Upload the Dataset

```
from google.colab import files
upload=files.upload()
```

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 House Price Prediction Dataset.csv to House Price Prediction Dataset.csv
```

Load the Dataset

```
import pandas as pd
# Read the dataset
df = pd.read_csv('House Price Prediction Dataset.csv')
```

Data Exploration

```
# Display first few rows
df.head()
```

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage	Price
0	1	1360	5	4	3	1970	Downtown	Excellent	No	149919
1	2	4272	5	4	3	1958	Downtown	Excellent	No	424998
2	3	3592	2	2	3	1938	Downtown	Good	No	266746
3	4	966	4	2	2	1902	Suburban	Fair	Yes	244020
4	5	4926	1	4	2	1975	Downtown	Fair	Yes	636056

```
# Shape of the dataset
print
(
"Shape:"
, df.shape)
# Column names
print
(
"Columns:"
, df.columns.tolist())
# Data types and non-null values
df.info()
```

```
# Summary statistics for numeric features
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Id         2000 non-null   int64
 1   Area       2000 non-null   int64
 2   Bedrooms   2000 non-null   int64
 3   Bathrooms  2000 non-null   int64
 4   Floors     2000 non-null   int64
 5   YearBuilt  2000 non-null   int64
 6   Location   2000 non-null   object
 7   Condition  2000 non-null   object
 8   Garage     2000 non-null   object
 9   Price      2000 non-null   int64
dtypes: int64(7), object(3)
memory usage: 156.4+ KB
```

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Price
count	2000.000000	2000.000000	2000.000000	2000.00000	2000.000000	2000.000000	2000.000000
mean	1000.500000	2786.209500	3.003500	2.55250	1.993500	1961.446000	537676.855000
std	577.494589	1295.146799	1.424606	1.10899	0.809188	35.926695	276428.845719
min	1.000000	501.000000	1.000000	1.00000	1.000000	1900.000000	50005.000000
25%	500.750000	1653.000000	2.000000	2.00000	1.000000	1930.000000	300098.000000
50%	1000.500000	2833.000000	3.000000	3.00000	2.000000	1961.000000	539254.000000
75%	1500.250000	3887.500000	4.000000	4.00000	3.000000	1993.000000	780086.000000
max	2000.000000	4999.000000	5.000000	4.00000	3.000000	2023.000000	999656.000000

Check for Missing Values and Duplicates

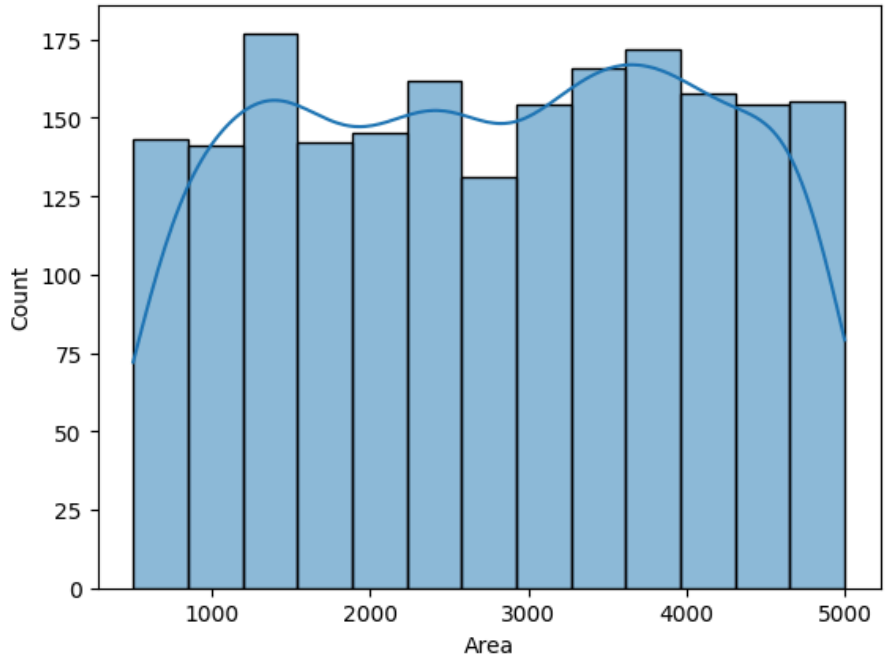
```
# Check for missing values
print(df.isnull().sum())
# Check for duplicates
print("Duplicate rows:", df.duplicated().sum())
```

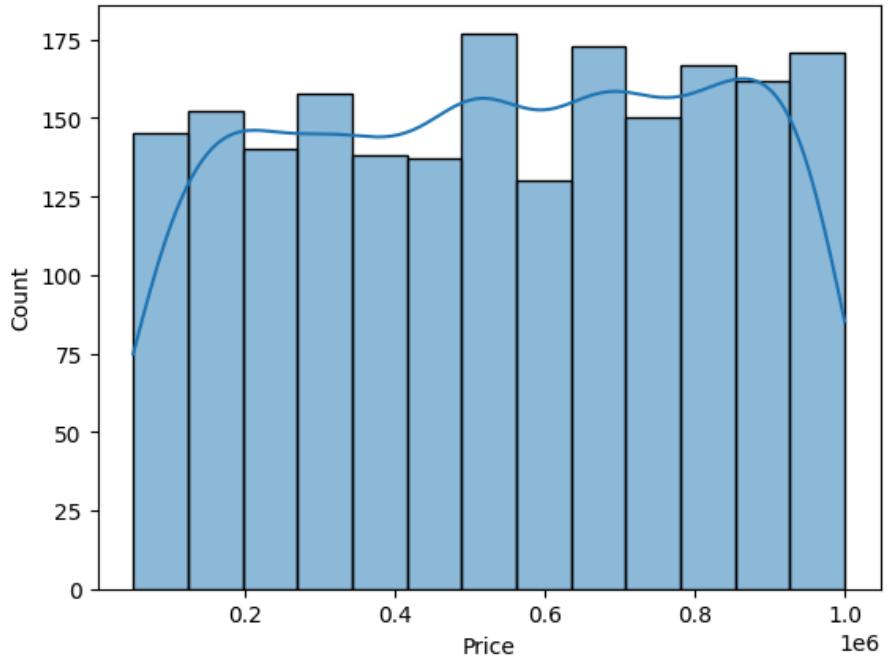
```
Id           0
Area         0
Bedrooms     0
Bathrooms    0
Floors       0
YearBuilt    0
Location     0
```

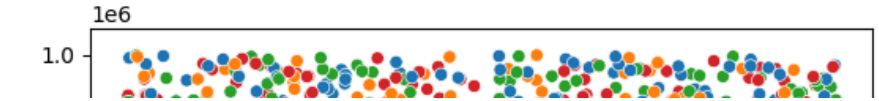
```
Condition    0
Garage       0
Price        0
dtype: int64
Duplicate rows: 0
```

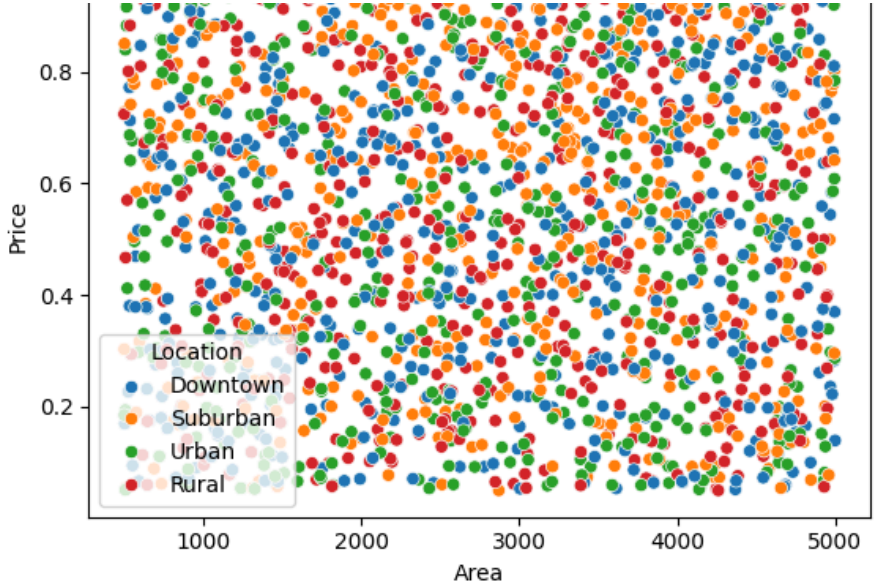
Visualize a Few Features

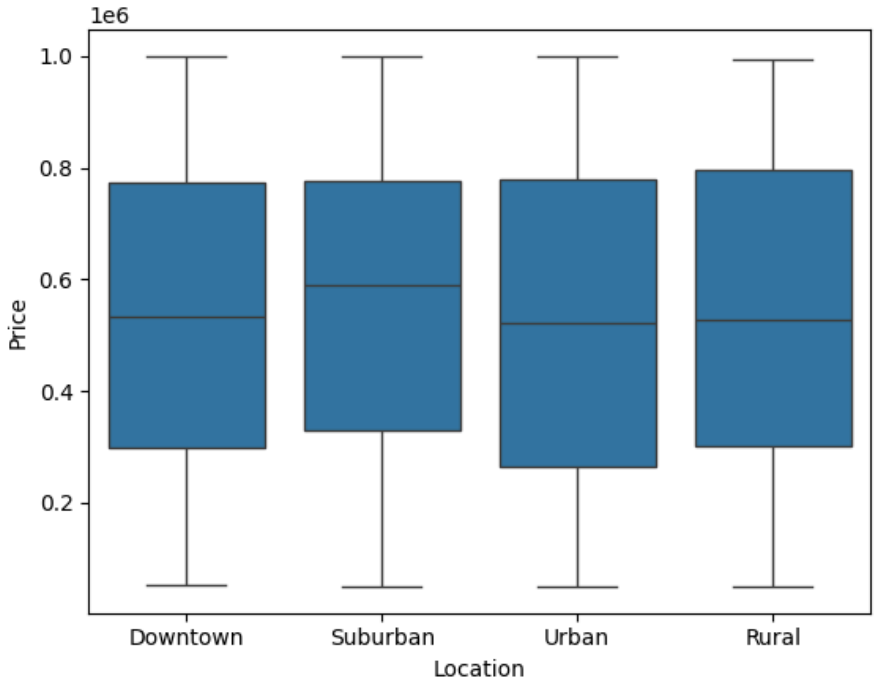
```
import seaborn as sns, matplotlib.pyplot as plt
sns.histplot(data=df, x='Area', kde=True); plt.show()
sns.histplot(data=df, x='Price', kde=True); plt.show()
sns.scatterplot(data=df, x='Area', y='Price', hue='Location'); plt.show()
sns.boxplot(data=df, x='Location', y='Price'); plt.show()
```


Count
Area
0
25
50
75
100
125
150
175
1000
2000
3000
4000
5000


Count
Price
1e6
0
25
50
75
100
125
150
175
0.2
0.4
0.6
0.8
1.0


1e6
1.0


0.8
0.6
0.4
0.2
Price
Location
Downtown
Suburban
Urban
Rural
1000
2000
3000
4000
5000
Area


1e6
1.0
0.8
0.6
0.4
0.2
Price
Downtown
Suburban
Urban
Rural
Location

Identify Target and Features

```
import pandas as pd
df = pd.read_csv("House Price Prediction Dataset.csv")
y = df['Price']                          # Target variable
X = df.drop(columns=['Price', 'Id'])    # Feature variables
```

Convert Categorical Columns to Numerical

```
import pandas as pd
df = pd.read_csv("House Price Prediction Dataset.csv")
print(df.columns.tolist())
```

```
['Id', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Location', 'Condition', 'Garage', 'Price']
```

Feature Scaling

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[['Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt']] = scaler.fit_transform(df[['Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt']])
```

Train-Test Split

```
X_train: (1600, 5)
X_test: (400, 5)
y_train: (1600,)
y_test: (400,)
```

Model Buliding

```
import pandas as pd; from sklearn.linear_model import LinearRegression
df = pd.read_csv("House Price Prediction Dataset.csv")
df = pd.get_dummies(df.drop(columns=['Id']), drop_first=True)
model = LinearRegression().fit(df.drop(columns=['Price']), df['Price'])
```

Evaluation

```
from sklearn.metrics import r2_score, mean_absolute_error
y_true = df['Price']; y_pred = model.predict(df.drop(columns=['Price']))
```

```
print("R2 Score:", r2_score(y_true, y_pred))
print("MAE:", mean_absolute_error(y_true, y_pred))
```

```
R2 Score: 0.008285421018850991
MAE: 237786.40455379031
```

Make Predictions from New Input

```
import pandas as pd
new_data = pd.DataFrame([{'Area': 2000, 'Bedrooms': 3, 'Bathrooms': 2, 'Floors': 2, 'YearBuilt': 2010, 'Location': 'Downtown', 'Condition': 'Good', 'Garage':
new_data = pd.get_dummies(new_data).reindex(columns=model.feature_names_in_, fill_value=0)
print("Predicted Price:", model.predict(new_data)[0])
```

```
Predicted Price: 532981.4584826692
```

convert to DataFrame and Encode

```
import pandas as pd
df = pd.read_csv('House Price Prediction Dataset.csv')
df_encoded = pd.get_dummies(df)
print(df_encoded.head())
```

```
   Id  Area  Bedrooms  Bathrooms  Floors  YearBuilt   Price  \
0   1  1360         5          4       3       1970  149919
1   2  4272         5          4       3       1958  424998
2   3  3592         2          2       3       1938  266746
3   4   966         4          2       2       1902  244020
4   5  4926         1          4       2       1975  636056

   Location_Downtown  Location_Rural  Location_Suburban  Location_Urban  \
0               True           False              False           False
1               True           False              False           False
2               True           False              False           False
3              False           False               True           False
4               True           False              False           False

   Condition_Excellent  Condition_Fair  Condition_Good  Condition_Poor  \
0                 True           False           False           False
1                 True           False           False           False
2                False           False            True           False
3                False            True           False           False
4                False            True           False           False

   Garage_No  Garage_Yes
0       True       False
1       True       False
2       True       False
3      False        True
4      False        True
```

Predict the Final Grade

```
import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.get_dummies(pd.read_csv("House Price Prediction Dataset.csv").drop(columns=['Id']), drop_first=True)
model = LinearRegression().fit(df.drop(columns=['Price']), df['Price'])
print(model.predict(df.drop(columns=['Price']))[:5])
```

```
[542606.87406134 542277.50243372 549168.70328683 579186.68894125
 555940.43334211]
```

Deployment-Building an Interactive App

```
pip install gradio
```

```
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-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==1.10.1 (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-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)
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.18)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.2.2)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.2.1)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
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 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-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==1.10.1->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.1->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)
```

Create a Prediction Function

```
def predict_price(input_df):
    input_df = pd.get_dummies(input_df).reindex(columns=model.feature_names_in_, fill_value=0)
    return model.predict(input_df)
```

Create the Gradio Interface

```
pip install pandas scikit-learn gradio
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages (5.29.1)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (24.1.0)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.9.0)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.115.12)
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (from gradio) (0.5.0)
```

```
Requirement already satisfied: gradio-client==1.10.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (1.10.1)
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.2)
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)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (3.10.18)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from gradio) (24.2)
Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.2.1)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.0.20)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.11.9)
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.1.6)
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.46.2)
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.13.2)
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)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.34.2)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.1->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.1->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: 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: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
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: 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.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1->gradio) (3.
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)
```

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import gradio as gr

# Load and prepare data
df = pd.read_csv("House Price Prediction Dataset.csv")
```

```
df = df.drop(columns=["Id"])

# Encode categorical features
label_encoders = {}
for col in ['Location', 'Condition', 'Garage']:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le

# Split data
X = df.drop(columns=["Price"])
y = df["Price"]
model = RandomForestRegressor()
model.fit(X, y)

# Prediction function
def predict(area, bedrooms, bathrooms, floors, year_built, location, condition, garage):
    # Ensure inputs match the order and types expected by the model training
    input_data = pd.DataFrame([[
        area, bedrooms, bathrooms, floors, year_built,
        label_encoders['Location'].transform([location])[0],
        label_encoders['Condition'].transform([condition])[0],
        label_encoders['Garage'].transform([garage])[0]
    ]], columns=X.columns)
    prediction = model.predict(input_data)[0]
    return f"Estimated Price: ${int(prediction):,}"

# Gradio interface
gr.Interface(
    fn=predict,
    inputs=[
        gr.Number(label="Area (sq ft)"),
        gr.Number(label="Bedrooms"),
        gr.Number(label="Bathrooms"),
        gr.Number(label="Floors"),
        gr.Number(label="Year Built"),
        # Convert the numpy array of classes to a list
        gr.Dropdown(label_encoders['Location'].classes_.tolist(), label="Location"),
        # Convert the numpy array of classes to a list
        gr.Dropdown(label_encoders['Condition'].classes_.tolist(), label="Condition"),
        # Convert the numpy array of classes to a list
        gr.Dropdown(label_encoders['Garage'].classes_.tolist(), label="Garage")
    ],
    outputs="text",
    title="House Price Predictor"
).launch()
```

```
It looks like you are running Gradio on a hosted a Jupyter notebook. For the Gradio app to work, sharing must be enabled. Automatically setting `share=Tr

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://c61887ef4f0c4483ca.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
```

House Price Predictor