

colab.research.google.com/drive/1JJHE7d0CQ8M31Ck1BED6EluD8u_zdBaa?authuser=0#scrollTo=MONB-5G7M-20

Untitled1.ipynb

File Edit View Insert Runtime Tools Help

Files

Analyse your files with code written by Gemini Upload

Kenya Road Accidents Dataset...

Double-click (or enter) to edit

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
import joblib

# Load the dataset
file_path = "/content/drive/MyDrive/Colab CSV Files/Kenya Road Accidents Dataset.csv"
df = pd.read_csv(file_path, delimiter="\t")

# Define dependent and independent variables
X = df.drop(columns=["Accident_ID", "Accident_Severity_(1_10)"]) # Features
y = df["Accident_Severity_(1_10)"] # Target

# Identify categorical and numerical features
categorical_features = ["Weather", "Road_Type", "Time_of_Day", "Vehicle_Type", "Road_Condition", "Location"]
numerical_features = ["Speed_Limit_(km/h)", "Driver_Age"]

# Preprocessing: One-hot encoding for categorical variables
preprocessor = ColumnTransformer(
    transformers=[
        ("num", "passthrough", numerical_features),
        ("cat", OneHotEncoder(), categorical_features)
    ]
)

# Create a pipeline with preprocessing and regression model
```

0s completed at 04:24

```
"Speed_Limit_(km/h)": [90],
"Driver_Age": [30],
"Time_of_Day": ["Morning"],
"Vehicle_Type": ["Car"],
"Road_Condition": ["Paved"],
"Location": ["Nairobi"]
})
```

0s completed at 04:24

colab.research.google.com/drive/1JJHE7d0CQ8M31Ck18ED6EluD8u_zdBaa?authuser=0#scrollTo=MONB-5G7M-20

Untitled1.ipynb

File Edit View Insert Runtime Tools Help

Files

Analyse your files with code written by Gemini Upload

Kenya Road Accidents Dataset...

Commands + Code + Text

```
# Split 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)

# Train the model
model.fit(X_train, y_train)

# Save the trained model for future use
joblib.dump(model, "road_accident_severity_model.pkl")

# Define a hypothetical accident scenario for prediction
hypothetical_data = pd.DataFrame({
    "Weather": ["Rainy"],
    "Road_Type": ["Highway"],
    "Speed_Limit_(km/h)": [90],
    "Driver_Age": [30],
    "Time_of_Day": ["Morning"],
    "Vehicle_Type": ["Car"],
    "Road_Condition": ["Paved"],
    "Location": ["Nairobi"]
})

# Ensure hypothetical data goes through the same transformation pipeline
predicted_severity = model.predict(hypothetical_data)

# Display the prediction result
print(f"Predicted Accident Severity: {predicted_severity[0]:.2f}")
```

Predicted Accident Severity: 7.38

0s completed at 04:24

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
import joblib

# Load the dataset
file_path = "/content/drive/MyDrive/Colab CSV Files/Kenya Road Accidents Dataset.csv"
df = pd.read_csv(file_path, delimiter="\t")

# Define dependent and independent variables
X = df.drop(columns=["Accident_ID", "Accident_Severity_(1_10)"]) # Features
y = df["Accident_Severity_(1_10)"] # Target

# Identify categorical and numerical features
categorical_features = ["Weather", "Road_Type", "Time_of_Day", "Vehicle_Type", "Road_Condition",
                        "Location"]
numerical_features = ["Speed_Limit_(km/h)", "Driver_Age"]

# Preprocessing: One-hot encoding for categorical variables
preprocessor = ColumnTransformer(
    transformers=[
        ("num", "passthrough", numerical_features),
        ("cat", OneHotEncoder(handle_unknown="ignore"), categorical_features) # Handle unknown
        categories
    ]
)

# Create a pipeline with preprocessing and regression model
model = Pipeline(steps=[
    ("preprocessor", preprocessor),
```

```
("regressor", LinearRegression())
])

# Split 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)

# Train the model
model.fit(X_train, y_train)

# Save the trained model for future use
joblib.dump(model, "road_accident_severity_model.pkl")

# Define a hypothetical accident scenario for prediction
hypothetical_data = pd.DataFrame({
    "Weather": ["Rainy"],
    "Road_Type": ["Highway"],
    "Speed_Limit_(km/h)": [90],
    "Driver_Age": [30],
    "Time_of_Day": ["Morning"],
    "Vehicle_Type": ["Car"],
    "Road_Condition": ["Paved"],
    "Location": ["Nairobi"]
})

# Ensure hypothetical data goes through the same transformation pipeline
predicted_severity = model.predict(hypothetical_data)

# Display the prediction result
print(f'Predicted Accident Severity: {predicted_severity[0]:.2f}')
```

Benefits of a Road Accident Severity Prediction Model for Underdeveloped Countries

1. **Data-Driven Decision Making** – Helps authorities identify high-risk areas and allocate resources efficiently.
2. **Improved Emergency Response** – Enables faster ambulance dispatch and hospital preparedness for severe cases.
3. **Enhanced Road Safety Regulations** – Supports better traffic laws, safer speed limits, and awareness campaigns.
4. **Cost Reduction** – Reduces healthcare, vehicle repair, and insurance costs for governments and citizens.
5. **Better Infrastructure Planning** – Identifies unsafe roads for improved construction, traffic signals, and safety measures.
6. **Support for Law Enforcement** – Helps police monitor accident-prone areas and enforce road safety rules effectively.
7. **Accident Prevention Alerts** – Can be integrated into navigation apps to warn drivers about dangerous road conditions.