Project 2 Summary - Vancouver and Seattle Home Price Prediction
Web Scraping and Linear Regression Project
Xinru (Flora) Cheng

For this project, my main objective was to predict listing prices of homes for sale in both Vancouver and Seattle. My initial goal was to compare listing prices in the two cities, given a set of features such as home type, number of bedrooms, etc. Insights gained from this project could be useful for potential homebuyers considering to relocate to either of these cities, as well as to real estate investors.

However, after performing web scraping and exploratory data analysis, I realized the Vancouver and Seattle datasets have different features, and I would not have enough time to thoroughly analyze the effects of those features for each city's home listing prices. I decided to focus on model selection and tuning using my current dataset, setting aside the comparison for later.

I wrote the web scraping data using Selenium. Main takeaways include how versatile "find element by XPath" is, as well as how websites with similar functions and appearance can differ by a lot when we inspect the HTML code.
Exploratory data analysis was performed on both datasets, the scraped Vancouver data needed more cleaning (such as using string methods and regular expressions to replace column names). I decided to log transform the target column (price) because pairplots showed it being really positively-skewed, and log transform would help it look more Gaussian. Later I transformed it back to calculate the Root Mean Squared Error (RMSE) so the error would have realistic units in dollars.

For modeling I first did a 80-20 train-test split for the mvp (minimum viable product) for this project, and plotted the actual versus predicted price (both log-transformed). Then I performed cross-validation with 5 and 10 folds, for more stable models and more reliable results. During this phase, I compared linear, 2nd and 3rd degree polynomial models using R^2 as my metric (shows how much of the variation in the target variable can be explained by our model). The simple linear regression model performed best, with a value of 0.713 for Vancouver and 0.501 for Seattle. Because the training R^2 was slightly larger than the testing R^2 for all models, I then performed Ridge and Lasso regularization to adjust for overfitting. Because the number of features used is already small, I chose Ridge over Lasso since Lasso could eliminate useful features.

Before calculating the RMSE, I decided to remove outliers (> 3 million USD) to see how much it improved prediction results. For the Seattle dataset the improvement was significant: from 411k (~57% of median) to 298k (~40% of median).

For standardization of the data, I explored a number of scalers available in Scikit-learn (Standard, MinMax, Robust, QuantileTransform, PowerTransform). But for these datasets and models I did not observe any significant improvement over the StandardScaler.

Data

I used a combination of web-scraped data and downloaded flat files. The Vancouver dataset was scraped from redfin.ca, and the Seattle dataset was downloaded from redfin.com. They were the latest data available on their website during the first week of October 2019. The two websites differ slightly in their layouts which made it harder to scrape them with the same code. They also differ in features The two datasets combined have about 450 observations (rows) and 27 unique features (columns). For the analysis as of mid-October 2019, I focused on 5 numerical features (price, size in square feet, number of bedrooms, number of bathrooms, and days listed on the market).

After the analysis, I came to suspect there was some underlying bias in the data. This is because I could only access a small sample of the listings that were shown (100 of 3456 homes for sale in the Vancouver case, 350 of 2064 in the Seattle case). It was unclear from the website how those listings were selected. I didn’t realize there was this limitation until after I scraped the data (also see Next Steps).

Tools

Python, Pandas, NumPy, Scikit-learn, Seaborn
HTML, Selenium for web scraping
JupyterNotebook, Visual Studio Code

Next Steps

Given more time and resources, I would continue by examining the possible underlying bias in the dataset by plotting the residuals and looking for patterns indicating heteroskedasticity (the circumstance in which the variability of a variable is unequal across the range of values of a second variable that predicts it).

If the dataset from redfin is clearly biased, I would get more data from another website. Then I would improve upon and regularize my model, and compare coefficients of different features to find out how much impact each categorical feature had on the price. I also plan to perform more feature engineering on my models - in particular, adding “dummy variables” for the many categorical variables; as well as introducing 2nd and 3rd degree polynomial terms and interaction terms (ratio of number of bathrooms to bedrooms, for example) once I have a larger dataset. Once I introduce more features to the model, it could be beneficial to use Adjusted R^2 as the metric, since it only increases if the new term improves the model accuracy.

The ultimate goal that I will work towards is to develop a web API that potential homebuyers could use to compare home prices in two different cities, after selecting a particular set of filters (2 bed 2 bath, with garage and balcony, for example).

Lessons Learned (Project Design)

This was my first experience with web scraping, as well as the first project with complete freedom in choosing the topic and data. The dataset I wrote the web scraping code for was small and did not have enough features. Once I got the scraped data, I decided to work with what I had in the interest of time, only introducing one additional Seattle dataset. For future projects, I would examine the data sources in more detail when choosing a topic and formulating my research questions, and ideally combine data from multiple sources.

In terms of project design, it would have been more efficient if I chose to focus on one city, instead of trying to compare Vancouver and Seattle data. During the second and last week of this project, I found myself trying to repeat the same workflow in two different notebooks, then cleaning the code and combining them into one notebook. Narrowing down the scope and focusing on home prices in one city would allow me to go deeper in the analysis, or possibly scrape data from multiple real estate websites.

For future projects, I would also improve my file organization and version control process, and possibly use a cookiecutter template.