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

Choose files accident ag.csv

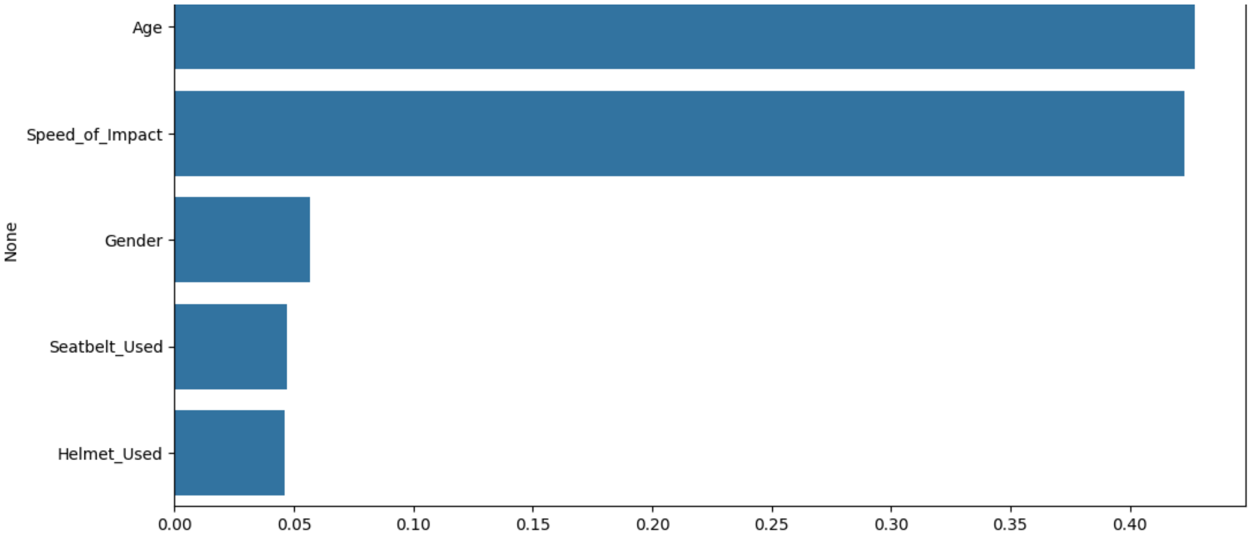
- **accident ag.csv**(text/csv) - 4542 bytes, last modified: 15/05/2025 - 100% done

```
Saving accident ag.csv to accident ag.csv
     Age  Gender  Speed_of_Impact Helmet_Used Seatbelt_Used  Survived
0     56  Female             27.0          No            No         1
1     69  Female             46.0          No           Yes         1
2     46    Male             46.0         Yes           Yes         0
3     32    Male            117.0          No           Yes         0
4     60  Female             40.0         Yes           Yes         0
..   ...     ...              ...         ...           ...       ...
195   69  Female            111.0          No           Yes         1
196   30  Female             51.0          No           Yes         1
197   58    Male            110.0          No           Yes         1
198   20    Male            103.0          No           Yes         1
199   56  Female             43.0          No           Yes         1

[200 rows x 6 columns]
Age                0
Gender             1
Speed_of_Impact    3
Helmet_Used        0
Seatbelt_Used      0
Survived           0
dtype: int64
```

```
import pandas as pd
import numpy as np
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, accuracy_score, confusion_matrix

# Load dataset
df = pd.read_csv('accident ag.csv')

# Display basic info
print("Dataset Info:")
print(df.info())
print("\nMissing values:\n", df.isnull().sum())

# Fill or drop missing values (basic approach)
df = df.dropna()  # or you can fillna() based on strategy

# Encode categorical features
label_encoders = {}
```

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

    # Check correlation to find a target (assume last column is target if unknown)
    print("\nCorrelation with target:")
    print(df.corr())

    # For demo, let’s assume the last column is the target
    X = df.iloc[:, :-1]
    y = df.iloc[:, -1]

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

    # Random Forest Classifier
    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)

    # Prediction
    y_pred = model.predict(X_test)

    # Evaluation
    print("\nAccuracy:", accuracy_score(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))
    print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

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

    # Plot feature importances
    plt.figure(figsize=(12, 6))
    sns.barplot(x=importances[indices], y=X.columns[indices])
    plt.title("Feature Importances")
    plt.show()
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Age              200 non-null    int64  
 1   Gender           199 non-null    object 
 2   Speed_of_Impact  197 non-null    float64
 3   Helmet_Used      200 non-null    object 
 4   Seatbelt_Used    200 non-null    object 
 5   Survived         200 non-null    int64  
dtypes: float64(1), int64(2), object(3)
memory usage: 9.5+ KB
None

Missing values:
 Age                0
Gender             1
Speed_of_Impact    3
Helmet_Used        0
Seatbelt_Used      0
Survived           0
dtype: int64

Correlation with target:
                      Age    Gender  Speed_of_Impact  Helmet_Used  \
Age              1.000000 -0.049222         0.106833     0.104798   
Gender          -0.049222  1.000000        -0.047972     0.012760   
Speed_of_Impact  0.106833 -0.047972         1.000000    -0.010659   
Helmet_Used      0.104798  0.012760        -0.010659     1.000000   
Seatbelt_Used   -0.016214 -0.058491        -0.000240     0.099879   
Survived         0.119213  0.115288         0.042902    -0.053661   

                 Seatbelt_Used  Survived  
Age                  -0.016214  0.119213  
Gender               -0.058491  0.115288  
Speed_of_Impact      -0.000240  0.042902  
Helmet_Used           0.099879 -0.053661  
Seatbelt_Used         1.000000  0.059183  
Survived              0.059183  1.000000  

Accuracy: 0.425

Classification Report:
               precision    recall  f1-score   support

           0       0.58      0.28      0.38        25
           1       0.36      0.67      0.47        15

    accuracy                           0.42        40
   macro avg       0.47      0.47      0.42        40
weighted avg       0.50      0.42      0.41        40


Confusion Matrix:
 [[ 7 18]
 [ 5 10]]
```



```
!pip install gradio

import pandas as pd
import numpy as np
import gradio as gr
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import joblib

# Load data
df = pd.read_csv("accident (1).csv")

# Preprocess
df = df.dropna()

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

# Features and target (you can customize target column here)
X = df.iloc[:, :-1]
y = df.iloc[:, -1]

# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Save model
joblib.dump(model, "accident_model.pkl")

# Gradio interface
def predict_accident(*inputs):
    inputs = np.array(inputs).reshape(1, -1)
    prediction = model.predict(inputs)[0]
    return f"Predicted Class: {prediction}"

# Dynamically generate inputs
input_components = []
for col in X.columns:
    col_min = df[col].min()
    col_max = df[col].max()
    input_components.append(gr.Slider(minimum=col_min, maximum=col_max, step=1, lab

demo = gr.Interface(
    fn=predict_accident,
    inputs=input_components,
    outputs="text",
    title="AI-Driven Traffic Accident Severity Predictor",
    description="Enter feature values to predict the accident class."
)

if __name__ == "__main__":
    demo.launch()
```

```
Collecting gradio
  Downloading gradio-5.29.1-py3-none-any.whl.metadata (16 kB)
Collecting aiofiles<25.0,>=22.0 (from gradio)
  Downloading aiofiles-24.1.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-pack
Collecting fastapi<1.0,>=0.115.2 (from gradio)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.5.0-py3-none-any.whl.metadata (3.0 kB)
Collecting gradio-client==1.10.1 (from gradio)
  Downloading gradio_client-1.10.1-py3-none-any.whl.metadata (7.1 kB)
Collecting groovy~=0.1 (from gradio)
  Downloading groovy-0.1.2-py3-none-any.whl.metadata (6.1 kB)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/c
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-pac
Collecting ruff>=0.9.3 (from gradio)
  Downloading ruff-0.11.10-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.me
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
  Downloading safehttpx-0.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting semantic-version~=2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting starlette<1.0,>=0.40.0 (from gradio)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.11/di
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (frc
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist
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: annotated-types>=0.6.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-package
```