

```
# prompt: handling values for the above dataset
```

```
from google.colab import files
import pandas as pd
import io
```

```
uploaded = files.upload()
```

```
try:
```

```
    # The following line reads the uploaded file into a Pandas DataFrame
    df = pd.read_csv(io.BytesIO(uploaded['synthetic_patient_data.csv']))
```

```
except KeyError:
```

```
    print("Error: 'synthetic_patient_data.csv' not found among uploaded files.")
```

```
    # Instead of exiting, assign an empty DataFrame to 'df'
```

```
    df = pd.DataFrame() # Create an empty DataFrame if the file is not found
```

```
except pd.errors.ParserError:
```

```
    print("Error: Could not parse the uploaded CSV file. Please check the file format.")
```

```
    # Instead of exiting, assign an empty DataFrame to 'df'
```

```
    df = pd.DataFrame() # Create an empty DataFrame if parsing fails
```

```
except Exception as e:
```

```
    print(f"An unexpected error occurred: {e}")
```

```
    # Instead of exiting, assign an empty DataFrame to 'df'
```

```
    df = pd.DataFrame() # Create an empty DataFrame if any other error occurs
```

```
# Now 'df' is defined and you can work with it (even if it's empty)
```

```
#Example operations
```

```
print(df.head()) # Display first few rows (will be empty if there was an error)
```

```
print(df.info()) # Get info about the DataFrame
```

```
# Check if the DataFrame has columns before calling describe
```

```
if not df.empty:
```

```
    print(df.describe()) # Descriptive statistics
```

```
else:
```

```
    print("DataFrame is empty, cannot generate descriptive statistics.")
```

```
# ... other data handling ...
```

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 synthetic_patient_data.xlsx to synthetic_patient_data (9).xlsx
Error: 'synthetic_patient_data.csv' not found among uploaded files.

Empty DataFrame

Columns: []

Index: []

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 0 entries

Empty DataFrame

None

DataFrame is empty, cannot generate descriptive statistics.

Double-click (or enter) to edit

[4] # prompt: handling values for the above dataset

Example: Handling missing values

print(df.isnull().sum()) # Count of missing values in each column

df.fillna(0, inplace=True) # Fill missing values with 0 (Example)

Example: Handling inconsistent data types (Example: converting a column to numeric)

Assuming 'age' is a column that might have inconsistent data types

if 'age' in df.columns:

try:

df['age'] = pd.to_numeric(df['age'], errors='coerce') # Convert to numeric, setting invalid parsing to NaN

df.dropna(subset=['age'], inplace=True) # Remove rows with NaNs in the converted column

except KeyError:

print("Error: 'age' column not found in the DataFrame.")

else:

print("Warning: 'age' column not found. Skipping type conversion.")

Example: Removing duplicates

df.drop_duplicates(inplace=True)

Example: Removing rows based on conditions

Assuming 'blood_pressure' is a column

if 'blood_pressure' in df.columns:

df = df[df['blood_pressure'] > 70] # Example: keep rows where blood pressure is above 70


```
print("Warning: 'blood_pressure' column not found, cannot filter.")  
print(df.head())
```

```
Series([], dtype: float64)  
Warning: 'age' column not found. Skipping type conversion.  
Warning: 'blood_pressure' column not found, cannot filter.  
Empty DataFrame  
Columns: []  
Index: []
```

✓ [5] # prompt: remove duplicate values for the above dataset

```
# Example: Removing duplicates  
df.drop_duplicates(inplace=True)  
  
print(df.head())
```

```
Empty DataFrame  
Columns: []  
Index: []
```

✓ [10] # prompt: visual heatmaps

```
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# ... (Your existing code for file upload and data preprocessing) ...  
  
# Example: Creating a heatmap of correlations between numerical features  
if not df.empty:  
    numerical_features = df.select_dtypes(include=['number']) # Select only numerical columns  
    if not numerical_features.empty:  
        correlation_matrix = numerical_features.corr()  
        plt.figure(figsize=(10, 8))  
        sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")  
        plt.title('Correlation Heatmap of Numerical Features')
```

```

plt.title('Correlation Heatmap of Numerical Features')
plt.show()
else:
    print("No numerical features found in the DataFrame to create a correlation heatmap.")
else:
    print("DataFrame is empty, cannot generate a heatmap.")

```

↳ DataFrame is empty, cannot generate a heatmap.

[9] # prompt: build a model

```

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Assuming 'df' is your DataFrame and you have a target variable column named 'target'
if 'target' in df.columns and not df.empty: # Check if 'target' exists and the DataFrame isn't empty
    # Separate features (X) and target (y)
    X = df.drop('target', axis=1)
    y = df['target']

    # Handle potential errors during splitting
    try:
        # Split data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    except ValueError as e:
        print(f"Error during train_test_split: {e}")
        print("Check if your features (X) and target (y) have compatible shapes or contain invalid values.")
        exit() # Or handle the error differently

    # Initialize and train a logistic regression model
    model = LogisticRegression(max_iter=1000) # Increased max_iter

    try:
        model.fit(X_train, y_train)
    except ValueError as e:
        print(f"Error during model training: {e}")
        print("Check your data for issues like infinite values or incorrect data types.")
        exit()

    # Make predictions on the test set

```

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
else:
    print("Error: 'target' column not found in the DataFrame or the DataFrame is empty.")
```

Error: 'target' column not found in the DataFrame or the DataFrame is empty.

[13] !pip install gradio

```
import gradio as gr

def greet(name):
    return "Hello " + name + "!"



iface = gr.Interface(
    fn=greet,
    inputs=gr.Textbox(lines=2, placeholder="Enter your name here..."),
    outputs="text",
    title="Simple Greeting App",
)

iface.launch()
```

Collecting gradio
 Downloading gradio-5.29.0-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-packages (from gradio) (4.9.0)
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==1.10.0 (from gradio)
 Downloading gradio_client-1.10.0-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-packages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.31.1)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.0.2)

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-packages (from gradio) (6.0.2)
Collecting ruff<0.9.3 (from gradio)
Downloading ruff-0.11.9-py3-none-manylinux_2_17_x86_64_manylinux2014_x86_64.whl.metadata (25 kB)
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 tomkit<0.14.0,>=0.12.0 (from gradio)
Downloading tomkit-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-packages (from gradio) (0.15.3)
Requirement already satisfied: typing-extensions==4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.13.2)
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 (from gradio-client==1.10.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (15.0.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0.16.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.67.1)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (1.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.4.0)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio) (1.17.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio) (2.19.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1->gradio) (3.4.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1->gradio) (2.4.0)
Requirement already satisfied: mdurl==0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio) (0.1.2)

Commands	+ Code	+ Text
----------	--------	--------

 Share
  Gem

```

Downloaded gradio_client-1.10.0-py3-none-any.whl (322 kB)
54.1/54.1 MB 13.1 MB/s eta 0:00:00
Downloaded aiofiles-24.1.0-py3-none-any.whl (15 kB)
322.9/322.9 kB 22.2 MB/s eta 0:00:00
Downloaded fastapi-0.115.12-py3-none-any.whl (95 kB)
95.2/95.2 kB 7.0 MB/s eta 0:00:00
Downloaded groovy-0.1.2-py3-none-any.whl (14 kB)
Downloaded python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloaded ruff-0.11.9-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
11.5/11.5 MB 104.8 MB/s eta 0:00:00
Downloaded safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloaded semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloaded starlette-0.46.2-py3-none-any.whl (72 kB)
72.0/72.0 kB 5.8 MB/s eta 0:00:00
Downloaded tomkit-0.13.2-py3-none-any.whl (37 kB)
Downloaded uvicorn-0.34.2-py3-none-any.whl (62 kB)
62.5/62.5 kB 4.8 MB/s eta 0:00:00
Downloaded ffmpeg-0.5.0-py3-none-any.whl (6.0 kB)
Downloaded pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub, uvicorn, tomkit, semantic-version, ruff, python-multipart, groovy, ffmpeg, aiofiles, starlette, safehttpx, gradio-client, fastapi, gradio
Successfully installed aiofiles-24.1.0 fastapi-0.115.12 ffmpeg-0.5.0 gradio-5.29.0 gradio-client-1.10.0 groovy-0.1.2 pydub-0.25.1 python-multipart-0.0.20 ruff-0.11.9 safehttpx-0.1.6 semantic-version-2.10.0 starlette-0.46.2 tomkit-0.13.2 uvicorn-0.34.2
It looks like you are running Gradio on a hosted Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automatically setting 'share=True' (you can turn this off by setting 'share=False' in 'launch()' explic

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://b33f8eabca4ea1956a.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run 'gradio deploy' from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spaces)

```

Simple Greeting App

<div><div>name</div><div><div>Enter your name here...</div></div><div><div>Clear</div><div>Submit</div></div></div>	<div><div>output</div><div></div><div>Flag</div></div>
---	--