Default title text

```
# @title Default title text
from google.colab import files
uploaded = files.upload()
```

Choose Files datafile.xls

- **datafile.xls**(application/vnd.ms-excel) - 842752 bytes, last modified: 5/2/2025 - 100% done

Saving datafile.xls to datafile (1).xls

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
df = pd.read_csv('traffic_accidents.csv')
print("Dataset Preview:")
print(df.head())
```

```
Dataset Preview:
  Reference Number  Grid Ref: Easting  Grid Ref: Northing  Number of Vehicles  \
0          1112091             429175              431904                   2
1          1180869             430429              431025                   3
2          1180869             430429              431025                   3
3          11A0238             424660              427582                   3
4          11A0238             424660              427582                   3

   Number of Casualties Accident Date  Time (24hr)  1st Road Class  \
0                     1    01/01/2014         1840               6
1                     2    08/01/2014         1430               1
2                     2    08/01/2014         1430               1
3                     2    10/01/2014          817               1
4                     2    10/01/2014          817               1

   Road Surface  Lighting Conditions  Weather Conditions  Casualty Class  \
0             2                    4                   2               1
1             2                    1                   1               1
2             2                    1                   1               1
3             1                    1                   1               1
4             1                    1                   1               1

   Casualty Severity  Sex of Casualty  Age of Casualty  Type of Vehicle
0                  3                1               58                9
1                  3                1               69                9
2                  3                2               41                9
3                  3                1               35                9
4                  3                1               25                9
```

```
df.dropna(thresh=len(df)*0.5, axis=1, inplace=True)
df.fillna(df.median(numeric_only=True), inplace=True)
df.fillna(df.mode().iloc[0], inplace=True)
```

```
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
```

```
print([col for col in df.columns if 'accident' in col.lower()])
```

```
['Accident Date']
```

```
print("Available columns:", df.columns.tolist())
target_col = 'Accident_Severity'  # use exact match
```

```
Casualties', 'Accident Date', 'Time (24hr)', '1st Road Class', 'Road Surface', 'Lighting Conditions', 'Weather Conditions', 'Casualty C
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-32-3ab15314309e> in <cell line: 0>()
----> 1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

NameError: name 'X' is not defined
```

Next steps: Explain error

```
# 1. Upload the dataset (Google Colab)
from google.colab import files
uploaded = files.upload()

# 2. Import required libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# 3. Load the dataset
df = pd.read_csv('traffic_accidents.csv')  # Replace with your filename
print("Columns in dataset:", df.columns.tolist())  # Check column names

# 4. Handle missing values
df.dropna(thresh=len(df) * 0.5, axis=1, inplace=True)  # Drop columns with >50% missing
df.fillna(df.median(numeric_only=True), inplace=True)  # Fill numeric NaNs with median
df.fillna(df.mode().iloc[0], inplace=True)  # Fill remaining NaNs with mode

# 5. Encode categorical columns
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# 6. Define target column using correct name from your dataset
target_col = 'Casualty Severity'
if target_col not in df.columns:
    raise ValueError(f"'{target_col}' column not found. Available columns: {df.columns.tolist()}")

# 7. Split features and labels
X = df.drop(target_col, axis=1)
y = df[target_col]

# 8. Feature scaling
scaler = StandardScaler()
X = scaler.fit_transform(X)

# 9. Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 10. Train model
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)

# 11. Predict and evaluate
y_pred = clf.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# 12. Confusion matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# 13. Feature importances
importances = clf.feature_importances_
feature_names = df.drop(target_col, axis=1).columns

plt.figure(figsize=(10, 6))
sns.barplot(x=importances, y=feature_names)
plt.title('Feature Importances')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.tight_layout()
plt.show()
```

Choose Files dataset.csv

- **dataset.csv**(text/csv) - 157246 bytes, last modified: 5/12/2025 - 100% done

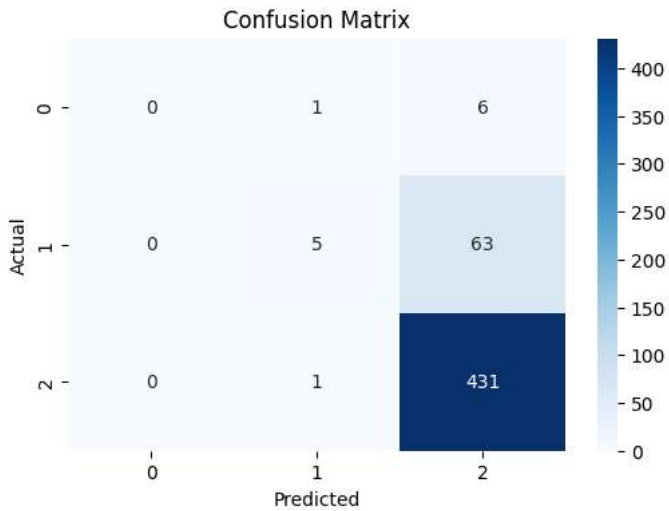
```
Saving dataset.csv to dataset (1).csv
Columns in dataset: ['Reference Number', 'Grid Ref: Easting', 'Grid Ref: Northing', 'Number of Vehicles', 'Number of Casualties', 'Accic
Accuracy: 0.8599605522682445

Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         7
           2       0.71      0.07      0.13        68
           3       0.86      1.00      0.92       432

    accuracy                           0.86       507
   macro avg       0.53      0.36      0.35       507
weighted avg       0.83      0.86      0.81       507

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and be
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```



Feature Importances