

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report

file_path = "/content/pollution dataset.xlsx"
df = pd.read_excel(file_path)

df['PM10'] = df['PM10'].apply(lambda x: max(x, 0))
df['SO2'] = df['SO2'].apply(lambda x: max(x, 0))

def categorize_air_quality(pm25):
    if pm25 <= 12:
        return "Good"
    elif 12 < pm25 <= 35:
        return "Moderate"
    elif 35 < pm25 <= 55:
        return "Unhealthy for Sensitive Groups"
    elif 55 < pm25 <= 150:
        return "Unhealthy"
    else:
        return "Hazardous"

df['Air_Quality_Level'] = df['PM2.5'].apply(categorize_air_quality)

label_encoder = LabelEncoder()
df['Air_Quality_Level'] = label_encoder.fit_transform(df['Air_Quality_Level'])

t
features = ['Temperature', 'Humidity', 'PM10', 'NO2', 'SO2', 'CO',
'Proximity_to_Industrial_Areas', 'Population_Density']
target = 'Air_Quality_Level'
X = df[features]
y = df[target]

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

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

```

```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
print(classification_report(y_test, y_pred,
target_names=label_encoder.classes_))

def predict_air_quality(new_data):
    new_df = pd.DataFrame([new_data], columns=features)
    prediction = model.predict(new_df)
    return label_encoder.inverse_transform(prediction)[0]

sample_data = {'Temperature': 70, 'Humidity': 20, 'PM10': 80, 'NO2': 25,
'SO2': 10, 'CO': 1.5, 'Proximity_to_Industrial_Areas': 8,
'Population_Density': 500}
predicted_quality = predict_air_quality(sample_data)
print(f"Predicted Air Quality Level: {predicted_quality}")
```