Random Forest and XGBoost determination of Zillow Prize dataset feature importance

Martin Skarzynski

October 25, 2017

1 Introduction

1.1 Zillow Prize

The Zillow Prize is a data science and machine learning competition organized by the Zillow real estate company. The competition is host by Kaggle, a company that provides datasets and computational kernels for data science challenges. Kaggle was purchased by Google earlier this year. The Zillow Prize competition has recently declared to be the second largest data science challenge on Kaggle. The purpose of the Zillow prize is to inspire data scientists around the world to work on improving the accuracy of the Zillow "Zestimate" [1] home price estimate algorithm.

1.2 Zestimate Algorithm

Zillow's Zestimate home valuation was first released in 2006 [2] and has since had a major impact on the United States real estate industry. The Zestimate algorithm created a new standard of providing free, publicly available housing data and home price estimates. The Zestimate algorithm relies on "7.5 million statistical and machine learning models" that have been refined over the years to have a margin of error on only 5%.

The real estate industry is a major contributor to the U.S economy and was worth roughly 30 trillion dollars in 2016. Homeownership equity is a major form of wealth that Americans hold, while mortgages are a major type of American private debt. It is extremely important for U.S. homeowners and for the U.S. economy that home prices are estimated correctly. Without proper estimates, lender and borrowers cannot confidently monitor their assets and liabilities. Overevaluation of real estate assets and the potential returns on real estate debt for lenders has been linked to the great recession. [3]

1.3 My Goal

My goal in working with the Zillow Prize data was to answer the question of which features in the dataset are the most important determinants of the major evaluation metric for the Zillow Prize. This metric is called "logerror" in the data and is defined as the difference between the log of Zestimate price and the log of the actual sales price. To achieve this goal, I cleaned the provided housing valuation data and employed two different machine learning methods, random forest and xgboost, to calculate importance scores for each of the features in the cleaned dataset.

2 Methods

2.1 Data

I obtained the data from Kaggle website. The data consisted of the following files:

- properties_2016.csv.zip
- properties_2017.csv.zip
- sample_submission.csv
- train_2016_v2.csv.zip
- train_2017.csv.zip
- zillow_data_dictionary.xlsx

The `zillow_data_dictionary.xlsx` is a code book that explains the data. The data are available on the Kaggle website, but I also made the data available on figshare where they can be accessed without the need to create/enter a username and password. The data can also be accessed as part of my Zillow Kaggle kernel.

2.2 Analysis

Data analysis was done in Jupyter Notebook (formerly known as IPython Notebook) [4] Integrated Development Environment using the Python language [5] and a number of software packages. I used NumPy [6] and Pandas [7] for data wrangling. To calculate the importance scores, I used the Scikit-learn [8] and XGBoost [9] machine learning libraries. Finally, I visualized the data with the Matplotlib [10] and Seaborn [11] libraries.

2.3 Reproducibility

Reproducibility is extremely important in scientific research yet many examples of problematic studies exist in the literature [12]. To make the analysis reproducible for users of the Anaconda distribution of Python. The names and versions of each package used herein are listed in the accompanying `env.yml` file in the `config` folder. The computational environment used to analyze the data can be recreated using this `env.yml` file and the `conda` package and environment manager. Additionally, details on how to setup a Docker image capable of running the analysis is included in the `README.md` file in the `config` folder. The code in the form of a jupyter notebook can also be run on the Kaggle website (this requires logging in with a username and password) by accessing the Zillow Kaggle kernel I created. More information on the details of how this project was created and the computational environment was configured can be found in the accompanying `README.md` file.

3 Results

3.1 Missing values

The dataset provided for the Zillow challenge consisted of 58 features. I first determined the percent representation of missing for each of these features (Supplemental Figure S1). I then used median imputation to fill in the missing values for the all of the features. I decided not to remove any features based on the percent missing values, because I was more concerned with the feature importance calculation.

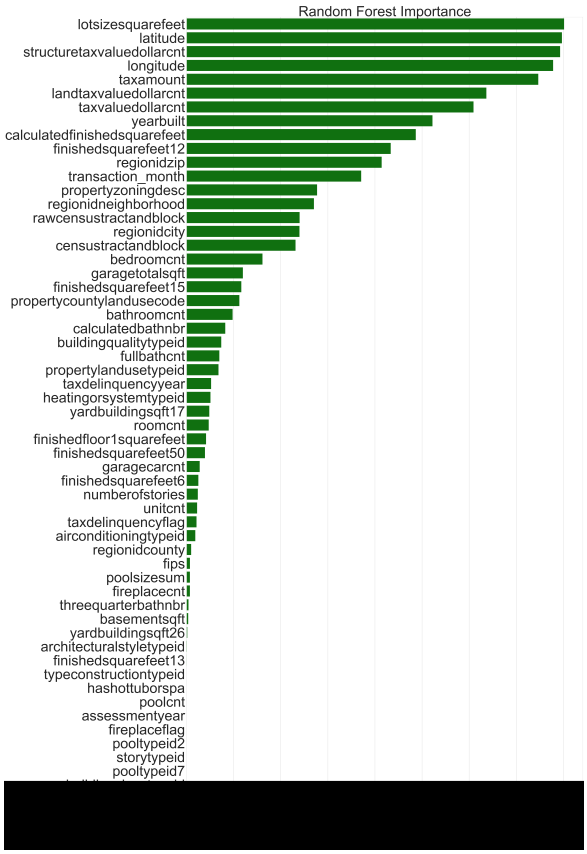

Figure 1. Random Forest Importance Scores. The Random Forest method was used to obtain importance scores for each of the features in the Zillow dataset. The features were ranked by Random Forest importance and then plotted with their corresponding importance score. The Random Forest method calculates importance in terms of the proportion of total variance explained by each feature.

3.2 Feature Importance

Next, I used the Random Forest and XGBoost methods to obtain importance scores for each of the features in the dataset. When comparing the feature importance scores calculated by Random Forest (Figure 1) and XGBoost (Figure 2) methods, it is important to keep in mind that these scores use different units. The Random Forest method calculates importance in terms of the proportion of variance explained and produces values between 0 and 1. In contrast, the XGBoost method calculates importance as an F score.

I made a final list of features by removing any features that had an importance of less than 0.001 when obtained from the Random Forest model or an XGBoost F score importance of less than 10. Of the initial 58 features, 39 features remained in final feature list.

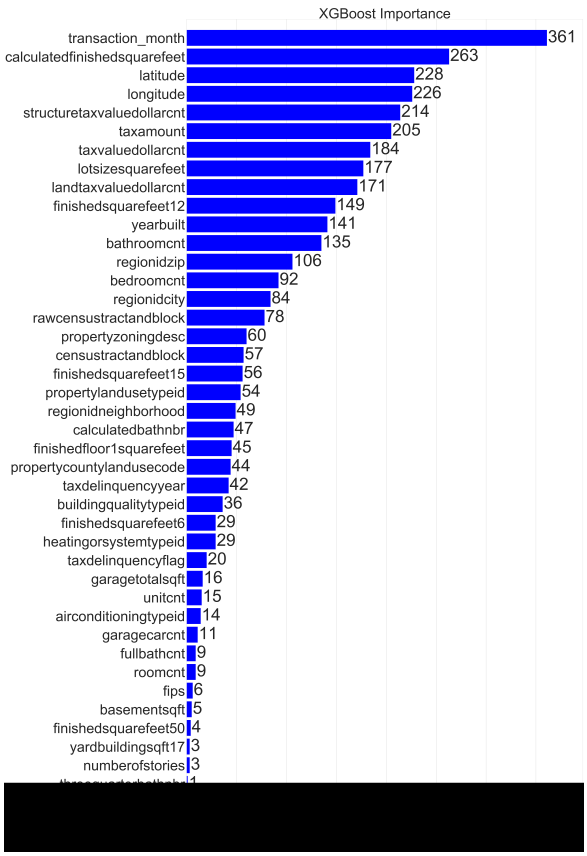

Figure 2. XGBoost Importance Scores. The XGBoost method was used to obtain importance scores for each of the features in the Zillow dataset. The features were ranked by XGBoost importance and plotted with their corresponding XGBoost score. The XGBoost method calculates importance as an F score.

4 Conclusions

4.1 Limitations

For my first Kaggle competition, I was only interested in how to determine which features would be most likely to be useful in making a prediction. One major limitation of the analysis I did was that I did not validate my calculated importance scores using data set aside for this purpose. Another major limitation was that I did not show that a model with a smaller set of high-importance features could perform at a similar level as a model with all of the features.

4.2 Lessons Learned

In addition to learning, how to calculate importance scores using the Random Forest and XGBoost methods. I learned a great deal about Kaggle competition while working on this project. For example, I learned how to use the Kaggle Jupyter Notebook interface and make and publish my own Kaggle kernel. This knowledge will allow me to compete in future Kaggle challenges and share my code in the form of Kaggle kernels. I also learned how to use Docker images for reproducibility, which is a useful means of recreating the environment in which a data analysis was completed. Most importantly, I learned how to create a reproducible report that contains citations, links, plots, code and text in a single Jupyter Notebook source file that can generate various types of output files including a LaTeX formatted PDF.

References

[1] Diane Tuman. *What is a Zestimate?* Apr, 2013.

[2] James R Hagerty. How good are zillows estimates? *Wall Street Journal*, 2007.

[3] Sher Verick and Iyanatul Islam. The great recession of 2008-2009: causes, consequences and policy responses. 2010.

[4] F. Pérez and B. E. Granger. Ipython: a system for interactive scientific computing. *Computing in Science & Engineering*, 9:2129, 2007.

[5] F. Pérez, B. E. Granger, and J. D. Hunter. Python: an ecosystem for scientific computing. *Computing in Science & Engineering*, 13:1321, 2011.

[6] S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13:2230, 2011.

[7] W. McKinney. Data structures for statistical computing in python, 2010.

[8] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2014.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system, 2016.

[10] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9:9095, 2007.

[11] J VanderPlas. Data visualization with seaborn. *O'Reilly Media. Retrieved April*, 27:2016, 2015.

[12] J. Couzin-Frankel. Cancer research. as questions grow, duke halts trials, launches investigation. *Science*, 329:6145, 2010.