```
! pip install gradio
```

```
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/dis
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.
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
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.1
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.1
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/d
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-pack
Downloading gradio-5.29.0-py3-none-any.whl (54.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 54.1/54.1 MB 18.9 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 322.9/322.9 kB 19.4 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 95.2/95.2 kB 5.4 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.5/11.5 MB 94.4 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)
Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.0/72.0 kB 4.3 MB/s eta 0:00:00
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.5/62.5 kB 3.6 MB/s eta 0:00:00
Downloading ffmpy-0.5.0-py3-none-any.whl (6.0 kB)
Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Installing collected packages: pydub, uvicorn, tomlkit, semantic-version, ruff,
Successfully installed aiofiles-24.1.0 fastapi-0.115.12 ffmpy-0.5.0 gradio-5.29.
```

```
import pandas as pd

# Read the excel file into a pandas dataframe.
# Note that content/dataset (1).xlsx should be the actual path to your excel file.
df = pd.read_excel('/content/dataset (1).xlsx')

# Display the dataframe.
display(df)
```

NameError NameError FileNotFound •••

Please explain the error:

```
FileNotFoundError: [Errno 2] No such file or di
```

Hypothesis

The error message `FileNotFoundError: [Errno 2] No such file or directory: 'dataset.xlsx'` indicates that the Python script is unable to locate the file named `dataset.xlsx` in the current working directory. It's likely that the file is either not present in the directory where the script is being executed, or the path specified in the `pd.read_excel()` function is incorrect.

Suggested Changes

Ensure the 'dataset.xlsx' file is in the same directory as your Python script or provide the full path to the file in `pd.read_excel()`.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import gradio as gr
from sklearn.model_selection import train_test_
from sklearn.linear_model import LinearRegressi
from sklearn.metrics import mean_squared_error,

# ============================================
# 1. Load Data
# ============================================
# Assuming 'dataset.xlsx' is in the current dir
df = pd.read_excel("dataset.xlsx")
# If the file is in a different directory, use:
# df = pd.read_excel("/path/to/your/dataset.xls
df.columns = df.columns.str.strip().str.title()
df.drop_duplicates(inplace=True)
df.dropna(subset=["Bedrooms", "Bathrooms", "Flo
df["TotalRooms"] = df["Bedrooms"] + df["Bathroo
df.reset_index(drop=True, inplace=True)

# ... (rest of the code remains the same)
```

Use code with caution

▶ Sources

Rate this answer

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage
0	1	1000	5	4	3	1970	Downtown	Excellent	No
1	2	2000	5	4	3	1958	Downtown	Excellent	No
2	3	3000	2	2	3	1938	Downtown	Good	No
3	4	4000	4	2	2	1902	Suburban	Fair	Yes
4	5	5000	1	4	2	1975	Downtown	Fair	Yes
...	...	...	...	...	...	...	...	...	...
77	78	6000	3	3	1	1994	Rural	Good	Yes
78	79	5000	3	3	2	1977	Urban	Fair	No
79	80	4000	2	2	2	1975	Urban	Good	Yes
80	81	6000	3	3	3	1976	Rural	Good	Yes
81	82	34000	2	2	3	1876	Suburban	Good	No

82 rows × 10 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import gradio as gr
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# ================================================
# 1. Load Data
# ================================================
df = pd.read_excel("dataset (1).xlsx")
df.columns = df.columns.str.strip().str.title()
df.drop_duplicates(inplace=True)
df.dropna(subset=["Bedrooms", "Bathrooms", "Floors", "Garage", "Condition", "Price"], in
df["TotalRooms"] = df["Bedrooms"] + df["Bathrooms"]
df.reset_index(drop=True, inplace=True)

# ================================================
# 2. Model Training
# ================================================
features = ["Bedrooms", "Bathrooms", "Floors", "Garage", "Condition", "TotalRooms"]
X = pd.get_dummies(df[features], drop_first=True)
y = df["Price"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

model = LinearRegression()
model.fit(X_train, y_train)

# ================================================
# 3. Prediction Function
# ================================================
def predict_price(bedrooms, bathrooms, floors, garage, condition):
    total_rooms = bedrooms + bathrooms
    input_data = pd.DataFrame([{
        "Bedrooms": bedrooms,
        "Bathrooms": bathrooms,
        "Floors": floors,
        "Garage": garage,
        "Condition": condition,
        "TotalRooms": total_rooms
    }])
    input_data = pd.get_dummies(input_data, drop_first=True)
    input_data = input_data.reindex(columns=X.columns, fill_value=0)
    prediction = model.predict(input_data)[0]
    return f"${prediction:,.2f}"

# ================================================
# 4. Gradio Interface
# ================================================
iface = gr.Interface(
```

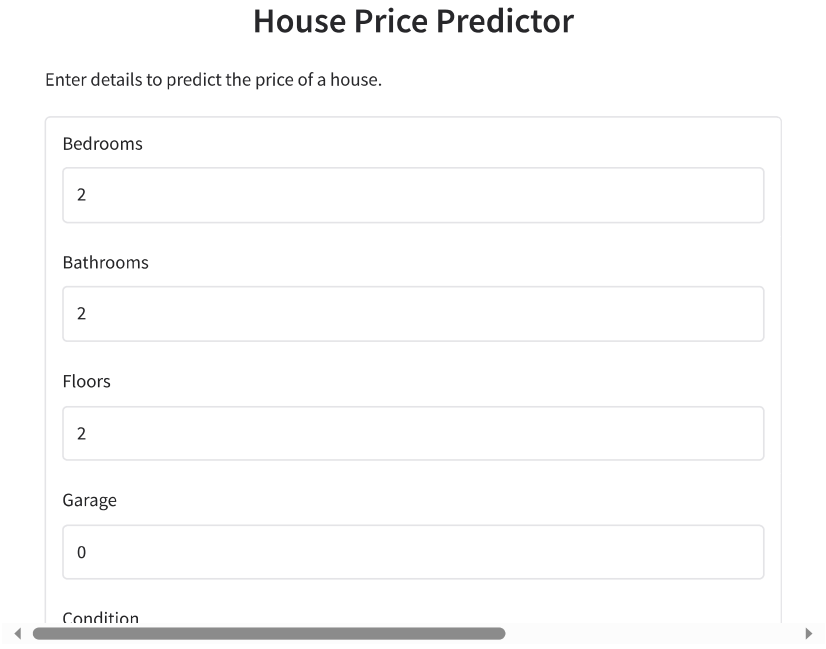
```
iface = gr.Interface(
    fn=predict_price,
    inputs=[
        gr.Number(label="Bedrooms"),
        gr.Number(label="Bathrooms"),
        gr.Number(label="Floors"),
        gr.Number(label="Garage"),
        gr.Dropdown(["Excellent", "Good", "Fair"], label="Condition")
    ],
    outputs="text",
    title="House Price Predictor",
    description="Enter details to predict the price of a house."
)

iface.launch()
```

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

Colab notebook detected. To show errors in colab notebook, set debug=True in launc
* Running on public URL: https://6560adc2d0a157d356.gradio.live

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



Enter a prompt here

0 / 2000

Gemini can make mistakes so double-check responses and use code with caution. Learn more