# STEP 1: Upload your .xlsb file

from google.colab import files

uploaded = files.upload()  # Select your file: road\_accident\_dataset.xlsb

# STEP 2: Install the package to read .xlsb files

!pip install pyxlsb

# STEP 3: Import required libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import classification\_report, confusion\_matrix

# STEP 4: Load the .xlsb file

df = pd.read\_excel('road\_accident\_dataset.xlsb', engine='pyxlsb')

df.head()

# STEP 5: Basic Preprocessing

# Drop missing values

df = df.dropna()

# Encode categorical variables

label\_encoders = {}

for col in df.select\_dtypes(include='object').columns:

    le = LabelEncoder()

    df[col] = le.fit\_transform(df[col])

    label\_encoders[col] = le

df.head()

# STEP 6: Exploratory Data Analysis

print("Data Summary:")

print(df.describe())

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title("Feature Correlation Heatmap")

plt.show()

# STEP 7: Split Features and Target

# Assuming last column is the target

X = df.iloc[:, :-1]

y = df.iloc[:, -1]

print("Features:", X.columns.tolist())

print("Target:", y.name)

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error

import matplotlib.pyplot as plt

# Sample dataset

data = pd.DataFrame({

    'feature1': [1, 2, 3, 4, 5],

    'feature2': [5, 4, 3, 2, 1],

    'target': [1, 3, 5, 7, 9]

})

# Split features and target

X = data[['feature1', 'feature2']]

y = data['target']

# Split train and test

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

# Train model

model = LinearRegression()

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

# Predict

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

# Evaluate

print("Mean Absolute Error (MAE):", mean\_absolute\_error(y\_test, y\_pred))

print("Mean Squared Error (MSE):", mean\_squared\_error(y\_test, y\_pred))

print("Root Mean Squared Error (RMSE):", np.sqrt(mean\_squared\_error(y\_test, y\_pred)))

# Plot top features

coefficients = pd.Series(model.coef\_, index=X.columns)

coefficients.abs().nlargest(2).plot(kind='barh', color='skyblue')

plt.title("Top Important Features (by Coefficient)")

plt.xlabel("Coefficient Value")

plt.ylabel("Feature")

plt.show()

coefficients = pd.Series(model.coef\_, index=X.columns)

coefficients.abs().nlargest(10).plot(kind='barh', color='skyblue')

plt.title("Top 10 Important Features (by Absolute Coefficient Value)")

plt.xlabel("Coefficient Magnitude")

plt.ylabel("Feature")

plt.show()