```
import pandas as pd
from google.colab import files
uploaded= files.upload()
df=pd.read_csv("guna.csv")
print(df)
print(df.isnull().sum())
```

Choose files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving guna.csv to guna.csv
          Time Day_of_week Age_band_of_driver Sex_of_driver  \
0      17:02:00      Monday              18-30          Male
1      17:02:00      Monday              31-50          Male
2      17:02:00      Monday              18-30          Male
3      01:06:00      Sunday              18-30          Male
4      01:06:00      Sunday              18-30          Male
...         ...         ...                ...           ...
12311  16:15:00   Wednesday              31-50          Male
12312  18:00:00      Sunday            Unknown          Male
12313  13:55:00      Sunday            Over 51          Male
12314  13:55:00      Sunday              18-30        Female
12315  13:55:00      Sunday              18-30          Male

        Educational_level Vehicle_driver_relation Driving_experience  \
0       Above high school                Employee             1-2yr
1      Junior high school                Employee         Above 10yr
2      Junior high school                Employee             1-2yr
3      Junior high school                Employee            5-10yr
4      Junior high school                Employee             2-5yr
...                   ...                     ...               ...
12311                 NaN                Employee             2-5yr
12312   Elementary school                Employee            5-10yr
12313  Junior high school                Employee            5-10yr
12314  Junior high school                Employee         Above 10yr
12315  Junior high school                Employee            5-10yr

           Type_of_vehicle Owner_of_vehicle Service_year_of_vehicle  ...  \
0              Automobile            Owner              Above 10yr  ...
1      Public (> 45 seats)            Owner                 5-10yrs  ...
2          Lorry (41?100Q)            Owner                     NaN  ...
3      Public (> 45 seats)     Governmental                     NaN  ...
4                      NaN            Owner                 5-10yrs  ...
...                    ...              ...                     ...  ...
12311       Lorry (11?40Q)            Owner                     NaN  ...
12312          Automobile            Owner                     NaN  ...
12313               Bajaj            Owner                  2-5yrs  ...
12314      Lorry (41?100Q)            Owner                  2-5yrs  ...
12315               Other            Owner                  2-5yrs  ...

      Vehicle_movement   Casualty_class Sex_of_casualty Age_band_of_casualty  \
0       Going straight               na              na                    na
1       Going straight               na              na                    na
2       Going straight  Driver or rider            Male                 31-50
3       Going straight       Pedestrian          Female                 18-30
4       Going straight               na              na                    na
...                ...              ...             ...                   ...
12311   Going straight               na              na                    na
12312            Other               na              na                    na
12313            Other  Driver or rider            Male                 31-50
12314            Other               na              na                    na
12315         Stopping       Pedestrian          Female                     5

      Casualty_severity Work_of_casuality Fitness_of_casuality  \
0                    na              NaN                  NaN
1                    na              NaN                  NaN
2                     3           Driver                  NaN
3                     3           Driver               Normal
```

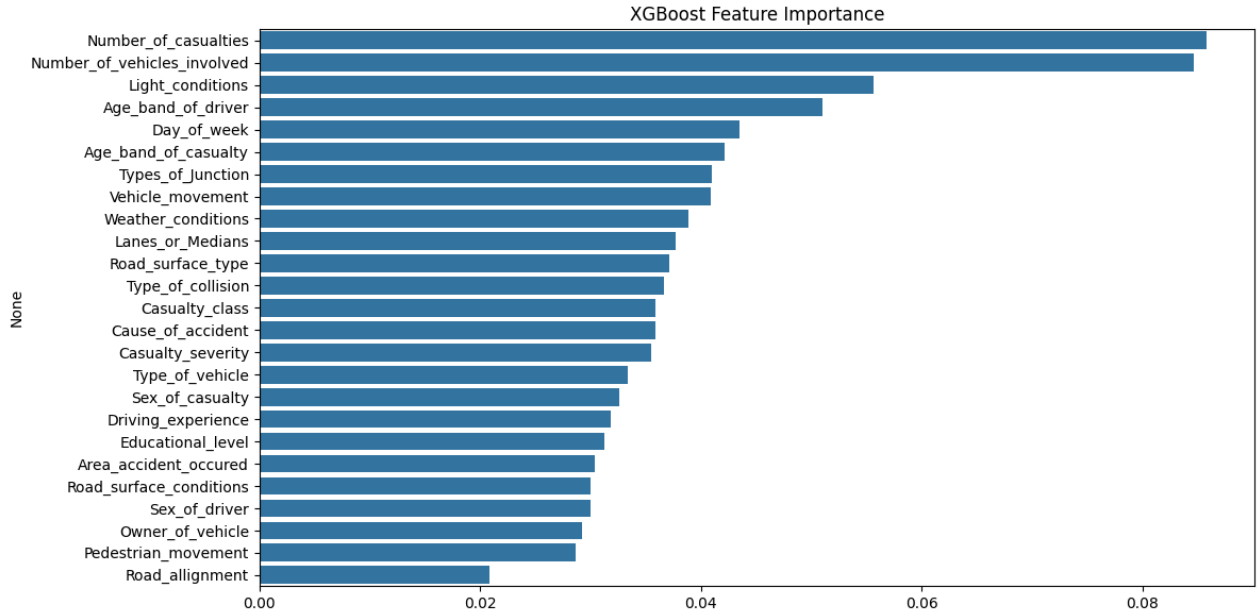
```
4                    na               NaN                    NaN
...                 ...               ...                    ...
12311                na            Driver                Normal
12312                na            Driver                Normal
12313                 3            Driver                Normal
12314                na            Driver                Normal
12315                 3            Driver                Normal

                                  Pedestrian_movement  \
0                                    Not a Pedestrian
1                                    Not a Pedestrian
2                                    Not a Pedestrian
3                                    Not a Pedestrian
4                                    Not a Pedestrian
...                                                ...
12311                                Not a Pedestrian
12312                                Not a Pedestrian
12313                                Not a Pedestrian
12314                                Not a Pedestrian
12315  Crossing from nearside - masked by parked or s...

                         Cause_of_accident Accident_severity
0                          Moving Backward     Slight Injury
1                                Overtaking     Slight Injury
2                 Changing lane to the left    Serious Injury
3                Changing lane to the right     Slight Injury
4                                Overtaking     Slight Injury
...                                     ...               ...
12311                       No distancing     Slight Injury
12312                       No distancing     Slight Injury
12313          Changing lane to the right    Serious Injury
12314  Driving under the influence of drugs     Slight Injury
12315          Changing lane to the right     Slight Injury

[12316 rows x 32 columns]
Time                            0
Day_of_week                     0
Age_band_of_driver              0
Sex_of_driver                   0
Educational_level             741
Vehicle_driver_relation       579
Driving_experience            829
Type_of_vehicle               950
Owner_of_vehicle              482
Service_year_of_vehicle      3928
Defect_of_vehicle            4427
Area_accident_occured         239
Lanes_or_Medians              385
Road_allignment               142
Types_of_Junction             887
Road_surface_type             172
Road_surface_conditions         0
Light_conditions                0
Weather_conditions              0
Type_of_collision             155
Number_of_vehicles_involved     0
Number_of_casualties            0
Vehicle_movement              308
Casualty_class                  0
Sex_of_casualty                 0
```

```
Age_band_of_casualty             0
Casualty_severity                0
Work_of_casualty              3198
Fitness_of_casualty           2635
Pedestrian_movement              0
Cause_of_accident                0
Accident_severity                0
dtype: int64
```

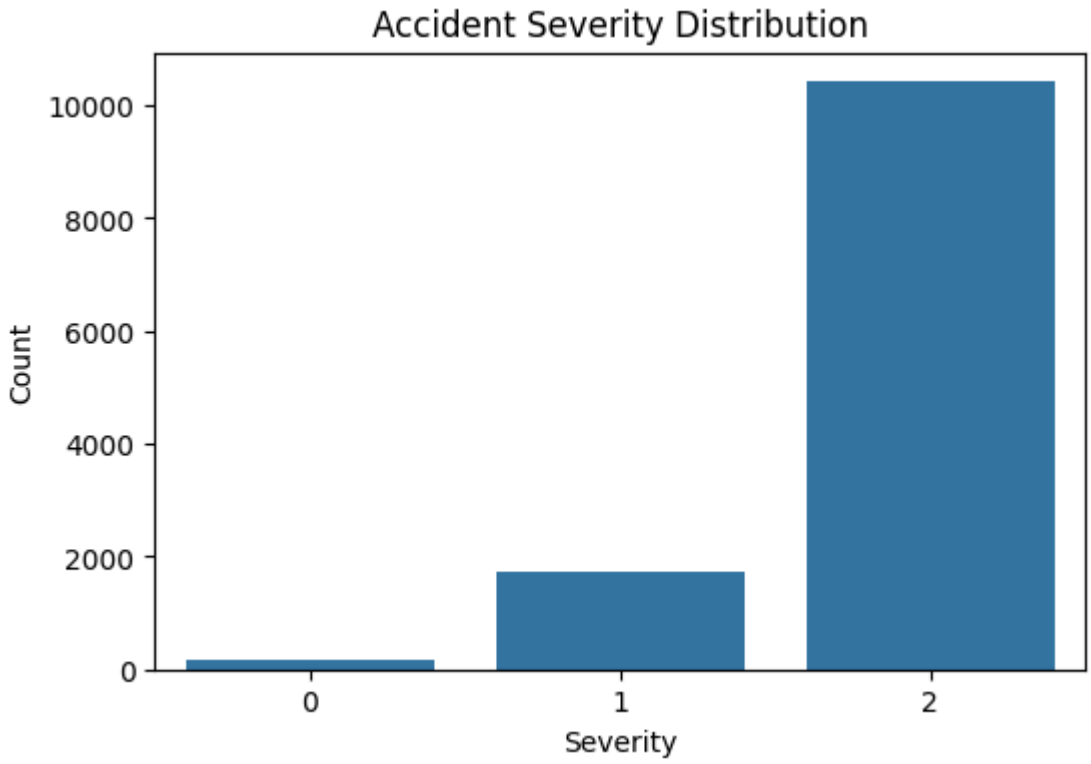
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
from xgboost import XGBClassifier
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv("guna.csv")

# Drop irrelevant or sparse columns
df.drop(columns=[
    'Work_of_casuality', 'Fitness_of_casuality', 'Service_year_of_vehicle',
    'Defect_of_vehicle', 'Time', 'Vehicle_driver_relation'
], inplace=True)

# Handle missing values
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

# 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

# Define features and target
X = df.drop('Accident_severity', axis=1)
y = df['Accident_severity']

# Scale numeric features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Handle class imbalance using sample weights
class_counts = y.value_counts()
total = len(y)
class_weights = {i: total / c for i, c in class_counts.items()}
sample_weights = y.map(class_weights)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

# Train the XGBoost model with class weights
model = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', num_class=3)
model.fit(X_train, y_train, sample_weight=sample_weights.iloc[y_train.index])

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

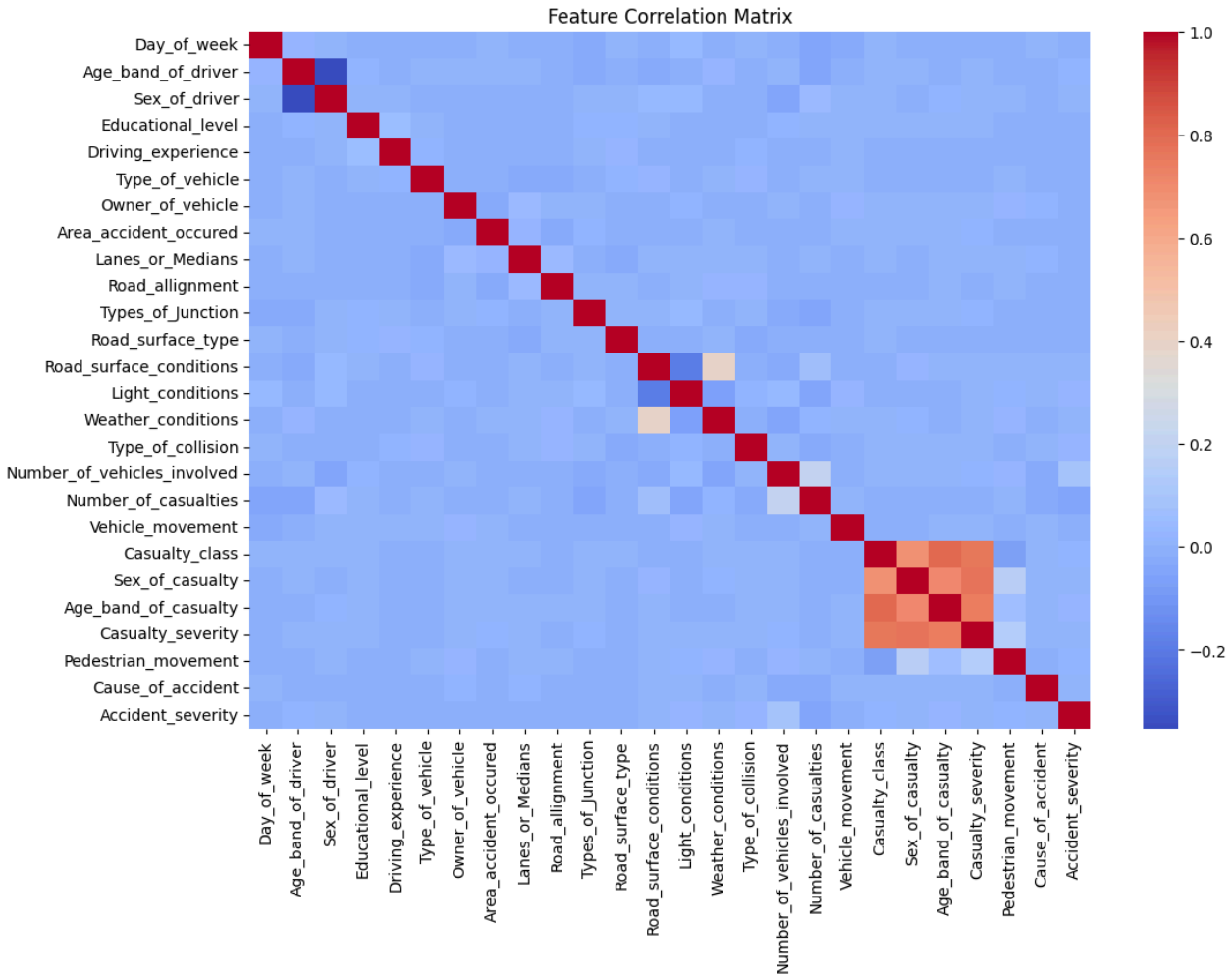
```
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))

    # Feature importance
    importances = model.feature_importances_
    features = X.columns
    indices = np.argsort(importances)[::-1]

    # Plot
    plt.figure(figsize=(12, 6))
    sns.barplot(x=importances[indices], y=features[indices])
    plt.title("XGBoost Feature Importance")
    plt.tight_layout()
    plt.show()
```

```
<ipython-input-18-eac69553470e>:21: FutureWarning: A value is trying to be set on
The behavior will change in pandas 3.0. This inplace method will never work becaus

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.metho


  df[col].fillna(df[col].mode()[0], inplace=True)
/usr/local/lib/python3.11/dist-packages/xgboost/core.py:158: UserWarning: [13:41:0
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
Confusion Matrix:
 [[   6   11   20]
 [   2  100  261]
 [  16  294 1754]]

Classification Report:
              precision    recall  f1-score   support

           0       0.25      0.16      0.20        37
           1       0.25      0.28      0.26       363
           2       0.86      0.85      0.86      2064

    accuracy                           0.75      2464
   macro avg       0.45      0.43      0.44      2464
weighted avg       0.76      0.75      0.76      2464
```



```
# Distribution of accident severity
plt.figure(figsize=(6,4))
sns.countplot(x='Accident_severity', data=df)
plt.title("Accident Severity Distribution")
plt.xlabel("Severity")
plt.ylabel("Count")
plt.show()

# Bivariate analysis: Speed vs Severity (if speed feature exists)
if 'Speed_limit' in df.columns:
    plt.figure(figsize=(6,4))
    sns.boxplot(x='Accident_severity', y='Speed_limit', data=df)
    plt.title("Speed Limit vs Accident Severity")
    plt.show()

# Correlation matrix
plt.figure(figsize=(12,8))
sns.heatmap(df.corr(), cmap='coolwarm', annot=False)
plt.title("Feature Correlation Matrix")
plt.show()
```


Accident Severity Distribution
Count
Severity
0
2000
4000
6000
8000
10000
0
1
2


Feature Correlation Matrix
Day_of_week
Age_band_of_driver
Sex_of_driver
Educational_level
Driving_experience
Type_of_vehicle
Owner_of_vehicle
Area_accident_occured
Lanes_or_Medians
Road_allignment
Types_of_Junction
Road_surface_type
Road_surface_conditions
Light_conditions
Weather_conditions
Type_of_collision
Number_of_vehicles_involved
Number_of_casualties
Vehicle_movement
Casualty_class
Sex_of_casualty
Age_band_of_casualty
Casualty_severity
Pedestrian_movement
Cause_of_accident
Accident_severity
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

```
import joblib

# Save trained model and scaler
joblib.dump(clf, 'random_forest_accident_model.pkl')
joblib.dump(scaler, 'scaler.pkl')
print("Model and scaler saved successfully.")
```

```
Model and scaler saved successfully.
```

```
# Load model and scaler
clf_loaded = joblib.load('random_forest_accident_model.pkl')
scaler_loaded = joblib.load('scaler.pkl')

# Example: New data input (replace with real values)
sample = pd.DataFrame([X.iloc[0]])  # using a sample from your dataset
sample_scaled = scaler_loaded.transform(sample)

# Predict
prediction = clf_loaded.predict(sample_scaled)
print("Predicted Severity:", prediction[0])
```

```
Predicted Severity: 2
```

```
!pip install streamlit
```

```
Collecting streamlit
  Downloading streamlit-1.45.0-py3-none-any.whl.metadata (8.9 kB)
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: blinker<2,>=1.5.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numpy<3,>=1.23 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: packaging<25,>=20 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: pillow<12,>=7.1.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: protobuf<7,>=3.20 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.11/dist-pa
```

```
Requirement already satisfied: tenacity<10,>=8.1.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: typing-extensions<5,>=4.4.0 in /usr/local/lib/python3.
Collecting watchdog<7,>=2.1.5 (from streamlit)
  Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl.metadata (44 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.3/44.3 kB 1.9 MB/s eta 0:00:00
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib/python3
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.9.1-py2.py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (fro
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: narwhals>=1.14.2 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/di
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
Downloading streamlit-1.45.0-py3-none-any.whl (9.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.9/9.9 MB 52.3 MB/s eta 0:00:00
Downloading pydeck-0.9.1-py2.py3-none-any.whl (6.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.9/6.9 MB 68.9 MB/s eta 0:00:00
Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl (79 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.1/79.1 kB 6.5 MB/s eta 0:00:00
Installing collected packages: watchdog, pydeck, streamlit
Successfully installed pydeck-0.9.1 streamlit-1.45.0 watchdog-6.0.0
```

```
import streamlit as st
import joblib
import pandas as pd

# Load model and scaler
model = joblib.load("random_forest_accident_model.pkl")
scaler = joblib.load("scaler.pkl")

st.title("Accident Severity Prediction")

# Input form
input_data = {}
features = ['Number_of_casualties', 'Number_of_vehicles_involved', 'Age_band_of_driver',
            'Sex_of_driver', 'Educational_level', 'Vehicle_type', 'Weather_conditions',
            'Road_surface_conditions', 'Light_conditions', 'Cause_of_accident']

for feature in features:
    input_data[feature] = st.number_input(f"Enter value for {feature}", value=0)
```

```
# Predict
if st.button("Predict Severity"):
    df_input = pd.DataFrame([input_data])
    df_scaled = scaler.transform(df_input)
    pred = model.predict(df_scaled)
    severity_map = {0: "Slight", 1: "Serious", 2: "Fatal"}
    st.success(f"Predicted Severity: {severity_map.get(pred[0], 'Unknown')}")
```

```
2025-05-07 14:17:03.440 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.444 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.446 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.446 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.447 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.448 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.449 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.450 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.450 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.451 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.452 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.452 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.453 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.453 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.454 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.455 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.455 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.456 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.456 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.457 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.458 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.458 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.459 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.459 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.460 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.460 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.461 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.462 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.462 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.463 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.463 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.464 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.465 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.465 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.466 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.466 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.467 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.468 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.468 Thread 'MainThread': missing ScriptRunContext! This warnin
2025-05-07 14:17:03.469 Thread 'MainThread': missing ScriptRunContext! This warnin
```

```
2025-05-07 14:17:03.475 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.476 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.477 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.477 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.478 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.478 Thread 'MainThread': missing ScriptRunContext! This warning
2025-05-07 14:17:03.479 Thread 'MainThread': missing ScriptRunContext! This warning
```

```
# Drop unused or sparse columns
df.drop(columns=[
    'Work_of_casuality', 'Fitness_of_casuality', 'Service_year_of_vehicle',
    'Defect_of_vehicle', 'Time', 'Vehicle_driver_relation'
], inplace=True, errors='ignore')

# Fill missing values
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

# Encode categorical variables
from sklearn.preprocessing import LabelEncoder

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

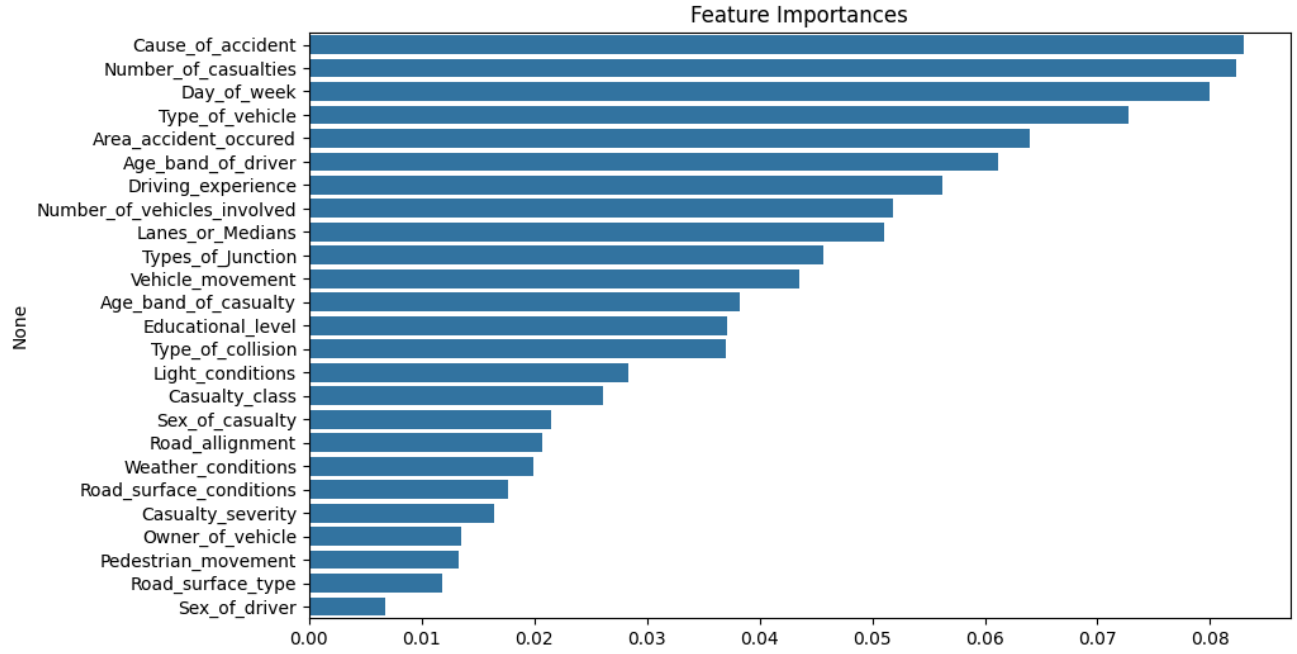
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.utils.class_weight import compute_class_weight

# Handle class imbalance
classes = y.unique()
weights = compute_class_weight('balanced', classes=classes, y=y)
class_weights = dict(zip(classes, weights))

clf = RandomForestClassifier(n_estimators=100, class_weight=class_weights, random_state=4
clf.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(class_weight={np.int64(0): np.float64(25.9831223628692),
                                     np.int64(1): np.float64(2.3553260661694395),
                                     np.int64(2): np.float64(0.39417506801088176)},
                       random_state=42)
```

```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

y_pred = clf.predict(X_test)
```

```
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))

    # Feature importance
    importances = clf.feature_importances_
    indices = importances.argsort()[::-1]
    feature_names = X.columns

    plt.figure(figsize=(10, 6))
    sns.barplot(x=importances[indices], y=feature_names[indices])
    plt.title("Feature Importances")
    plt.show()
```

```
Confusion Matrix:
 [[   0    0   37]
 [   0    3  360]
 [   0    1 2063]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        37
           1       0.75      0.01      0.02       363
           2       0.84      1.00      0.91      2064

    accuracy                           0.84      2464
   macro avg       0.53      0.34      0.31      2464
weighted avg       0.81      0.84      0.77      2464

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



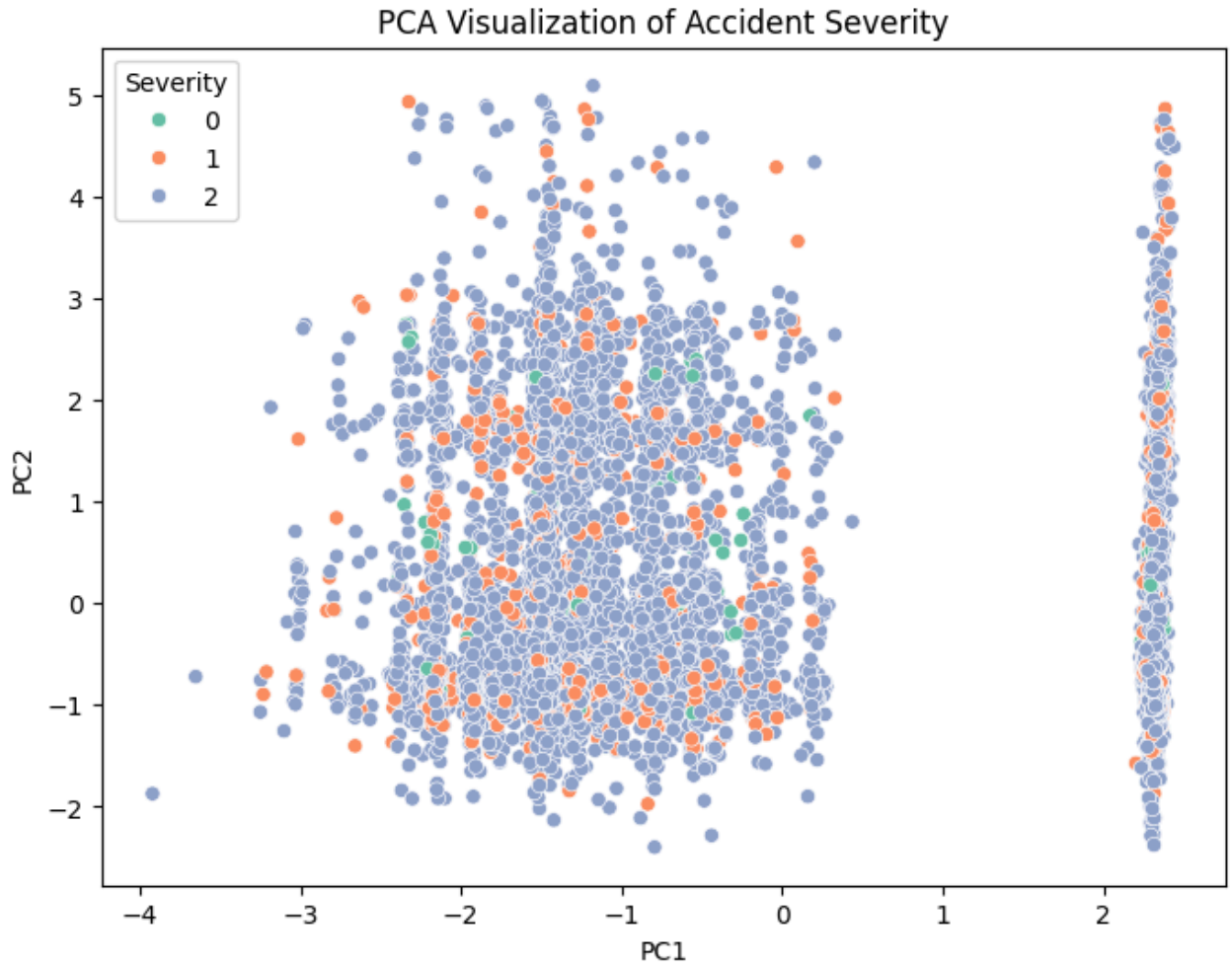

```
import joblib

joblib.dump(clf, 'random_forest_accident_model.pkl')
```

```
joblib.dump(scaler, 'scaler.pkl')

files.download('random_forest_accident_model.pkl')
files.download('scaler.pkl')
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Load dataset
df = pd.read_csv("guna.csv")

# 2. Drop irrelevant columns
df.drop(columns=[
    'Work_of_casuality', 'Fitness_of_casuality', 'Service_year_of_vehicle',
    'Defect_of_vehicle', 'Time', 'Vehicle_driver_relation'
], inplace=True, errors='ignore')

# 3. Fill missing categorical data
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

# 4. 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

# 5. Prepare features and target
X = df.drop("Accident_severity", axis=1)
y = df["Accident_severity"]

# 6. Normalize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 7. Train/test split
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42
)

# 8. Train Support Vector Classifier
svm = SVC(kernel='rbf', C=1.0, gamma='scale', class_weight='balanced')  # RBF Kernel
svm.fit(X_train, y_train)
```

```
# 9. Predict and Evaluate
y_pred = svm.predict(X_test)

print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

# 10. Optional: PCA Visualization (if >2 features)
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
df_viz = pd.DataFrame(X_pca, columns=["PC1", "PC2"])
df_viz["Severity"] = y

plt.figure(figsize=(8,6))
sns.scatterplot(data=df_viz, x="PC1", y="PC2", hue="Severity", palette="Set2")
plt.title("PCA Visualization of Accident Severity")
plt.show()
```

```
<ipython-input-36-846de00636b4>:21: FutureWarning: A value is trying to be set on
The behavior will change in pandas 3.0. This inplace method will never work becaus

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.metho


  df[col].fillna(df[col].mode()[0], inplace=True)
Confusion Matrix:
 [[  10   15   12]
 [  19  150  194]
 [  85  544 1435]]

Classification Report:
              precision    recall  f1-score   support

           0       0.09      0.27      0.13        37
           1       0.21      0.41      0.28       363
           2       0.87      0.70      0.77      2064

    accuracy                           0.65      2464
   macro avg       0.39      0.46      0.40      2464
weighted avg       0.76      0.65      0.69      2464
```



```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Load dataset
df = pd.read_csv('guna.csv')

# 2. Drop irrelevant columns
df.drop(columns=[
    'Work_of_casuality', 'Fitness_of_casuality', 'Service_year_of_vehicle',
    'Defect_of_vehicle', 'Time', 'Vehicle_driver_relation'
], inplace=True, errors='ignore')

# 3. Fill missing values
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)

# 4. 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

# 5. Prepare features and target
X = df.drop("Accident_severity", axis=1)
y = df["Accident_severity"]

# One-hot encode the target for NN
y_encoded = to_categorical(y)

# 6. Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 7. Train/test split
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y_encoded, test_size=0.2, random_state=42
)

# 8. Build Neural Network model
model = Sequential([
    Dense(64, input_dim=X_train.shape[1], activation='relu'),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(y_encoded.shape[1], activation='softmax')  # Output layer with softmax
])
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

# 9. Train the model
history = model.fit(X_train, y_train, epochs=50, batch_size=32, validation_split=0.1)

# 10. Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {accuracy:.2f}")

# 11. Confusion Matrix
y_pred_probs = model.predict(X_test)
y_pred_classes = np.argmax(y_pred_probs, axis=1)
y_true = np.argmax(y_test, axis=1)

print("Classification Report:\n", classification_report(y_true, y_pred_classes))

conf_matrix = confusion_matrix(y_true, y_pred_classes)
plt.figure(figsize=(6,5))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues")
plt.title("Neural Network Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("True")
plt.show()
```

```
<ipython-input-37-929e48adb18c>:23: FutureWarning: A value is trying to be set on
The behavior will change in pandas 3.0. This inplace method will never work becaus

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.metho


  df[col].fillna(df[col].mode()[0], inplace=True)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWar
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	1,664
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 3)	99

Total params: 3,843 (15.01 KB)
Trainable params: 3,843 (15.01 KB)
Non-trainable params: 0 (0.00 B)

```
Epoch 1/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.6905 - loss: 0.8005 - val_a
Epoch 2/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.8483 - loss: 0.5015 - val_a
Epoch 3/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8445 - loss: 0.4867 - val_a
Epoch 4/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 3s 11ms/step - accuracy: 0.8404 - loss: 0.5015 - val_
Epoch 5/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 3s 3ms/step - accuracy: 0.8491 - loss: 0.4716 - val_a
Epoch 6/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8447 - loss: 0.4735 - val_a
Epoch 7/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8446 - loss: 0.4731 - val_a
Epoch 8/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8446 - loss: 0.4617 - val_a
Epoch 9/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8463 - loss: 0.4581 - val_a
Epoch 10/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8444 - loss: 0.4673 - val_a
Epoch 11/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8462 - loss: 0.4558 - val_a
Epoch 12/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8487 - loss: 0.4468 - val_a
Epoch 13/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8458 - loss: 0.4487 - val_a
Epoch 14/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - accuracy: 0.8486 - loss: 0.4436 - val_a
Epoch 15/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8443 - loss: 0.4521 - val_a
Epoch 16/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8481 - loss: 0.4487 - val_a
Epoch 17/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8530 - loss: 0.4328 - val_a
```

```
Epoch 18/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8429 - loss: 0.4459 - val_a
Epoch 19/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8512 - loss: 0.4348 - val_a
Epoch 20/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8420 - loss: 0.4476 - val_a
Epoch 21/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8502 - loss: 0.4345 - val_a
Epoch 22/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8517 - loss: 0.4271 - val_a
Epoch 23/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8515 - loss: 0.4195 - val_a
Epoch 24/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8421 - loss: 0.4489 - val_a
Epoch 25/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 2s 4ms/step - accuracy: 0.8565 - loss: 0.4212 - val_a
Epoch 26/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8478 - loss: 0.4334 - val_a
Epoch 27/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8551 - loss: 0.4219 - val_a
Epoch 28/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8455 - loss: 0.4386 - val_a
Epoch 29/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8498 - loss: 0.4323 - val_a
Epoch 30/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8425 - loss: 0.4349 - val_a
Epoch 31/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8511 - loss: 0.4255 - val_a
Epoch 32/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8472 - loss: 0.4304 - val_a
Epoch 33/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8457 - loss: 0.4311 - val_a
Epoch 34/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8524 - loss: 0.4247 - val_a
Epoch 35/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8435 - loss: 0.4374 - val_a
Epoch 36/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8489 - loss: 0.4229 - val_a
Epoch 37/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8525 - loss: 0.4183 - val_a
Epoch 38/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8604 - loss: 0.3979 - val_a
Epoch 39/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8505 - loss: 0.4230 - val_a
Epoch 40/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8458 - loss: 0.4342 - val_a
Epoch 41/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8456 - loss: 0.4215 - val_a
Epoch 42/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8501 - loss: 0.4194 - val_a
Epoch 43/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8554 - loss: 0.4147 - val_a
Epoch 44/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8526 - loss: 0.4165 - val_a
Epoch 45/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8499 - loss: 0.4162 - val_a
Epoch 46/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8550 - loss: 0.4121 - val_a
Epoch 47/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8474 - loss: 0.4187 - val_a
Epoch 48/50
```

```
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8503 - loss: 0.4158 - val_a
Epoch 49/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.8534 - loss: 0.4076 - val_a
Epoch 50/50
278/278 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.8533 - loss: 0.4178 - val_a
77/77 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8413 - loss: 0.4702
Test Accuracy: 0.84
77/77 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        37
           1       0.83      0.01      0.03       363
           2       0.84      1.00      0.91      2064

    accuracy                           0.84      2464
   macro avg       0.56      0.34      0.31      2464
weighted avg       0.83      0.84      0.77      2464

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