


MMA 867: Predictive Modelling

Individual Assignment 1

Matthew Gottlieb

20259474

Due Date: April 30th, 2021



Matthew Gottlieb

Student at Smith School of Business, Queen's University
Toronto, Ontario, Canada
Joined a year ago · last seen in the past day

[Home](#) [Competitions \(2\)](#) [Datasets](#) [Code](#) [Discussion](#) [Followers](#) [Notifications](#)

Competitions
Novice

Unranked

0

0

0

Housing Prices ...
9 years to go
Top 3%

1,591st
of 65681

House Prices - ...
Ongoing
Top 23%

2,112th
of 9283

Datasets
Novice

Unranked

0

0

0

No dataset results

Notebooks
Novice

Unrank

0

0

No notebook

Competition Selection

The aim of this project is to select a Kaggle competition which is suitable for predictive modelling using regression analyses. Specifically, we are to choose three competitions with a minimum of 200 entrants in each and narrow it down. When it came to deciding which competition to select, I had a few ideas in mind. To start, I wanted to ensure that the data provided was robust enough to create a predictive model. For example, a dataset with 200 rows and <5 predictor variables would not be enough.

Conversely, I also did not want the dataset to be too large. This would mean millions of rows and thousands of predictors. Although this amount of data would allow for a strong model if done correctly, I also know the limitations of the hardware I am using, which would likely not be able to process this amount of data in a timely or effective manner.

The first dataset that I came across was 'House Prices – Advanced Regression Techniques'. This is a popular dataset on Kaggle, with many participants (over 9,000 currently), so I knew that the competition would be strong. After reviewing the dataset further, it appeared to fall into a so-called 'sweet spot'. That is, there was enough data to build an effective predictive model, but not too much that would overwhelm my computer. This ultimately would allow me to challenge myself to go above and beyond to build a strong model, as we had yet to deal with a dataset of this nature in class. I also knew that many of my classmates would be choosing this competition and that having my name show up near the top of the leaderboard would look good on me. Finally, I felt that working on a dataset like this would allow me to develop technical acumen that could be applied to real-world situations rather than some datasets which are more theoretically driven. This was ultimately the dataset that I chose to proceed with.

The following dataset that I looked at was 'New York Taxi Fare Prediction'. This dataset was also of great interest to me, as it solves a problem I always wondered (prior to the pandemic); How much will my Uber/Taxi fare cost? It would be interesting to uncover what the key drivers are of fare prices for taxis in New York City and understand what the actual distribution of fares looks like. When reviewing the data itself, it appeared highly robust, with about 55,000,000 rows in the training dataset and about 10,000 rows in the testing dataset. This is a significant amount of data, however, likely too much for my computer to handle. For that reason, I decided not to select this dataset.

The final dataset I looked at was 'M5 Forecasting – Accuracy'. The goal of this competition was to predict point forecasts of unit sales of various products sold at Walmart. In my professional career, I have built numerous sales forecasts for apparel retailers, so naturally, this contest sparked my interest. After reviewing the data, it appeared to be robust and fall within the sweet spot that I was looking for. However, as I have experience with forecasting in the retail industry, I opted not to choose this data set as I wanted to broaden my horizons.

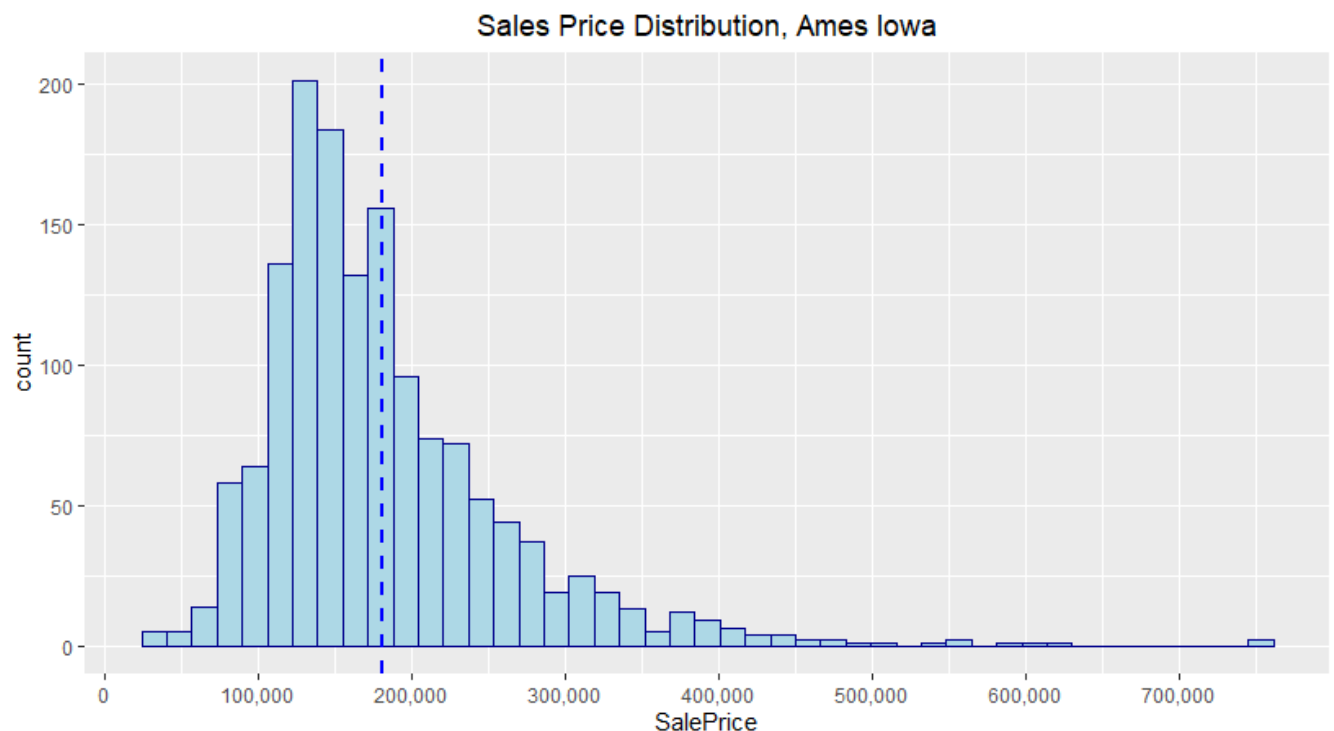
Consulting Report

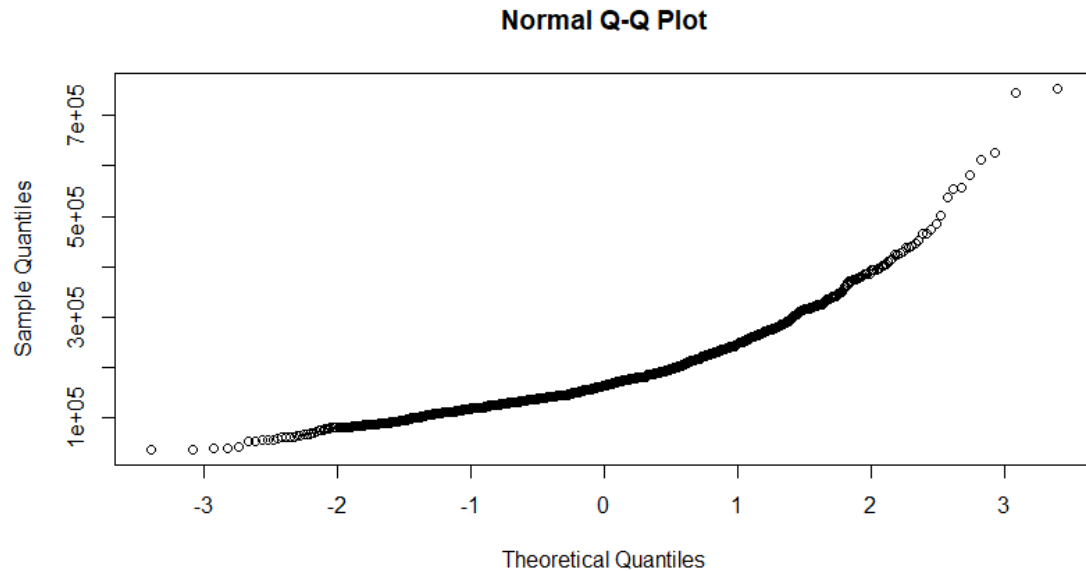
Business Problem

The business problem in this competition is to build a predictive model that can determine the selling prices of homes in Ames, Iowa, based on 79 predictor variables. As a home buyer/seller, real-estate agent, or housing developer, having insight into the predicted selling prices of homes is extremely valuable to have. For example, this model could help prospective home buyers identify the price they should expect to pay for a home given specific criteria. For a real estate agent, having access to this model could allow them to identify over/undervalued properties for their clients based on the expected selling price. Finally, a land developer could use this model to generate forecasts for the bank/investors to secure financing for a future project. Overall, there are multiple ways in which this predictive model could be leveraged to solve a variety of potential business problems.

Exploratory Data Analysis (EDA)

When beginning the exploratory data analysis for this project, the first task I wanted to do was gain an understanding of the variable that we are trying to predict; Sale Price. By understanding the minimum, maximum and average selling prices, I would be able to understand what the housing market looks like in Ames based on selling price. Below is a graph detailing the distribution of Sales Price from the raw data:





The first thing that comes to mind when viewing the above distribution is that it is not normally distributed. It appears to be positively skewed, resulting in the mean being larger than the median, which is larger than the mode. In order for our model to generate accurate predictions, this skew needs to be fixed with a log transformation, which will be done during the data prep portion of this analysis. Regarding the specifics of the selling price data itself, the mean sales price was observed to be \$180,921, while the median was \$163,000. In addition, the maximum selling price was \$755,000, and the minimum selling price was \$34,900. Finally, a standard deviation of \$79,442 was observed, indicating a wide spread of selling prices in the data.

In the next part of the EDA, I wanted to understand the variables that shared the highest level of correlation with the target variable, Sales Price. Specifically, I filtered all of the numeric variables in the dataset and ran a correlation analysis to discover which variables shared the strongest relationship with Sales Price. Below are the top 10 variables:

feature	correlation
SalePrice	1.00000000
OverallQual	0.79098160
GrLivArea	0.70862448
GarageCars	0.64040920
GarageArea	0.62343144
TotalBsmtSF	0.61358055
X1stFlrSF	0.60585218
FullBath	0.56066376
TotRmsAbvGrd	0.53372316
YearBuilt	0.52289733

From this chart, it is evident that OverallQual shares a very strong positive linear relationship at 0.79. This makes sense logically, as typically, higher quality homes will sell for higher prices. GrLivArea also appears, which refers to the above-ground living area in square feet (sqft). Similarly, this makes sense from a logical perspective as larger homes typically sell for higher prices. Other variables that appear also relate to the total size of the garage and how many cars can fit in it, total basement sqft, the number of full bathrooms in the house, and finally, the year it was built. These are variables that I will keep an eye on when creating the final predictive models. In addition, it is important to note that there were no variables that shared a significant negative linear relationship with Sales Price. The largest negative linear relationship observed was Kitchen above Ground with -0.13.

Data Prep

Initial Cleaning

The next portion of this analysis focuses on data prep. This step is extremely crucial to the success of our predictive models, as without appropriate and thorough data cleaning and feature engineering, we would not be able to create an effective model.

The first thing I was interested in with data prep is to check how many NA values there are in the data. This is important to understand, as machine learning models will not process NA values, and thus they need to be accounted for. Below is a chart detailing the variables that had over 10% missing values (2,919 total observations):

Variable	Quantity	%Missing
PoolQC	2909	0.9965741692
MiscFeature	2814	0.9640287770
Alley	2721	0.9321685509
Fence	2348	0.8043850634
FireplaceQu	1420	0.4864679685
LotFrontage	486	0.1664953751

The first variable that jumps out with over 99% missing is PoolQC. After investigating why these are missing, it appears that NA values refer to homes that do not have pools. Similarly, the NA values in MiscFeature do not have the features that are listed in the description; this includes elevators, a second garage, as well, as tennis courts. Alley, Fence, and FireplaceQU are all similar in that NA refers to the house not having the specified feature. Lot Frontage, the linear feet of street-connected to the property, is a different story, however. As this is a continuous variable, I decided to impute the missing values using the KNN approach, which is completed at a later stage of data prep.

When digging deeper into the data, there were additional adjustments that were necessary to make. For example, dates that came in numeric format were converted into characters and then ultimately factors. This

applied to Year Sold, Month Sold, Year Built, Garage, Year Built, and Year Remodeled. The reasoning behind this is so that the predictive model does not interpret increasing years as a measure of growth (although it could be argued that newer houses on average sell for more money), but rather a categorical variable represented as a level in a factor. I also noticed some issues with the Veneer variables. I converted the NA values to appear as 'None' as supported in the data documentation and adjusted the square footage for NA Veneer values to be 0.

Feature Engineering

The next stage of data prep was dedicated to feature engineering. This is one of the most critical portions of building an effective predictive model, as immense value can be created. By applying business acumen to the data, variables can be defined that improve the accuracy of the model drastically.

The first feature that I created was to define the age of the house. The idea behind creating this feature is that from market experience, newer homes typically have higher selling prices than older ones. Although this is a generalization and may not be the case in Ames, Iowa, it was a variable that I believed could improve the accuracy of the model. On the same note, I also added a dummy variable to indicate if the house was brand new (defined as being sold in the same year it was built). This was created as a dummy variable, with 1 indicating that it was brand new and 0 indicating that it was not.

The next variable I created was to define the total square footage available in the house. Typically the size of the house can play a prominent role in determining the selling price, and I wanted to make sure that the model was able to pick up on this relationship. This feature was defined by taking the total square feet available on the upper floors and adding them to the total basement square feet. Combined, this would indicate the total square footage available throughout the entire house.

Similarly, I also created a variable to define the total square footage available on the house porches. When examining the variables in the dataset, I discovered there to be five different variables in relation to the porch; Wood Deck sqft, Open Porch sqft, Enclosed Porch sqft, 3 Season Porch sqft, and Screen Porch sqft. By combining these all into one aggregate variable, it may make it easier for the model to pick up on the significance of the porch square footage available.

The final feature I created was in relation to the total number of bathrooms available at the house. Like the above features, typically having more bathrooms in a house can lead to higher prices, and I wanted the model to pick up on this relationship. This feature was defined as the number of full bathrooms + the number of half bathrooms.

Data Segregation & Recipes Function

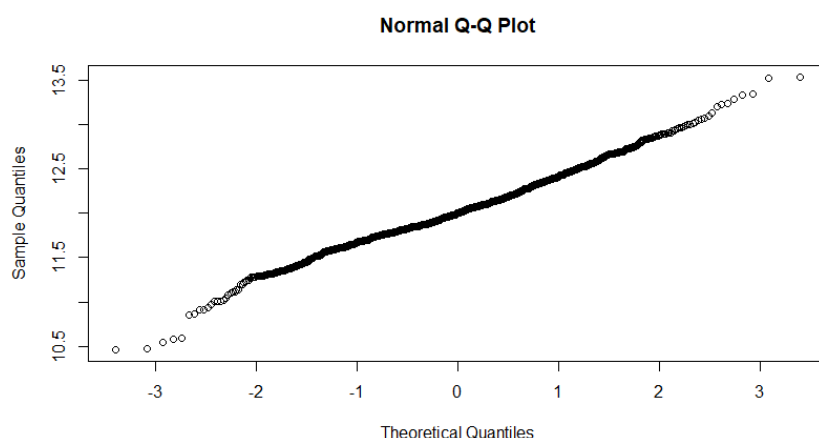
Once the new features were created, the next stage of the data prep was to convert all existing character variables into factors. The reasoning behind this was to make it easier to transform the categorical variables

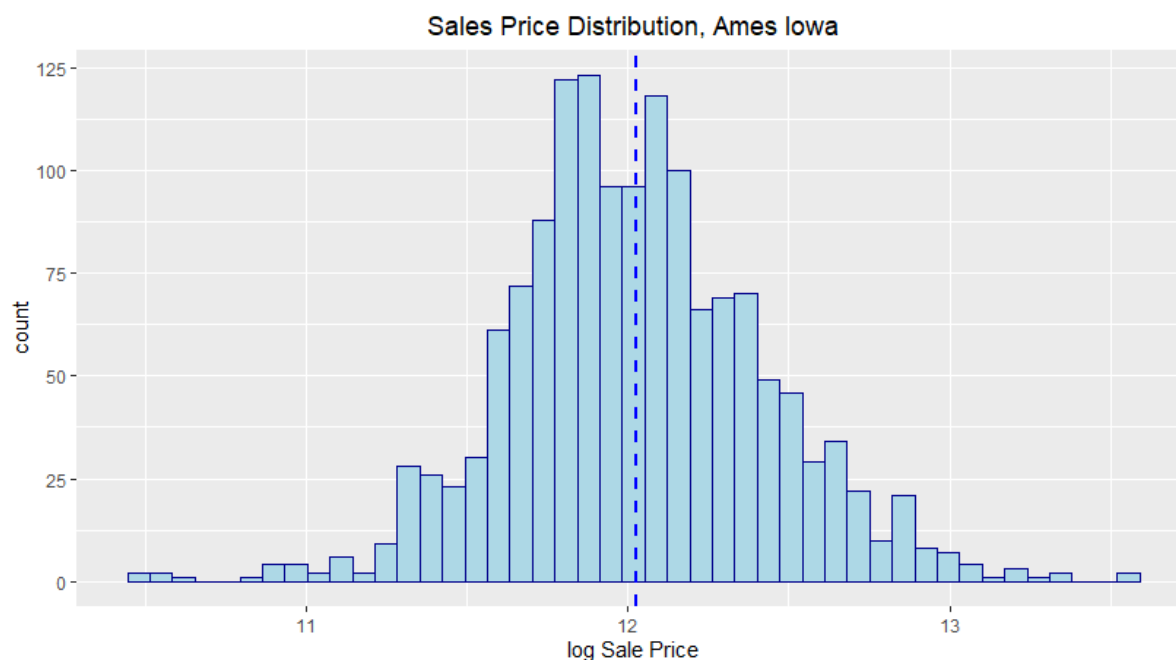
into dummy variables. Once complete, I merged the numeric variables and factor variables into one dataset containing all features and then split them back into train and test datasets.

The next step is one of the most crucial in the analysis. The recipes library was utilized as a centralized method to clean, impute and scale the data. To take care of the NA values in the data, KNN impute was used on all predictor variables. I chose this method instead of simply imputing mean/median values in order to receive more robust results specifically for the continuous variables. Once the data was imputed, the next step was to transform the factor variables into dummy variables. By doing so, the predictive model can differentiate between the different 'levels' within each category, essentially creating an individual intercept for each. This would allow the model to define different baselines for each level of the category.

Additionally, I also wanted to create interaction terms within the model. Doing so would allow the model to create adjusted slopes for the interacted variables, ultimately allowing it to consider different rates of growth. The first interaction that was created was between Neighbourhood and Overall Quality. The main idea behind this again stems from general acumen regarding real estate prices. One would typically expect the price of an excellent quality home in a top neighborhood to have a steeper slope than that of a poor quality home in a bad neighborhood. Furthermore, I also created a 3-way interaction between MSZoning (general zoning classification), MsSubClass (type of dwelling), and Overall Quality. The same intuition applies to this interaction. For example, the selling price of a 2-story home of good quality in a residential high-density area will likely increase at a different rate than a 1-story home of poor quality in an industrial area.

The final steps taken in data prep was in regards to scaling the data. First, a Box-Cox transformation was applied to all predictors. This step is vital as it allows us to transform non-normal variables to follow a normal distribution. In addition, I then centered and scaled the data to normalize it further. The variables in the raw data came in many different scales, so by normalizing and scaling it, the coefficients in the predictive model will be easier to interpret. We then apply the zero-variance filter to our data, which will remove variables that contain only a single value. In this case, this would represent a dummy variable without any observations, which would ultimately be a list of zeroes. Finally, we apply a log transformation to ensure that the Sale Price variable will be corrected to normality, as represented in the graphs below:





Model Training & Selection

Model Setup & Hyper Parameter Tuning

After preprocessing the data, the next step was to build an effective regression machine learning model to predict sales price. As the data was now very robust with predictor variables, relying on traditional OLS regression by manually adding and evaluating features was not an option. Instead, I looked towards testing regularized linear models, including LASSO and Ridge Regression. By doing so, I would be able to avoid the complicated task of manually selecting model features, and at the same time, could ensure that the model generalizes well.

For the model building itself, I opted to use the Caret package. The main reason behind this was due to its ease of use to toggle between different machine learning models in a relatively short amount of code. The first step was to define a range of lambda values for the models to cycle through. The Caret package would determine the optimal Lambda to use and apply it to the model. Furthermore, rather than simply training the model on the training data, I opted to use the repeated K-fold cross-validation method. Specifically, this would repeat 10-fold cross-validation five times to provide an accurate estimate of how the model would perform against new data. I chose the repeated method to increase the accuracy of estimates of model performance, as I knew that my hardware could handle the task. However, if the dataset were larger (i.e., millions of rows), this approach would not have been an option.

LASSO Model

The final step was to run the actual models. The first model I chose to test was LASSO regression. It resulted in a 0.12960 RMSE based on the repeated cross-fold validation, which was a strong result in the context of the competition (enough to represent the top 25%). Additionally, it reported an R^2 of 0.8934, signifying that the model explained a large amount of variation in Sale Price. When reviewing the coefficients that made it into the model, there were a couple of key takeaways. First, Overall Quality was one of the strongest positive coefficients in the model at 8.25. This was largely expected, as the overall quality of a home is usually a good indicator of its value. Furthermore, it was also noted that some of the features that were engineered during the data prep stage also played a significant role in the predictive power of the model. For example, Total Porch Area made it into the model with an 8.81 coefficient, as well as Total Sqft at 7.70. Total Bathrooms also made it into the model with a 2.21 coefficient. As a LASSO model will only select the features that provide the most value, this signified that the feature engineering process was a success. It was also observed that some of the interaction terms created made it into the model. The interaction between Neighbourhood and Overall Quality had the most significant impact, with 8/21 terms included. In addition, the 3-way interaction between Overall Quality, Sub Class, and Zoning Classification also had an effect with 5/35 interaction terms making it into the model.

Ridge Regression Model

When running the Ridge Regression model, the results were not as strong. It is important to note that, unlike LASSO, Ridge will not remove coefficients but rather shrink them down to near 0 depending on their importance. After using the same repeated cross-fold validation approach, the Ridge model produced a 0.1420 RMSE and 0.8732 R^2 , significantly worse than the LASSO method. After reviewing the coefficients of the model, the critical predictors that played a large role in the LASSO method were dulled in the Ridge method to make room for the fact that all coefficients needed to be included in some capacity. In this case, it was evident that the LASSO method was the clear winner in terms of strength of predictive power.

Competition Results

Ultimately, the LASSO model was used to make predictions against the testing data. When uploaded to the competition itself, the results were even stronger than anticipated. The model scored an RMSE of 0.12673, placing it inside the top 23% of the competition (at the time of submission). This provided the validation necessary to confirm that the model built was a useful one.

Conclusion


Overall, this project developed an effective model that can be used by homebuyers, real estate agents, and land developers to predict the sales price of homes. It began through an exploratory data analysis, where an understanding of the distribution of selling prices of homes in Ames, Iowa, was developed, as well as gaining insight into some critical variables that had the highest degree of correlation with our target variable, Sale Price. The next step was to prepare the data, first by identifying variables that had large numbers of missing values. It was shown that features such as Swimming Pools, Fencing and Tennis Courts had large quantities of NA values simply because the homes did not have those features, rather than it genuinely being missing information. Features were then created to provide additional insight, including Total SQFT, Total Bathrooms,

Total Porch Area, as well as the age of the house and whether or not it was brand new. These features would go on to play a significant role in the predictive power of the final model. The data was then cleaned, imputing missing values while also applying transformations, including Box-Cox, centering, and scaling, and taking the log value. Additionally, interactions were created between Overall Quality and Neighbourhood, as well as a 3-way interaction between Overall Quality, SubClass, and Zoning Area to represent different rates of growth. Finally, the predictive models were tested, opting to choose between regularized models with LASSO and Ridge Regression. It was ultimately shown that the LASSO model was the more effective in predicting Sale Price (0.12960 RMSE and 0.8934 R², vs. the Ridge Regression model (0.1420 RMSE and 0.8732 R²), resulting in a competition finish in the top 23% (0.12673 RMSE). Overall, this was evidence to show that the project was a success and that the LASSO predictive model generated could be an effective tool to solve the business problem at hand.

Appendix

Link to Kaggle competition: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

Screenshot of Leaderboard Position:



Matthew Gottlieb

Student at Smith School of Business, Queen's University
Toronto, Ontario, Canada
Joined a year ago · last seen in the past day

[Home](#)
[Competitions \(2\)](#)
[Datasets](#)
[Code](#)
[Discussion](#)
[Followers](#)
[Notifications](#)

Competitions
Novice

Unranked

0

0

0

Housing Prices ...

9 years to go

Top 3%

1,591st

of 65681

House Prices - ...

Ongoing

Top 23%

2,112th

of 9283

Datasets
Novice

Unranked

0

0

0

No dataset results

Notebooks
Novice

Unranked

0

0

No notebook

2112	Matthew Gottlieb		0.12673	8	18h
<div> <div>Your Best Entry ↑</div> <div> Your submission scored 0.12673, which is an improvement of your previous score of 0.12736. Great job! <div> Tweet this! </div> </div> </div>					
2113	Joe Harding		0.12673	1	1mo
2114	Satoru_Oishi		0.12673	5	2mo