**EVAT Blackspot Prediction API - Detailed Documentation**

**1. Introduction**

The EVAT Blackspot Prediction API is designed to predict traffic blackspots based on location data (latitude, longitude) and other relevant features. The primary objective is to integrate this API with the EVAT mobile application to provide real-time blackspot prediction and visualization, enhancing user safety and experience.

**2. Objective**

The purpose of the API is to predict whether a given location is a blackspot based on:

* Latitude and Longitude
* Traffic Count
* Nearest Charger Distance

**3. How the API Works**

**Input:**

* **Latitude and Longitude:** Mandatory input for the prediction.
* **Optional Parameters:**
  + Traffic Count
  + Distance to Nearest Charger

**Output:**

* **Prediction:** Whether the location is a blackspot (True/False).
* **Risk Level:** High, Medium, or Low.
* **Nearest Charger Distance:** Calculated dynamically.
* **Explanation:** Reason for blackspot prediction.

**4. Working Logic**

**Data Source:**

The API uses preprocessed traffic and charging station data stored in CSV files:

* **Traffic Data:** Contains latitude, longitude, traffic count, and other features.
* **Charger Data:** Contains latitude and longitude of EV charging stations.

**Real-Time Data Fetching:**

The API dynamically computes:

* **Traffic Count:** Based on the nearest traffic data point.
* **Nearest Charger Distance:** Uses a KDTree for fast lookup of the closest charging station.

**Prediction Logic:**

The API uses a pre-trained **Random Forest model** loaded from a file (best\_blackspot\_model.pkl). The model predicts the risk level and outputs the following:

* Blackspot status (True/False)
* Risk Level (High/Low)
* Confidence Score

**5. Complete API Code**

from flask import Flask, request, jsonify

import joblib

import pandas as pd

import numpy as np

from scipy.spatial import cKDTree

from geopy.distance import geodesic

# Load model and data

model = joblib.load("best\_blackspot\_model.pkl")

traffic\_data = pd.read\_csv("processed\_traffic\_data.csv")

chargers\_data = pd.read\_csv("charging\_stations.csv")

# Build KDTree for charger locations

charger\_coords = chargers\_data[['Latitude', 'Longitude']].values

charger\_tree = cKDTree(charger\_coords)

app = Flask(\_\_name\_\_)

def nearest\_charger\_distance(lat, lon):

dist, idx = charger\_tree.query([[lat, lon]], k=1)

nearest\_lat, nearest\_lon = charger\_coords[idx[0]]

return geodesic((lat, lon), (nearest\_lat, nearest\_lon)).km

@app.route('/')

def home():

return "🚦 Blackspot Prediction API is running!"

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

def predict():

try:

data = request.get\_json()

latitude = float(data['latitude'])

longitude = float(data['longitude'])

traffic\_count = get\_nearest\_traffic\_count(latitude, longitude)

nearest\_charger\_dist = nearest\_charger\_distance(latitude, longitude)

features = np.array([[latitude, longitude, traffic\_count, nearest\_charger\_dist]])

prediction = model.predict(features)[0]

risk\_level = "High" if prediction == 1 else "Low"

response = {

"latitude": latitude,

"longitude": longitude,

"traffic\_count": traffic\_count,

"nearest\_charger\_distance": nearest\_charger\_dist,

"is\_blackspot": bool(prediction),

"risk\_level": risk\_level,

"message": "High-risk area!" if prediction == 1 else "Safe area."

}

return jsonify(response)

except Exception as e:

return jsonify({"error": str(e)})

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, host='0.0.0.0', port=5000)

**6. Integration with EVAT App**

The API integrates seamlessly with the EVAT app using RESTful communication:

* **Frontend:** Uses a map interface to capture user location and sends coordinates to the API.
* **Backend:** Calls the API with location data and displays predictions on the map.

**7. Benefits of the API Integration**

* **Real-Time Safety Alerts:** Warns users of high-risk areas.
* **Enhanced User Experience:** Offers a dynamic safety feature.
* **Scalability:** Easily expandable to include new risk factors.

**8. How to Deploy the API**

1. Install Flask:

pip install flask

1. Run the API:

python blackspot\_api.py

1. Test the API:

curl -X POST http://localhost:5000/predict -H "Content-Type: application/json" -d '{"latitude": -37.8136, "longitude": 144.9631}'

**9. Conclusion**

The EVAT Blackspot Prediction API is a powerful tool that enhances the safety and usability of the EVAT app. With real-time predictions and seamless integration, it significantly contributes to a safer driving experience for EV users.