Source code

MODEL BUILDING:

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report

from sklearn.preprocessing import LabelEncoder, StandardScaler

df = pd.read_csv("feature_engineered_data.csv")

# Label encoding for target

le = LabelEncoder()

df['AQI_Bucket'] = le.fit_transform(df['AQI_Bucket'])

X = df.drop(columns=['AQI_Bucket'])

y = df['AQI_Bucket']

# Standardize

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
stratify=y, random_state=42)

# Logistic Regression

logreg = LogisticRegression(max_iter=1000)

logreg.fit(X_train, y_train)

y_pred_logreg = logreg.predict(X_test)
```

```
print("Logistic Regression Report:")
print(classification_report(y_test, y_pred_logreg, target_names=le.classes_))
# Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest Report:")
print(classification_report(y_test, y_pred_rf, target_names=le.classes_))
```

FEATURE ENGINEERING:

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.decomposition import PCA
df = pd.read_csv("preprocessed_data.csv")
# Create Pollution Load
df['Pollution_Load'] = df[['PM2.5', 'PM10', 'NO2', 'SO2', 'CO',
'O3']].sum(axis=1)
# THI: Temperature-Humidity Index
df['THI'] = df['Temperature'] - (0.55 - 0.0055 * df['Humidity']) *
(df['Temperature'] - 14.5)
# PM Ratio
df['PM_Ratio'] = df['PM2.5'] / (df['PM10'] + 1e-5)
```

```
# Polynomial features

poly = PolynomialFeatures(degree=2, include_bias=False)

poly_features = poly.fit_transform(df[['PM2.5', 'NO2']])

poly_df = pd.DataFrame(poly_features,
columns=poly.get_feature_names_out(['PM2.5', 'NO2']))

df = pd.concat([df, poly_df], axis=1)

# PCA

pca = PCA(n_components=2)

pca_features = pca.fit_transform(df[['PM2.5', 'PM10', 'NO2', 'SO2', 'CO',
'O3']])

df['PCA1'], df['PCA2'] = pca_features[:, 0], pca_features[:, 1]

df.to_csv("feature_engineered_data.csv", index=False)
```

DATA PREPROCESSING:

```
import pandas as pd

from sklearn.preprocessing import LabelEncoder, StandardScaler

df =
pd.read_csv("Air_Quality_Measures_on_the_National_Environmental_Health_
Tracking_Network.csv")

# Drop duplicates

df.drop_duplicates(inplace=True)

# Handle missing values (simple imputation or dropping)

df.fillna(method='ffill', inplace=True)
```

```
# Encode categorical variables

le = LabelEncoder()

if 'AQI_Bucket' in df.columns:

    df['AQI_Bucket_Encoded'] = le.fit_transform(df['AQI_Bucket'])

# Standardization

scaler = StandardScaler()

numeric_cols = df.select_dtypes(include=['float64', 'int64']).columns

df[numeric_cols] = scaler.fit_transform(df[numeric_cols])

df.to_csv("preprocessed_data.csv", index=False)
```