Enhancing road safety through AI driven Traffic analysis and Prediction

Source code:

```
#Upload the dataset

From google.colab import files

Uploaded = files.upload()

Import pandas as pd

# Load the dataset

Df = pd.read_csv('RTA Dataset.csv')

# Basic inspection

Print(df.shape)

Df.head()

Print("columns:",df.columns.tolist())

Df.info()

Df.describe()

#import necessary libraries

Import pandas as pd

Import numpy as np

Import matplotlib.pyplot as plt

Import seaborn as sns

From sklearn.model_selection import train_test_split

From sklearn.preprocessing import LabelEncoder, StandardScaler

From sklearn.ensemble import RandomForestClassifier

From sklearn.metrics import classification_report, confusion_matrix

# Handle missing values

Df.dropna(inplace=True)

# Encode categorical columns

Le = LabelEncoder()
```

```
Categorical_cols = df.select_dtypes(include='object').columns
For col in categorical_cols:
 Df[col] = le.fit_transform(df[col])
# Define features and target
X = df.drop('Accident_severity', axis=1)
Y = df['Accident_severity']
# Normalize/standardize (optional)
Scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
Random_state=42, stratify=y)
Model = RandomForestClassifier(random_state=42)
Model.fit(X_train, y_train)
Y_pred = model.predict(X_test)
Print("Classification Report:\n", classification_report(y_test, y_pred))
Sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')
Plt.title("Confusion Matrix")
Plt.show()
!pip install gradio
Import gradio as gr
Def predict_severity(*inputs):
 Input_array = np.array(inputs).reshape(1, -1)
Input_scaled = scaler.transform(input_array)
 Prediction = model.predict(input_scaled)
 Return f"Predicted Severity: {prediction[0]}"
Input_features = list(X.columns)
Interface = gr.Interface(fn=predict_severity,
 Inputs=[gr.Number(label=col) for col in input_features],
Outputs="text",
```

Title=”Traffic Accident Severity Predictor”)

Interface.launch(share=True,debug=True)