**Our source code for the project using Python and AI to fetch data from Vultr's cloud, analyze air quality, and send alerts when needed.**

**Setting Up Vultr API for fetching data**

**Python:**

import requests

API\_KEY = ' our\_vultr\_api\_key'

url = 'https://api.vultr.com/v2/ our\_endpoint'

headers = {

'Authorization': f'Bearer {API\_KEY}'

}

response = requests.get(url, headers=headers)

if response.status\_code == 200:

air\_quality\_data = response.json() # Air quality data

print("Data fetched successfully:", air\_quality\_data)

else:

print(f"Failed to fetch data. Status code: {response.status\_code}")

**Analyzing Air Quality Data Using AI**

Once we fetch the air quality data, we can use Python libraries such as scikit-learn or TensorFlow to analyze the data. Detecting poor air quality:

**Python**

import numpy as np

from sklearn.ensemble import RandomForestClassifier

air\_quality\_data = np.array([[35, 60, 25, 40, 10, 0.8]])

training\_data = np.array([[30, 50, 20, 35, 8, 0.5], [70, 90, 45, 55, 25, 1.5], [15, 30, 12, 25, 5, 0.4]])

labels = np.array([0, 1, 0]) # Corresponding labels

model = RandomForestClassifier()

model.fit(training\_data, labels)

prediction = model.predict(air\_quality\_data)

if prediction == 1:

print("Air quality is unhealthy! Sending alert...")

else:

print("Air quality is within safe limits.")

**Sending Alert to the Government**

we can use an email service like SMTP to send alerts to government officials or decision-makers.

**Python**

import smtplib

from email.mime.text import MIMEText

def send\_alert():

sender\_email = " our\_email@example.com"

receiver\_email = "government\_email@example.com"

subject = "Air Quality Alert: Immediate Action Required"

body = "The air quality in our area has reached hazardous levels. Immediate action is required."

msg = MIMEText(body)

msg["Subject"] = subject

msg["From"] = sender\_email

msg["To"] = receiver\_email

try:

with smtplib.SMTP("smtp.example.com", 587) as server:

server.starttls()

server.login(" our\_email@example.com", " our\_password")

server.sendmail(sender\_email, receiver\_email, msg.as\_string())

print("Alert email sent successfully!")

except Exception as e:

print(f"Failed to send email: {e}")

if prediction == 1:

send\_alert()

**Deploying the System on Vultr Cloud**

Once our python code is ready, we can deploy it on a Vultr cloud instance. we can set it up as a service that continuously fetches and monitors air quality, sending alerts when needed. Make sure to use tools like cron jobs to schedule regular checks and alerts.

This framework will help us in fetching data, analyzing it with AI, and sending alerts. we can extend it by adding more sophisticated models, more data sources, and additional alert mechanisms like SMS, push notifications, or integrations with local authority systems.