# Import libraries

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

import joblib

# Load dataset

data = pd.read\_csv("//content/sample\_data/RTA Dataset.csv.zip")

# Convert 'Time' to seconds since midnight

def time\_to\_seconds(time\_str):

    if pd.isna(time\_str):

        return None

    h, m, s = map(int, time\_str.split(':'))

    return h \* 3600 + m \* 60 + s

data['Time\_seconds'] = data['Time'].apply(time\_to\_seconds)

# Define dependent and independent variables

X = data[['Time\_seconds', 'Age\_band\_of\_driver', 'Weather\_conditions','Cause\_of\_accident','Road\_surface\_conditions','Vehicle\_movement']]

y = data['Accident\_severity']

# Handle categorical variables by one-hot encoding for features

X = pd.get\_dummies(X, columns=['Age\_band\_of\_driver', 'Weather\_conditions','Cause\_of\_accident', 'Road\_surface\_conditions', 'Vehicle\_movement'], drop\_first=True)

# Handle categorical target variable by one-hot encoding

y = pd.get\_dummies(y, columns=['Accident\_severity'], drop\_first=True)

# Split dataset

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=50)

# Create model

model = LinearRegression()

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

# Save model for future use

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

# Evaluate model

score = model.score(X\_test, y\_test)

print(f"Model R² Score: {score:.2f}")

# Load saved model

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

# Hypothetical input

hypothetical\_input\_data = {

    'Time\_seconds': [36000], # Example: 10:00:00 AM

    'Age\_band\_of\_driver': ['18-30'],

    'Weather\_conditions': ['Normal'],

    'Cause\_of\_accident': ['No distancing'],

    'Road\_surface\_conditions': ['Dry'], # Added Road\_surface\_conditions to hypothetical input

    'Vehicle\_movement': ['Going straight']

}

hypothetical\_input = pd.DataFrame(hypothetical\_input\_data)

# Apply one-hot encoding to hypothetical input, ensuring all columns from training data are present

hypothetical\_input = pd.get\_dummies(hypothetical\_input, columns=['Age\_band\_of\_driver', 'Weather\_conditions','Cause\_of\_accident', 'Road\_surface\_conditions', 'Vehicle\_movement'], drop\_first=True)

# Reindex hypothetical input to match the columns of the training data (X\_train) and fill missing with 0

hypothetical\_input = hypothetical\_input.reindex(columns=X\_train.columns, fill\_value=0)

# Prediction

predicted\_severity = model.predict(hypothetical\_input)

# Print predicted values for each category

for i, col in enumerate(y\_train.columns):

    print(f"Predicted probability for {col}: {predicted\_severity[0][i]:.2f}")