Machine Learning Engineer Nanodegree

House Prices: Advanced Regression Techniques

Dirk Honda

March 9th 2018

Project Overview

For most individuals, home ownership is the largest and most important financial decision they will make. With approximately 64% of Americans owning a home [1] determining the fair market value of a home is critical. Historically, the value of homes has been predicated on the sale of homes in the area with similar features (e.g., beds, baths, living square footage). This approach is susceptible to outliers and variances that could skew the value of homes. The Kaggle competition, House Prices: Advanced Regression Techniques provides a dataset to which feature engineering, feature selection, and machine learning techniques can be applied to gain intuition and develop a model capable of producing future fair market values with greater accuracy.

Zillow, Inc. a leader in real estate analytics, has demonstrated that modeling with machine learning techniques can successfully be applied to the real estate domain. Zillow's Chief Analytics Officer, Stan Humphries outlined the approach and metrics in an article written at the end of July of 2017 [2]. The article states, "...2006 when we launched, we were at about 14% median absolute percent error on ... homes [prices]." Mr. Humphries is quoted as saying,"So maybe it's a decision tree, thinking about it from what you may call a 'hedonic' or housing characteristics approach, or maybe it's a support vector machine looking at prior sale prices." when speaking about the types of machine learning algorithms used to predict home prices.

Problem Statement

As stated in the project overview, house prices are generally modeled on recently sold comparable homes (sales comps). This approach is based on the opinions and experience of brokers and Realtors. As such it is susceptible to fluctuations, outliers, and human bias. By applying machine learning algorithms, a more robust and accurate regression model can be developed which can be evaluated by making sale price predictions on data prior to sale and then comparing them to their respective future purchase price.

Metrics

The objective of this Kaggle competition is to predict the sales price for each house in the dataset and submit the results for evaluation on the Kaggle website. Submissions are evaluated by root mean squared error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price. (Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally.)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Data Exploration

The housing dataset includes four files for use in the challenge: train.csv, test.csv, data_description.txt, and sample_submission.csv. The train.csv contains a mixture of categorical and numerical data features for training models comprised of 1,460 data points and 81 features.

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	Overall
1	60	RL	65	8450	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	
2	20	RL	80	9600	Pave	NA	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	
3	60	RL	68	11250	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	
4	70	RL	60	9550	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	
5	60	RL	84	14260	Pave	NA	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	
6	50	RL	85	14115	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	Norm	Norm	1Fam	1.5Fin	
7	20	RL	75	10084	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Somerst	Norm	Norm	1Fam	1Story	
8	60	RL	NA	10382	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	PosN	Norm	1Fam	2Story	
9	50	RM	51	6120	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	OldTown	Artery	Norm	1Fam	1.5Fin	
10	190	RL	50	7420	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Artery	Artery	2fmCon	1.5Unf	
11	20	RL	70	11200	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	1Fam	1Story	
12	60	RL	85	11924	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	Norm	1Fam	2Story	
13	20	RL	NA	12968	Pave	NA	IR2	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	1Fam	1Story	
14	20	RL	91	10652	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	1Story	
15	20	RL	NA	10920	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	1Story	
16	45	RM	51	6120	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Norm	Norm	1Fam	1.5Unf	
17	20	RL	NA	11241	Pave	NA	IR1	Lvl	AllPub	CulDSac	Gtl	NAmes	Norm	Norm	1Fam	1Story	
18	90	RL	72	10791	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	Duplex	1Story	
19	20	RL	66	13695	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	SawyerW	RRAe	Norm	1Fam	1Story	
20	20	RL	70	7560	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	
21	60	RL	101	14215	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NridgHt	Norm	Norm	1Fam	2Story	
22	45	RM	57	7449	Pave	Grvl	Reg	Bnk	AllPub	Inside	Gtl	IDOTRR	Norm	Norm	1Fam	1.5Unf	
23	20	RL	75	9742	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	1Story	
24	120	RM	44	4224	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	MeadowV	Norm	Norm	TwnhsE	1Story	

Figure 1: Data from the train.csv file

The test.csv, which will be used to evaluate how well we have modeled the data, has the same structure as train.csv. The SalePrice is removed and applied only after submission of the predicted Sale Price is uploaded to the Kaggle website.

	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB
1	GarageCond	PavedDrive	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition
2	TA	Y	140	0	0	0	120	0	NA	MnPrv	NA	0	6	2010	WD	Normal
3	TA	Y	393	36	0	0	0	0	NA	NA	Gar2	12500	6	2010	WD	Normal
4	TA	Y	212	34	0	0	0	0	NA	MnPrv	NA	0	3	2010	WD	Normal
5	TA	Y	360	36	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
6	TA	Y	0	82	0	0	144	0	NA	NA	NA	0	1	2010	WD	Normal
7	TA	Y	157	84	0	0	0	0	NA	NA	NA	0	4	2010	WD	Normal
8	TA	Y	483	21	0	0	0	0	NA	GdPrv	Shed	500	3	2010	WD	Normal
9	TA	Y	0	75	0	0	0	0	NA	NA	NA	0	5	2010	WD	Normal
10	TA	Y	192	0	0	0	0	0	NA	NA	NA	0	2	2010	WD	Normal
11	TA	Y	240	0	0	0	0	0	NA	MnPrv	NA	0	4	2010	WD	Normal
12	TA	Y	203	68	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
13	TA	Y	275	0	0	0	0	0	NA	NA	NA	0	2	2010	COD	Normal
14	TA	Y	0	0	0	0	0	0	NA	NA	NA	0	3	2010	WD	Normal
15	TA	Y	173	0	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
16	TA	Y	0	30	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
17	TA	Y	144	133	0	0	0	0	NA	NA	NA	0	1	2010	New	Partial
18	TA	Y	0	35	0	0	0	0	NA	NA	NA	0	6	2010	New	Partial
19	TA	Y	192	74	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
20	TA	Y	0	119	0	0	0	0	NA	NA	NA	0	2	2010	WD	Normal
21	TA	Y	220	150	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
22	TA	Y	238	130	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal
23	TA	Y	120	49	0	0	0	0	NA	NA	NA	0	4	2010	WD	Normal
24	TA	Y	36	23	0	0	0	0	NA	NA	NA	0	1	2010	WD	Normal
25	TA	Y	100	116	0	0	0	0	NA	NA	NA	0	1	2010	WD	Normal
26	TA	Y	100	0	0	0	0	0	NA	NA	NA	0	6	2010	WD	Normal

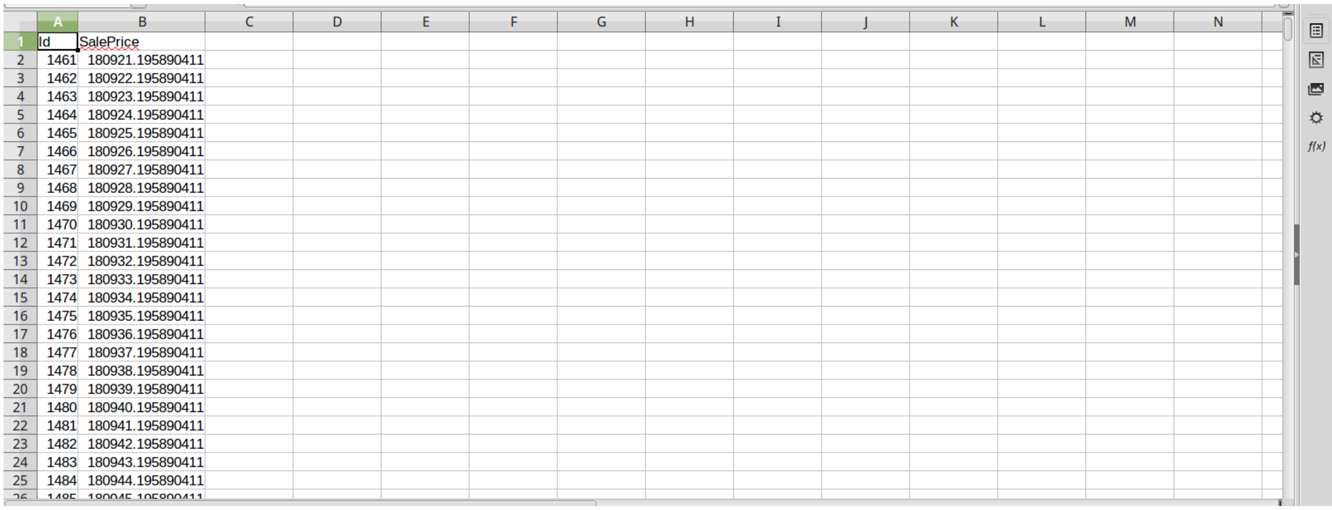
Figure 2: Data from the test.csv file

An additional document, data_description.txt, is a full description of each column, originally prepared by Dean De Cock but lightly edited to match the column names used here.

```
       TwnhsE   Townhouse End Unit
       TwnhsI   Townhouse Inside Unit

HouseStyle: Style of dwelling

       1Story   One story
       1.5Fin   One and one-half story: 2nd level finished
       1.5Unf   One and one-half story: 2nd level unfinished
       2Story   Two story
       2.5Fin   Two and one-half story: 2nd level finished
       2.5Unf   Two and one-half story: 2nd level unfinished
       SFoyer   Split Foyer
       SLvl Split Level

OverallQual: Rates the overall material and finish of the house

       10   Very Excellent
       9    Excellent
       8    Very Good
       7    Good
       6    Above Average
       5    Average
       4    Below Average
       3    Fair
       2    Poor
       1    Very Poor

OverallCond: Rates the overall condition of the house

       10   Very Excellent
       9    Excellent
       8    Very Good
       7    Good
       6    Above Average
```

Figure 3: Data from data_descriptions.txt

The last document included is the sample_submission.csv. It is a benchmark submission from a linear regression on year and month of sale, lot square footage, and number of bedrooms. The submission serves a dual purpose in demonstrating the format in which submissions must be uploaded to be evaluated by the Kaggle API.



Id	SalePrice
1461	180921.195890411
1462	180922.195890411
1463	180923.195890411
1464	180924.195890411
1465	180925.195890411
1466	180926.195890411
1467	180927.195890411
1468	180928.195890411
1469	180929.195890411
1470	180930.195890411
1471	180931.195890411
1472	180932.195890411
1473	180933.195890411
1474	180934.195890411
1475	180935.195890411
1476	180936.195890411
1477	180937.195890411
1478	180938.195890411
1479	180939.195890411
1480	180940.195890411
1481	180941.195890411
1482	180942.195890411
1483	180943.195890411
1484	180944.195890411

Figure 4: Sample Submission

With all datasets, certain abnormalities and characteristics germane to the dataset need to be addressed. In the case of the housing dataset three areas were addressed: categorical data, outliers, and missing values. The categorical data is incompatible with machine learning algorithms so to alleviate the issue, one hot encoding was employed. This increased the number of dimensions per record from 80 to 290, but allowed for the application of many different models. Outliers were identified by scatter plot diagrams which plotted SalePrice against the top 10 correlated features. Obvious outliers were removed if and only if they improved the test score as ranked by the Kaggle website. The third area of concern was missing data. To ensure that data was balanced and as normally distributed as possible, the median value was used for each feature to backfill any NaN or missing values.

Exploratory Visualization

The feature space after one hot encoding included 290 different dimensions. In order to explore the correlation and patterns associated with those features, correlation values were found between each feature and the target variable (i.e., SalePrice). T he rational being the greatest impact on the SalePrice would be those features with the strongest correlation. In the interest of time, the results were limited to the top 10 strongest correlations and then plotted. See figures 5 and 6 below.

Figure 5: Scatter-plot of top 5 features correlated to SalePrice

Figure 6: Scatter-plot of top 10 features correlated to SalePrice

Examination of the scatter-plots identified some areas with likely outliers. These features were individually compared against SalePrice to give a visual scatter-plot with greater granularity. All figures below are before any removal of outliers.

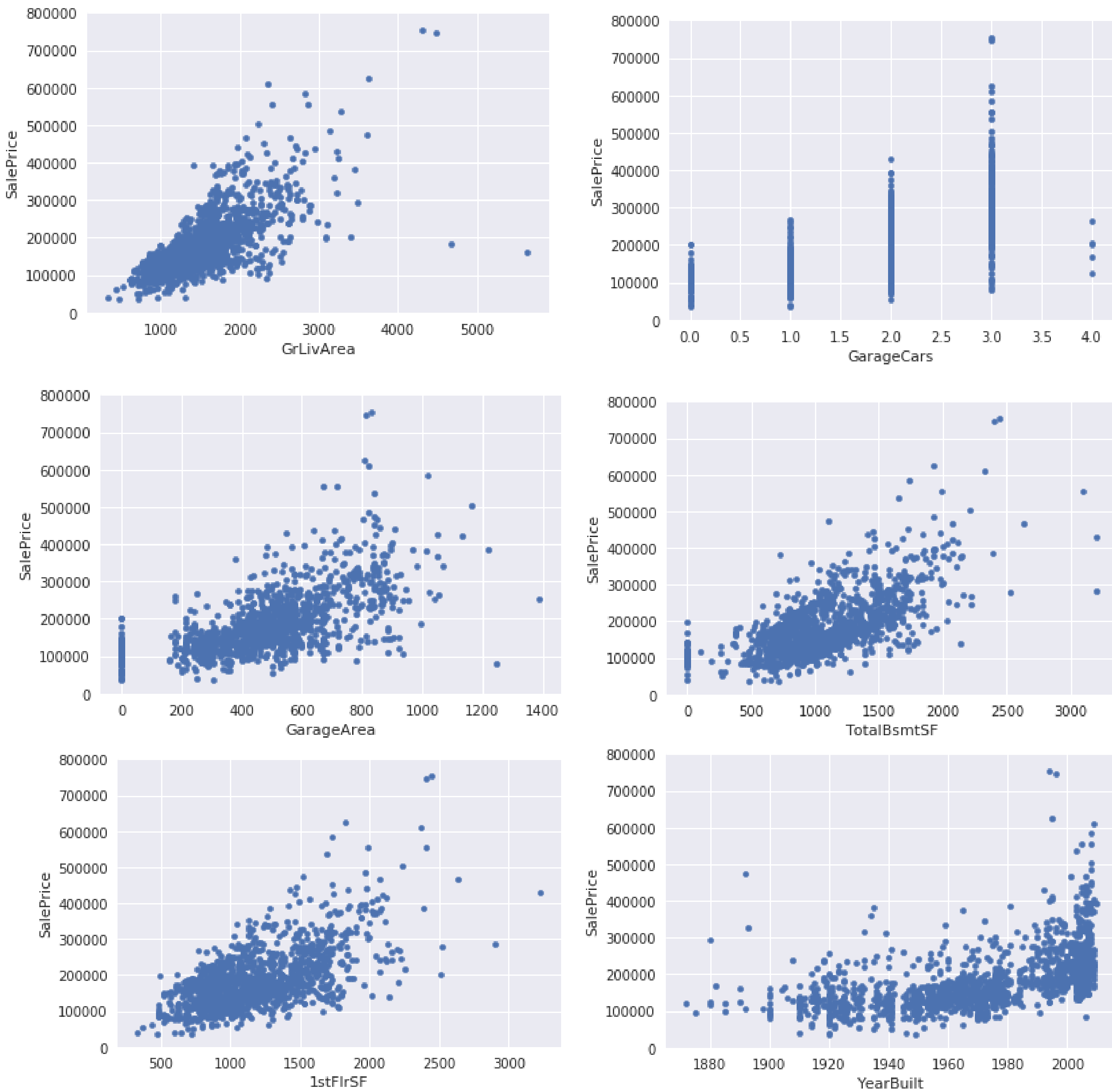


Figure 7: Scatter-plots with possible outliers

Algorithms and Techniques

The capstone project employed four different learning models: linear regression, SVM, Decision Tree, Random Forest, and XGBoost. The learning models were selected to provide a diverse and robust set of learners ranging from simplistic and fast (e.g., linear regression) to the more complicated ensemble methods with boosting (e.g. XGBoost). Initial evaluation was based on default parameter values of each learner and scoring was done with RMSE evaluation of 20% of the training data which was held back.

Benchmark

Zillow was to be the benchmark in evaluating the effectiveness of the learning model. Unfortunately, the test.csv does not include the SalePrice information. This would be necessary calculate the median absolute percent error. Instead the benchmark will be a RMSE of 0.43045 which was obtained by taking the mean of all the properties in the training dataset and using that value for all records. Ultimately, the goal is to improve this score and rank in the top 20% of the public leader board on Kaggle. In order to rank successfully in the top 20% the submission would require a RMSE of 0.12049.

Data Preprocessing

The data preprocessing consisted of the following steps:

1. One hot encoded training and test data to remove categorical data
2. Identified the strongest correlations between features and SalePrice
3. Scatter plot strongest correlated features and SalePrice and remove outliers which negatively impacted score of models
4. Backfill NaN values with mean value for ta given attribute

Implementation

Pandas, Numpy, Scikitlearn, Seaborn, and Math packages were used extensively in the processing and exploration of the housing data. Certain tasks required supplemental functions and algorithms. These areas include both model evaluation and data preprocessing.

rmlse(y_true, y_pred) – implements the root mean log square error for model evaluation purposes. In some cases, linear regression would fail to converge resulting in "nan" scenarios. These cases will cause issues with the rmlse result.

evaluate_model(model, X_train, X_test, y_train, y_test) – implements model fitting with the test_train_split parameters. Additionally, it will automatically score the results and return the model and score as a tuple

feature_cat_conversion(column, data, debug=0) – implements a custom one hot encoding approach. After many iterations, this effort was abandoned as the default Pandas encoding produced more consistent and reliable results.

detect_outliers(column, data) – implements analysis to find outliers based on a a given column and dataset. Informative statistics are also displayed in addition to having a list of record identifiers returned.

grid_search_cv(model, params, X_train, X_test, y_train, y_test, verbose=False) – implements grid search cross validation on a given model with the training and testing data of a train_test_split dataset. The function also scores the models based on the defined function rmlse().

Refinement

As previously stated several models were evaluated in baseline form with both training and test data. The evaluation was performed with the root mean log squared error of data. After which, each model was parameterized and run through grid search cross validation. The best of each model configuration was then evaluated against the other "tuned" models. Further refinement required the removal of outliers and the predicted price data uploaded to the Kaggle site for evaluation. In some cases, efforts to generalize the data better by removing outliers actually resulted in lower scores. These cases have been documented in the notebook for completeness.

Model Evaluation and Validation

Model evaluation and validation have been two-fold. Initial evaluation was done with 20% of the training data being excluded from the model fitting and later used as a ground truth in a root mean log squared error (RMLSE) evaluation function. After final model selection and tuning, a second test dataset was used to predict 1,060 property prices. The true value is not visible, but results can be obtained by uploading the predicted prices to the Kaggle website. The Kaggle website evaluation scores were sufficiently close to feel confident that the model is generalized well enough to unseen data. The RMSLE results in some cases are only 3% off from the Kaggle score.

Justification

The best performing model was XGBRegressor. The model performed significantly better than other models; in some cases performing 3.4x better. These results are based on the localized RMLSE scoring system with the train test split approach. Further, these findings were consistent when evaluated by the Kaggle site. The best score obtained on the site, .13007, falls short of the goal of .12049. These findings are significant and do suggest that this approach could be feasible for solving this problem. Unfortunately the amount of error is too large to say that this project has done so. After viewing other submissions on the Kaggle site, it's clear that very accurate (< .01 RMSE) predictions can be made with the correct model, feature selection, and feature engineering.

Reflection

Below is the process overview for the House Prices: Advanced Regression Techniques project.

1. Data Exploration: Familiarize myself with the data through an explorative process including statistical analysis and data visualization.
2. Data Cleansing: Remove outliers, fill null values with mean, one hot encode categorical values, check relevance of every feature to the target feature.
3. Train Model: Train several supervised machine learning models and tune with techniques such as cross validation and GridSearchCV.
4. Test Model And Optimize: Optimize the model offline with the test dataset until satisfactory RMSE is achieved.
5. Submit: Finally submit results to Kaggle for evaluation.
6. Repeat as needed to meet appropriate benchmark.

The project had many facets, the one that I found most interesting related to identifying and removing key outliers. The concept that a few records can have a dramatic impact on a model as they did in this project was fascinating. I found most aspects of this project challenging as most of the techniques and packages were completely unknown to me prior to this course. If time spent is any indicator of difficulty, visualizations consumed the greatest amount of time. The final model, XGBoost, performed better than the other models unsurprisingly. It's far more powerful than the other models and wins many competitions so I expected it to do well in this application as well. It is well suited for problems in this domain.

Improvement

To see further improvement in the model, additional outliers and anomalies should be removed. While I did spend many hours tweaking the selections, a more experienced engineer can likely identify records which are skewing results. This would allow the model to generalize better and improve te predictive power of the model.

References

https://www.census.gov/housing/hvs/files/currenthvspress.pdf

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

http://www.zdnet.com/article/zillow-machine-learning-and-data-in-real-estate/