@@ -0,0 +1,143 @@

HTTP:

python codes

To generate code for your Traffic Volume Estimation project, you can break it into four major components:

---

✅ 1. Data Preprocessing & Model Training

File: traffic\_volume.ipynb

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestRegressor

from sklearn.preprocessing import LabelEncoder

import pickle

# Load dataset

df = pd.read\_csv("traffic volume.csv")

# Preprocessing (example)

df = df.dropna()

le = LabelEncoder()

df['weather\_main'] = le.fit\_transform(df['weather\_main'])

# Feature and target

X = df[['temp', 'rain\_1h', 'snow\_1h', 'clouds\_all', 'weather\_main']]

y = df['traffic\_volume']

# Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# Train model

model = RandomForestRegressor()

model.fit(X\_train, y\_train)

# Save model and encoder

pickle.dump(model, open('Flask/model.pkl', 'wb'))

pickle.dump(le, open('Flask/encoder.pkl', 'wb'))

---

✅ 2. Flask Web App

File: Flask/app.py

from flask import Flask, request, render\_template

import pickle

import numpy as np

app = Flask(\_name\_)

model = pickle.load(open('model.pkl', 'rb'))

encoder = pickle.load(open('encoder.pkl', 'rb'))

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict', methods=['POST'])

def predict():

features = [float(x) for x in request.form.values()]

features[-1] = encoder.transform([features[-1]])[0] # encode 'weather\_main'

final\_features = np.array([features])

prediction = model.predict(final\_features)

return render\_template('index.html', prediction\_text=f'Traffic Volume: {int(prediction[0])}')

if \_name\_ == "\_main\_":

app.run(debug=True)

---

✅ 3. HTML Form

File: Flask/templates/index.html

<!DOCTYPE html>

<html>

<head><title>Traffic Volume Predictor</title></head>

<body>

<h2>Enter the details</h2>

<form method="POST" action="/predict">

Temperature: <input type="text" name="temp"><br>

Rain 1h: <input type="text" name="rain\_1h"><br>

Snow 1h: <input type="text" name="snow\_1h"><br>

Clouds: <input type="text" name="clouds\_all"><br>

Weather: <input type="text" name="weather\_main"><br>

<input type="submit" value="Predict">

</form>

<h3>{{ prediction\_text }}</h3>

</body>

</html>

---

✅ 4. Requirements File

File: Requirements.txt

Flask

pandas

scikit-learn

numpy

---

✅ 5. IBM Watson Integration (Optional)Add commentMore actions

In IBM/traffic\_volume\_lbm\_scoring\_end\_point.ipynb, you use IBM Watson ML API to deploy your model. Here's a basic template:

import requests

import json