



# Project#1 Extension

BJ Bae & Ben Weiskopf



# Contents

**01**

Introduction

**02**

Naïve Approach

**03**

Takeaways

**04**

Proposed  
Improvements

**05**

Data Collection

**06**

New Model

**07**

Business Impact

**08**

Future Work

---

# Introduction

- Our Company LotwiZe, a competitor of Zillow, is trying to create an Automated Valuation Model for real estate in California
  - The purpose of this extension was to add a new level of complexity to improve our model's predictive power and the business impact from the data
-

# Naïve Approach

## Features

- Lot Size
- Living Area
- Has Garage\*
- Has Attached Property\*
- Has View\*
- Has HOA\*
- Annual HOA Fee
- Full Bathrooms
- Half Bathrooms
- Home Type
- Property Tax Rate
- Year Built
- Has Home Warranty\*
- Latitude
- Longitude

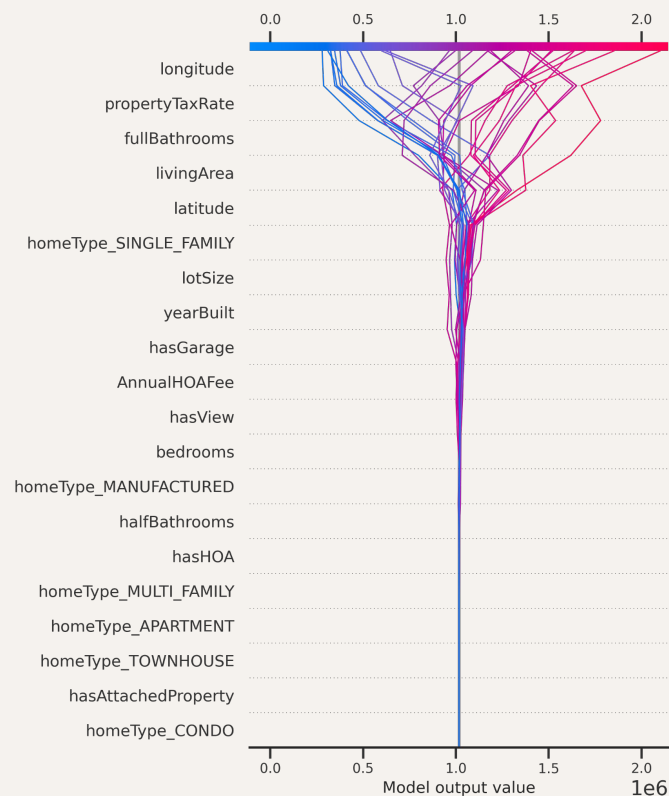
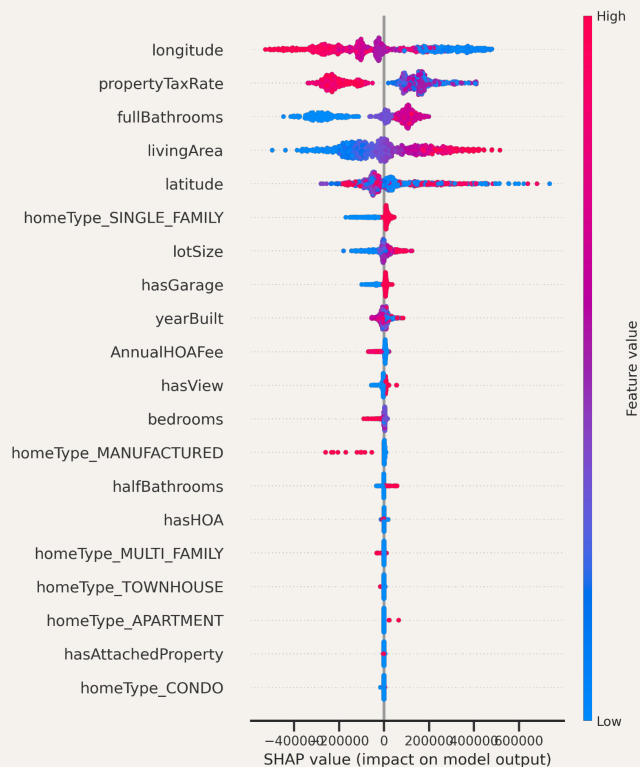
\* Are Boolean Values (True or False)

# Naïve Approach

## Method

- 5263 data points out of 9142 total used
- XGBoost with 1000 estimators
  - Handles NA values well (especially for inference)
  - More powerful than Random Forest
- Metrics
  - $R^2$  Train: 0.928
  - $R^2$  Test: 0.869
  - RMSE Train: \$157,376.15
  - RMSE Test: \$212,353.91

# Model Interpretation

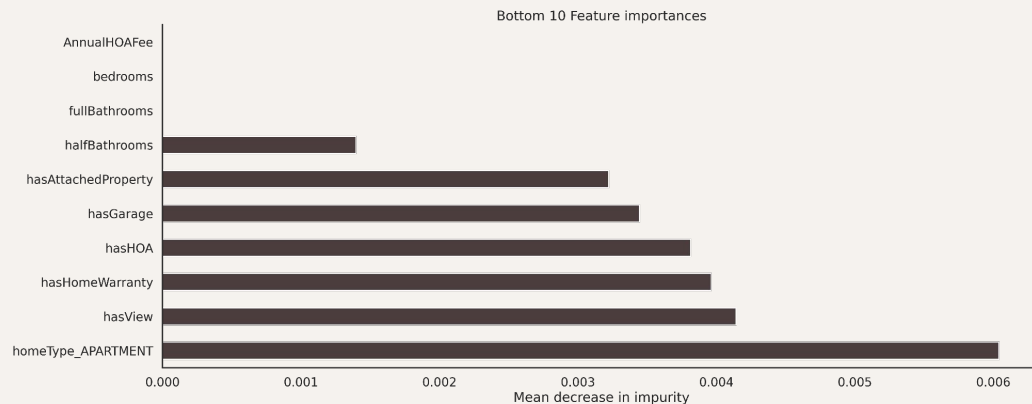
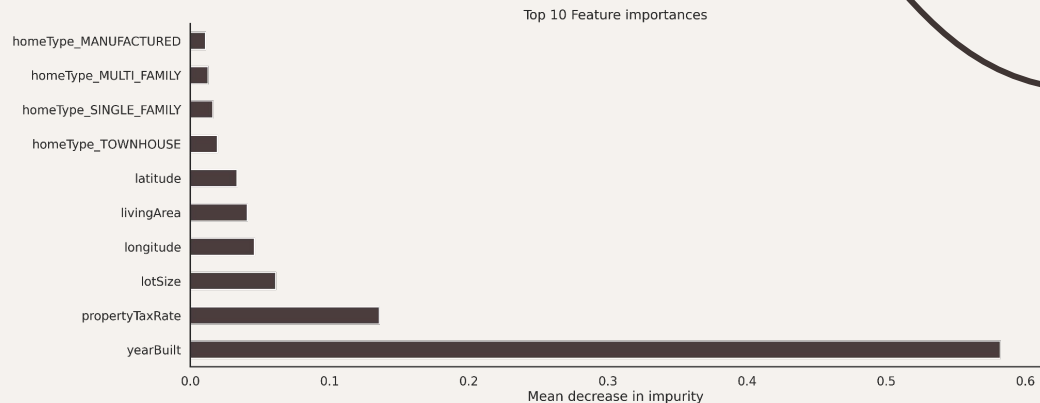


# Some Takeaways

So what is important?

- Age of a house might be consistently important
- Property Tax Rates is a strong indicator and might encode other hidden information about location, population, wealth, etc.
- Number of bedrooms and bathrooms **do not** have a great impact when determining the price of a property

We hypothesis that pricing of homes have to do more with external factors (location) more than internal ones (layout)



# Proposed Improvements

## Web Scraping for City-Level Data

- Elevation
- Average Temperatures
- Rainfall
- Population Density

## Geospatial Data from OpenStreetMaps API

- Nearest Highway
- Nearest Park
- Nearest Grocery Store
- Nearest Costco

- ❑ Adding these features aims to account for factors the original dataset lacked, offering more context around each property's location and improving the predictive power of the model.



---

# Data Collection



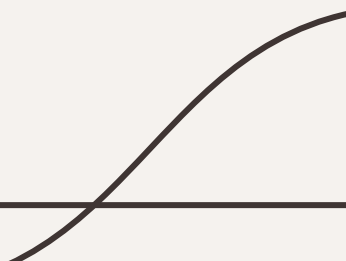
## Web Scraping

Web scrape city statistics  
from Wikipedia



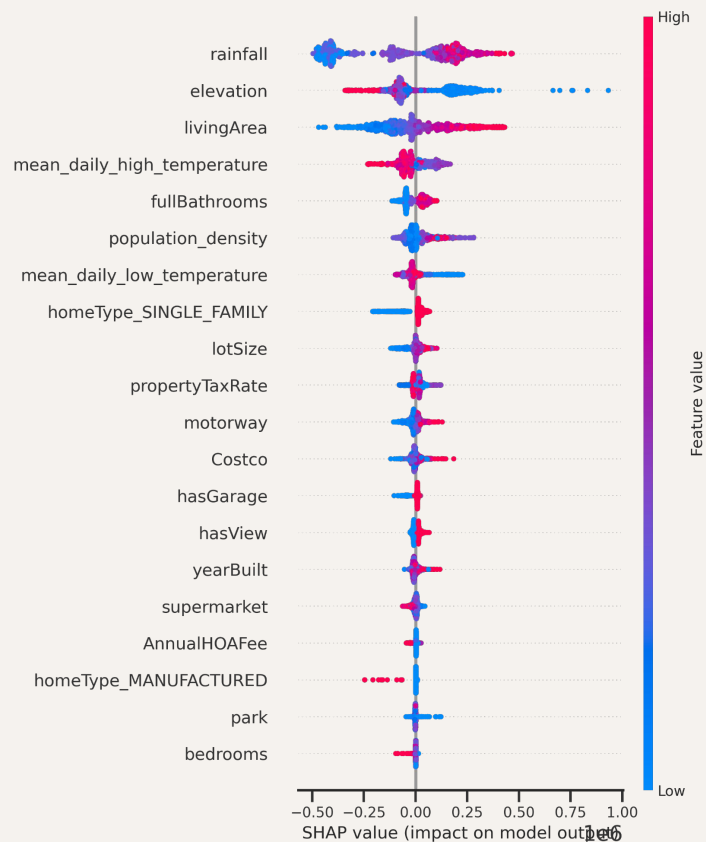
## Geocoded Location

Obtain distances from  
property to different  
places nearby

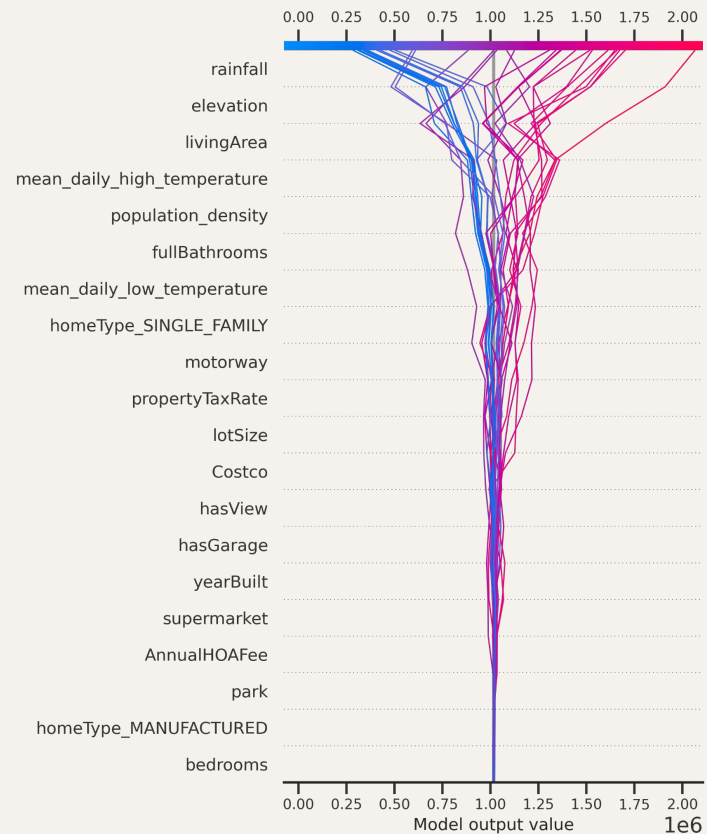
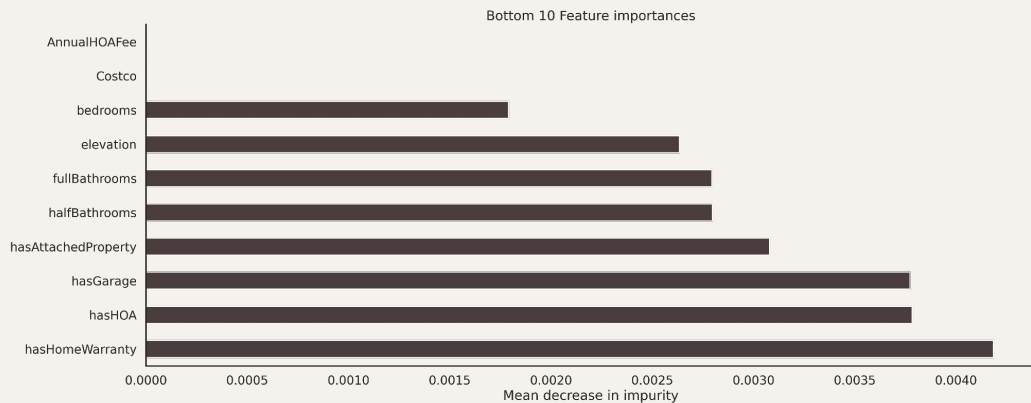
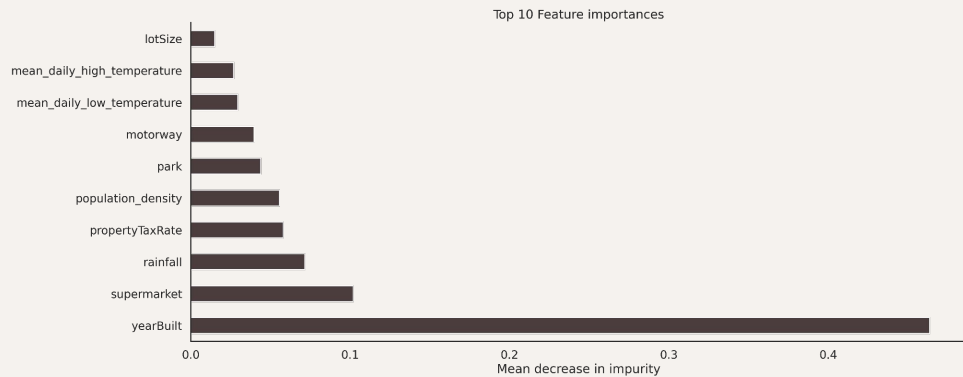
A decorative dark grey curved line starting from the bottom right corner and extending towards the center of the slide.

# New Model

- PCA Metrics
  - $R^2$  Train: 0.907
  - $R^2$  Test: 0.788
  - RMSE Train: \$179,045
  - RMSE Test: \$270,230
- Non-PCA Metrics
  - $R^2$  Train: 0.940
  - $R^2$  Test: 0.882
  - RMSE Train: \$144,353
  - RMSE Test: \$201,778
- New factors added a layer of predictive complexity to the model



# New Model



# Business Impact



## Year Built

Leverage this to target under priced modern & recently remodel/built properties.



## Differentiation

LotwiZe can provide a better prediction and explanations while preserving the nuances compared to than employing black box models.



## Property Tax Rate

High-tax neighborhoods correlate with better public services, giving clients nuanced investment opportunities.



## Robust Inference

By using our model, gaps in data don't render the useful parts of the data useless.

---

# Future Work

01

## Google Maps API

Use Google Maps Routes API to get more accurate and a larger variety of Points of Interests

02

## Robust to Future Trends

Track trends in deployment to make sure model is still relevant

# Appendix

## Source of Data:

- The climate data was scraped from Wikipedia for the cities in the dataset. We retrieved values such as temperature, elevation, population density, and rainfall.
- Data about location features was sourced from OpenStreetMaps APIs

## Unique Features Added:

- 'motorway', 'park', 'supermarket', 'Costco', 'population\_density', 'elevation', 'mean\_daily\_high\_temperature', 'mean\_daily\_low\_temperature', 'rainfall', were collected to improve the model's predictive power by including more complex factors.

## Climate Data Scraping Logic:

- The Python BeautifulSoup library was used to scrape climate data from Wikipedia for each city. The climate data was normalized and aggregated where necessary. Missing values were handled using techniques.

# Appendix

## Geocode Datascraping

- Searches for nearest Points of Interest 1km around the property coordinates.
- If it doesn't find a certain POI, increases search radius by 10km and searches again until 50km.
- This tries to optimize for the number of API calls
  - For example, it tries to only repeat calls for categories that are missing in the increasing radius search and removes calls that already got a valid result from the query
- The distances are measured in meters

# Appendix

Some assumptions made:

- Houses in the same area have the same property tax rates
- lotSize includes livingArea
- Condos and Manufactured homes don't have lotSize outside of livingArea (lotSize == livingArea)
- We don't try to train or predict properties that don't have a building on them (homeType == "Lot") because they are missing too many features about the house
- We impute missing latitude and longitude values from nearby homes
- We fill missing bathroom, bedroom data with 0
- To fill in missing data from climate data that we webscraped, we imputed the values of temperatures, precipitation, and elevation from the nearest cities, since we assumed that nearby cities would have similar attributes.



# Appendix

Data cleaning:

- We impute missing latitude and longitude values from nearby homes
- To fill in missing data from climate data that we webscraped, we imputed the values of temperatures, precipitation, and elevation from the nearest cities, since we assumed that nearby cities would have similar attributes.
- We removed homes that weren't recently sold, because that means the homes do not have a price value associated with them
- We remove properties where yearBuilt is 0
- We remove data where propertyTaxRate is 0

# Appendix

PCA:

- Keeps components with 95% of the variance