MGT 6203 Group Project Proposal, Team 48

David Firrinicieli, Eric Limeback, Ashwin Spencer, Andrew Taylor

PROBLEM OVERVIEW

Context: Predicting house prices using information about the house (number of bedrooms, number of bathrooms, etc.) is a common machine learning problem. However, in many house price prediction datasets, additional information about the economic status of potential buyers is not included. When our team came across a publicly available dataset of income tax return data by zip code, we started to wonder – how might knowing various tax return fields by zip code, such as adjusted gross income, help predict the prices of homes for a given time period?

Problem Statement: When homeowners list their house for sale, it can be very difficult to determine a fair valuation that might sell well in the market. No two houses are alike, and the value of the home is derived from a large combination of features like square footage, construction material, number of bedrooms, number of bathrooms, and much more. More often than not, sellers will look at comparable units in the neighborhood first to get a baseline price, which reaffirms just how difficult it can be to determine a listing price based on an isolated review of the house features. We aim to improve this process by researching the extent to which additional zip code features might improve price prediction models.

Primary Research Question: How well can we predict house prices using **both** standard "listing" information about the house and income tax information about the residents of a zip code?

Supporting Research Questions:

1. How does the performance of models with income tax data added compare to those with only house features?
2. Which income tax features are statistically significant? What is their relationship to house price?
3. Which traditional features are statistically significant? What is their relationship to house price? How does the set of significant features change when income tax features are added?

Motivation and Novelty: This would be important to any company or individual who wishes to get an estimate of a home's price, without knowing what it would be listed at. This could then be used to help evaluate potential home purchases, understand the value of current assets, and inform a variety of other residential real estate investment-related activities. If I was a potential investor, I could use this to compare expected prices to actual and look for opportunities. This could also improve products which evaluate home prices, like Zestimates on Zillow.

Literary Survey: After surveying two literary sources, we found that other attempts have been made to both predict a listing price and predict a price using spatial data (e.g. accounting for submarket), but neither source leverages income tax data to account for spatial affluency.

1. In our first source [1], a model is made for Bengaluru housing that predicts price from a number of features, primarily listing-specific. It includes the type of area the house is in, the availability, price, size, society, total_sqft, bath, balcony, and location. In this page, the location characteristics such as affluency are not encoded, just the location itself.
2. In our second source [2], a number of approaches are tested using data from Louisville, KY, including using a dummy variable to represent the zip code, two features to represent distance from a central point in Louisville, and separate models for different submarkets. Again, location characteristics such as affluency are not encoded, just the location itself.

Data Sources:

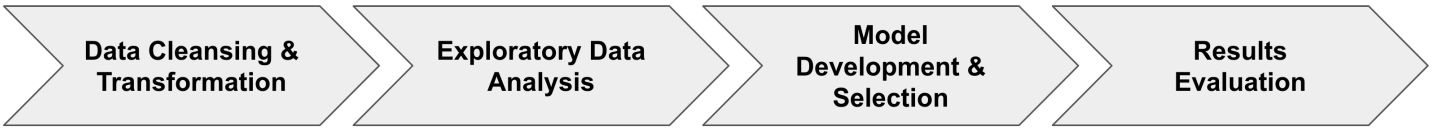
1. USA Real Estate Dataset – link – This is a listing and price-level dataset scraped from realtor.com, where each record represents a specific house listing (address), with multiple records if a listing's price changed. We'll use this for our dependent variable (the price) as well as the 'traditional' house listing features.
2. Individual Income Tax Statistics – link – This is a zip code and AGI-bucket-level dataset generated by the Internal Revenue Service (IRS), which includes summaries of every major field on individual federal income tax returns. This includes 128 fields, which we've narrowed down to those which are relevant (listed in the next section – screenshot

doesn't fit all fields). Relevant fields may then be further narrowed down using feature selection methods later in the project.

PLANNED APPROACH

Planned Approach

We plan to break this problem into four main steps, which may be iterative depending on our findings.



In the data cleansing and transformation step, our objective is to create the dataset which we'll study in exploratory data analysis and ultimately feed to the model. The activities in this step include performing null and duplicate checks, confirming data types, aggregating the daily datasets to an annual level, joining the datasets and checking join veracity, and creating calculated fields for our variables as needed.

Next, in the exploratory data analysis step, our objective is to summarize the data in a way that will help us understand anything we need to know before modeling. The activities in this step include univariate summaries (boxplots), dependent-independent variable summaries (pair plots, correlation heatmaps), and any other simple summaries to get to know the data better.

In the model development and selection step, our objective is to test various model types and select the model which provides the best results. We will likely test model types whose coefficients can be interpreted (linear regression, LASSO regression) as well as opaque models (Random Forest, Boosting). Activities here include splitting data into training/testing, scaling data as needed, performing PCA if needed for dimensionality reduction, feature selection, training and scoring models from different families, selecting a 'best' model type, and fitting that model on the final data. Specifically, we'll likely pick regression-appropriate error metrics for comparison (mean squared error, R-squared), and we'll use grid search and cross-validation to train and tune the hyperparameters for our models. We may also investigate interaction terms.

Finally, in the results evaluation step, our goal is to interpret the model to answer our research questions. The main activity will be evaluating model performance metrics to validate results and understand the predictive power of our model. In addition, it's during this step that we'll run a version of the model without the income tax features to understand the value that those features add. We expect that all analysis will be done in R Markdown files.

PROGRESS TO DATE

Data Cleansing & Transformation

To this point, we have completed data extraction, cleaning, and merging, although discoveries during model development may cause us to revisit this step (e.g. creating interaction terms).

During data extraction, we downloaded the two datasets from Kaggle, selected features from the income tax dataset, and aligned on the timeframe to concentrate on. The real estate dataset contained data from 1950 to 2014, inclusive, but we elected to concentrate on 2010-2014 as it provided a good enough sample size for our initial analysis, and was focused on the most recent years. We chose 4 years' worth to ensure we had adequate representation per zip code (5/zip code). We also discovered that the data is concentrated in the Dominican Republic and the Northeast (add picture). Within the income tax dataset, we hand-selected 15 features from the 128 available to test as features in our model, based on the features which were most contextually relevant to the problem.

Next, we cleaned the data, which meant addressing a few items. First, we saw that there were duplicates in the real estate data resulting in multiple listings per house. To account for this, we took the first record for each, assuming that was the initial listing. This was the best assumption we could make, since further context and data fields were not available to do root cause analysis. Next, we converted the zip codes from integer to string, and padded '0's on the left, as many Northeastern zip codes have leading zeros. We noticed some of the ZIP code data was tied to ZIP codes of

"00000" and "99999", which would not match to any of the real estate listings. We filtered these out as part of the cleaning process as well. Finally, we had to make some decisions about NULL values in our features and some missing records from the resulting join. We identified about 20 records for real estate listings with a ZIP code that was not included in the income tax data set. This meant that all of the relevant income tax statistics was missing for that listing as we used a left join to preserve all rows in the real estate data. The income tax data was a key piece of our analysis and model, so we decided to filter out these records. We were still left with null values in the real estate data, predominantly in the acre_lot feature. From further investigation, we suspected that this was not due to poor data collection but rather the type of listing; there appears to be a strong connection between the "NA" values and the listing address suggesting it is a condo or apartment. This makes sense, as these type of residences would not have their own housing lot. In some cases it would help to impute a value for the missing lot size, but in this context it would not make sense to do so. At present we have decided to keep the listings with null values in acre_lot, as we may compare models with and without the feature, and explore building out an additional feature listing_type by parsing the address for clues to determine if it is a house or condo style listing.

Finally, we had to aggregate and transform the income tax data to go from totals at the zip code – AGI stub (bucket of annual gross income) level to statistics at the zip code-level. The transformations primarily involved summing up a column and dividing by the number of returns to get a per-return statistic. If the column represented an amount (e.g. total income tax), this gave us a "$/return" statistic, and if the column represented the number of returns (e.g. number of returns with mortgage income), then this gave us a "proportion of returns" statistic.

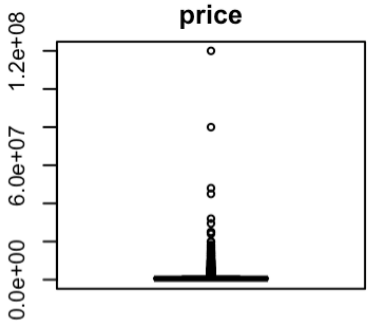
Finally, we merged the two datasets on zip code to get a listing-level dataset, with some listing-level features from our real estate data and some zip code-level features from the income tax dataset. This join was fully successful, with no dropped listings from the real estate dataset.

The final dataset looks like the following: The DV will be price, from our real estate dataset. The set of predictors is in the table to the right.

Predictor	Description
bed	Number of bedrooms on the listing
bath	Number of bathrooms on the listing
acre_lot	Size of the lot, in acres
house_size	Size of the house, in sq ft
state	State
total_credit_amt	Total tax credits amount per return
taxable_income_amt	Taxable income amount per return
mortgageint_amt	Mortgage interest paid amount per return
p_mortgageint_nr	Proportion of returns with mortgage interest paid
inctax_amt	Income tax amount per return
p_unemploy_nr	Proportion of returns with unemployment
agi_amt	Adjust gross income (AGI) [2]
num_dependents	Number of dependents per return
p_re_taxes_nr	Number of returns with real estate taxes

Exploratory Data Analysis

We've also completed an initial pass at EDA for the model, ignoring real estate features for now as we determine how to handle nulls in the bed, bath, acre_lot, and house_size fields. We conducted univariate analysis to see if there were any outliers in the dataset, investigated the relationship between the features and the DV using pairplots and correlation plots, and evaluated multicollinearity between predictors using a correlation heatmap.

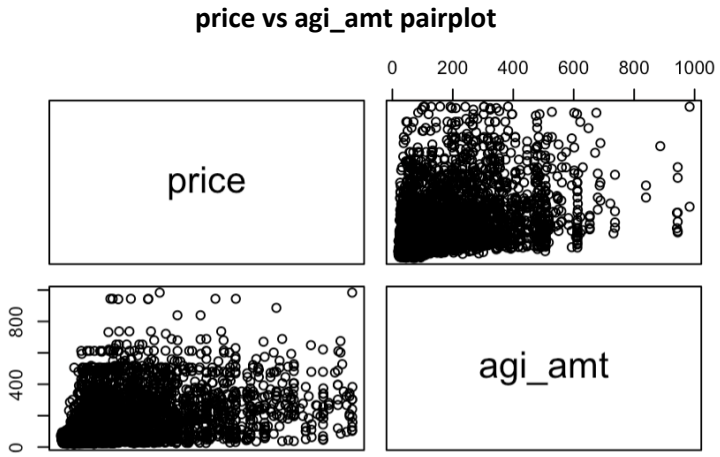


Univariate analysis revealed there are potential outliers in all except the agi_bucket field, which we will need to investigate. In particular, price has a number of outliers far beyond the whiskers of the box-and-whisker plot which we will want to investigate.

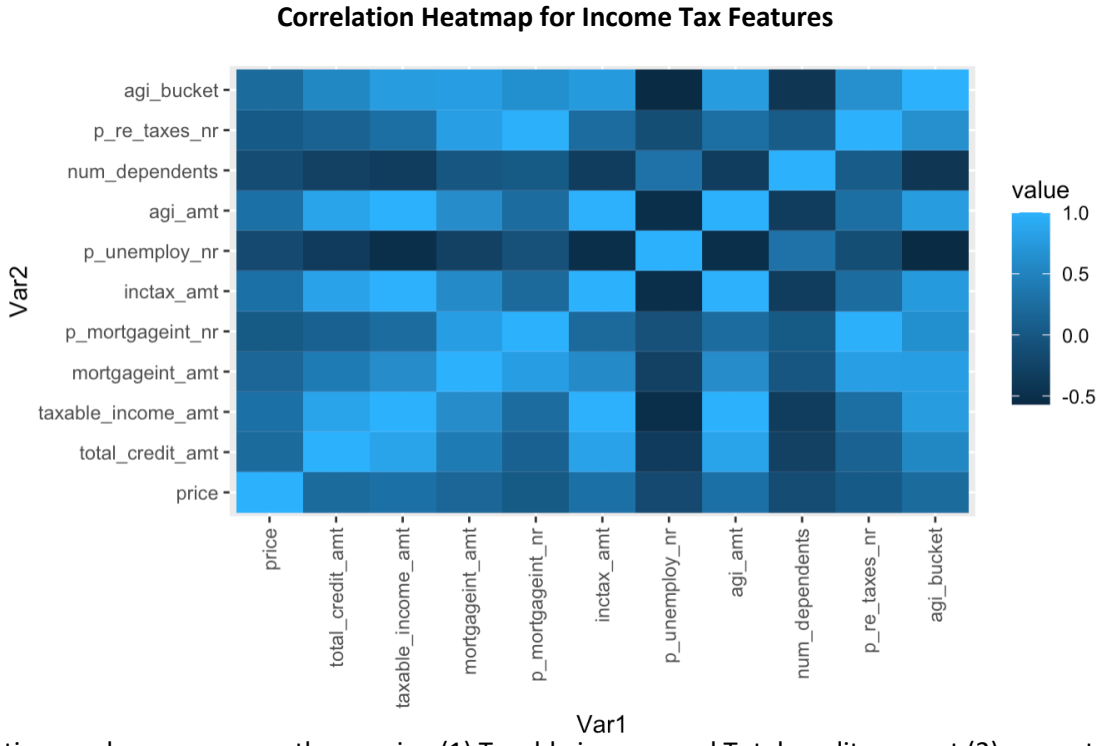
Pairplots showed a few places where a clear relationship could be seen. Within the tax predictors, we could see that taxable income amount, income tax amount, annual gross income, and p_re_taxes all had a weak relationship with

house price. The rest of the predictors had no correlation with the price. This aligned with their correlations, as show in the table below. As we can see from the agi_amt pairplot, there's still quite a bit of noise in the strongest relationships.

Feature	Correlation with Price
total_credit_amt	0.215
taxable_income_amt	0.292
mortgageint_amt	0.174
p_mortgageint_nr	0.020
inctax_amt	0.292
p_unemploy_nr	-0.176
agi_amt	0.288
num_dependents	-0.144
p_re_taxes_nr	0.032
agi_bucket	0.219



Evaluation of multicollinearity showed many places where we had correlation between predictors, particularly in the income tax dataset.



Strong correlation can be seen among these pairs: (1) Taxable income and Total credit amount (2) p_mortgageint_nr and mortgageint_amt (3) Income tax amount and Total credit amount (4) Income tax amount and Taxable income amount (5) Agi amount and Total credit amount (6) Agi amount and Taxable income amount (7) p_re_taxes_nar and Mortgageint amount. (8) Agi bucket and Taxable income amount (9) Agi bucket and Mortgageint amount.

Model Development and Selection

Finally, we have made significant initial progress on model development and selection. In addition to identifying the models we plan to test, we have also developed a rough framework in R for how we want to build and evaluate the models.

From a model type standpoint, we're going to use a combination of linear models and tree-based models. Specifically, we're going to build multiple linear regression, LASSO regression, random forest, and boosting regression models. While

the linear and LASSO regression models will help us better understand specific predictor-DV relationships through coefficient analysis, we expect our tree-based models to be stronger given their ability to account for nonlinear relationships between predictors and the response.

Our rough R framework is primarily focused on training each of these models, using both built-in R packages as well as other packages such as glmnet, dplyr, and stats. Up front, we'll split the data into training and test sets using an 80/20 split. Then, we'll train each model on the training set and compare using the test set. For the linear and LASSO models, we plan on performing PCA to reduce multicollinearity. Hyperparameters such as lambda for LASSO and number of trees for Random Forest will be selected using k-fold cross validation. Finally, we will compare the performance of our selected model to a model that does not contain tax data features. That way, we can better understand the incremental value our tax data adds when predicting house prices.

CHALLENGES & LESSONS LEARNED

The biggest challenges this far have been primarily during data cleansing. Understanding the meaning of different columns, working with the varying levels of detail in the data, cleansing the null values all presented their own unique challenges. We had to do additional digging through Kaggle and government sites to fully understand the units of the amount columns (thousands of dollars), as well as what each field meant in terms of tax forms. The tax data was also at a strange level of detail; each row represented a range of AGI within a zip code. This made aggregation math very important – after evaluating median, arithmetic mean, total, and weighted average, we settled on using the total amounts to get a balanced view of the aggregate. Finally, null values in the income tax data and the acre_lot field forced us to make decisions (some ongoing) about whether to impute or remove.

INITIAL HYPOTHESES

Based on the low correlations we saw between income tax features and price, it seems like the real estate features will still provide the strongest signal to the model given their relevance]. In addition, based on their correlations, the tax features related to affluency (AGI, taxable income amounts) seem to have the most potential to provide signal to the model. Finally, we theorize that tree-based models will far outperform linear models due to their ability to pick up nonlinear signal and predictor interactions without explicit interaction terms.

NEXT STEPS

Our next steps are primarily focused on finishing EDA, completing model development and selection, and evaluating the outputs. While we've completed EDA for our income tax features, we need to do the same for our real estate listing features once dealing with nulls. We've built the initial linear models, and we will be building out the models for LASSO, Random Forest, and Boosting in the next 1-2 weeks. This will allow us another week to evaluate the results. During that time, we'll compare performance metrics to select a top model. In addition, it's during this step that we'll run a version of the model without the income tax features to understand the value that those features add. We'll then use the final model and the versions with and without tax features to draw conclusions about the impact of real estate data on house price prediction.

LITERARY SOURCES

[1] "Machine Learning based Predicting House Prices using Regression Techniques"; J Manasa, Radha Guota, N S Narahari; https://ieeexplore.ieee.org/abstract/document/9074952

[2] "Predicting House Prices with Spatial Dependence: A Comparison of Alternative Methods"; Steven Bourassa, Eva Cantoni, & Martin Hoesli; https://www.tandfonline.com/doi/abs/10.1080/10835547.2010.12091276