

Predicting NY Real Estate Prices

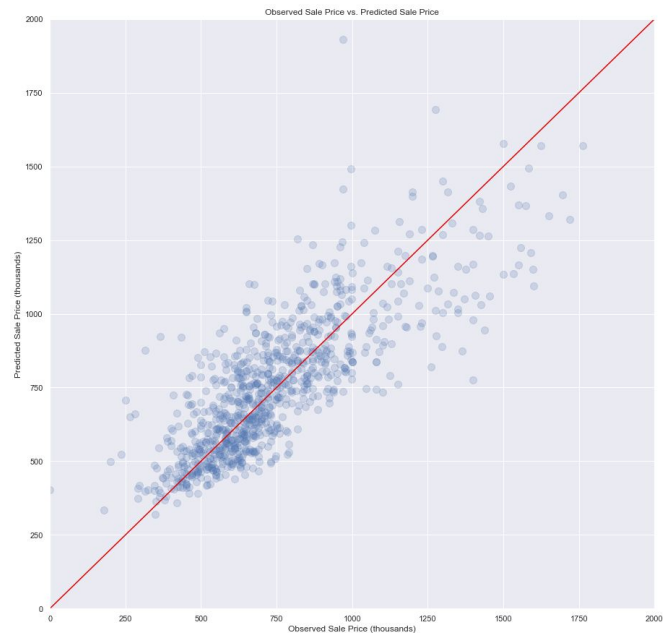
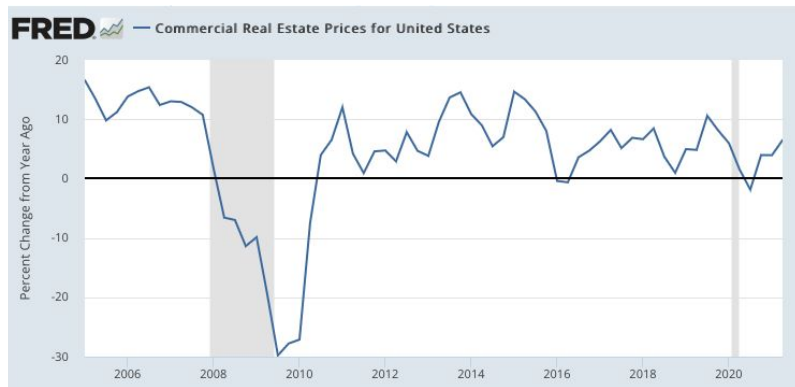
Using web scraping, geocoding, sentiment analysis, topic modeling, random forests, and the command line to create an automated data pipeline

Matthew Kwee
17 December 2021

Motivation

A few months ago, I took on this problem with only basic data analysis skills - and created a linear model for the data.

However, the housing market is far too complicated to be predicted by a single line.



Objective: Create an easily-accessible web application for predicting real-estate prices in New York City.

Methodology

Data: realtor.com

Initial scraping: 9000 listings; ~2400 usable

Daily updates: ~100 new listings/day

From raw HTML, 9 features (2
categorical, 1 text)

After processing/topic clustering: 83 features (9
numerical, 10 dummies, 64 NLP topics)

Data cleaning

Step 1: Raw HTML

```
<ul class="list-default">
  <li>Bedrooms: 6</li>
  <li>Bedrooms Possible: 6</li>
</ul>
<li>Bedrooms: 6</li>
<li>Bedrooms Possible: 6</li>
<ul class="list-default">
  <li>Bedrooms: 6</li>
  <li>Bedrooms Possible: 6</li>
</ul>
```

Step 2: Pull data into Pandas dataframe

	beds	baths	price	description	address	sqft	sale_date	lot_size	year_built	stories	rooms	property_type	neighborhood	borough
0	6	2	1810000	 - 2 family home in the Heart of Willa...	129 Devoe St, Ny, NY 11211	N/A	November 8, 2021	2500	1901	2	Total Rooms: 13	Single Family Home	Williamsburg	Brooklyn, NY
1	1	2	N/A	 - Primary residence only, no pied-a-te...	160 Bleecker St Apt 10LE, New York City, NY 10012	N/A	November 2, 2021	N/A	1896	10	Total Rooms: 4	N/A	SoHo	Manhattan, NY
2	2	3	2190000	 - Presenting 110 Summit Street; a coll...	110 Summit St Apt 1, New York City, NY 11231	2008	October 25, 2021	N/A	1899	3	Total Rooms: 5	N/A	others	Brooklyn, NY
3	3	3	665000	 - Beautiful 2 family home on a 30' x 1...	109 Station Ave, Staten Island, NY 10309	1350	November 10, 2021	3270	2002	3	Total Rooms: 6	Single Family Home	others	N/A
4	3	2	508000	 - Prime Arden Heights. Well Kept Singl...	92 Carlyle Grn, Staten Island, NY 10312	1080	November 16, 2021	2697	1975	2	N/A	Condo	others	BROOKLYN, NY

Data cleaning

Step 3: Cluster descriptions with topic model, create dummy variables, conduct sentiment analysis!

	beds	baths	price	sqft	stories	rooms	building_age	pol	sub	lda_topic0	...	Commercial	Condo	Multi-Family Home	Other	Single-Family Home	Bronx	
3	3.0	3.0	665000.0	1350.0	3.0	6.0	19.0	0.417532	0.530519	0.009693	...	0.0	0	0	0	1	0.0	[...]
8	1.0	1.0	459000.0	532.0	4.0	2.0	5.0	0.122857	0.495714	0.009472	...	0.0	1	0	0	0	0.0	[...]
15	5.0	5.0	1999999.0	3114.0	3.0	12.0	122.0	0.262141	0.528006	0.056832	...	0.0	1	0	0	0	0.0	[...]
18	6.0	5.0	1190000.0	3700.0	3.0	12.0	51.0	0.330556	0.701190	0.006829	...	0.0	0	1	0	0	0.0	[...]
20	2.0	1.0	849000.0	1200.0	6.0	5.0	104.0	0.303194	0.571307	0.066964	...	0.0	1	0	0	0	0.0	[...]
...	[...]
72	1.0	1.0	725000.0	786.0	32.0	3.0	36.0	0.133917	0.482117	0.019385	...	0.0	1	0	0	0	0.0	[...]
75	2.0	3.0	379000.0	1354.0	3.0	6.0	25.0	0.356746	0.542659	0.011675	...	0.0	0	0	0	1	0.0	[...]
79	2.0	2.0	1695000.0	2000.0	7.0	4.0	109.0	0.296512	0.593700	0.032381	...	0.0	1	0	0	0	0.0	[...]
80	2.0	1.0	760750.0	793.0	6.0	4.0	122.0	0.151384	0.348802	0.028637	...	0.0	1	0	0	0	0.0	[...]
81	7.0	2.0	1470000.0	1100.0	2.0	12.0	101.0	0.116667	0.275000	0.009032	...	0.0	0	1	0	0	0.0	[...]

2375 rows x 83 columns

Methodology

Tools (a lot!):

Selenium for data scraping

Pandas and NumPy for data storage and formatting

Google Maps Geocoding API for filling in missing location data

NLTK for tokenization of listing descriptions

TextBlob for sentiment analysis of descriptions

Gensim for topic modeling descriptions in order to perform soft clustering

Scikit-Learn for regression models

TensorFlow's Keras module for constructing a MLP Regressor

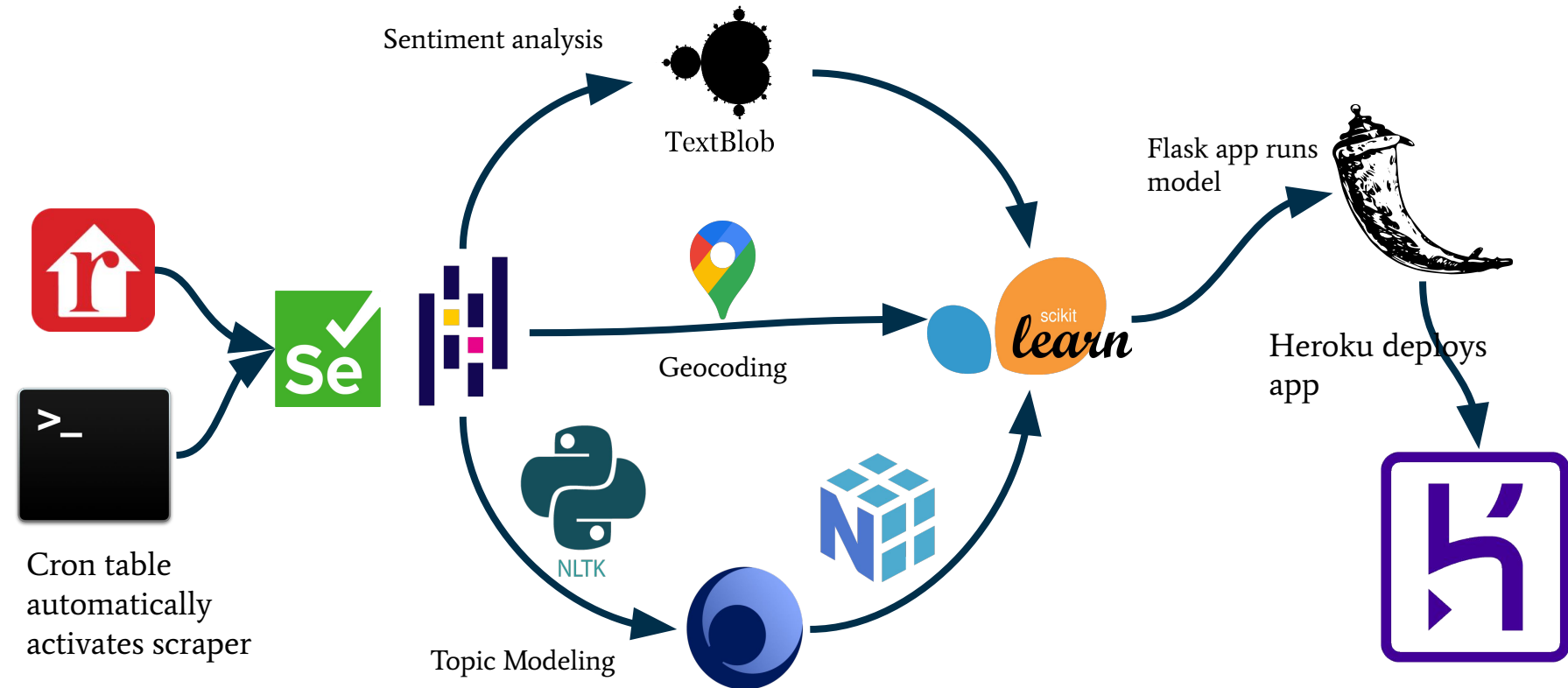
MacOS Command Line scripts (cron) for automated data collection and model updates

GitHub API for automatically pushing the Random Forest model to a backup repository

Flask for creating web app

Heroku for deploying Flask app

Data Pipeline - Tools



Model:

Random Forest
Regressor

256 trees

Train-Test 85-15 split

Train: 2000 listings

Test: 353 listings

Training data r^2 : 0.88

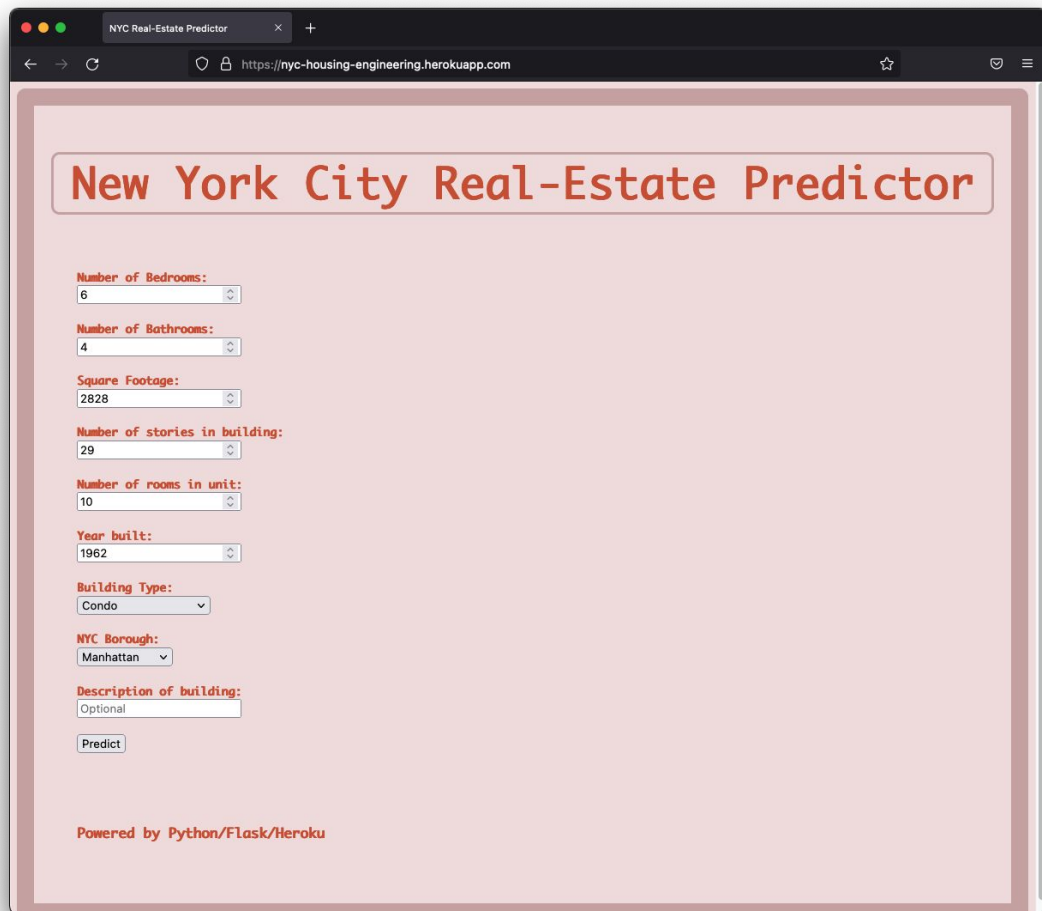
Test data r^2 : 0.57

As the pipeline acquires more and more data, the problems caused by overfitting will diminish.

The Flask App

The model can be used and tested at:

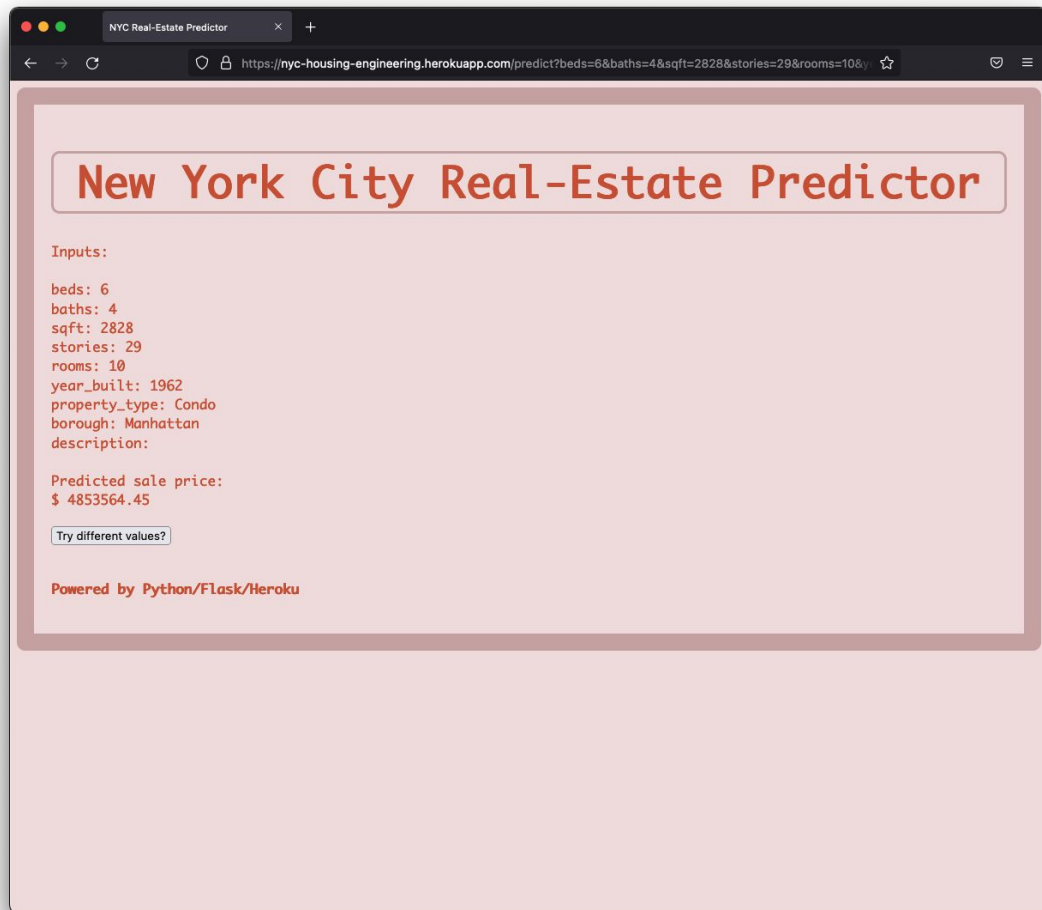
<https://nyc-housing-engineering.herokuapp.com>



The screenshot shows a web browser window with the title "NYC Real-Estate Predictor" and the URL "https://nyc-housing-engineering.herokuapp.com". The page has a light pink background and a title "New York City Real-Estate Predictor" in a rounded rectangle. Below the title are several input fields with red labels: "Number of Bedrooms:" (value 6), "Number of Bathrooms:" (value 4), "Square Footage:" (value 2828), "Number of stories in building:" (value 29), "Number of rooms in unit:" (value 10), "Year built:" (value 1962), "Building Type:" (dropdown menu showing "Condo"), "NYC Borough:" (dropdown menu showing "Manhattan"), and "Description of building:" (text input with "Optional"). A "Predict" button is located below the description field. At the bottom of the page, it says "Powered by Python/Flask/Heroku".

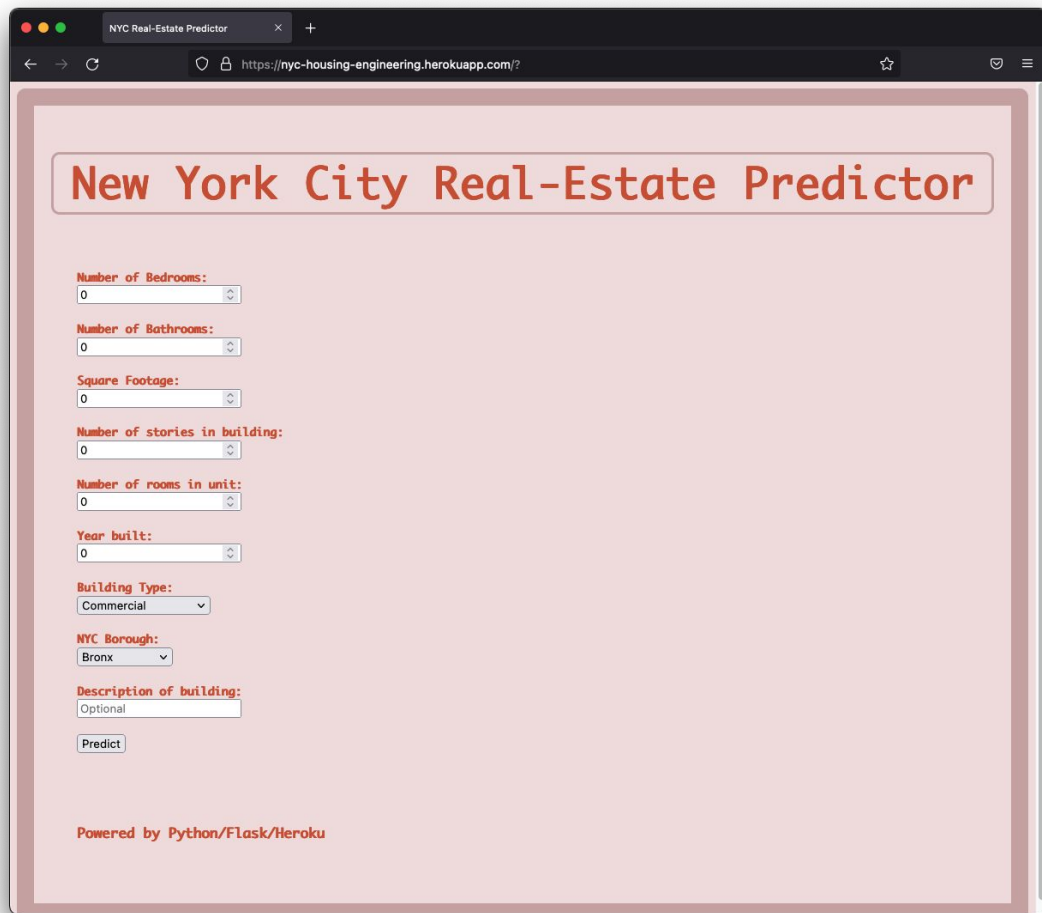
The Flask App

The app takes several real-estate features and a written description (optional), then predicts the predicted sale price.



The Flask App

Entering out-of-range values prompts the user to input valid ones instead.



The screenshot shows a web browser window with the title "NYC Real-Estate Predictor" and the URL "https://nyc-housing-engineering.herokuapp.com/?". The page has a light pink background and a title "New York City Real-Estate Predictor" in a rounded rectangle. Below the title are several input fields, each with a red label and a white input box with a small up/down arrow on the right. The fields are: "Number of Bedrooms:" (0), "Number of Bathrooms:" (0), "Square Footage:" (0), "Number of stories in building:" (0), "Number of rooms in unit:" (0), "Year built:" (0), "Building Type:" (Commercial), "NYC Borough:" (Bronx), and "Description of building:" (Optional). A "Predict" button is located below the description field. At the bottom of the page, it says "Powered by Python/Flask/Heroku".

NYC Real-Estate Predictor

<https://nyc-housing-engineering.herokuapp.com/?>

New York City Real-Estate Predictor

Number of Bedrooms:
0

Number of Bathrooms:
0

Square Footage:
0

Number of stories in building:
0

Number of rooms in unit:
0

Year built:
0

Building Type:
Commercial

NYC Borough:
Bronx

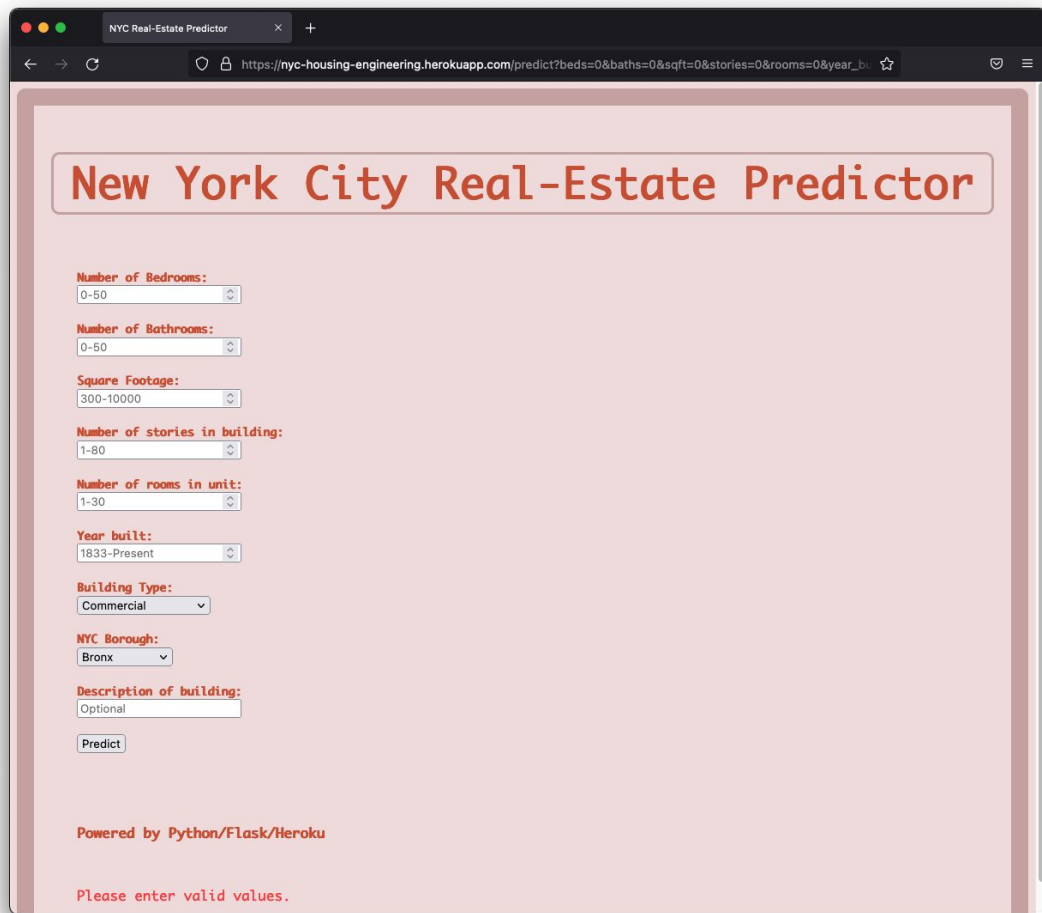
Description of building:
Optional

Predict

Powered by Python/Flask/Heroku

The Flask App

Entering out-of-range values prompts the user to input valid ones instead.



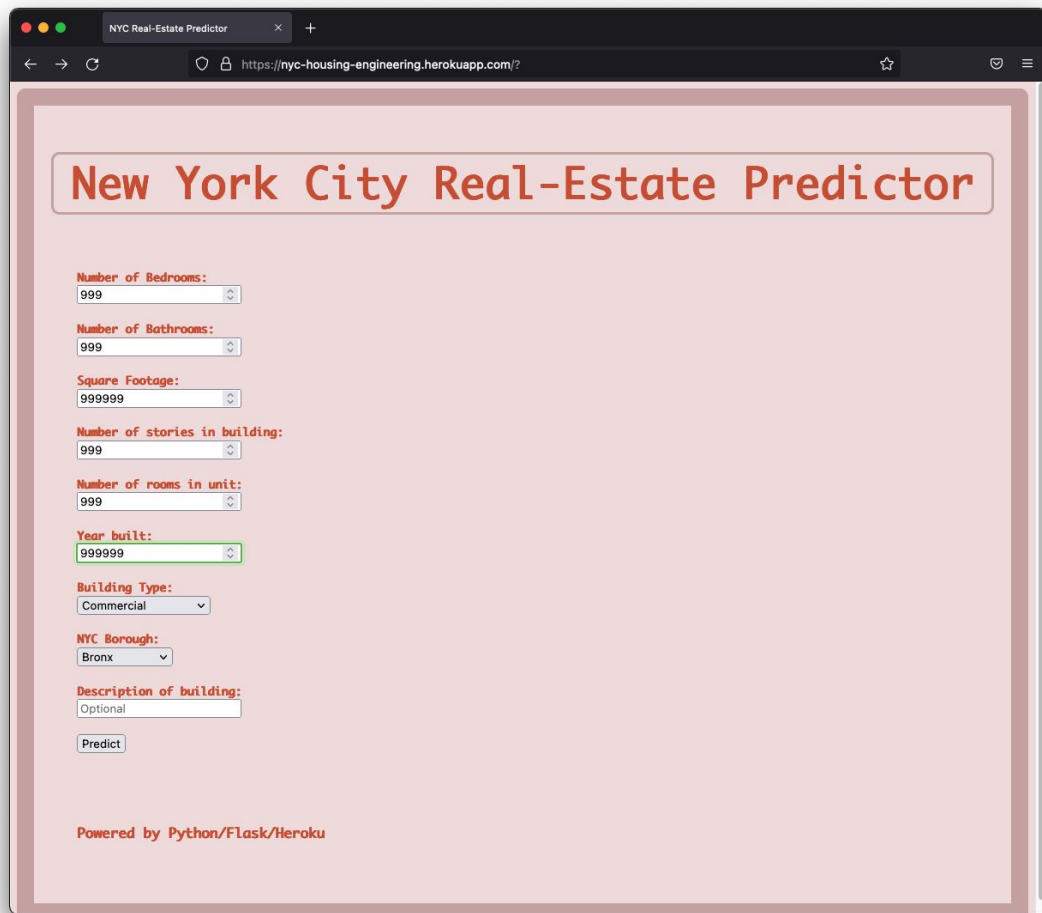
The screenshot shows a web browser window with the title "NYC Real-Estate Predictor". The address bar shows the URL: https://nyc-housing-engineering.herokuapp.com/predict?beds=0&baths=0&sft=0&stories=0&rooms=0&year_built=0. The main content area has a light pink background and a title "New York City Real-Estate Predictor" in a rounded rectangle. Below the title are several input fields with red labels and ranges:

- Number of Bedrooms:** 0-50 (range: 0-50)
- Number of Bathrooms:** 0-50 (range: 0-50)
- Square Footage:** 300-10000 (range: 300-10000)
- Number of stories in building:** 1-80 (range: 1-80)
- Number of rooms in unit:** 1-30 (range: 1-30)
- Year built:** 1833-Present (range: 1833-Present)
- Building Type:** Commercial (dropdown menu)
- NYC Borough:** Bronx (dropdown menu)
- Description of building:** Optional (text input)

Below the input fields is a "Predict" button. At the bottom of the form, it says "Powered by Python/Flask/Heroku" and "Please enter valid values.".

The Flask App

Entering out-of-range values prompts the user to input valid ones instead.



The screenshot shows a web browser window with the title "NYC Real-Estate Predictor" and the URL "https://nyc-housing-engineering.herokuapp.com/?". The page has a light pink background and a title "New York City Real-Estate Predictor" in a rounded rectangle at the top. Below the title are several input fields, each with a red label and a white input box with a small up/down arrow on the right. The inputs are: "Number of Bedrooms:" with value "999", "Number of Bathrooms:" with value "999", "Square Footage:" with value "999999", "Number of stories in building:" with value "999", "Number of rooms in unit:" with value "999", and "Year built:" with value "999999" (this field has a green border). Below these are two dropdown menus: "Building Type:" with "Commercial" selected, and "NYC Borough:" with "Bronx" selected. There is also a text input field for "Description of building:" with the value "Optional". At the bottom of the form is a "Predict" button. At the very bottom of the page, it says "Powered by Python/Flask/Heroku".

NYC Real-Estate Predictor

<https://nyc-housing-engineering.herokuapp.com/?>

New York City Real-Estate Predictor

Number of Bedrooms:
999

Number of Bathrooms:
999

Square Footage:
999999

Number of stories in building:
999

Number of rooms in unit:
999

Year built:
999999

Building Type:
Commercial

NYC Borough:
Bronx

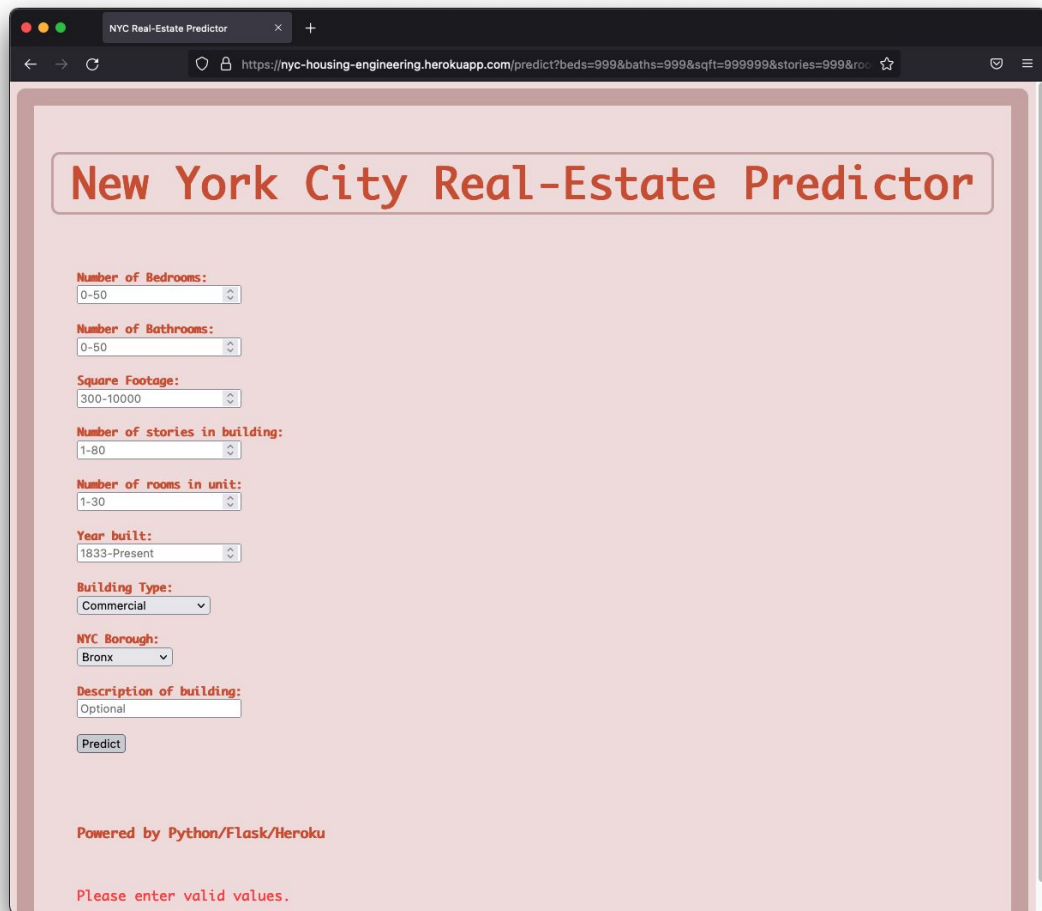
Description of building:
Optional

Predict

Powered by Python/Flask/Heroku

The Flask App

Entering out-of-range values prompts the user to input valid ones instead.



The screenshot shows a web browser window with the title "NYC Real-Estate Predictor". The address bar shows the URL: <https://nyc-housing-engineering.herokuapp.com/predict?beds=999&baths=999&sqft=999999&stories=999&rooms=999&year=999&borough=Bronx&description=Optional>. The page has a light pink background and a title "New York City Real-Estate Predictor" in a rounded rectangle at the top.

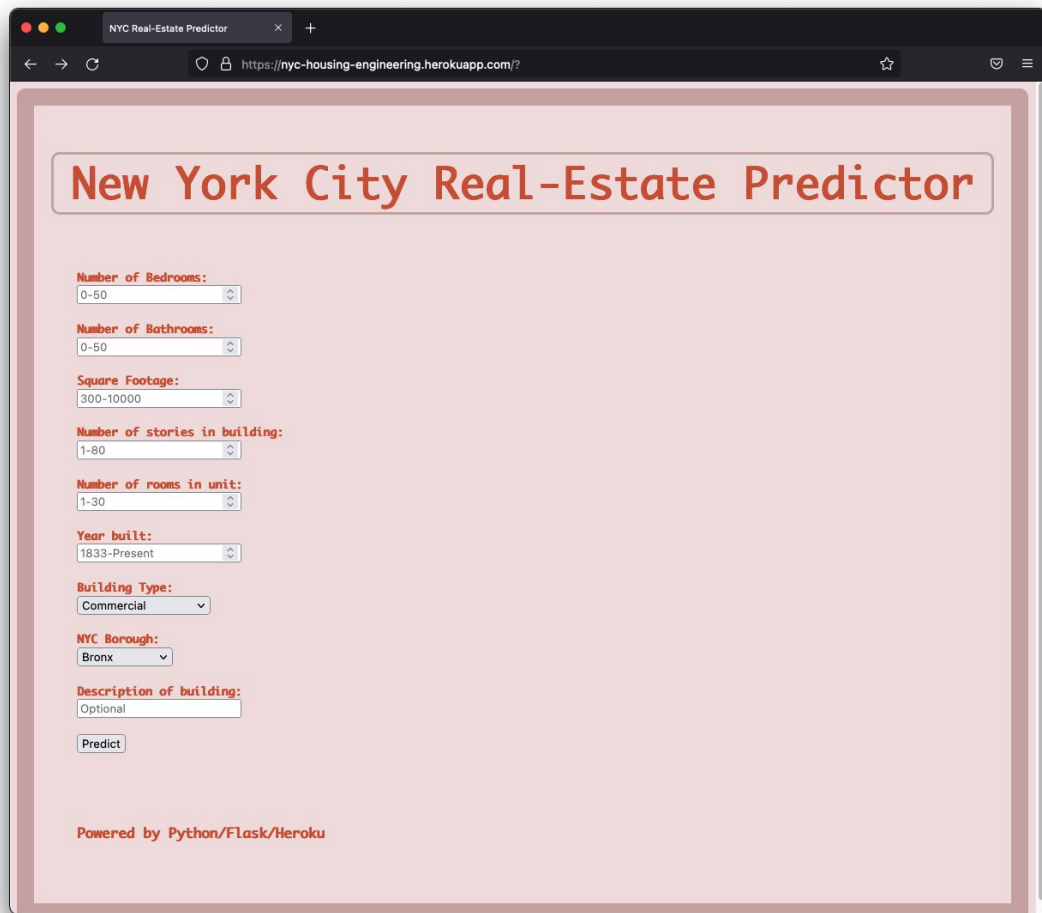
The form contains the following fields:

- Number of Bedrooms:** A range input with "0-50".
- Number of Bathrooms:** A range input with "0-50".
- Square Footage:** A range input with "300-10000".
- Number of stories in building:** A range input with "1-80".
- Number of rooms in unit:** A range input with "1-30".
- Year built:** A range input with "1833-Present".
- Building Type:** A dropdown menu with "Commercial" selected.
- NYC Borough:** A dropdown menu with "Bronx" selected.
- Description of building:** A text input with "Optional" entered.

Below the form is a "Predict" button. At the bottom of the page, it says "Powered by Python/Flask/Heroku" and "Please enter valid values." in red text.

The Flask App

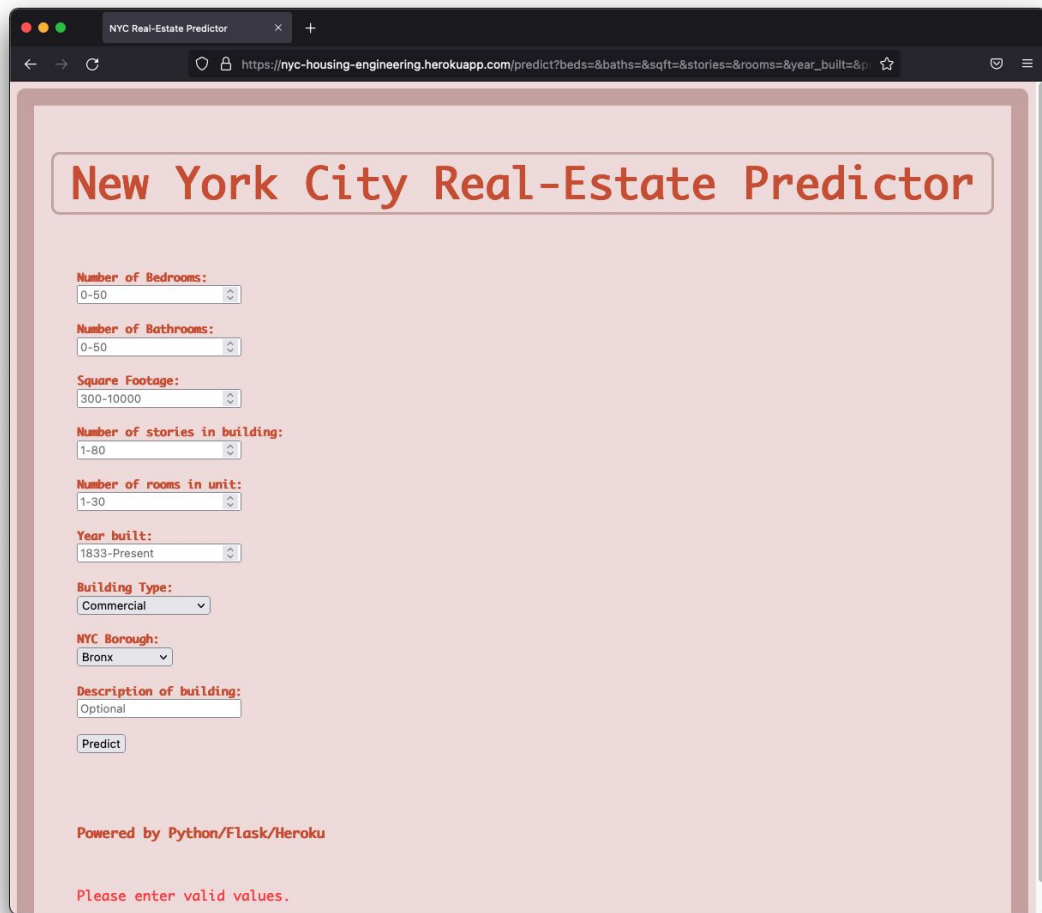
So does leaving values blank.



The screenshot shows a web browser window with the title "NYC Real-Estate Predictor" and the URL "https://nyc-housing-engineering.herokuapp.com/?". The page has a light pink background and a title "New York City Real-Estate Predictor" in a red-bordered box. Below the title are several input fields with red labels: "Number of Bedrooms:" (range 0-50), "Number of Bathrooms:" (range 0-50), "Square Footage:" (range 300-10000), "Number of stories in building:" (range 1-80), "Number of rooms in unit:" (range 1-30), "Year built:" (range 1833-Present), "Building Type:" (dropdown menu with "Commercial" selected), "NYC Borough:" (dropdown menu with "Bronx" selected), and "Description of building:" (text input with "Optional" entered). A "Predict" button is located below the description field. At the bottom of the page, it says "Powered by Python/Flask/Heroku".

The Flask App

So does leaving values blank.



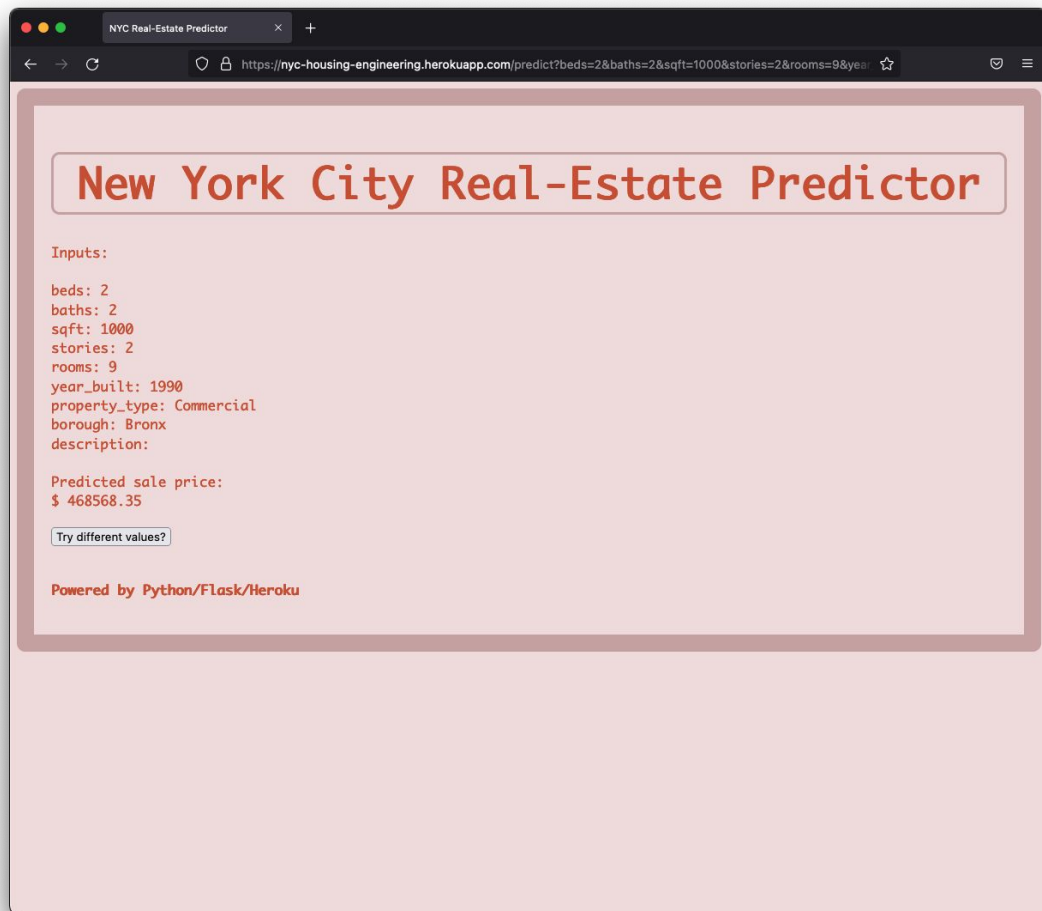
The screenshot shows a web browser window with the title "NYC Real-Estate Predictor". The URL in the address bar is https://nyc-housing-engineering.herokuapp.com/predict?beds=&baths=&sft=&stories=&rooms=&year_built=&building_type=&borough=&description=. The main heading is "New York City Real-Estate Predictor". Below the heading are several input fields, each with a red label and a range of values:

- Number of Bedrooms:** 0-50
- Number of Bathrooms:** 0-50
- Square Footage:** 300-10000
- Number of stories in building:** 1-80
- Number of rooms in unit:** 1-30
- Year built:** 1833-Present
- Building Type:** Commercial (dropdown menu)
- NYC Borough:** Bronx (dropdown menu)
- Description of building:** Optional (text input field)

Below the input fields is a "Predict" button. At the bottom of the form, it says "Powered by Python/Flask/Heroku" and "Please enter valid values."

The Flask App

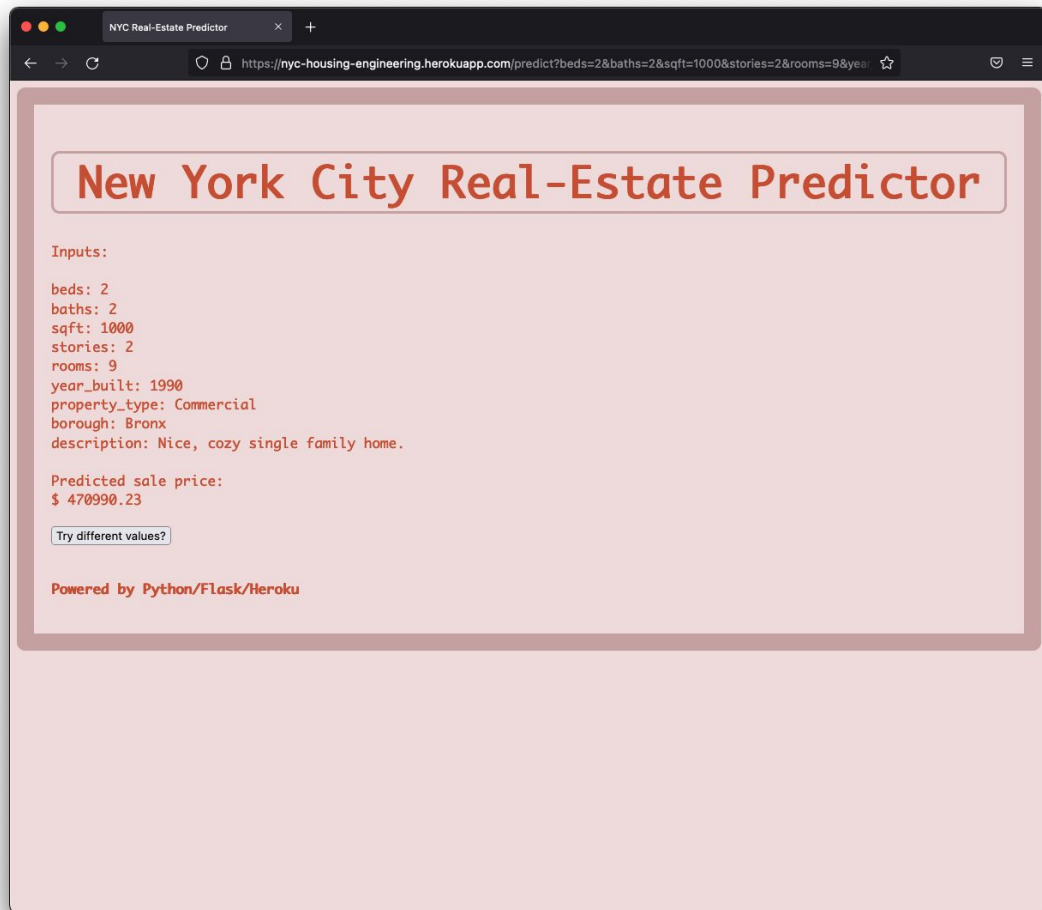
Writing a positive description can help increase your property's sale price!



The Flask App

Writing a positive description can help increase your property's sale price!

Disclaimer: Writing this exact sentence may not increase your home's price by \$2,000.



Future Work

- Refine website; make it better looking and more user-friendly
- Create an API that can process more than one listing per user at a time
- Migrate away from Pandas, which is not scalable for large data sets

Questions?