MGT 6203 Group Project Final Report, Team 48

Real Estate Listing Price Prediction Enhanced with Income Tax Data

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. Often, 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 aimed 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 several features, primarily listing-specific. It includes the type of area the house is in, the availability, price, size, society, total square feet`, 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], several 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

Data Sources (links, attachments, etc.): Both datasets were downloaded from Kaggle, with the intent to join them on the zip code field to create our merged dataset for analysis.

1. USA Real Estate Dataset - https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset?resource=download
2. Individual Income Tax Statistics - https://www.kaggle.com/datasets/irs/individual-income-tax-statistics

Data Description (describe each of your data sources, include screenshots of a few rows of data):

1. USA Real Estate Dataset – 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.

```
# A tibble: 6 × 12
  status    price   bed  bath acre_lot full_address                                                  street  city  state zip_c…¹ house…² sold_date
  <chr>     <dbl> <dbl> <dbl>    <dbl> <chr>                                                         <chr>   <chr> <chr>   <dbl>   <dbl> <date>
1 for_sale 105000     3     2     0.12 Sector Yahuecas Titulo # V84, Adjuntas, PR, 00601             Sector… Adju… Puer…     601     920 NA
2 for_sale  80000     4     2     0.08 Km 78 9 Carr # 135, Adjuntas, PR, 00601                       Km 78 … Adju… Puer…     601    1527 NA
```

2. Individual Income Tax Statistics – 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.

```
# A tibble: 6 × 128
  state…¹ state zipcode agi_s…²     n1  mars1  mars2  mars4   prep     n2 numdep total…³  vita   tce a00100 n02650 a02650 n00200 a00200 n00300 a00300
  <chr>   <chr> <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 01      AL    00000         1 850050 481840 115070 240450 479900 1.40e6 548630   24840 16660  8180 1.10e7 850050 1.12e7 682860 8.75e6  95140  64688
2 01      AL    00000         2 491370 200750 150290 125560 281350 1.02e6 375670   10850  7080  3780 1.77e7 491370 1.78e7 425830 1.45e7  92610  69421
```

Key Variables:

Our dependent variable was the 'price' field from the real estate dataset. We expected to predict the actual sale price of a home, based on the other features.

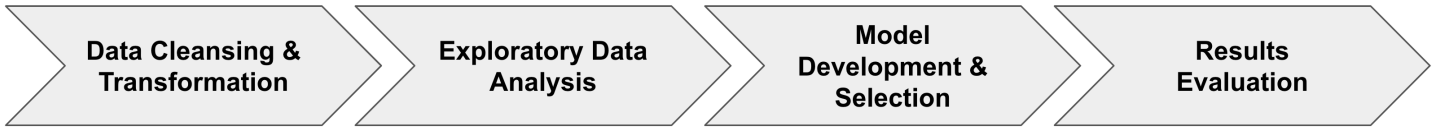
Our independent variables were a combination of the provided variables in the real estate dataset, which provide listing-level data, and others from the income tax dataset, which provide zip code-level data. Since the income tax dataset was initially at the zip code and AGI-bucket level, we needed to perform aggregations to get to our features, further described in the 'Process & Key Findings' section. These aggregations generally will calculate either an amount per return or a proportion of total returns in a zip code. Since the real estate dataset may have multiple records for a home, we also needed to remove duplicates and take the most recent price for a home. More detail, including a field list is included in the 'Data Cleansing & Transformation' subsection of the 'Process & Key Findings' section.

From the real estate dataset, we expected that the number of beds, number of baths, and house size would all be significant. From the income tax dataset, there's a lot to evaluate. Although EDA provided more specific insights, we initially expected the most significant features to be those related to affluency – taxable income, AGI, total income tax, and total income amount.

APPROACH

Planned Approach

We broke this problem into four main steps.



In the data cleansing and transformation step, our objective was to create the dataset which we'll study in exploratory data analysis and ultimately feed to the model. The activities in this step included 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 was to summarize the data in a way that will help us understand anything we need to know before modeling. The activities in this step included 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 was to test various model types and select the model which provides the best results. We tested both transparent models whose coefficients can be interpreted (linear regression, LASSO regression) and opaque models (Random Forest, Boosting). Activities here included 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 picked regression-appropriate error metrics for comparison (e.g., mean squared error, R-squared), and we used cross-validation to train and tune the hyperparameters for our models.

Finally, in the results evaluation step, our goal was to interpret the model to answer our research questions. The main activity was evaluating model performance metrics to validate results and understand the predictive power of our model. In addition, it's during this step that we ran a version of the model without the income tax features to understand the value that those features add. All analysis was completed in R Markdown files.

PROCESS & KEY FINDINGS

Data Cleansing & Transformation

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 were 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 types of residences would not have their own housing lot.

Field	Number of Nulls
house_size	2,202 (29%)
acre_lot	87 (1.2%)
bed	25 (0.3%)
bath	15 (0.2%)

To address these null values, we took an approach of imputing where we had enough insight to do so and dropping the remaining records. First, we noticed that a significant number of nulls occurred in acre_lot wherever the listing was for an apartment or condo, presumably where the unit does not come with land. To address this, we used string matching to identify records with 'unit' or 'apt' in the address, and we created a binary flag column ('house') representing whether a listing was a house (with its own lot) or not a house (without a lot). We then created a column representing the product of this binary column and the acre_lot column ('house.acre.lot'). This new column effectively showed the lot size if the listing was a house, and 0 otherwise. Next, we noticed that studio apartments often had null values for the number of bedrooms and 1 for the number of bathrooms. In these instances, we set the value for `bed` to 0. Once these two imputations were performed, we removed any remaining records with null values in any of the predictor columns, as there were not reliable methods for imputing the remaining nulls.

Our second-to-last transformation

step was to aggregate and transform the income tax from records at the *zip code – AGI bucket* 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.

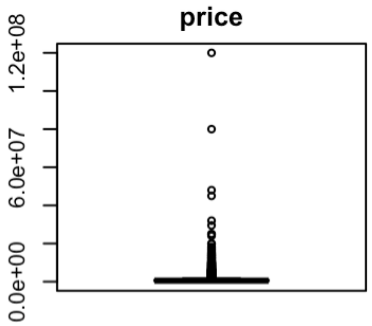
Predictor	Description
bed	Number of bedrooms on the listing
bath	Number of bathrooms on the listing
house	Binary flag set to 1 if a house with its own lot, 0 if a building unit (apartment or condo)
house.acre.lot	Size of the lot, in acres for houses, set to 0 for apartments/condos
house_size	Size of 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

The Dependent Variable (DV) was the 'price' field, from our real estate dataset. The set of predictors is in the table above.

Exploratory Data Analysis

During EDA, 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.

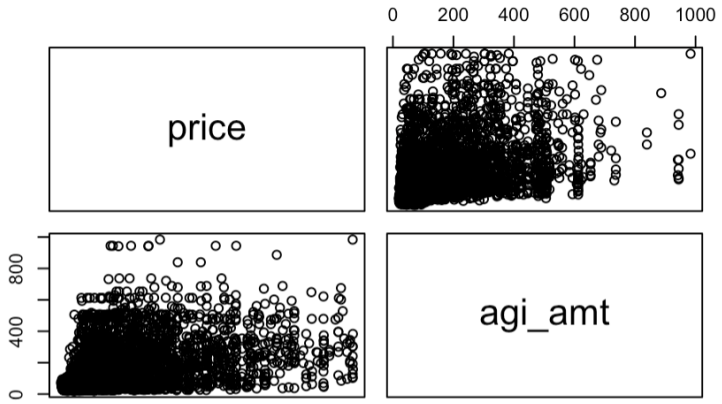
Univariate analysis revealed there are potential outliers in all except the agi_bucket field, which we will need to investigate. In particular, price has a number of outliers far beyond the whiskers of the box-and-whisker plot which we will want to investigate.



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.

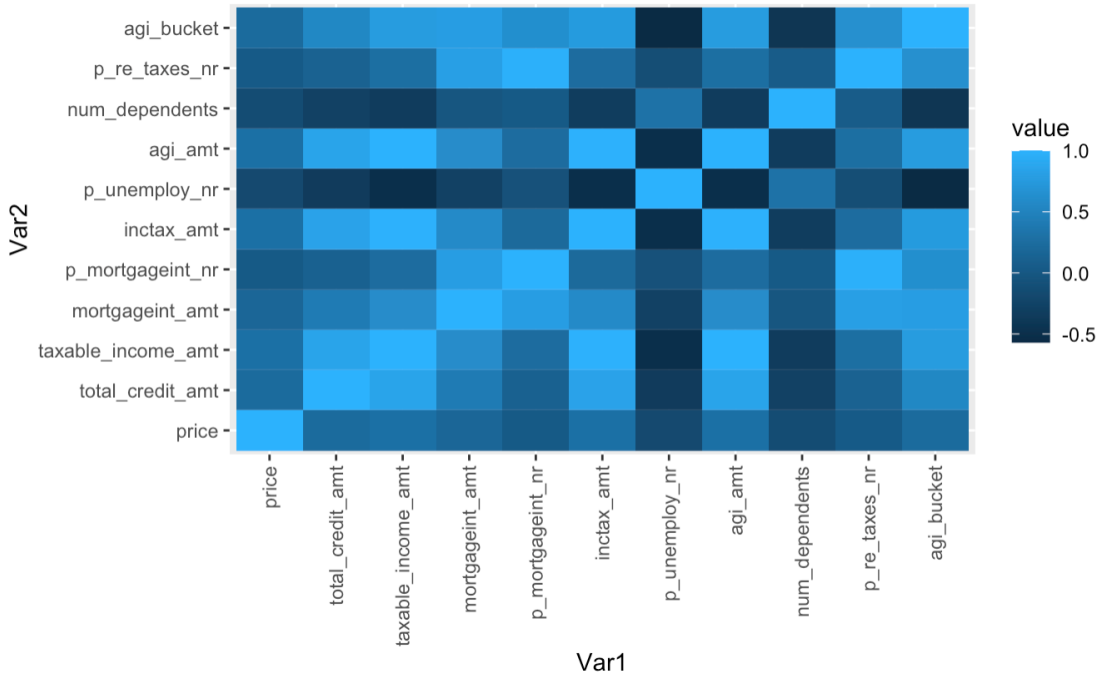
Feature	Correlation with Price
total_credit_amt	0.290
taxable_income_amt	0.375
mortgageint_amt	0.234
p_mortgageint_nr	0.024
inctax_amt	0.378
p_unemploy_nr	-0.215
agi_amt	0.372
num_dependents	-0.179
p_re_taxes_nr	0.046
agi_bucket	0.278
bed	0.218
house_size	0.395
house_acre_lot	0.030
bath	0.490

price vs agi_amt pairplot



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

Correlation Heatmap for Income Tax Features



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.

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.

Model Development and Selection

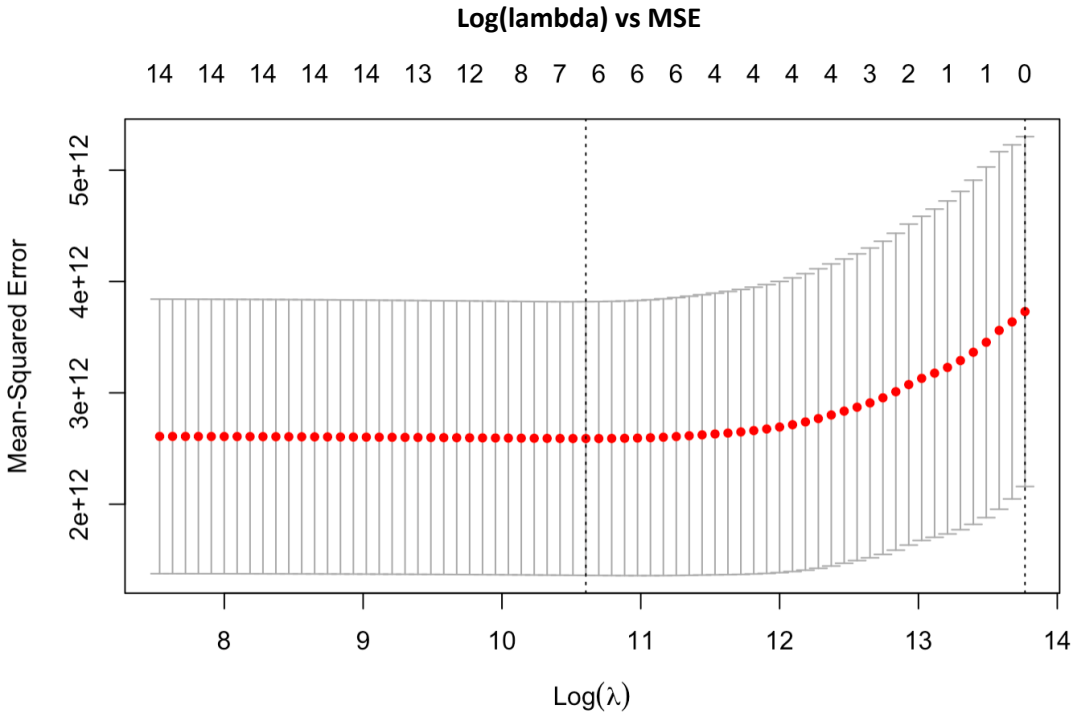
Third, we performed model development and selection. During this step, we identified the types of models we wanted to test, prepared the data for modeling, and evaluated each model.

From a model type standpoint, we used a combination of linear models and tree-based models. Specifically, we built 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. We also planned to run a final version of each model without the income tax data, to compare the performance when we add this data.

To prepare the data, we read in the relevant R packages (stats, dplyr, glmnet, caret, randomForest, xgboost, fastDummies) and the data from the output of the transformation step. Then, we created dummy variables from the state names, using the `fastDummies` package. Next, we scaled all non-binary numeric columns. Finally, we randomly split the data into an 80/20 training and test set.

During model evaluation, we trained each of the four model types on the training set and evaluated them on the test set. For each model type, we trained a model with all features as well as a model with only the traditional real estate features. There were also a few model-specific preprocessing steps needed:

- For linear regression, we performed PCA before model fit to account for multicollinearity
- For LASSO regression, we used cross-validation with 5 folds to determine the best value of lambda
- For Random Forest, we iteratively tested different hyperparameter values before ultimately selecting 128 trees
- For XGBoost, we iteratively tested different hyperparameter values before ultimately selecting 50 trees and a maximum tree depth of 3



Results Evaluation

Finally, we could review the results to answer our research questions. This step was broken into two parts. First, we reviewed the R-squared values of each model both with and without the full feature set. This allowed us to determine the strength of the best model, as well as understand how much value the income tax features added. Next, we looked at the results of the LASSO model to get an understanding of the most important features.

In reviewing overall model performance, we saw that our random forest model performed highest with an R-squared value of 0.57, moderate strength. We also saw that adding income tax features to the model consistently improved the R-squared value, adding between 0.05 and 0.23 to the metric.

	R-SQUARED	
	Real Estate Features Only	**All Features**
Linear Regression	0.3408	0.3938
LASSO Regression	0.3336	0.3939
Random Forest	0.3450	0.5702
XGBoost	0.2282	0.3073

Next, in reviewing the LASSO results, we found several interesting insights:

- The number of baths was selected, while beds was not
- A listing that's a house is ~$50k lower than one that is not, if all other predictors are held constant
- A listing that's in New York is ~$472k higher than one that is not, if all other predictors are held constant
- The income tax amount and total number of returns in a zip code both have a positive relationship with price
- The number of dependents and proportion of returns with real estate taxes both have a negative relationship with price

Field	LASSO Coefficient
(Intercept)	777652.34
bed	.
bath	736419.1
house	-53034
house.acre.lot	.
state_Connecticut	.
state_Delaware	.
state_Maine	.
state_Massachusetts	.
state_New.Hampshire	.
state_New.Jersey	.
state_New.York	472845.74
state_Pennsylvania	.
state_Rhode.Island	.
state_Vermont	.
house_size	231909.44
n1_total	28145.47
total_credit_amt	.
taxable_income_amt	.
mortgageint_amt	.
p_mortgageint_nr	.
inctax_amt	334825.93
p_unemploy_nr	.
agi_amt	.
num_dependents	-226096.28
p_re_taxes_nr	-61115.92
agi_bucket	.

CHALLENGES & LESSONS LEARNED

The biggest challenges were primarily encountered during data cleansing. Understanding the meaning of different columns, working with the varying levels of detail in the data, and 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.

POTENTIAL MODEL IMPROVEMENTS

When we thought about potential model improvements, two primary actions came to mind.

First, there are a few improvements to the model data that improve model results. For one thing, collecting more data might be helpful. While we built the model on approximately 5,000 records, more records might better capture zip code-to-zip code differences. In addition, adding more detailed real estate features from other datasets, such as build quality, could improve model accuracy. Finally, adding temporal features such as most recent comparable sale, could be helpful.

Second, improving the LASSO model by adding interaction terms could be interesting to investigate. There are a few potential interaction terms that might improve the model, such as the typical number of dependents on a return for a zip code multiplied by the number of bedrooms, to represent the additional value of bedrooms to families.

CONCLUSIONS

In response to our primary research question, we can see that our most predictive model (random forest) provides a moderate R-squared value of 0.57. Adding income tax features to the model improve the model, and increase the R-Squared value by 0.05-0.23. The most significant real estate features appear to be the size of the house, number of baths, and whether the house is in NY or MA. Finally, the most significant income tax features appear to be the number of returns, total tax credit amount, number of dependents, and the proportion of returns with real estate tax.

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