Eric Ens

Capstone 1: Zillow Home Price Indicator

Building a predictor model for Zestimate errors.

Introduction

Buying a house is a massive investment and can carry with it a great deal of risk. Having accurate models which forecast home values based on empirical data can help both home buyers and financial institutions make the sound decisions.

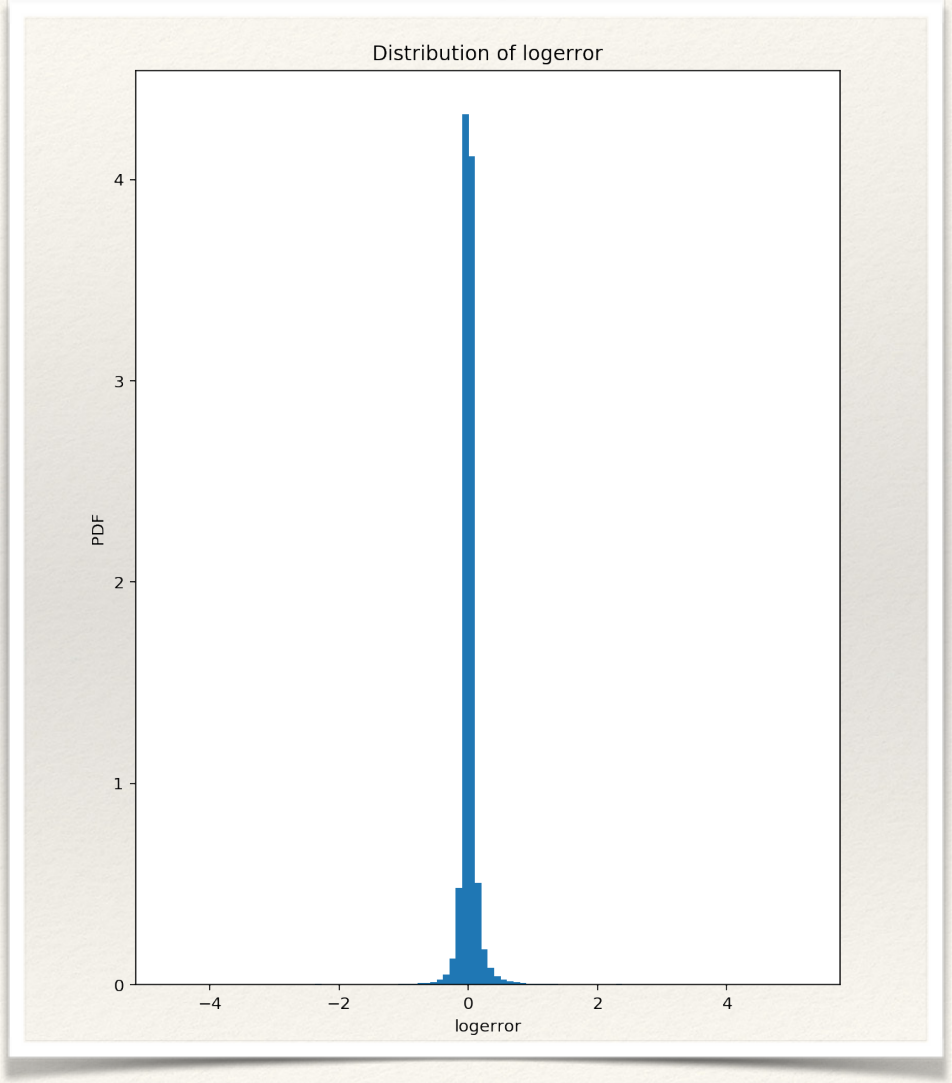
Zillow's Zestimate is an industry leading tool for making these predictions. It is my aim to predict how accurate their model will be using a dataset provided by Zillow on Kaggle.com

The Problem

- The *Zestimate* model is already very accurate, the prediction target has very little variance making prediction difficult.
- Features which are normally useful for predicting prices may not be useful for prediction.

The Data

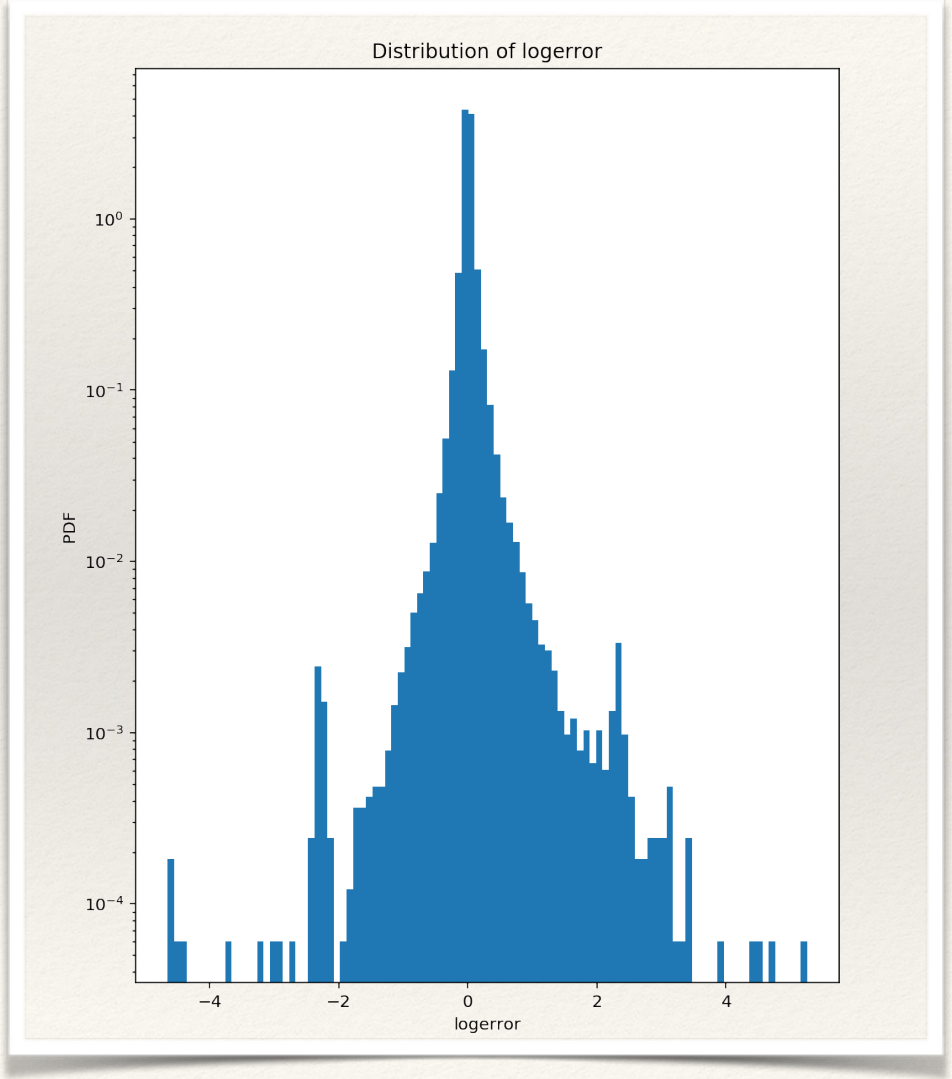
- The value we seek to predict is ‘logerror.’
- We can see that the nearly all values are very close to zero.
- The data has many missing values.



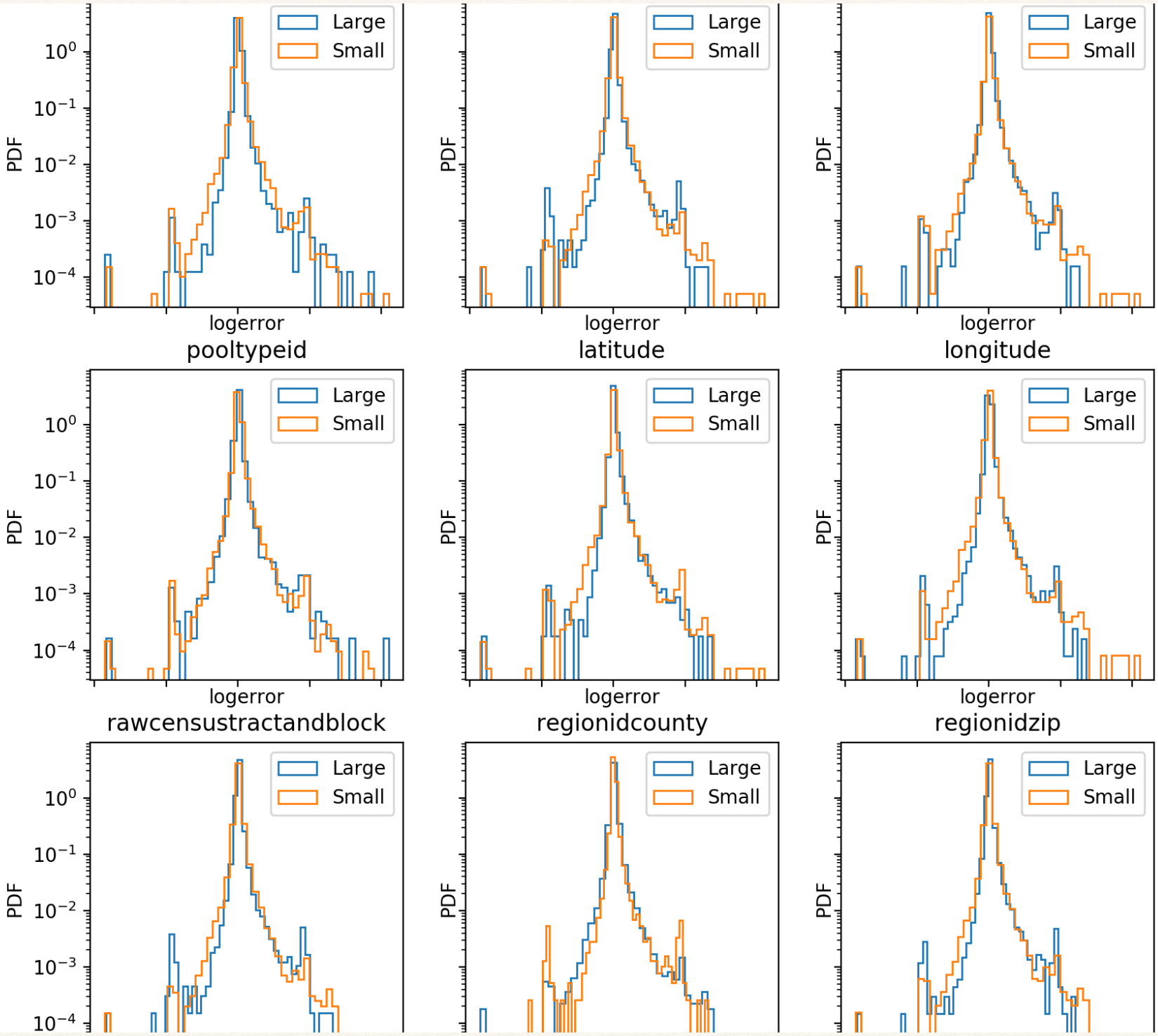
Data Cleaning

- Data Cleaning mainly takes the form of filling missing values
- Easy values to fill are ones where missing values represent a zero or a false value.
- For more difficult ones we used K-nearest-neighbours to guess at values, using latitude and longitude. Assuming that nearby properties have similar features like neighbourhood. This is done in the notebook Capstone_1_data_cleaning.ipynb

Data Analysis



- While the distribution of ‘logerror’ is mostly close to zero, if we look closer (using log scale) we can see some interesting.
- In particular, two clusters around +2 and -2.
- Using boosting algorithms we find a number of features track closely with them


Large
Small
PDF
logerror
pooltypeid
latitude
longitude
rawcensustractandblock
regionidcounty
regionidzip

Machine learning model

- A simple regressor did not work very well since most of the data is symmetric around 0, so a simple linear regression of $y = 0$ would work the best.
- So we built an ensemble model which first used a gradient booster classifier to predict a 'logerror' bin and then a regressor within each bin.
- This also proved difficult. Training a classifier on this dataset was heavily biased towards the bin containing 0. So we needed to oversample the other bins and undersample the 0 bin.
- The model can be found in the notebooks Linear_Regression_pt2.ipynb, Classifier.ipynb and Model.ipynb

Conclusions

- Unfortunately the problem is quite difficult and we were unable to substantially improve on the line $y = 0$.
- While the accuracy of the initial classifier was 91% the regressors on each bin performed poorly