# Import necessary libraries

import pandas as pd

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

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error

import datetime

# Load historical traffic data (you would need a dataset for this)

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

# Preprocess the data (feature engineering, data cleaning, etc.)

def preprocess\_data(data):

# Add date and time features

data['datetime'] = pd.to\_datetime(data['timestamp'])

data['hour'] = data['datetime'].dt.hour

data['day\_of\_week'] = data['datetime'].dt.dayofweek

data['month'] = data['datetime'].dt.month

data['year'] = data['datetime'].dt.year

# Drop unnecessary columns

data = data.drop(['timestamp', 'datetime'], axis=1)

return data

processed\_data = preprocess\_data(traffic\_data)

# Split the data into training and testing sets

X = processed\_data.drop('traffic\_volume', axis=1)

y = processed\_data['traffic\_volume']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train a machine learning model (Random Forest Regressor in this case)

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

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

# Evaluate the model

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

# Use the model for traffic forecasting (you can input your own data)

def forecast\_traffic(data):

data = preprocess\_data(data)

traffic\_forecast = model.predict(data)

return traffic\_forecast

# Example usage:

new\_data = pd.DataFrame({'timestamp': ['2023-09-11 12:00:00'], 'other\_features': [values]})

predicted\_traffic = forecast\_traffic(new\_data)

print(f"Predicted Traffic Volume: {predicted\_traffic[0]}")