Real Estate Sales Analysis to Predict Mill Rates in Connecticut

Introduction

In real estate, the sale price of a property is one of the most important figures for prospective buyers and sellers. Before a property can be transacted, however, a local government assessor has to provide an assessed value for the property.[1] The assessed value tends to be determined as a percentage of the fair market value of the property.[1] The mill rate, which is equivalent to 1 dollar of tax per 1,000 dollars of assessed value, can be combined with the assessed value to determine the property tax.[4] The equation to calculate property taxes in Connecticut is shown below:

Property tax = (Assessed Value x Mill Rate) / 1,000.[4]

The mill rate of properties in Connecticut is decided by the Board of Finance and was the focus of this project since knowing the mill rate directly helps in calculating the tax obligation of a property.[6] In Connecticut, the median property value is 291,200 dollars, the average property tax rate is 1.63 percent, the median property tax is 4,738 dollars per year, and the median household income is 85,993 dollars.[1] This means that the typical household spends about 5.5 percent of their annual household income on property taxes. Homeowners are responsible for paying property taxes every year as long as they maintain ownership of a property. Thus, the property tax associated with a home is an important factor when deciding on buying a property and has considerable influence on the personal finances of prospective buyers.

Predicting the sale price of real estate tends to be the subject of many projects. This project, however, adopts a novel approach by focusing on predicting the mill rates of individual properties to help prospective buyers understand the annual tax burden of real estate. Two different data sets assembled by the Connecticut government were combined to complete this analysis in a novel way. While mill rates and property taxes can change over time, these attributes are important considerations when purchasing property and were the focus of this project.

Problem Statement

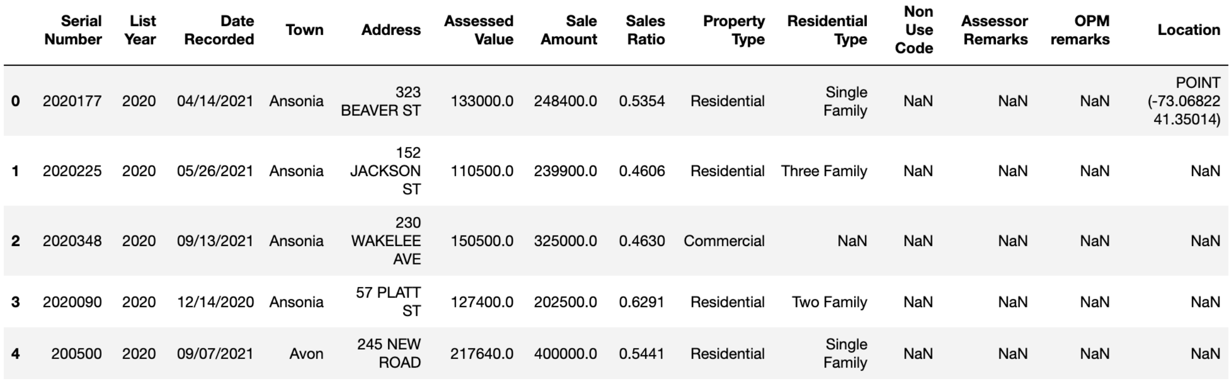
The goal of this project was to build regression models that can predict the mill rate of individual properties in the state of Connecticut with a high degree of certainty. The null model, which utilizes the average mill rate from the training data for its predictions, achieved an r-squared of approximately 0 and mean squared error (MSE) of 121.559. The r-squared represents the variability in mill rate that can be explained by the features included in the model, whereas the MSE represents the average squared difference between the predicted values and actual values.[3] These two evaluation metrics were utilized in tandem to assess the performance of various machine learning models. Therefore, the primary hypothesis that this project seeks to answer is the following: **Can a regression model be constructed to predict the mill rate of individual properties in Connecticut with an R-squared greater than 0.50 and an MSE less than 121.559.** A supporting hypothesis for this project includes determining the features that play important roles in making predictions.

In this project, feature engineering was performed, insightful visualizations were created, and strong predictive models were built to enable prospective home buyers to make informed decisions when transacting on homes and understand the implications on the property taxes.

Data Source

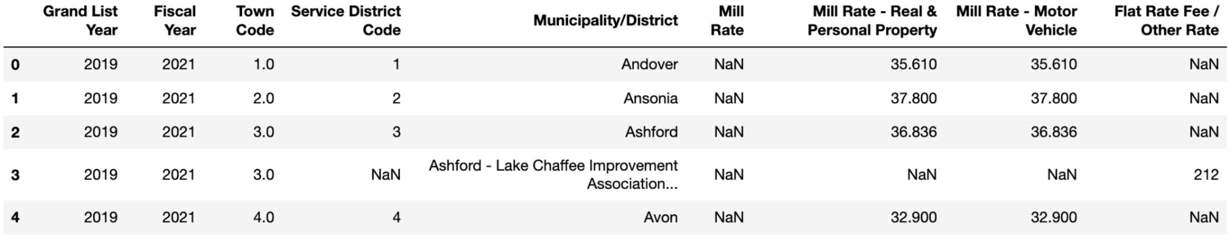
Two data sets compiled by the Connecticut government (i.e. ct.gov) were used in this project. The first data set includes real estate sales from 2001 to 2020 grand list years (i.e. October 1 through September 30), whereas the second data set includes mill rates from 2014 to 2024 fiscal years. Both data sets were downloaded as csv files. The real estate sales data set consists of 14 features and 997,213 observations, and a small sample is shown below in **Figure 1**. The mill rates data set consists of 9 features and 4,117 observations, and a small sample is shown below in **Figure 2**. The response variable used in the analysis is the variable labeled "Mill Rate - Real & Personal Property" in the mill rates data set.

Figure 1. Real Estate Sales Data



	Serial Number	List Year	Date Recorded	Town	Address	Assessed Value	Sale Amount	Sales Ratio	Property Type	Residential Type	Non Use Code	Assessor Remarks	OPM remarks	Location
0	2020177	2020	04/14/2021	Ansonia	323 BEAVER ST	133000.0	248400.0	0.5354	Residential	Single Family	NaN	NaN	NaN	POINT (-73.06822 41.35014)
1	2020225	2020	05/26/2021	Ansonia	152 JACKSON ST	110500.0	239900.0	0.4606	Residential	Three Family	NaN	NaN	NaN	NaN
2	2020348	2020	09/13/2021	Ansonia	230 WAKELEE AVE	150500.0	325000.0	0.4630	Commercial	NaN	NaN	NaN	NaN	NaN
3	2020090	2020	12/14/2020	Ansonia	57 PLATT ST	127400.0	202500.0	0.6291	Residential	Two Family	NaN	NaN	NaN	NaN
4	200500	2020	09/07/2021	Avon	245 NEW ROAD	217640.0	400000.0	0.5441	Residential	Single Family	NaN	NaN	NaN	NaN

Figure 2. Mill Rates Data



	Grand List Year	Fiscal Year	Town Code	Service District Code	Municipality/District	Mill Rate	Mill Rate - Real & Personal Property	Mill Rate - Motor Vehicle	Flat Rate Fee / Other Rate
0	2019	2021	1.0	1	Andover	NaN	35.610	35.610	NaN
1	2019	2021	2.0	2	Ansonia	NaN	37.800	37.800	NaN
2	2019	2021	3.0	3	Ashford	NaN	36.836	36.836	NaN
3	2019	2021	3.0	NaN	Ashford - Lake Chaffee Improvement Association...	NaN	NaN	NaN	212
4	2019	2021	4.0	4	Avon	NaN	32.900	32.900	NaN

Methodology

The entire methodology involves the following steps: (1) data collection, (2) data cleaning, (3) data preparation, (4) feature engineering, (5) EDA on training data, (6) modeling, and (7) evaluation.

To collect the data, the two data sets were downloaded as csv files from two ct.gov pages.

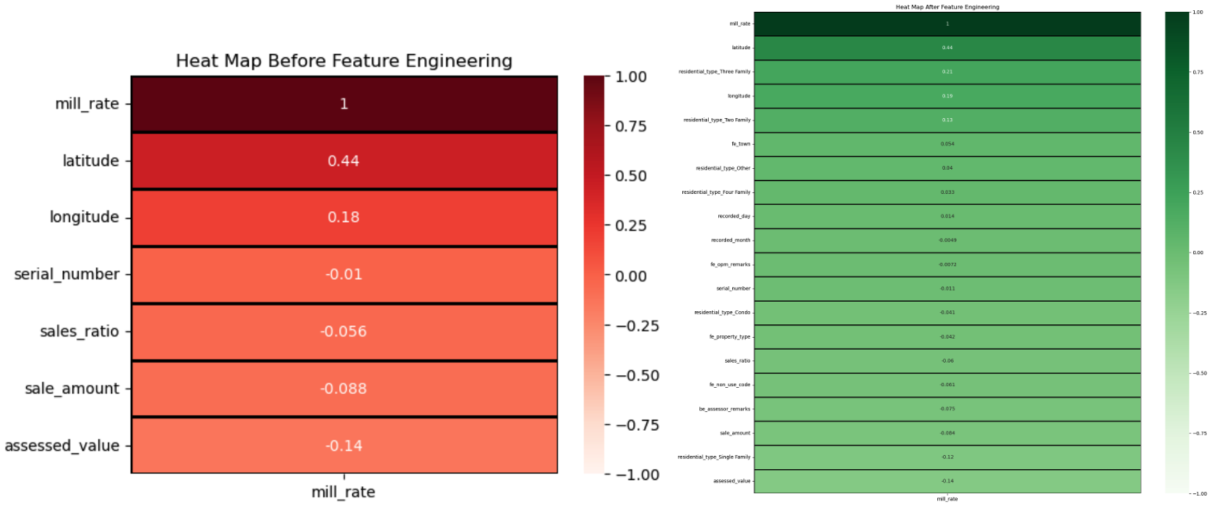
To clean the data, multiple steps were performed to make the data easier to use for analysis. For both data sets, the columns were renamed for easier access, the variables were converted to the appropriate data types, the variables that were not needed for the analysis were dropped, and missing values were handled through various methods. Handling missing values included methods like removing rows where missing values comprised less than 1 percent of the full data set, imputing null values with indicator values, and removing rows where location data was missing.

To prepare the data for modeling, additional steps were taken. Regular expressions were utilized to convert location (i.e. originally a categorical variable) into latitude and longitude data. Due to the large number of missing values, this meant that only a fraction of the original data could be used. In addition, the data was limited to include the latest data from 2021 (i.e. the 'date_recorded' variable included transactions from 2021) since the mill rate data had many missing values for previous years. The final data set contained about 2.2 percent, or 21,747 observations, after all of the data cleaning and preparation was finished.

The real estate sales data set contained 5 numerical features and 9 categorical features, and the mill rates data set contained 6 numerical features and 3 categorical features, so it was imperative to perform feature engineering to design features that can improve the results of predictive models. Techniques like expanding variables, creating dummy variables for categorical data, encoding frequencies and binary labels, and geocoding, were applied to expand the feature space. Furthermore, feature engineering was completed separately on the training and testing data to prevent data leakage.

Once the data was prepared for modeling, EDA was then conducted on the training data using the engineered features. Summary statistics were reviewed to observe if any unusual patterns could be detected. Multiple visualizations were created, such as heat maps before and after feature engineering, one-dimensional kernel density estimation (KDE) plots, one-dimensional histograms, box plots, and scatter plots. **Figure 3** below illustrates the impact of feature engineering on the feature space in the data set.

Figure 3. Heat Map Before (Left) and After (Right) Feature Engineering



In the modeling phase, 4 different machine learning models were utilized to solve this regression problem. The data was randomly split, with 80 percent assigned to the training data and 20 percent assigned to the testing data. The r-squared and MSE of each model were evaluated to determine their performances. Cross validation was conducted on the training data for each model to estimate model performance on unseen data. The null model, which predicts the average mill rate from the training data for every unseen data point, was used in setting the targets for the two evaluation metrics. The null model achieved an r-squared of 0 and MSE of 121.559, performing poorly as expected.

The first model explored in this project was the multiple linear regression model with recursive feature elimination. The recursive feature elimination allowed the multiple linear regression model to select the 5 most important features to use. After selecting 5 features, the linear model achieved an r-squared of 0.0719 and an MSE of 112.814, both of which were slightly better than those of the null model.

The second model was the random forest model. The parameters, n_estimators and max_features, were tuned via a grid search, and the best estimator had n_estimators equal to 150 and max_features equal to None. The random forest model achieved an r-squared of 0.994 and an MSE of 0.713, far exceeding the target performance stated in the **Problem Statement** and the performance of all the other models.

The third model was a neural network model called the multi-layer perceptron (MLP) regressor. Hyperparameter tuning was performed on the parameters, hidden_layer_sizes and max_iter, with the best model having hidden_layer_sizes of (10, 10, 10) and max_iter of 500. Despite tuning the parameters, all iterations of the model yielded extremely poor results, and the best model achieved an r-squared of -13,748 and an MSE of 1,671,351. The MLP model performed worse than the null model and may have required additional tuning to achieve better results.

The fourth model was the k-nearest neighbor (KNN) model. For this model, a pipeline with a standard scaler was included in addition to hyperparameter tuning. The parameters, n_neighbors, weights, and p, were all tuned, and the best model had n_neighbors equal to 5, weights equal to distance, and p equal to 1. The KNN model achieved an r-squared of 0.898 and an MSE of 12.388, making it the second best-performing model after random forest.

Evaluation and Final Results

As mentioned above, the two evaluation metrics utilized to assess the performance of all the models are r-squared and MSE. These evaluation metrics were chosen because the r-squared provides insights into how the features affect the variability in the response variable, and the MSE is a loss function that improves as it gets closer to 0. The results from the models can be seen below in **Figure 4**. The r-squared and MSE from the cross validations (denoted as CV) are also shown in **Figure 4**.

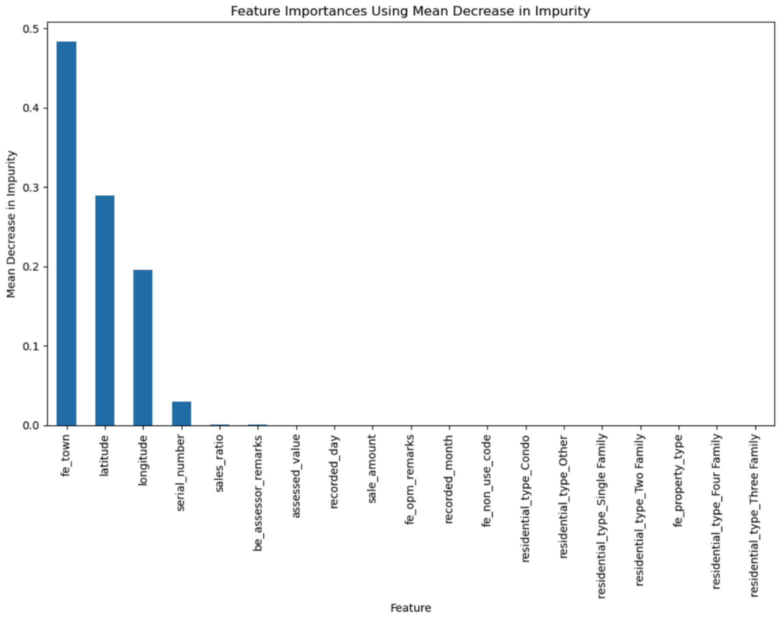
Figure 4. Model Results

Model	CV MSE	MSE	CV R-squared	R-squared
Null model	-	121.559	-	-2.737e-5
Multiple linear regression	113.098	112.814	0.0636	0.0719
Random forest	0.346	0.713	0.997	0.994
Multi-layer perceptron regressor	1.483e9	1.671e6	-1.217e7	-1.375e4
K-nearest neighbor	89.415	12.388	0.259	0.898

Overall, the random forest model performed the best on both metrics, followed by the KNN model, multiple linear regression model, null model, and MLP model. The random forest model is an ensemble model that utilizes trees to capture complex structures and appears to have performed very well on the unseen data. The KNN model utilizes the nearest neighbors to make predictions and also appears to have performed well on the test data. The multiple linear regression model, MLP model, and null model all performed poorly on the test data. The linear regression model may have resulted in poor performance because the nature of the problem was not linear. The MLP model may have performed poorly because the parameters may have required additional tuning.

Based on the optimal model (i.e. the random forest model), the most important features were: (1) fe_town (i.e. frequency encoded town), (2) latitude, (3) longitude, and (4) serial_number, as shown in **Figure 5**. It makes sense that these variables are important because the mill rates are determined based on the municipality of a property, and the first three variables all deal with geography.

Figure 5. Random Forest - Feature Importance



Returning to the **Problem Statement**, both the primary and support hypotheses were achieved. The random forest achieved an r-squared of 0.994 and an MSE of 0.713, both of which exceeded those of the null model and the specified targets. Furthermore, the random forest shows that (1) fe_town, (2) latitude, (3) longitude, and (4) serial_number were the most important features in predicting the mill rate on unseen data.

Python Libraries

Category Encoders
Datetime
IPython
Math
Matplotlib
NumPy
Pandas
Seaborn
Scikit-learn
Statistics
Time

References

[1] "Connecticut Property Taxes 2023". Tax-Rates.org, https://www.tax-rates.org/connecticut/property-tax. Accessed 4 Dec. 2023.

[2] Kagan, Julia. “Assessed Value: Definition, How It’s Calculated, and Example”. Investopedia, https://www.investopedia.com/terms/a/assessedvalue.asp#:~:text=Assessed%20value%20is%20the%20dollar,market%20value%20of%20the%20property. Accessed 12 Oct. 2023.

[3] "Mean Squared Error (MSE)". Statistics By Jim, https://statisticsbyjim.com/regression/mean-squared-error-mse/. Accessed 4 Dec. 2023.

[4] "Mill Rates". CT.gov, https://portal.ct.gov/OPM/IGPP/Publications/Mill-Rates. Accessed 4 Dec. 2023.

[5] “Real Estate Sales 2001-2020 GL”. Data.gov, https://catalog.data.gov/dataset/real-estate-sales-2001-2018. Accessed 11 Oct. 2023.

[6] "What is a mill rate and how is it established". Brookfield CT.gov, https://www.brookfieldct.gov/assessor/faq/what-mill-rate-and-how-it-established. Accessed 4 Dec. 2023.