Capstone Project -- Real Estate Valuation Predictions

Peter Grabowski, Hoang Phan, Anthony Ramirez, Frederic Suares

I. Introduction: Motivation, Background and Why This Matters

The world of real estate transactions is inherently uncertain for all parties involved -- in particular for buyers, sellers, and agents (not to mention lenders, appraisers, and title insurance agents). The uncertainty causes stress in the process of purchasing real estate, including, but not limited to: stress for sellers in choosing an appropriate sale price that maximizes potential value while minimizing time on the market, stress for buyers on what a true “value” is, and stress for real estate agents to provide helpful guidance to clients on how to price and bid on properties. While investments broadly speaking are inherently risky -- just ask Burton Malkiel or consult *A Random Walk Down Wall Street* -- anything that would ease the uncertainty in the transaction would be helpful for all parties involved. This intuition was confirmed after speaking with a number of people who have recently been involved in a real estate transaction, including both real estate agents and buyers.

Many attempts have already been made to reduce uncertainty in real estate transactions, particularly in assigning an accurate assessment to a property. The most notable come from both the biggest players in the online real estate market (e.g., Zillow, Redfin, et al.) and from academics in a variety of cities. That said, since a lot of the existing literature is either proprietary or performance-focused, there is still an opportunity to mine for additional insight, particularly into the following questions:

1. What ultimately matters when it comes to real estate valuation, weighing all factors against one another?
2. To what extent can accuracy of predicting real estate value be improved with data not available to the big online real estate players? What could the potential be for improved accuracy if this data were crowdsourced?
3. What would the impact be of applying additional engineering be on improving performance?
4. To what extent is it possible to predict time to sale?

The purpose of this analysis is to apply a variety of machine learning techniques to shed light on the above research questions using a proprietary data set covering three years of real estate transactions in Travis County, Texas (which contains Austin). This brief paper will briefly cover the background on how this problem has been approached in the past, walk through the data set and machine learning / engineering methodologies employed, present results relative to results others have achieved, and discuss how these results could be applied (e.g., building an application, selling to real estate hedge funds and other investment firms).

II. Background

A well-known and well-publicized attempt to improve the accuracy of real estate valuation was done via a Kaggle competition hosted by Zillow (https://www.kaggle.com/c/zillow-prize-1), which was organized around improving Zillow's residual error before taking a ground-up approach to building one's own algorithm (the winning entry improved on Zillow's original estimates through a combination of ensemble models and advanced feature engineering; publicly available write-ups on some teams' solutions include mentions of CatBoost and LightGBM). Zillow has claimed 5% median margin of error in estimating real estate prices nationally, though there is significant variation from geography to geography, since the data that are publicly available vary significantly from state to state.

In fact, and relevant to our analysis on Travis County data, Texas is one of fewer than ten states in the nation where the sale price of a house is not required to be reported to the state. This limited access to data creates numerous hurdles when attempting to accurately model the fair price of a given home. Indeed, Zillow's own Zestimate, arguably the most well known of the publicly available Automated Valuation Models (AVMs) reports that it is unable to compute[1] values for homes in Travis County (including the city of Austin). However, this same data disparity creates a wealth of opportunities for those who have access to the data, especially in a real estate market as dynamic as the city of Austin[2]. This is relevant since it lends credence to the idea that additional / crowdsourced data may have a leg up on what ZEstimates are able to offer.

Outside of Zillow, there are a number of other interesting papers on what has been tried in the past. Their approaches included:

1. Use an ensemble of models to maximize prediction performance, rather than rely on one model
2. Incorporate disparate sources of data (e.g., economic indicators, local pollution trends, etc.) to improve performance
3. Try a variety of approaches to deriving important geographic features

For more information, please see the following papers:

- Antipov, E. A., & Pokryshevskaya, E. B. (2012). Mass appraisal of residential apartments: An application of random forest for valuation and a cart-based approach for model diagnostics. Expert Systems with Applications, 39(2), 1772–1778.
- Matysiak, G. A. (2017, May). The Accuracy of Automated Valuation Models (AVMs) (Tech. Rep.). TEGoVA: The European Group of Valuers' Associations.
- Zurada, J., Levitan, A., & Guan, J. (2011). A comparison of regression and artificial intelligence methods in a mass appraisal context. Journal of Real Estate Research, 33(3), 349–387.

[1] See https://www.zillow.com/howto/DataCoverageZestimateAccuracyTX.htm

[2] See https://www.bizjournals.com/austin/news/2019/01/22/2018-set-another-record-for-home-sales-but-rising.html

This analysis has taken these lessons to heart in crafting an approach to estimating real estate values.

III. Data and Methods

In the end, this analysis rests on the comparison of a number of progressively more advanced machine learning techniques to predict sale price and time to sell after crafting a novel approach to neighborhood classification and incorporating rental information from Airbnb to provide further orthogonal insight on housing behavior. This section will cover the specifics of how the problem was formed, a brief overview of the data used in this analysis, and a survey of the methods covered in the feature engineering and regression models.

A. Problem Statement and Data Available

At the core of this analysis are two regression models, for a given residential property: 1) predict the sale price of a property if it were to be sold, and 2) predict the time to sale of a property based on its list price.

As our first task, we collected datasets from three distinct entities with vested real estate interests in Austin, including Zillow, the Travis County Appraisal District (TCAD), and the Austin Board of Realtors (ABOR). After months of extensive research and persistent follow ups, we were able to obtain a core dataset from each of the parties above. Work then began on data extraction, transformation, and loading (ETL), including a reconciliation layer that merged the disparate datasets by determining likely matches for homes based on address. This reconciled, merged, dataset formed the substrate for all future analyses.

Given the objective and the dataset, it is important to call out a number of key assumptions or limitations of this analysis:

1) **Population representation:** Properties that were actually sold in the last 3 years are used to extrapolate on all properties that could be sold in the future; one could argue that samples sold in the future comprise a sample from a different hypothetical population, and therefore not represented in the training sample for this analysis
2) **Market trends:** Overall market movements are captured only over the three years that are covered in the training dataset; major market disruptions such as recessions are not represented in the dataset, so using this model during a recession may lead to increased errors
3) **Data accuracy:** It likely goes without saying, but this analysis is reliant on the accuracy of the underlying data; while an exploratory analysis was completed to sense check the data, there is still a possibility for error that is not accounted for in this analysis

B. Exploratory Data Analysis

An exploratory analysis of the data yielded a number of insights, many of which resulted in changes we made to the dataset

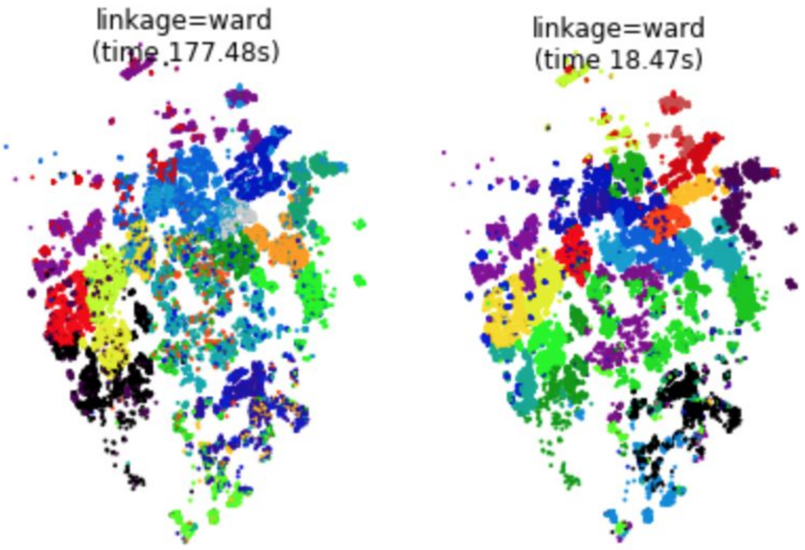
1. Built a “built %” variable, which is the construction size divided by lot size, which proxies which properties are condos versus on large plots of land
2. Used log(sales) due to the shape of the sales dependent variable (right-tailed)
3. There is clear seasonality at play, in addition to a general rising trend in prices year over year, which influenced how train/dev/test split was handled (more on this in the results section)
4. Sale price is almost always less than the list price, which suggests properties aren’t priced to inspire “bidding wars”; there is also a negative correlation between time to sale and the difference between sale and list prices, which is not surprising
5. Did not use a number of variables in the original merged dataset due to sparsity issues (e.g., number of fireplaces)

C. Feature Engineering and Models Used

Beyond the typical data transformations to process the datasets mentioned above into usable, two significant data engineering tasks were done through great trial and error: 1) construct “neighborhoods” from the ground up using unsupervised methodologies and compare how these match up to already-defined neighborhood boundaries, for use in both models, and 2) add in Airbnb rentals data to give additional insight.

For the neighborhood clustering, we used a combination of unsupervised learning techniques. First, we created a graph of which homes were geographically close to one another by running KNN on just the latitude and longitude values for the homes. This preserved geospatial locality, allowing us to mathematically represent the notion that houses in the same neighborhood should all be physically close to one another. After we had this connectivity matrix, we used that as the template for an agglomerative clustering model. More specifically, we restricted the agglomerative clustering model such that clusters could only expand along bounds defined by the KNN home graph. Agglomerative clustering is a “bottom up” clustering approach similar to many other clustering algorithms, where each home starts in it’s own cluster (neighborhood) and clusters of similar homes are successively merged. This structure adapts nicely to geospatial requirements, making it appropriate for use in real estate analysis.

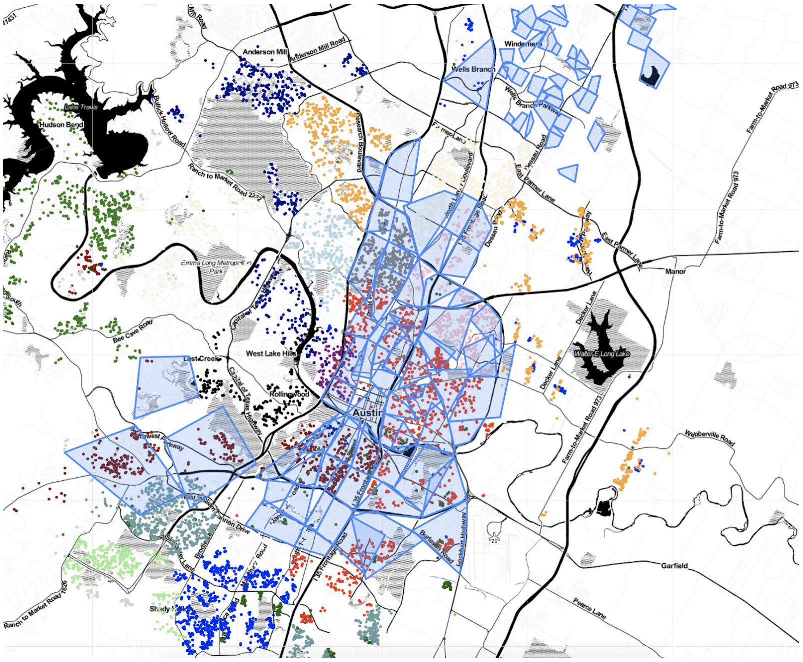
Figure 1: Two Variations on Geographic Clustering in Austin



A model without spatial locality restrictions on the left, and with spatial locality on the right. Note the mottled clusters on the left

Encouragingly, the emergent clusters we observed resembled well known neighborhoods in Austin. Our next step was to plot the models on a true map of Austin, so we could confirm that the emergent clusters resembled CoA's static neighborhoods.. Thankfully, the neighborhoods aligned as expected.

Figure 2: an overlay of Austin neighborhoods with a clustering output



From there, the next step was rigorously evaluating how these neighborhoods lined up with how others viewed them. In order to do so, we obtained a set of GeoJSON files representing the neighborhoods as viewed by the City of Austin (CoA) for planning purposes. We were then able to match each home in Austin both to a neighborhood as defined by CoA, as well as to a neighborhood detected by our emergent model. In this case, the intuition is that neighborhood boundaries may have shifted over time, and that there's extra predictive power to be realized by have more current neighborhood estimates. For each emergent neighborhood we identified, we looked at the most common CoA neighborhood label. If there was high agreement, we marked the neighborhood and the cluster as a match. In either case, both labels were made available downstream for further modeling efforts. Finally, we made the GeoJSON labelling code extensible, allowing us to work with all GeoJSON data sources, greatly expanding the number of data sources we were able to integrate with.

For the Airbnb data, we pulled from a website which already scrapes AirBnB data on a monthly basis (Insideairbnb.com). Monthly median revenue by zip code was extracted, as was the TF-IDF of the top 25 transportation terms (from the 'transportation' section of individual listings) on a per zip code basis. The sum of these TF-IDFs was calculated for all the listings in a zip code, which resulted in an additional feature column for our dataset, which we joined on zip code. We avoided adding the monthly median revenue in the end since we were worried about the time period that this dataset was recorded in being different from the real estate dataset.

Finally, for model choice, we used a variety of standard models to regress on (log) Sale Price and (log) Time to Sale. Based on our EDA, we found that using the log scale helped mitigate the right skewed distribution of the data. For both dependent variables, we started with standard Linear Regression and expanded to regularized Ridge and Lasso Regression. We also explored tree-based models (RandomForest, XGBoost) and Dense Neural Networks, but finally found that XGBoost model performed the best for both tasks.

We applied standard pre-processing for all our baseline models (scaling for numeric, one-hot encoding for categorical). Initially we split our training, validation, and test data randomly (which yielded very high accuracy), but eventually we updated our split criteria to Sold Date to ensure future data was not leaking into our training set. This, unsurprisingly, reduced our accuracy for both targets, but is still remained competitive to other sources given the lack of data available for our chosen geography.

To tune our models, we started with using Grid Search but eventually converted to Bayesian Optimization due to the efficiency gains. We optimized our parameters for both XGBoost models in this way, and ended up selecting the best parameters for our final models.

We decided to use the Median Absolute Percentage Error (MdAPE) to compare models because:

1. The metric is commonly used by Zillow and Redfin to assess their estimation errors

2. As a Median percentage, the metric is not as influenced by large outliers (compared to Mean percentage)
3. Using a percentage (as opposed to RMSE) helps readers more easily assess the magnitude of the error relative to the value predicted

Since we regressed on a log-scaled variable, we exponentiated the predictions and ran the error metric on the original (non-logged) units in order to produce a more interpretable error metric.

Ultimately, our regression on Sales Price was significantly more accurate than our regression on Days to Sale, so we will focus our results and error analysis on the former.

IV. Results

Regression on Sale Price

Overall, the regression model for the value prediction task performed up to par with many of the top models in the research. A few key insights from the modeling:

- XGBoost performed best across all of our modeling tasks, with RandomForest and Dense Neural Network relatively close
- Model performance differs significantly in how one performs the train/dev/test split
  - Random split performs the best, though this unfairly uses “future” properties to predict “past properties”; splitting by date makes more sense to avoid this issue
  - By splitting our train/test set using dates, our MdAPE increased by 3%, which was expected but not as severe as we initially thought
- Our EDA suggested that properties sold before 2017 may be inherently different than those sold after, and we saw this manifest in the increase in MdAPE when using older property data in the model
- The feature engineering yielded little to no benefit (increasing MdAPE in some cases)

Regression Results - Sale Price Model

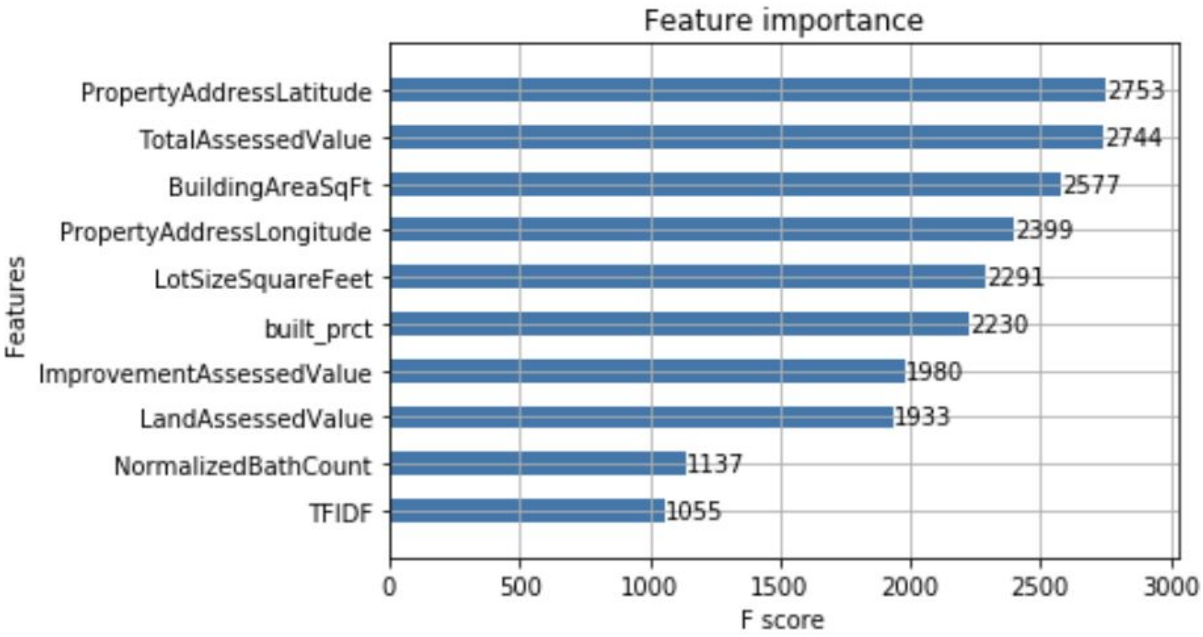
	Train (1yr) - Test (1yr)		Train (2yr) - Test (1yr)		Random Split	
Model	**Adj R^2**	**MdAPE**	**Adj R^2**	**MdAPE**	**Adj R^2**	**MdAPE**
XGBoost Baseline	0.901	8.209	0.854	11.538	0.942	6.066
XGBoost with TF-IDF	0.900	8.345	0.854	11.508	0.943	5.888
XGBoost with TF-IDF Clustering	0.901	8.247	0.862	11.104	0.943	5.938

As mentioned in previous sections, Zillow and Redfin's national average MdAPE is around 5%, but the estimates Zillow has are north of 10% for Austin. The results we achieved far exceed other estimates available for the Austin area, and are in line with leading models given the difficulty of acquiring data and the dynamic nature of the Austin real estate market.

The added features we engineered did not seem to have a strong impact on the model, but this is just one application of the latent neighborhood feature we produced. Similarly, the Airbnb data could also be implemented in future models for different prediction tasks.

Regression - Most Relevant Features

Below are the features that were most relevant for our full XGBoost model (by weight i.e. the number of times a feature appears in a tree).



In reviewing the most relevant features we see that our engineered features: built_prct and TF-IDF, were part of the top 10 most important features. The others were key elements one would expect in a model predicting Sales Price.

Regression Error Analysis

While the models performed relatively well regardless of the train/dev/test split, a few patterns emerged when probing where errors occurred:

- Errors tended to be above the MdAPE for higher-priced properties when compared to lower-priced properties -- likely due to the right-tailed skew of the data (the variation is significantly larger for higher-priced houses)
- Errors tended to be above the MdAPE for properties with fewer bedrooms and bathrooms when compared to properties with more bedrooms and bathrooms -- one possible reason is that, given the construction boom in condos in Austin, there is a sales premium for these properties that is not being fully picked up despite the built % and year built variables
- There is wide variation geographically on where errors cluster, with fewer errors in Northwest and West Austin, and more errors in North and East Austin

Regression on Time to Sale

We were not able to produce competitive predictive models for ‘Time to Sale’, mainly due to what we believe was poor data fit to the task. Our best model (XGBoost) resulted in a test MdAPE of 46%, which is far higher than our regression models. Similar to our regression models, we did not find a significant improvement in MdAPE with the inclusion of our Airbnb or

Cluster features. In order to improve these results to make them more usable in the actual market, we recommend adding additional features beyond those included in our research.

V. Conclusion

In the end, in keeping with the axiom ‘more data trumps smarter models’, having access to superior data was much more instrumental to approaching Zillow’s results than the feature engineering completed above. That said, there are a number of additional directions that could be taken to further improve performance of the models, including:

- Try neural nets (either CNN or RNN, or even fully connected) with trained embeddings for geographies (either satellite photographs or even Google Maps) that are fed forward to estimate sale price
- Devise new ways to use the Airbnb data (potentially revenue) beyond the TF-IDF analysis on the transportation section
- Look for additional datasets (including local economic indicators) to better control for market movements (in addition to tweaking the modeling approach)

That said, there is an important takeaway from the triumph of ‘more data’ over ‘smarter models’: if someone were able to source local data that were unavailable to Zillow or other national real estate estimators, then they could likely build superior models in certain local geographies. This would have helpful applications either if 1) Zillow or a firm like Zillow were to invest in the apparatus to source this data, or 2) one were to build a competitor that competed with Zillow and others in local markets where their data and estimation abilities were superior. One could even imagine an institutional investor (e.g., hedge funds, REITs) investing in data collection to outperform publicly available valuations.

So in the end -- it may be better to invest in better data than overinvesting in marginal benefits from smarter models.