


Predicting NY Real Estate Prices

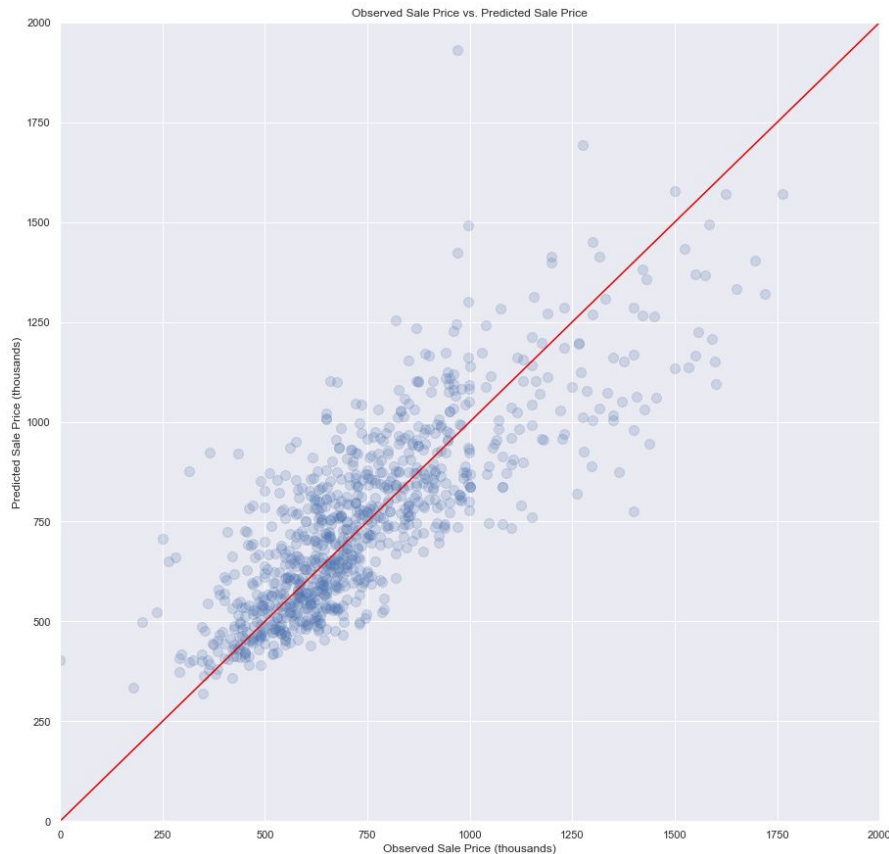
Using web scraping, geocoding, sentiment analysis, topic modeling, random forests, and the command line to create and update a data pipeline

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

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.

Methodology

Data:

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
(10 dummies, 64 NLP topics)

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

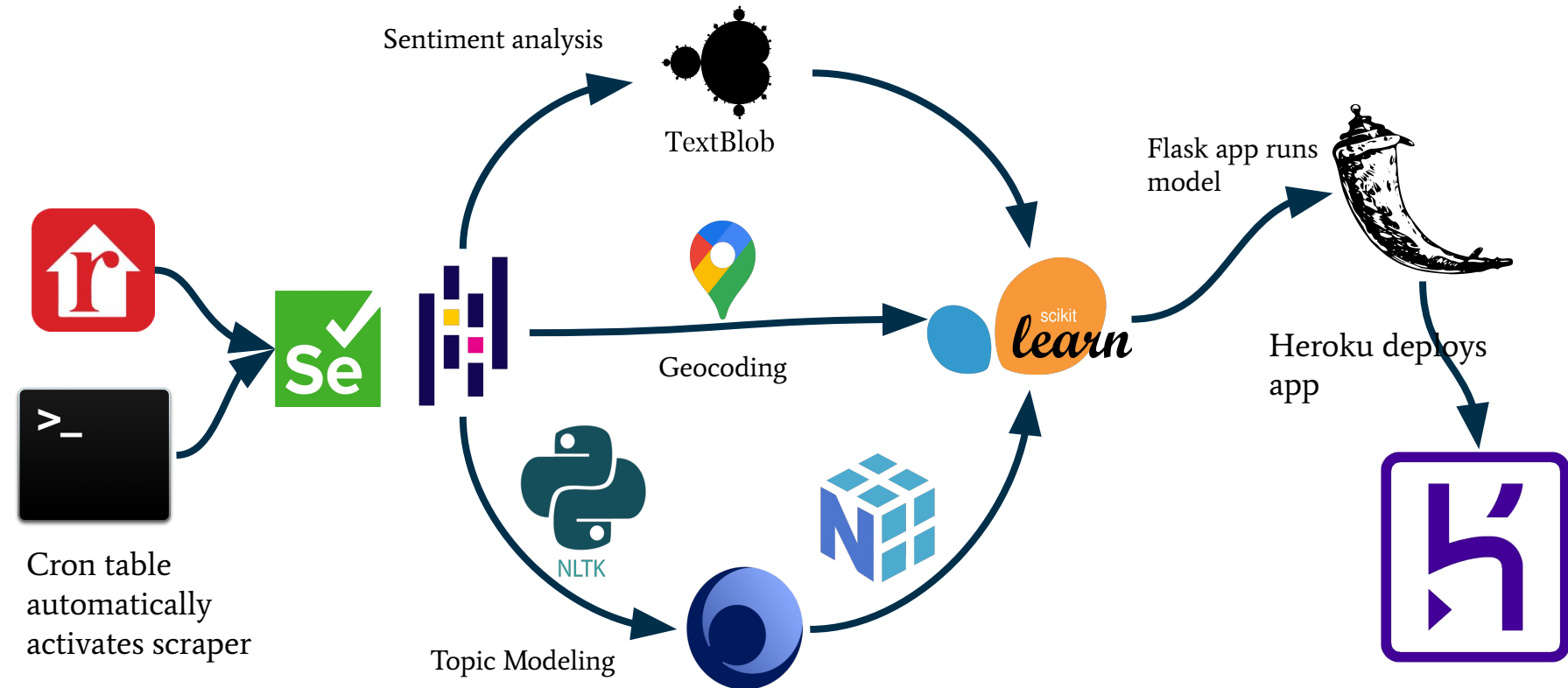
MacOS Command Line scripts for automated data collection and model updates

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

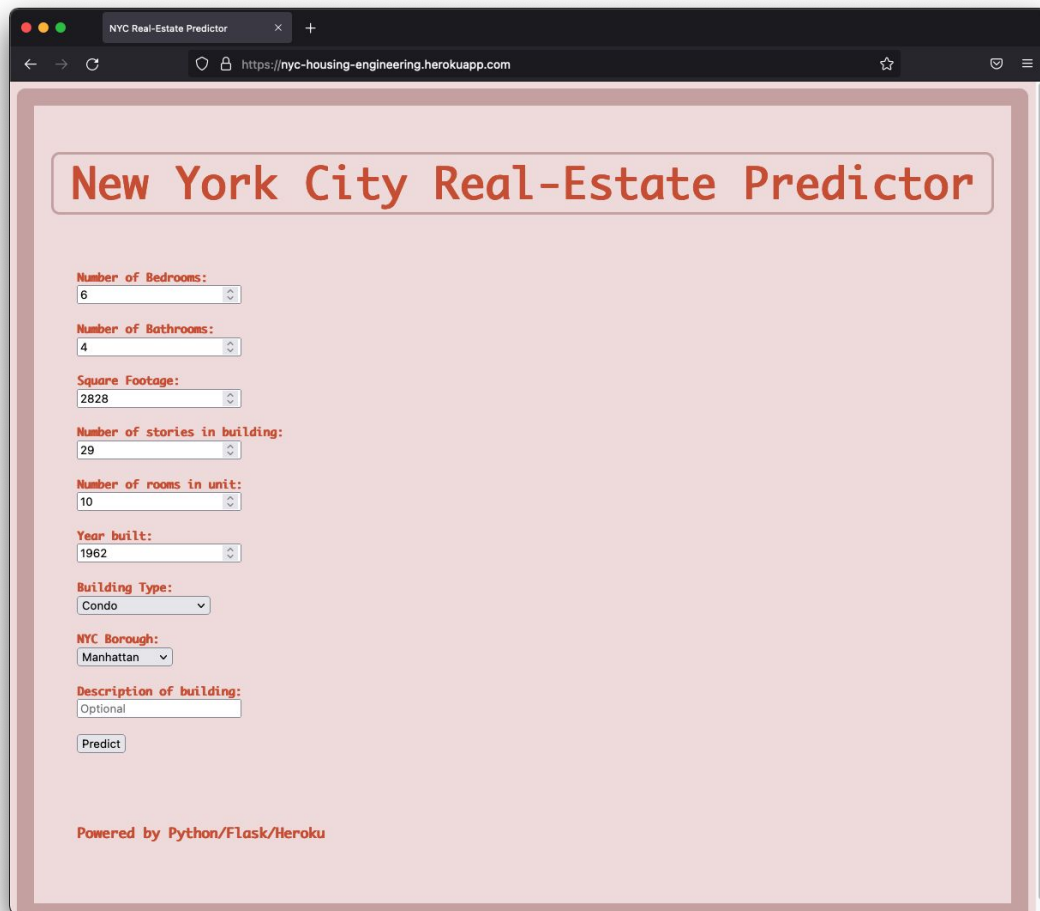
Test data r^2 : 0.57

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

What's the real price of your estate?

The model can be used and tested at:

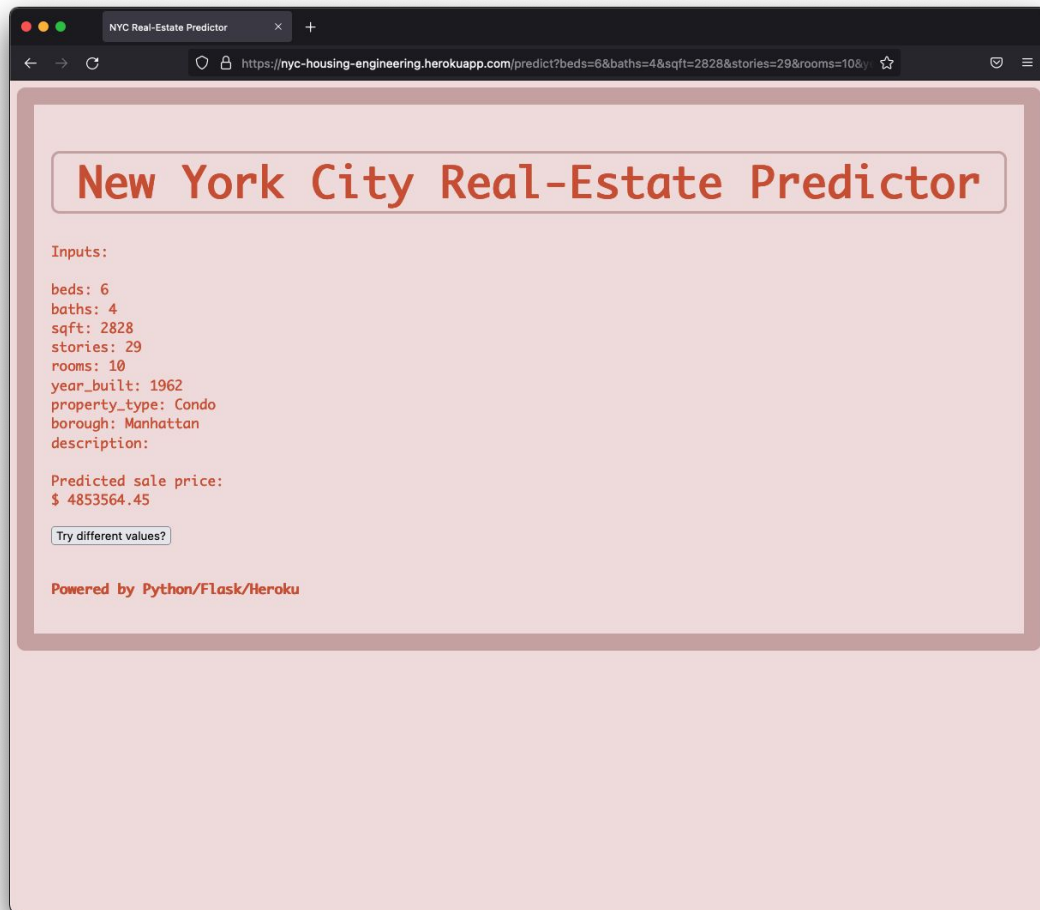
<https://nyc-housing-engineering.herokuapp.com>



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 red-bordered box. 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.



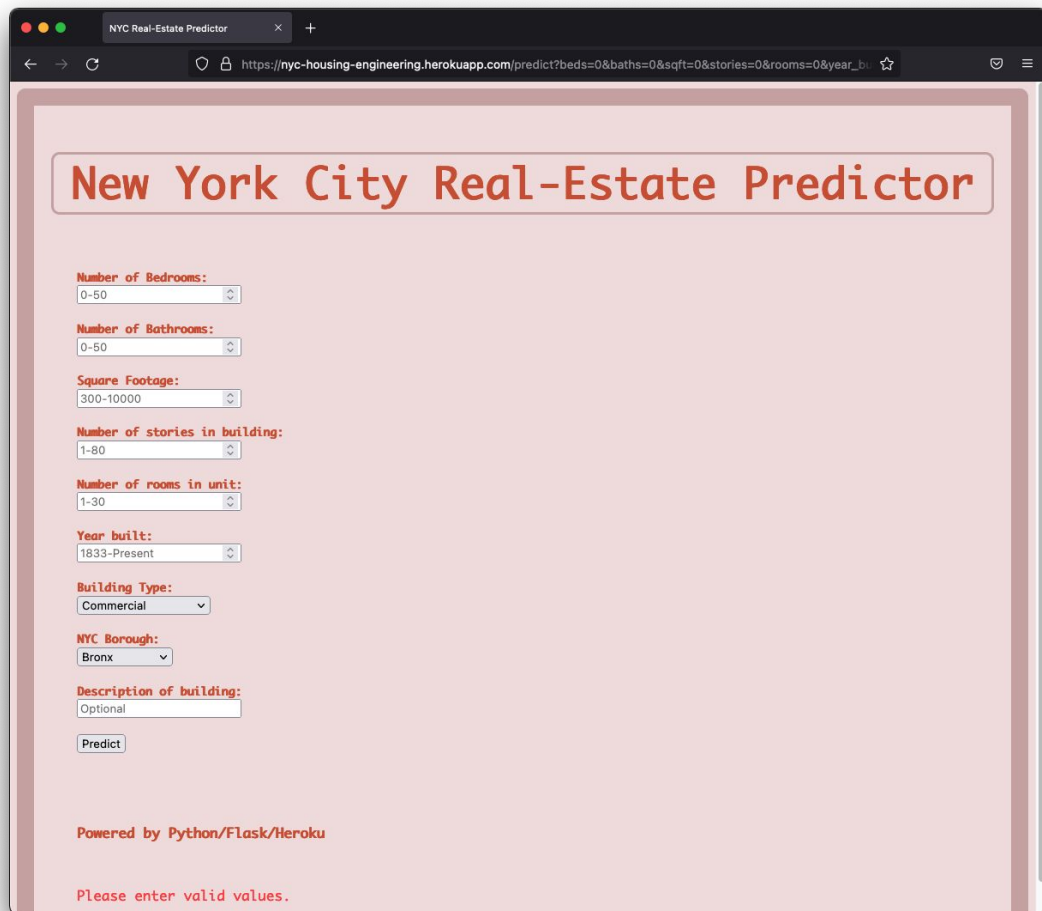
The Flask App

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 label in red text and a value of '0' (except for the dropdowns). The fields are: 'Number of Bedrooms:', 'Number of Bathrooms:', 'Square Footage:', 'Number of stories in building:', 'Number of rooms in unit:', 'Year built:', 'Building Type:' (a dropdown menu showing 'Commercial'), 'NYC Borough:' (a dropdown menu showing 'Bronx'), and 'Description of building:' (a text input field with 'Optional' entered). A 'Predict' button is located below the 'Description of building:' field. At the bottom of the page, it says 'Powered by Python/Flask/Heroku'.

The Flask App

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



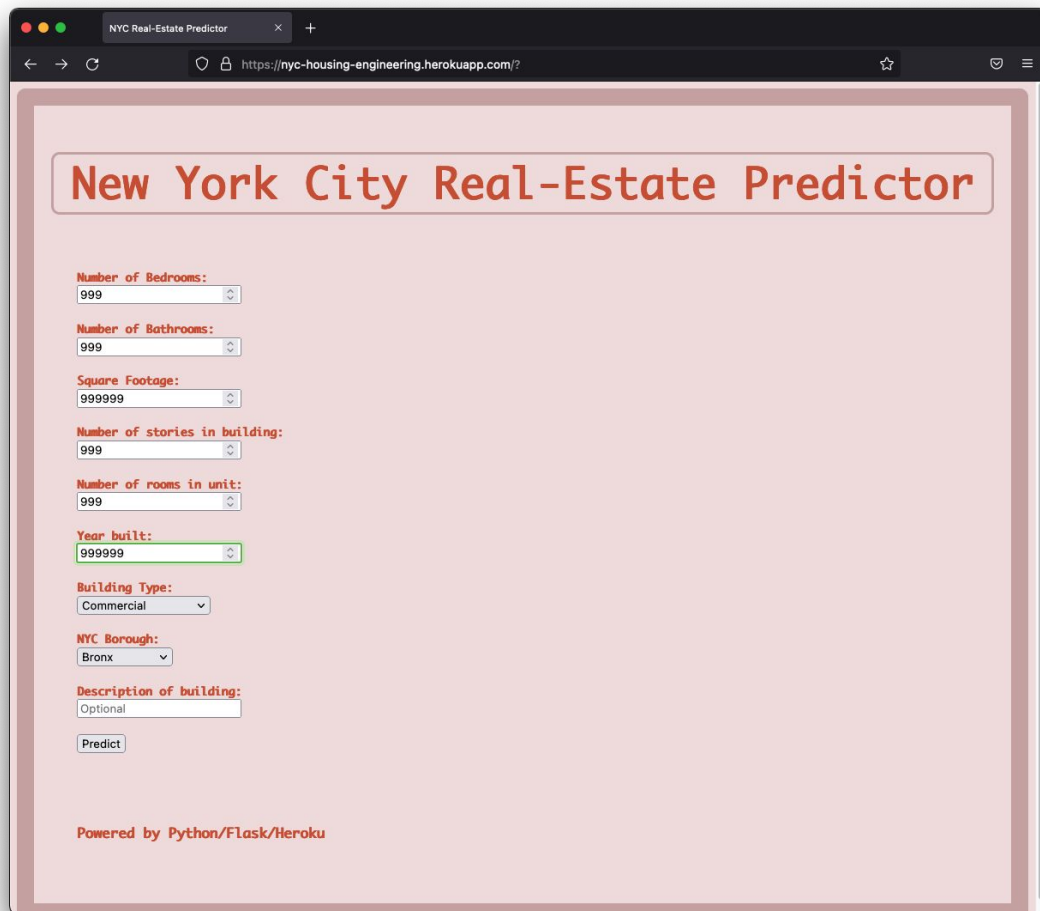
The screenshot shows a web browser window with the title "NYC Real-Estate Predictor". The address bar shows the URL: https://nyc-housing-engineering.herokuapp.com/predict?beds=0&baths=0&sft=0&stories=0&rooms=0&year_built=0. The main heading is "New York City Real-Estate Predictor". Below the heading are several input fields with red labels and red error messages:

- Number of Bedrooms:** Input field shows "0-50". Error message: "Number of Bedrooms must be between 1 and 50."
- Number of Bathrooms:** Input field shows "0-50". Error message: "Number of Bathrooms must be between 1 and 50."
- Square Footage:** Input field shows "300-10000". Error message: "Square Footage must be between 300 and 10000."
- Number of stories in building:** Input field shows "1-80". Error message: "Number of stories in building must be between 1 and 80."
- Number of rooms in unit:** Input field shows "1-30". Error message: "Number of rooms in unit must be between 1 and 30."
- Year built:** Input field shows "1833-Present". Error message: "Year built must be between 1833 and Present."
- Building Type:** Dropdown menu shows "Commercial". Error message: "Building Type must be either Residential or Commercial."
- NYC Borough:** Dropdown menu shows "Bronx". Error message: "NYC Borough must be either Manhattan, Bronx, Queens, Richmond, or Brooklyn."
- Description of building:** Input field shows "Optional". Error message: "Description of building is required."

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. 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:" (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), 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".

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

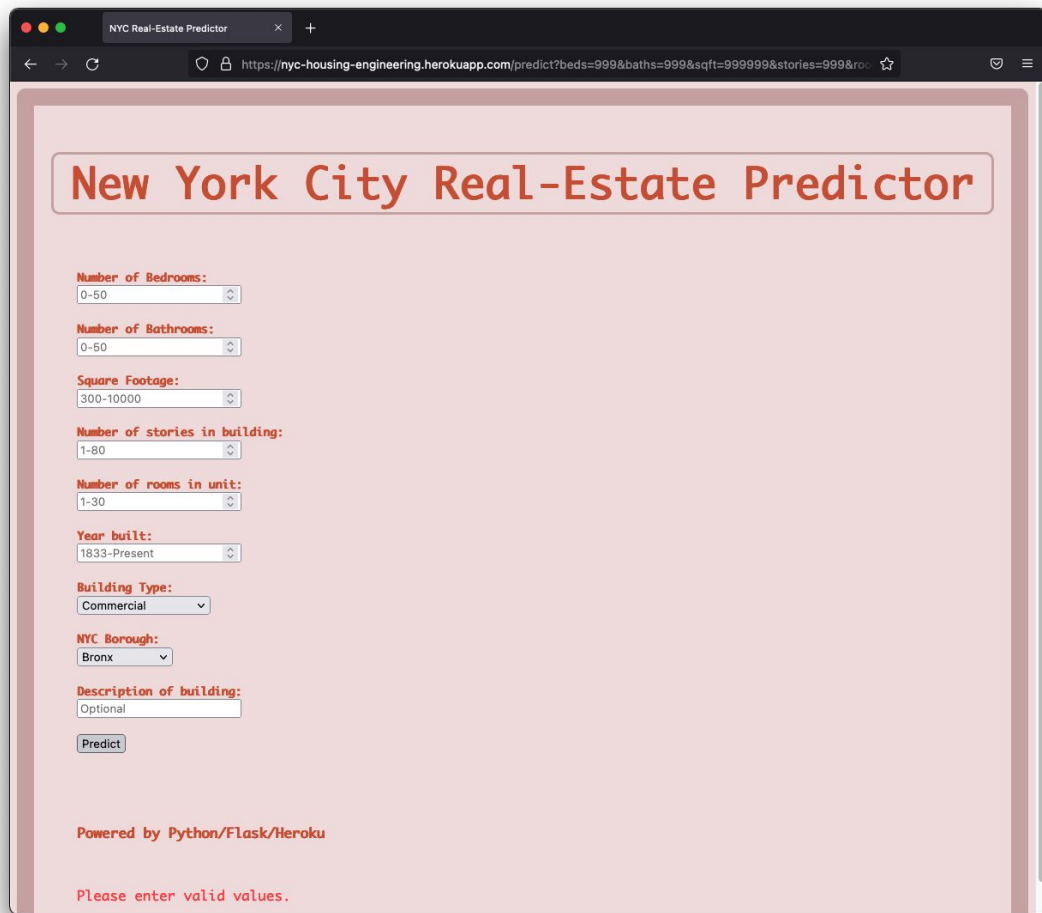
Description of building:
Optional

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 URL in the address bar is <https://nyc-housing-engineering.herokuapp.com/predict?beds=999&baths=999&sqft=999999&stories=999&rooms=999&year=999&type=Commercial&borough=Bronx&description=Optional>. 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 range: "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), "NYC Borough:" (Bronx), and "Description of building:" (Optional). A "Predict" button is at the bottom of the form. At the bottom of the page, it says "Powered by Python/Flask/Heroku" and "Please enter valid values." in red text.

New York City Real-Estate Predictor

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

NYC Borough:
Bronx

Description of building:
Optional

Predict

Powered by Python/Flask/Heroku

Please enter valid values.

The Flask App

So does leaving values blank.

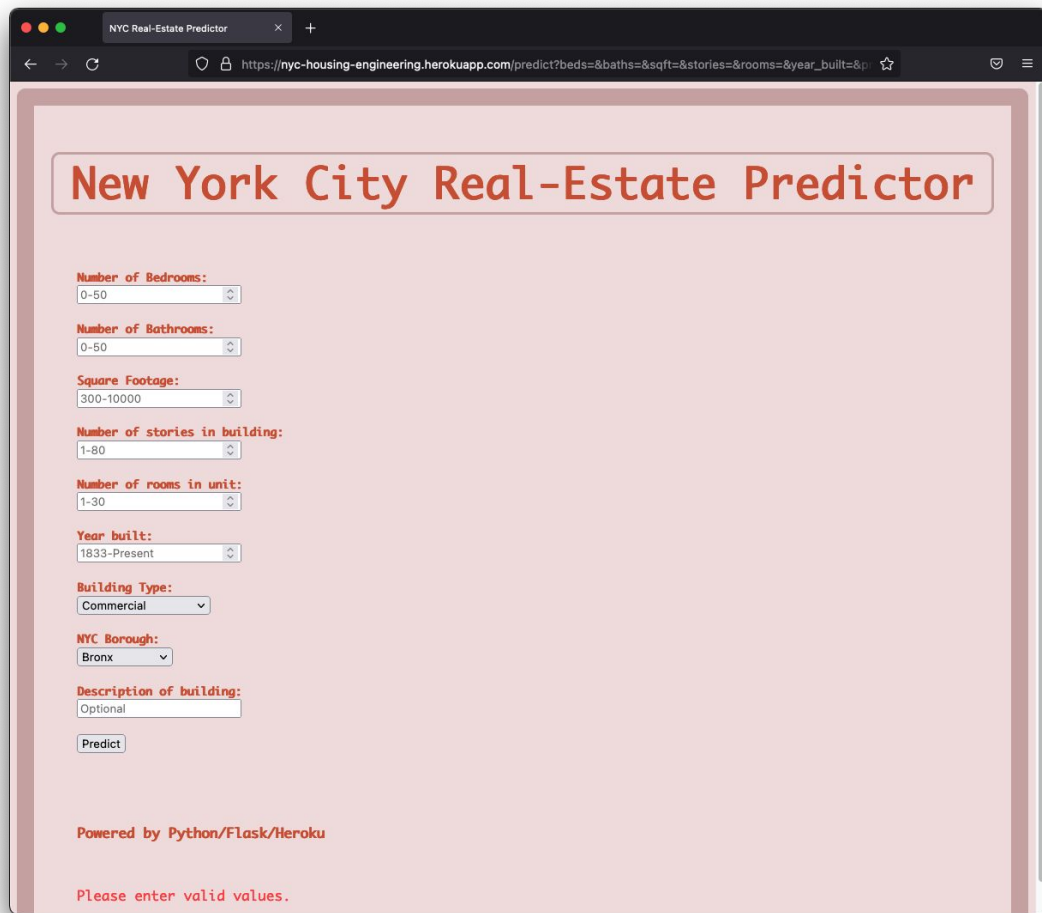
The screenshot shows a web browser window with the title 'NYC Real-Estate Predictor' and the URL 'https://nyc-housing-engineering.herokuapp.com/'. The page has a light pink background and a title 'New York City Real-Estate Predictor' in a red-bordered box. Below the title is a form with the following fields:

- Number of Bedrooms:** A range input field with '0-50'.
- Number of Bathrooms:** A range input field with '0-50'.
- Square Footage:** A range input field with '300-10000'.
- Number of stories in building:** A range input field with '1-80'.
- Number of rooms in unit:** A range input field with '1-30'.
- Year built:** A range input field 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 field with 'Optional'.

At the bottom of the form is a 'Predict' button. Below the form, it says 'Powered by Python/Flask/Heroku'.

The Flask App

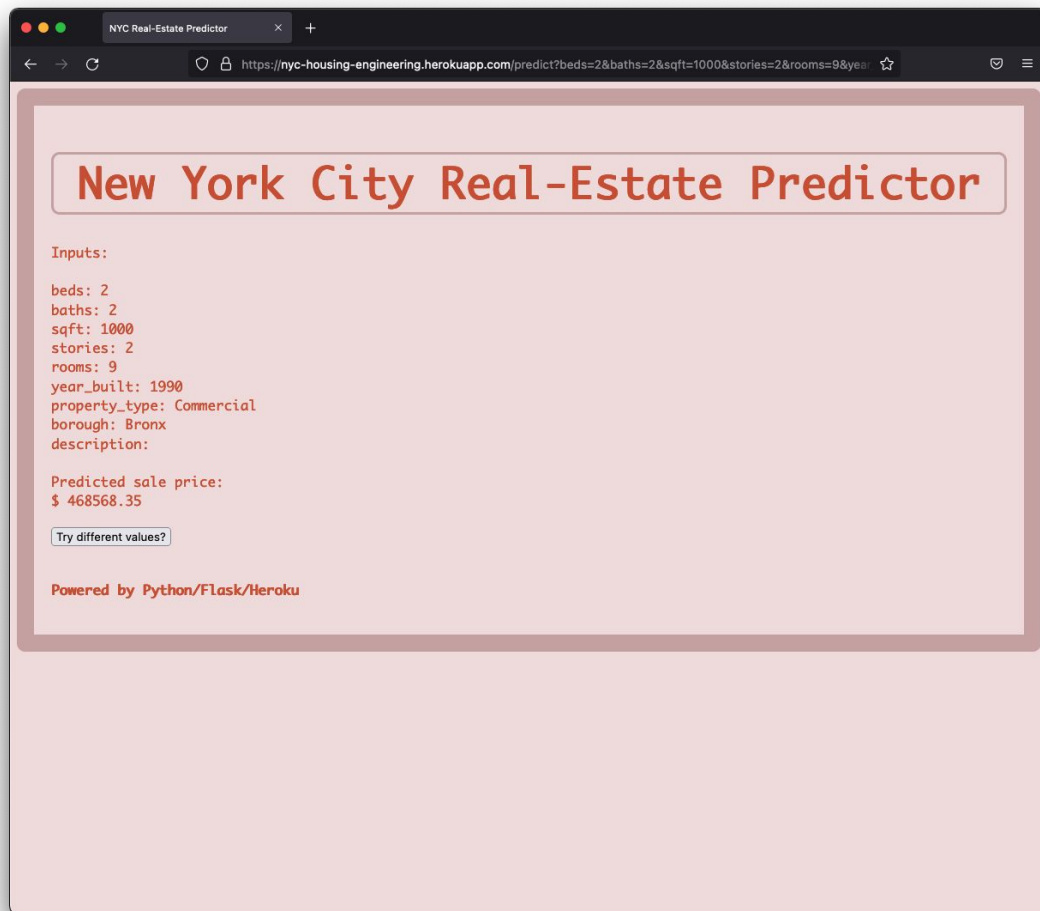
So does leaving values blank.



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 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" and "Please enter valid values."

The Flask App

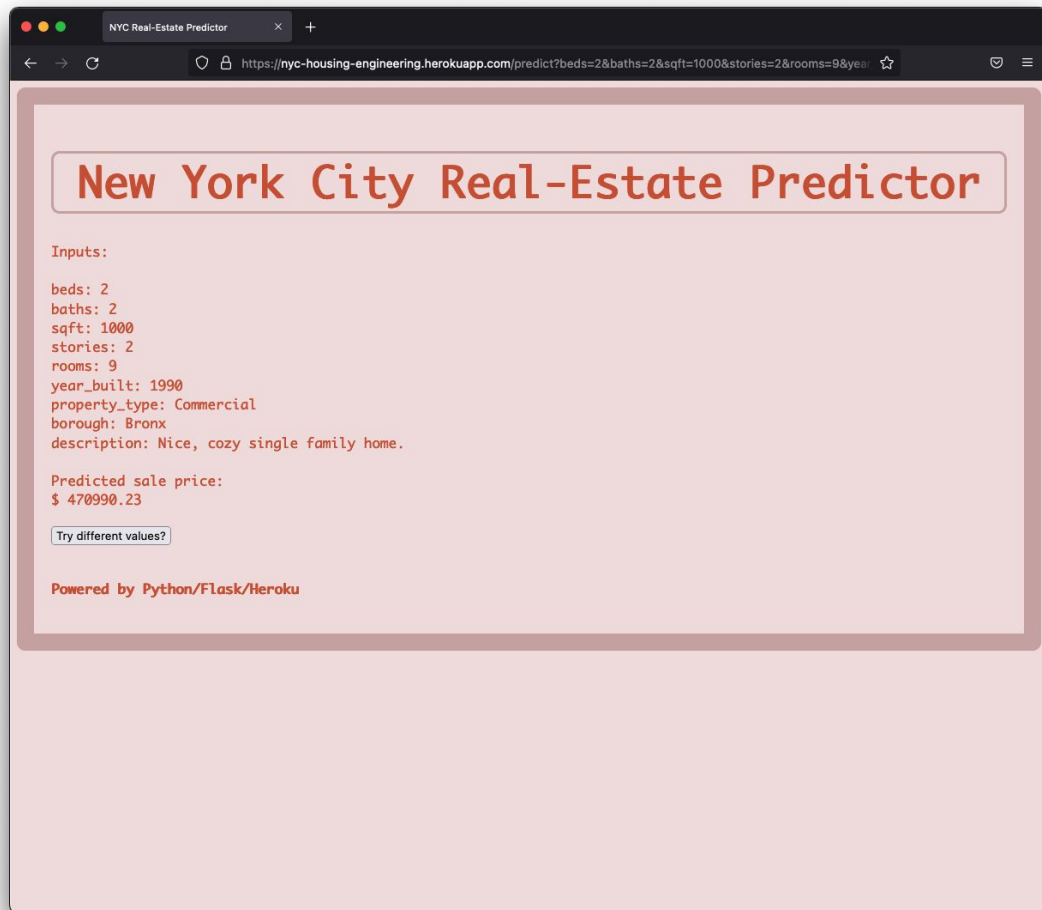
Writing a positive description can help increase your property's sale price!



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
- Allow users to give feedback on website for improvement

Questions?