**Step 1: Define Variables**

1. **Dependent Variable (Target):** Accident Severity
   * This variable represents the severity level of an accident, typically categorized as fatal, serious, or minor.
2. **Independent Variables (Features):**
   * **Weather Condition**: Weather conditions at the time of the accident (e.g., clear, rainy).
   * **Lighting Condition**: Whether the accident happened during the day, at dusk, or at night.
   * **Road Condition**: Conditions of the road (e.g., dry, wet, icy).
   * **Vehicle Speed**: Speed of the vehicle at the time of the accident.
   * **Number of Vehicles Involved**: Total number of vehicles in the accident.
   * **Driver Age and Experience**: Driver's age and driving experience.
   * **Alcohol Involvement**: Whether alcohol was involved in the accident.
   * **Urban or Rural Area**: Whether the accident occurred in an urban or rural setting.

**Step 2: Load Dataset and Preprocess**

Assume you have a dataset named accidents.csv which contains these fields.

python

Copy code

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import joblib

# Load dataset

data = pd.read\_csv('accidents.csv')

# Define dependent and independent variables

X = data[['Weather Condition', 'Lighting Condition', 'Road Condition',

'Vehicle Speed', 'Number of Vehicles Involved', 'Driver Age',

'Driver Experience', 'Alcohol Involvement', 'Urban or Rural']]

y = data['Accident Severity']

# Split the data into training and testing sets

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

**Step 3: Build the Linear Regression Model**

python

Copy code

# Initialize the model

model = LinearRegression()

# Train the model

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

# Test the model

predictions = model.predict(X\_test)

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

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

**Step 4: Save the Model for Future Use**

python

Copy code

# Save the model

joblib.dump(model, 'accident\_severity\_model.pkl')

**Step 5: Example Prediction**

Let's predict accident severity using a hypothetical set of independent variables.

python

Copy code

# Load the model

loaded\_model = joblib.load('accident\_severity\_model.pkl')

# Define hypothetical values for a new accident scenario

new\_data = pd.DataFrame({

'Weather Condition': [1], # E.g., 1 for clear weather

'Lighting Condition': [0], # E.g., 0 for daylight

'Road Condition': [1], # E.g., 1 for dry road

'Vehicle Speed': [60], # Speed in km/h

'Number of Vehicles Involved': [2],

'Driver Age': [30],

'Driver Experience': [5], # Years of driving experience

'Alcohol Involvement': [0], # 0 for no alcohol involved

'Urban or Rural': [1] # 1 for urban

})

# Predict the severity

predicted\_severity = loaded\_model.predict(new\_data)

print(f"Predicted Accident Severity: {predicted\_severity[0]}")

**Model Benefits for Traffic Accident Analysis in Underdeveloped Countries**

This linear regression model can provide insights into the contributing factors for road accidents, allowing authorities to implement data-driven safety measures. For instance:

1. **Resource Allocation**: Identifying high-risk areas (e.g., poor road or lighting conditions) for improvement.
2. **Policy Making**: Targeted regulations for speed limits, alcohol checks, and road safety campaigns based on critical factors identified by the model.
3. **Educational Campaigns**: Awareness about the common causes of severe accidents, focusing on issues prevalent in underdeveloped regions like poor lighting and road conditions.

**Uploading to GitHub**

1. After implementing the above code, create a GitHub repository.
2. Add the code files and trained model (accident\_severity\_model.pkl) to the repository.
3. Push the changes to GitHub.
4. Share the GitHub repository URL.