```
!pip install gradio
```

```
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.2
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dis
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: gradio-client==1.10.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packages
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/d
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-
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.11/
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.11/d
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.11/di
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.11/d
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
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (fro
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: 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
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dis
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages
```

```
import pandas as pd
from google.colab import files
uploaded= files.upload()
df=pd.read_csv("global_traffic_accidents.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 global_traffic_accidents.csv to global_traffic_accidents (1).csv
     Accident ID        Date   Time           Location   Latitude   Longitude  \
0       b0dd6f57  2023-04-19  06:39      Mumbai, India  13.488432  -73.290682
1       debfad09  2023-01-17  02:47  São Paulo, Brazil -37.798317  -32.244242
2       6d69aa36  2024-04-09  02:55  Sydney, Australia  33.767869  104.869018
3       425bb1f0  2023-10-10  11:23       Tokyo, Japan  -0.378031 -165.825855
4       90d5cf62  2023-01-02  12:07     Beijing, China  41.254879  -30.776959
...          ...         ...    ...                ...        ...         ...
9995    2d26c7e2  2023-01-10  18:41      Paris, France -41.344055  109.335620
9996    4d236cfd  2023-04-04  16:48  São Paulo, Brazil -60.765148  -10.432225
9997    1d32722f  2024-09-30  14:43     Beijing, China -11.161278  -72.164379
9998    64722572  2024-10-27  18:34  Sydney, Australia -17.153524   20.803006
9999    96272c1b  2024-10-26  10:30    Toronto, Canada  21.917486  100.486079

     Weather Condition      Road Condition  Vehicles Involved  Casualties  \
0                 Snow               Snowy                  5           7
1                Clear                 Icy                  4           1
2                 Rain               Snowy                  1           7
3                Storm                 Wet                  4           0
4                Storm               Snowy                  3           9
...                ...                 ...                ...         ...
9995             Storm                 Wet                  3          10
9996             Storm                 Dry                  3           9
9997              Snow  Under Construction                  2           3
9998             Storm  Under Construction                  4           3
9999             Storm                 Icy                  5           4

                   Cause
0       Reckless Driving
1          Drunk Driving
2       Reckless Driving
3          Drunk Driving
4       Reckless Driving
...                 ...
9995  Distracted Driving
9996  Weather Conditions
9997  Weather Conditions
9998       Drunk Driving
9999  Mechanical Failure

[10000 rows x 11 columns]
Accident ID          0
Date                 0
Time                 0
Location             0
Latitude             0
Longitude            0
Weather Condition    0
Road Condition       0
Vehicles Involved    0
Casualties           0
Cause                0
dtype: int64
```

```
import pandas as pd
import numpy as np
import gradio as gr
```

```
import gradio as gr
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import pickle

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

# Create Severity label based on Casualties
df["Severity"] = pd.cut(
    df["Casualties"],
    bins=[-1, 2, 5, float("inf")],
    labels=["Low", "Medium", "High"]
)

# Select features
features = ["Weather Condition", "Road Condition", "Vehicles Involved"]
X = df[features]
y = df["Severity"]

# Encode categorical variables
X = pd.get_dummies(X)
le = LabelEncoder()
y_encoded = le.fit_transform(y)

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

# Train model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Save model and encoder
with open("model.pkl", "wb") as f:
    pickle.dump(model, f)

with open("label_encoder.pkl", "wb") as f:
    pickle.dump(le, f)

# Define prediction function
def predict_severity(weather, road, vehicles):
    input_df = pd.DataFrame([[weather, road, vehicles]], columns=["Weather Conditic
    input_df = pd.get_dummies(input_df).reindex(columns=X.columns, fill_value=0)
    prediction = model.predict(input_df)[0]
    severity = le.inverse_transform([prediction])[0]
    return f"Predicted Accident Severity: {severity}"

# Gradio app
# Convert choices to strings to avoid the ValueError
interface = gr.Interface(
    fn=predict_severity,
    inputs=[
        gr.Dropdown(choices=[str(c) for c in df["Weather Condition"].unique()], lab
        gr.Dropdown(choices=[str(c) for c in df["Road Condition"].unique()], label=
        gr.Number(label="Vehicles Involved")
```

```
        ],
        outputs="text",
        title="AI Traffic Accident Severity Predictor"
    )

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

```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio a

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://afbc2a7b06d45214e0.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `
```