

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix


def main():

    # Load dataset - update 'road_traffic_accidents.csv' with your actual dataset filename/path
    try:

        data = pd.read_csv('road_traffic_accidents.csv')

    except FileNotFoundError:

        print("Dataset file 'road_traffic_accidents.csv' not found. Please make sure the file exists.")

        return


    # Basic data inspection

    print("Dataset Info:")

    print(data.info())

    print("\nFirst 5 rows:")

    print(data.head())


    # Check for missing values

    if data.isnull().sum().any():

        # Basic missing value imputation: forward fill, then backward fill as fallback

        data.fillna(method='ffill', inplace=True)

        data.fillna(method='bfill', inplace=True)

        print("\nMissing values detected and filled with forward/backward fill method.")
```

```
# Define features and target

# Replace 'Accident_severity' with the correct target column name from your dataset
target_column = 'Accident_severity'

if target_column not in data.columns:

    print(f"Target column '{target_column}' not found in dataset.")

    return


X = data.drop(columns=[target_column])

y = data[target_column]


# Convert categorical variables to dummy variables if any exist
X = pd.get_dummies(X)


# 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, stratify=y)


# Initialize and train Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)


# Predict on test set
y_pred = model.predict(X_test)


# Evaluate predictions
```

```
print("\nConfusion Matrix:")
```

```
print(confusion_matrix(y_test, y_pred))
```

```
print("\nClassification Report:")
```

```
print(classification_report(y_test, y_pred))
```

```
if __name__ == "__main__":
```

```
    main()
```