Analyzing Housing Affordability in Metro Vancouver

Gunnar Miller, Joshua Fantillo, Houli (Minty) Huang, Zhenmin He, Tayaba Abbasi
Simon Fraser University Graduate School
Burnaby, Canada

Abstract

This project develops an interactive tool and a machine learning model to analyze and predict housing and property values in the Greater Vancouver area, using data gathered from municipal sources and online databases. The final products aim to provide accessible insights into housing market trends and future price predictions, enhancing understanding and decision-making for residents, policymakers, and businesses.

I. MOTIVATION

Housing is important. Everybody needs a place to live, and the specifics of where you live can impact all sorts of other facets of life including work opportunities, social connections, health, ability to start a family and much more. In particular, employment and housing are very tightly connected: your income determines where you can live, and where you live determines what jobs are available. As a result, there is a marked tendency for economically productive cities to have soaring housing prices as an increasing number of people compete for housing in a limited geographic area. This trend can have many further undesirable downstream effects: increased traffic, increased pollution, increased poverty, and much more.

These trends are visible in many places, but Vancouver is at the forefront of them. It has the highest rents of any city in Canada and a median house price of $1,318,687, compared to a median after-tax salary of $50,441. This means that even if they were paying 0% mortgage interest and devoting fully 50% of their income to making payments (both obviously unlikely), it would take the average resident over 52 years to pay for a house.

II. PROBLEM STATEMENT

The overarching goal of this project is to collect and organize data about property and housing costs in the greater Vancouver area and present it in a centralized, accessible, easy-to-understand way. The goal is for anyone without either specialized computer skills or knowledge of the housing market to look at our final product and gain useful insights into the current state and overall trends of Vancouver housing prices. This could be useful in all sorts of context. It could be useful to people from outside considering moving into the Metro Vancouver area, for residents looking for housing, for companies looking where they want to base their operations out of and for policymakers looking to usefully address the affordable housing crisis. This study was challenging primarily because of the size, complexity and fragmentation of the data. There are a number of municipalities in Greater Vancouver, each one makes different data available in different ways, and all the datasets include both a large number of records and many different data types.

III. DATA SCIENCE PIPELINE

Overview

The overall flow of our pipeline is aggregating data from various online sources, turning it into a usable form and then integrating that form into our interactive map. In addition, some of the data gets used to train the machine learning model, which then makes predictions that can also be integrated into the map. The data from two of our sources, the Zillow rental data and the Langley property values data, needed to be retrieved with a web scraper before being integrated into the rest of the pipeline. The remainder was available in a tabular form from various municipal web pages.

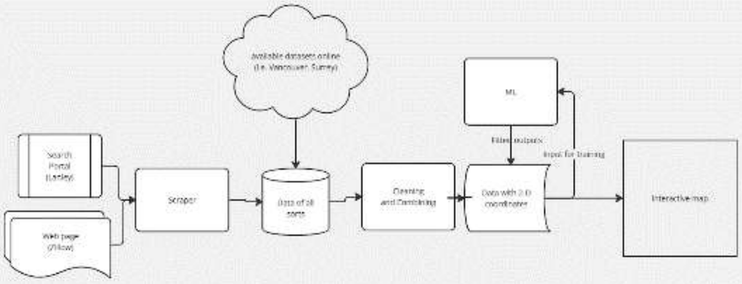

Figure 1: Data Pipeline

1. Data Gathering

Most of our data was sourced from municipal websites, particularly the open data portals for the cities of Vancouver, Surrey and Langley. Most of it was conveniently available as either CSV or JSON files, but the property value data from Langley needed to be web scraped from the map available on the portal and then joined with the postal code and address data that was also on the portal. Additionally, data on bus stop locations was retrieved from UBC's Abacus Data Repository, and the database relating postal code to latitude and longitude coordinates was sourced from the Service Object. Finally, data on rental rates was web scraped from Zillow. Both our immigration and migration data were taken from BC Statistics. Other cities in the Metro Vancouver area were also considered for taking data but were not chosen as they did not have either easily accessible data storage or were missing key values in their data such as location.

2. Data Cleaning and Processing

Most of our data was in the form of CSV files though a handful was in a JSON file form. Both of these files could be readily loaded into dataframes, which was done using either Pandas or PySpark depending on what processing that was needed. Pandas was mostly use for ease of use on smaller datasets as PySpark was used when dealing with the massive amount of property data for each city. Unwanted columns were dropped immediately to reduce the size of the data frames and improve speed when performing operations and calculations on the dataframes. Useful columns often needed to be reformatted using regular expressions to be usable in our code. This was because in most cities data is very messy and they have a lot of inconsistent addressing/location formatting inside a singular dataset. We used these regular expressions to extract the postal codes from these dataset and any other location data that we may need such as latitude and longitude coordinates.

After the initial cleaning and loading, we needed to standardize how we dealt with property locations. First, we aggregated every entry for a given postal code in a single year into a single entry, adding together the total property values and improvement values for that given year, and having each year's data be a separate entry in the row for in the given year. We also counted the number of properties in a singular postal code for that year so we could later get an estimate of how much each property value is worth by dividing the total property value for that postal code by the count of properties in that postal code. It should be noted that we chose to combine the properties because each property in the dataset only had a postal code attached to it and did not have individual property data that we could extract like lot size, housing size, etc.

Beyond this, a significant portion of the data didn't include latitude and longitude coordinates. To convert the postal codes into latitude and longitude coordinates we used the Service Objects database to find the latitude and longitude coordinates based on postal code and then appended that data to each row that required it. Finally, we removed outliers from the low-end of the land values and improvement values. These values that were under the threshold we set of $300,000 were treated as unreliable and set to zero. The zeroed values, approximately 800 rows worth out of 15,000 rows, were then filled in by a simple linear regression model so that we could plot those values for every year on our interactive map.

IV. METHODOLGY

1. Machine Learning Model

Once the data was appropriately cleaned and prepared, we extracted a subset of it to use in a random forest machine learning model to predict future property and improvement values. Because each property had data spanning back through multiple years, we wanted to make a model that could incorporate multiple previous years in order to make predictions, not just the most recent year. To that end, we included LOOKBACK_YEARS as a parameter in the code, determining how many previous years of data would be used in a prediction. The data for all the appropriate years was read into a dataframe; to keep the format consistent from year-to-year, a column was added for "current year," and then the column names for all the other time-dependent data were reformatted to "x years ago." With this done, we incorporated previous years into the feature vector while also mixing data from different starting years into the training data. For example, with LOOKBACK_YEARS set to 10 we could include feature vectors spanning 2015 to 2024, but also those spanning, for example, 2008 to 2017 or 2011 to 2020.

Besides year-by-year property value data we wanted to predict on a number of other data columns that did not depend on the year. To keep this flexible and not have to try to use our entire, massive dataframe at once, we wrote the code to read a list of static parameters out of a text file and to pick out those parameters to predict on.

This flexibility in code design paid off, as it quickly became apparent that some of our initial assumptions about how to build the model weren't fully accurate. Using the random forest feature importance property, we were able to assess the feature importance for our wide array of features. As we'd somewhat expected, almost all of the pertinent information was already encoded in the land value and improvement: the market price of an item should effectively aggregate all the information about factors that influence its desirability (making the other features largely redundant). Somewhat unexpectedly, even most of the price data was unimportant: only the most recent years mattered. They made up roughly 75% and 15% of the feature importance respectively. Realizing that previous years' prices were also rendered largely moot by the most recent data suggested two possibilities for refactoring the model. First a lightweight one that only used the most recent year's values. Second one that didn't include any previous property values at all, allowing us to better estimate which features the market used to determine that price. Fortunately, the previously written code was flexible enough to produce both of these variations with minimal friction merely by setting LOOKBACK_YEARS to 1 or 0 respectively. Unfortunately, looking back exactly one year caused a bug in the future predictions that wasn't present at other values, one we didn't have time to track down and fix. Nevertheless, it was still useful for understanding how the model incorporated the property-value data and how it changed over time.

2. Interactive Map

The back end of the interactive map made use of the Flask, pandas, numpy and matplot libraries. We used Flask to allow the backend to interface with the frontend and matplot to create graphs and visualizations to display alongside the map. We used Numpy and Pandas for scaling and calculations with the data, such as computing summary statistics of property values and making comparisons with region-wide demographic data such as immigration and construction rates.

Our overall dataset was large enough to be unwieldy to load directly into our front end. Instead, as soon as our script starts running, it instantiates a Flask object which loads the dataset into memory and performs our preliminary back-end calculations. Some of these calculations are used to generate our graphs, which are immediately saved as png files which can be displayed by the frontend on-demand. As a preliminary step to drawing the map, our backend divides the data into groups appropriate for any given zoom level of the graph and averages data values over all the members of a group, saving the result in memory. Computing all the levels ahead of time allows the user to zoom in and out of the graph to get a better idea of what is happening without leading to lag which would hurt the user experience. Because of the large size of the whole dataset, it is impractical to load multiple years in memory at once, so when a new year is selected there is a small amount of latency while the data loading and calculations are being re-done for that specific year.

For the front end the only minor calculations that occurred in the JavaScript are calls to leaflet and to our custom map we made using Mapbox. When we call the leaflet objects, we also call a leaflet marker cluster which is how the data points on the map are able to be displayed dynamically by clustering up in groups when you zoom out and unclustered when you zoom in. We were able to get our custom map into this front-end development by linking our mapbox key with the JavaScript to load it when it is called. Another calculation that the frontend development does is detected when to switch years and switch from land value type to improvement value type. Once it detects that a new year or new type has been chosen it will send a command to our Flask object in the backend and load the data points onto the map.

V. EVALUATION

Overall, we were satisfied with how our project turned out, though there are many additional tweaks, tests, refinements and extensions we'd be likely to try if we had more time. In the interactive map we made a useful, accessible and interactive tool to help people understand current and past patterns in Vancouver property data. It's much harder to confidently evaluate the predictions of future property values made by the machine learning model, because they are most likely at least as useful as doing a more traditional statistical extrapolation of property value trends, but it is difficult to guess how much value they add beyond that.

One area where we certainly fell short of our original objective was in dealing with rents. Rental prices are more directly pertinent to many people than raw property values, and we were unfortunately unable to incorporate them into either our map or model. Because of the way we aggregated postal codes, we couldn't make straightforward predictions of rents based on property value because there was no housing details or specific data related to housing details such as, square footage, rooms, bathrooms, etc. Our preferred workaround would be to also train the machine learning model on rent price data and use it to predict rents for the properties, both present and future that we didn't have data for. Unfortunately, we weren't able to find a broad enough sample size of rent price data to have any confidence that this approach would be generalized if we tried it. With more time we could have likely found an acceptable solution, but as things were, we decided we'd be better off leaving out rent prices in the map than to include either a highly incomplete dataset or a set of highly unreliable and low-confidence predictions.

Finally, while we gathered a large amount of data from several communities in Greater Vancouver, we fell significantly short of mapping the whole area. Several fairly large, central communities such as Burnaby, Coquitlam and Richmond are missing entirely, as are many smaller municipalities and outlying areas. Because of the non-standard nature of the datasets available, gathering data for some communities in the area doesn't necessarily speed up the process of gathering data for the others. Not only was the data available in different ways for different cities some cities in the Metro Vancouver area did not have any data available whatsoever publicly.

VI. DATA PRODUCTS

Our main data product is the interactive tool. In this tool the highest level is a home page, which includes links to two different pages, the interactive map and the graph page. In the graph page there is a set of graphs we can display such as the Improvement Value and Construction Value vs. the Year, Land Value vs. Improvement Value, as well as many others. We also have graphs comparing the percent difference per year of these values to see if there is any correlation between them. In the interactive map section of the webpages, we get a map that we can filter based on Property Value and Improvement Value. We can also filter by year going back to 2006 to 2024. The map displays the property/improvement value in a log base 2 format that is normalized based on the minimum and maximum values from every year in our dataset so that we can compare and contrast how prices have increased throughout the years. In addition, the user can select years up through 2034 to see predictions of future property values that were made by our machine learning model.

While our only immediate use for it is to create predictions for the map, the machine learning model is also a data product in its own right. It can be used independently to generate property price predictions for the lower mainland and can be easily modified to incorporate additional data and new features. Unlike the map, nothing about the model is specific to the Vancouver area as it could be used to make predictions for any urban area, provided it was appropriately provisioned with adequately formatted training data to allow it to be retrained to reflect the trends in that specific area.

VII. LESSONS LEARNT

One of the most pointed lessons we learned is how easily it is possible to have not enough data and too much data at the same time. As discussed, we were able to retrieve only a fraction of the data we'd hoped for, and yet even that fraction was at times significantly challenging due to both its size and variety. It was very striking how even the cities in a small geographic area, part of the same country, the same province and operating in the same language could have significant differences in how their data is stored, formatted and made available.

Another useful lesson is to pay attention to how markets inherently deal with data. It's a well-understood principle of economics that prices encode information about market conditions, but it's easy to overlook the ways that this shows up in practice. If we were starting the project over from the beginning, we might well want to rethink our approach to the machine learning model to consider how to best use the predictive power of land prices, rather than starting with an approach that effectively renders much of the data redundant.

VIII. SUMMARY

We gathered property and housing data for multiple municipalities in the Greater Vancouver area, as well as information on nearby geographic features, amenities and demographic trends. We used this data both to create a machine learning model for predicting future housing prices and for an interactive map to allow people to better navigate and understand the ins and outs of the Vancouver housing market.

REFERENCES

[1] Housing Data Vancouver, City of Vancouver Open Data Portal, Retrieved March 1 , 2024 from https://opendata.vancouver.ca/pages/home/

[2] Property Data Surrey, City of Surrey, Retrieved March 11th, 2024 from https://data.surrey.ca

[3] Rental Data, Zillow Inc, Retrieved March 14th, 2024 from https://www.zillow.com/homes/Vancouver,-BC_rb/

[4] Bus Stop Locations, Abacus Library UBC, Retrieved March 20th, 2024 from https://abacus.library.ubc.ca/file.xhtml?persistentId=hdl:11272.1/AB2/LMLPT1/IRPWHQ&version=2.0

[5] Postal Code Conversion, Service Objects, Retrieved March 11th, 2024 from https://www.serviceobjects.com/blog/free-zip-code-and-postal-code-database-with-geocoordinates/

[6] Langley Addresses. Langley Open Data Portal, Retrieved March 11th, 2024 from https://data-langleycity.opendata.arcgis.com/datasets/80dcd2c96b7f4fd8b80546494230fdf4_182/explore?location=49.099919%2C-122.654029%2C13.82

[7] Migration Data, BC Statistics, Retrieved March 15th, 2024 from https://catalogue.data.gov.bc.ca/dataset/inter-provincial-and-international-migration

[8] Immigration Data, BC Statistics, Retrieved March 15th, 2024 from https://catalogue.data.gov.bc.ca/dataset/inter-provincial-and-international-migration/resource/c99d63f6-5ec4-4ac0-9c07-c0352f2f1928