Abstract

Home price growth is rapidly on the surge due to tight supply and lower interest rates and borrowing costs. With the recovering economy, more buyers are entering the market. Due to a limited supply of houses available, homes are taken off real estate websites like Zillow, just as fast as they're placed on them. For this reason, I developed a linear regression model to predict the price of a home, given a set of features- so that when homebuyers see a home that's not yet listed, they can have an estimate of it's price just by the features it has.
Data was obtained from Zillow. I refined the linear regression model by transforming data, adding polynomial features, and creating dummy variables for categorical data.

Data

The data for this project was scraped from the website zillow.com, as well as gathered in csv format from census.gov (not-scraped). Data scraped focused on houses located in Staten Island and Brooklyn. Along with price as my target variable, I scraped the number of beds, baths, square footage, lot size, zip code, parking availability, cooling and heating. I then used census data to get information about population, density in a zip code, and average household income in that zipcode. I theorized that higher income areas would give me higher priced homes.

Tools and Algorithms

Web Scraping

I used BeautifulSoup to get urls from 20 pages, (including each house view url). I then parsed through the scraped htmls and appended data into many dictionaries. Lastly I created a list of dictionaries in order to convert into a pandas dataframe.

Data cleaning and feature engineering

Once data was in a dataframe, I merged census data with a left join on zipcodes. I then proceeded to clean data to fill in values intuitively and drop the rest of the nulls. I chose to drop nulls instead of filling them with means because I felt it would give my model better predictability if it was trained on real data rather than interpolated data.

The methodology I employed was to add polynomial terms, standard scale, add categorical data, fit a polynomial regression model and regularize with Lassocv.

Upon creating diagnostic plots for my model I discovered that the model predicted relatively well for homes between 400K and 800K and extremely poorly for homes over 2M. I decided to drop those values in order to prioritize predictability- which would've been more difficult with such large outliers. I used square feet as a means to filter out home values by some threshold square foot value. Unfortunately that did not adequately get rid of outliers- some homes are valued high due to other confounding variables. I decided to drop homes under 2M directly.

Linear regression modeling and evaluation

Linear regression fitting was performed by SciKit-Learn. Linear regression models were evaluted by SciKit-Learn and StatsModel.

Data visualization

Seaborn and matplotlib were used for data visualization.

Results/Design

I first examined the distribution of my target variable (home price) to ensure that the data was normally distributed or accurately represented home values based on my domain knowledge. To develop a linear regression model, I set aside 20% of the data as a test set. I then split the remaining 80% of the data into a training set (60% of all data) and validation set (20% of all data). I also cross-validated R^2 with a Kfold of 5. I ran polynomial features and did feature engineering on both numerical and categorical data.The model i chose had a high R^2, relatively high validation R^2 compared to other models and a lower error metric than other models. I then regularized the model with LassoCV, and finally, retrained the linear regression model using the same poly features and feature engineering as above.

With the results I obtained,, I concluded that this model has poor predictive power for higher priced homes, and potentially needs more data to add complexity to the model.

Communication

Results were presented to colleagues in the Metis data science bootcamp on June 11, 2021. All notebooks and datasets are in my github.