

## Source code

```
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

import seaborn as sns

import matplotlib.pyplot as plt


# 1. Sample dataset (embedded in code)

data = {

    'Speed_limit': [30, 40, 30, 50, 20, 60, 70, 30],

    'Weather_conditions': [1, 2, 1, 3, 1, 2, 3, 1],

    'Light_conditions': [1, 1, 2, 2, 1, 1, 2, 1],

    'Road_surface_conditions': [1, 2, 1, 2, 1, 2, 2, 1],

    'Urban_or_rural_area': [1, 2, 1, 2, 1, 2, 2, 1],

    'Accident_Severity': [0, 1, 0, 2, 0, 1, 2, 0] # 0: Slight, 1: Serious, 2: Fatal

}


df = pd.DataFrame(data)


# 2. Prepare features and target

X = df.drop('Accident_Severity', axis=1)

y = df['Accident_Severity']


# 3. Train/test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
```



## Source code

# 4. Train the model

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
```

# 5. Predict

```
y_pred = model.predict(X_test)
```

# 6. Evaluation

```
print("=== Classification Report ===")
```

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

```
print("=== Confusion Matrix ===")
```

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

# 7. Feature importance

```
importances = model.feature_importances_
```

```
features = X.columns
```

```
feat_imp = pd.Series(importances, index=features).sort_values(ascending=False)
```

```
print("\n=== Feature Importances ===")
```

```
print(feat_imp)
```

# 8. Plot feature importance

```
sns.barplot(x=feat_imp, y=feat_imp.index)
```



### Source code

```
plt.title("Feature Importance in Accident Severity Prediction")  
plt.xlabel("Importance Score")  
plt.ylabel("Feature")  
plt.tight_layout()  
plt.show()
```

