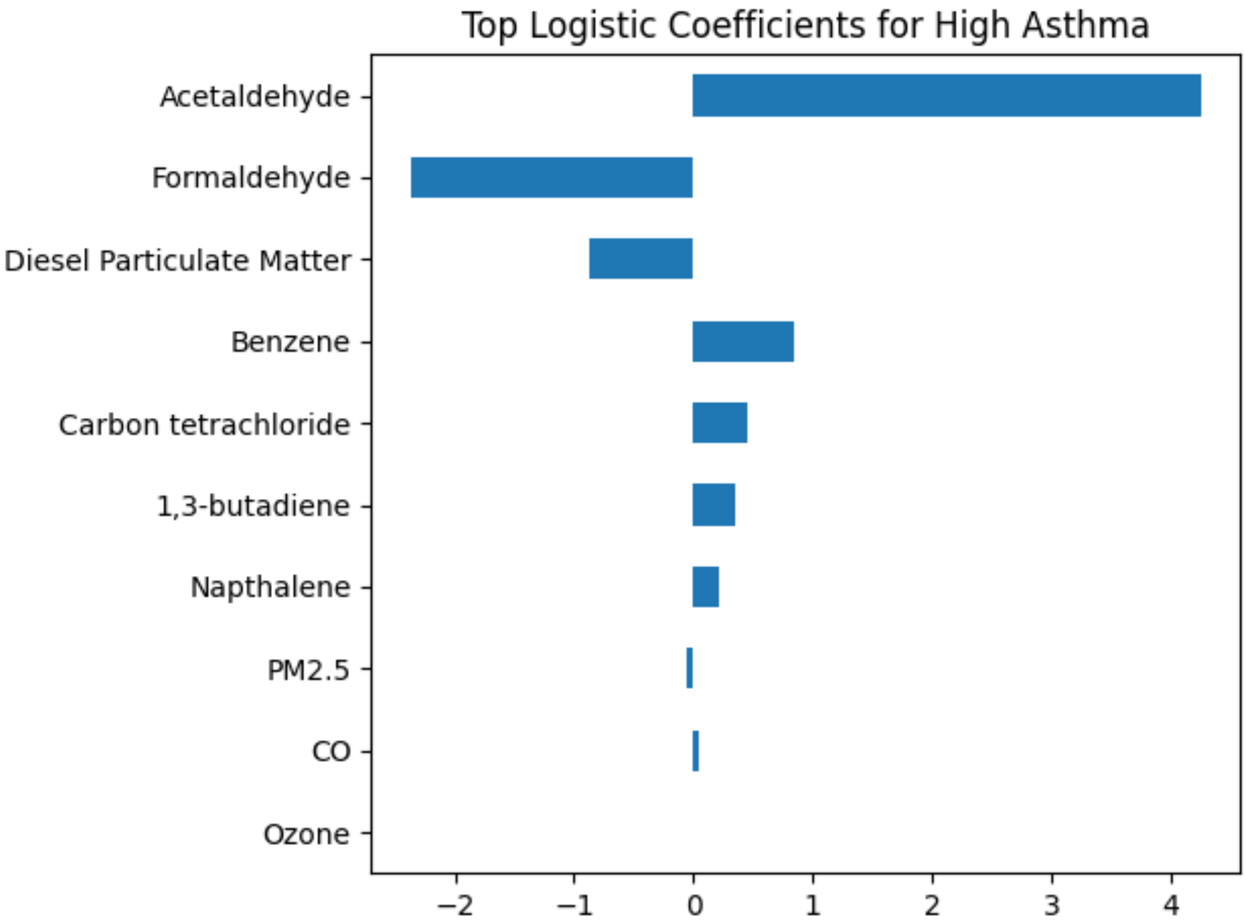
In [113…

In [114…

```
df_wide = (
    df_prep
    .pivot_table(
        index=['State','County'],
        columns='Defining Parameter',
        values='AQI',
        aggfunc='mean'
    )
    .reset_index()
)


df_health = (
    df[['State','County','Value Asthma','Value COPD','High_Asthma','High_COP
    .drop_duplicates(subset=['State','County'])
)

df_ml = df_wide.merge(df_health, on=['State', 'County'], how='inner')


X = df_ml.drop(columns=['State', 'County', 'Value Asthma', 'Value COPD', 'Hi
y = df_ml['High_Asthma']


imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=


clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)


y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)[:, 1]

print(classification_report(y_test, y_pred, target_names=["Low-Risk", "High-
print(f"ROC AUC: {roc_auc_score(y_test, y_prob):.3f}")

disp = ConfusionMatrixDisplay(confusion_matrix(y_test, y_pred), display_labe
disp.plot(cmap='Blues')
plt.title("High Asthma — Logistic Regression Confusion Matrix")
plt.show()

# Top features
feature_names = X.columns
coefs = pd.Series(clf.coef_[0], index=feature_names).sort_values(key=abs, as
coefs.head(10).plot(kind='barh', title="Top Logistic Coefficients for High A
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```

```
              precision    recall  f1-score   support

    Low-Risk       0.67      0.74      0.70       454
   High-Risk       0.72      0.64      0.68       468

    accuracy                           0.69       922
   macro avg       0.69      0.69      0.69       922
weighted avg       0.69      0.69      0.69       922

ROC AUC: 0.740
```



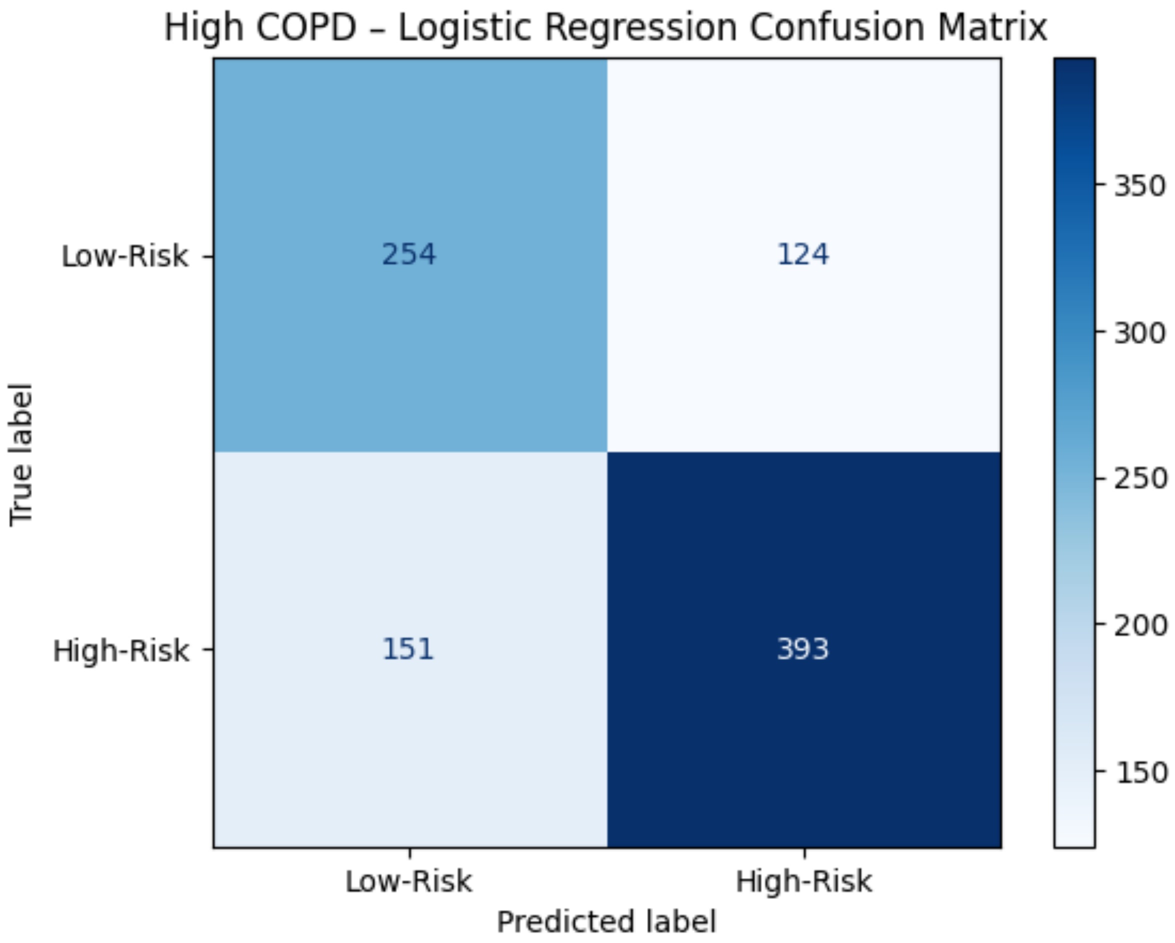


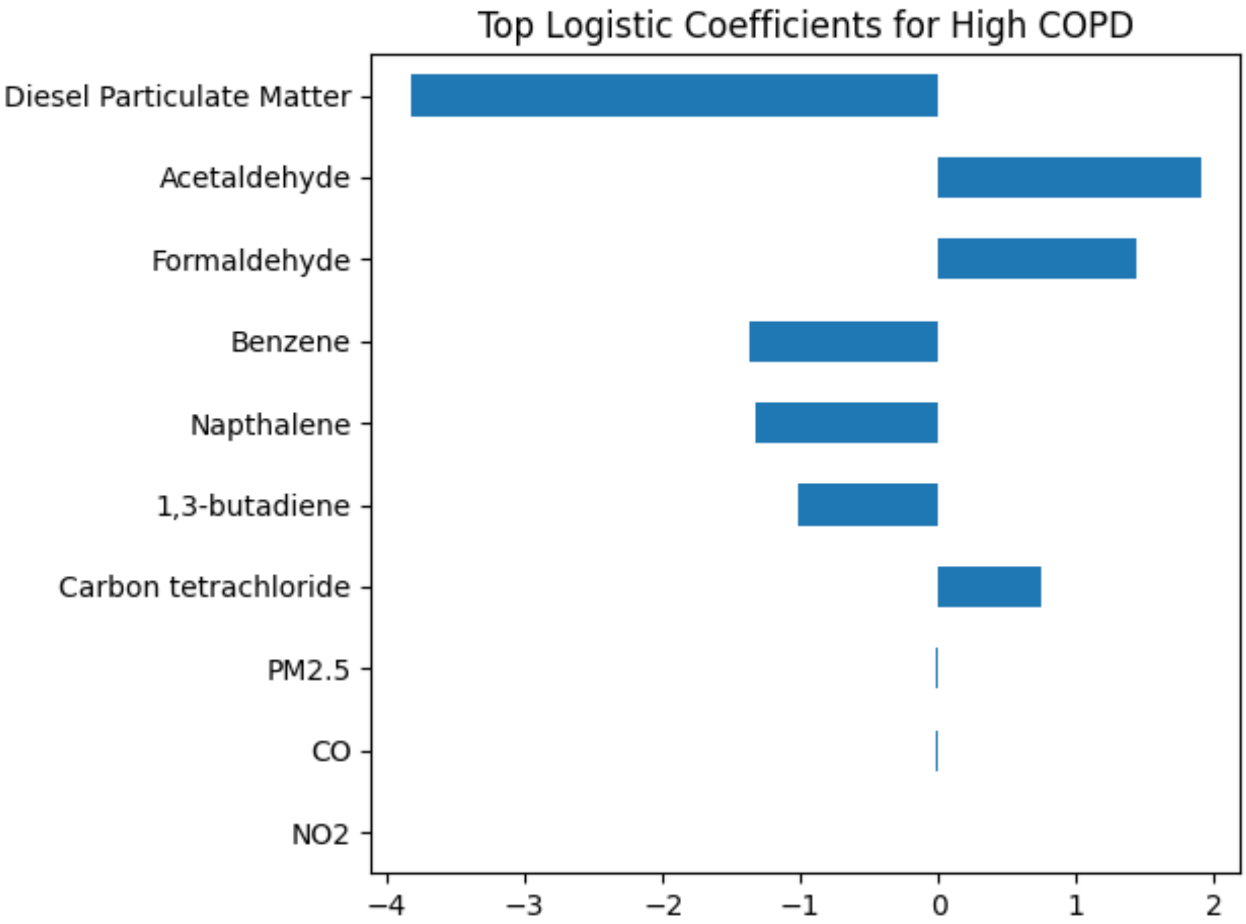
In [117…

```
df_wide = (
    df_prep
    .pivot_table(
        index=['State','County'],
        columns='Defining Parameter',
        values='AQI',
        aggfunc='mean'
    )
    .reset_index()
)


df_health = (
    df[['State','County','Value Asthma','Value COPD','High_Asthma','High_COP
    .drop_duplicates(subset=['State','County'])
)

df_ml = df_wide.merge(df_health, on=['State', 'County'], how='inner')


X = df_ml.drop(columns=['State', 'County', 'Value Asthma', 'Value COPD', 'Hi
y = df_ml['High_COPD']
```

```
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=


clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)


y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)[:, 1]

print(classification_report(y_test, y_pred, target_names=["Low-Risk", "High-
print(f"ROC AUC: {roc_auc_score(y_test, y_prob):.3f}")

disp = ConfusionMatrixDisplay(confusion_matrix(y_test, y_pred), display_labe
disp.plot(cmap='Blues')
plt.title("High COPD — Logistic Regression Confusion Matrix")
plt.show()

# Top features
feature_names = X.columns
coefs = pd.Series(clf.coef_[0], index=feature_names).sort_values(key=abs, as
coefs.head(10).plot(kind='barh', title="Top Logistic Coefficients for High C
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```

```
              precision    recall  f1-score   support

    Low-Risk       0.63      0.67      0.65       378
   High-Risk       0.76      0.72      0.74       544

    accuracy                           0.70       922
   macro avg       0.69      0.70      0.69       922
weighted avg       0.71      0.70      0.70       922

ROC AUC: 0.755
```


High COPD – Logistic Regression Confusion Matrix
True label
Low-Risk
High-Risk
254
124
151
393
Predicted label
Low-Risk
High-Risk
350
300
250
200
150


Top Logistic Coefficients for High COPD
Diesel Particulate Matter
Acetaldehyde
Formaldehyde
Benzene
Napthalene
1,3-butadiene
Carbon tetrachloride
PM2.5
CO
NO2
−4
−3
−2
−1
0
1
2

```
In [ ]:
```