Weather in Chicago

Blake Wallace
Capstone Technical Report

May 14, 2019

Data science objectives:

i. Is it possible to build a model that is easily re-implemented?

ii. Can we get a better score on each new submission to Kaggle?

Data:

See the data dictionary in Kaggle

- This data is from the Amos, IA housing market between the years 2006 and 2010
- There are 2051 data points in the training data, each having up to 80 descriptors.
- There are null values in the data. More will be said about this in what follows.

Data Cleaning/Data Manipulation/EDA:

Going from raw data to the modeling stage required a few transformations

- Null values

  Three rows, 616, 1327, 1712 were determined to have 10 troublesome null values. They were dropped.

  The 'Garage Yr Blt' feature had 114 cells that were null. There was not any clear way of understanding these empty measurements. So, the rows were dropped.

  The 'Mas Vnr Area' column had 22 cells that were null. In this instance it is reasonable to assume a null in this instance is representing 0 square feet. So, the cells here were filled with the number zero.

- Dropped Columns

  The 'Lot Frontage' feature was dropped from the dataframe. It was a hard feature to even interpret, let alone determine what the null values might represent.

- Data Conversion

  The 'Central Air' feature was converted from an object series to an int series by replacing any Yes ('Y') with a 1 and any No ('N') with a 0.

  The 'Paved Drive' feature was converted from an object series to a float series by populating 1 for 'Paved' 0 for 'not paved gravel/dirt', and 0.5 for 'Partial Paved.'

Note that, central to our research question, we started by doing little work to the descriptors aside from handling nulls, and gradually increased the amount of complexity present in our model creation. Ultimately, the biggest trend noticed was, all things being considered, the data tends to increase in accuracy as the number of features being used increases. That is to say, with more data comes better predictions. There was one exception to this rule, but in this exception we note that there was not...

Feature Engineering:

We did not use feature engineering during this project. However, we do believe there is plenty of room to increase our models accuracy, and that feature engineering could definitely be a primary contributor.

Features Matrix:

The first analysis used 10 features. They were

'Overall Qual', 'Year Built', 'Year Remod/Add', 'Total Bsmt SF', '1st Flr SF', 'Gr Liv Area', 'Full Bath', 'TotRms AbvGrd', 'Garage Cars', 'Garage Area'

The last analysis used 39 features. They are list below.

'Id', 'PID', 'MS SubClass', 'Lot Area', 'Overall Qual', 'Overall Cond', 'Year Built', 'Year Remod/ Add', 'Mas Vnr Area', 'BsmtFin SF 1', 'BsmtFin SF 2', 'Bsmt Unf SF', 'Total Bsmt SF', 'Central Air', '1st Flr SF', '2nd Flr SF', 'Low Qual Fin SF', 'Gr Liv Area', 'Bsmt Full Bath', 'Bsmt Half Bath', 'Full Bath', 'Half Bath', 'Bedroom AbvGr', 'Kitchen AbvGr', 'TotRms AbvGrd', 'Fireplaces', 'Garage Yr Blt', 'Garage Cars', 'Garage Area', 'Paved Drive', 'Wood Deck SF', 'Open Porch SF', 'Enclosed Porch', '3Ssn Porch', 'Screen Porch', 'Pool Area', 'Misc Val', 'Mo Sold', 'Yr Sold'

A careful look at these two lists shows that every item contained in the first data set is also included in the last. As the analysis proceeded, except for one instance, the analyses got better.

Cross-Validation:

Yes! Train/Test split and CV were key elements in all iterations of the model construction.

- During this project we stuck exclusively with an 80/20 ratio between the train and test sets during cross-validation. This was merely a product of trying a few at the beginning and it being the option that produced the best fitting model. So, we stuck with it throughout the completion of the assignment.
- We did also modify the random state parameter to be 75. This number was settled on after doing a for-loop to find a "good" candidate.
- We used CV with 5 folds in all cases simply because the recommendation from Riley was either 3 or 5.

Preprocessing:

We performed six different analyses. Five of them used a Standard Scale. After the initial model construction it was determined that a Power Transformation might be helpful in determining a best fir model. However, the RMSE score that came back was significantly higher than the score being generated by the other model. So, we abandoned the Power Transformation approach for the Standard Scale in all subsequent models. However, we take the time to note, since the target data "SalePrice" is right skewed, a power transformation could actually yield positive and useful results within future attempts at updating or improving our models.

Model:

This project used ordinary linear regression along with Lasso and Ridge regularization. It was not computationally expensive to use all of these models. The final submission, which yielded the lowest RMSE score came from performing a grid search on a pipeline through a Lasso Regularization to determine an optimal alpha.

Model Evaluation:

The success of our models was determined by the RMSE score. The Kaggle.com listing used this metric, and we deferred to it's calculation. It should be noted that within the Jupyter Notebooks containing the analyses the listed score is the R^2. In all of our models the training and CV scores were close and the testing scores are a little higher. We believe this is an indication of the potential to improve our model.

Recommendations/Insights?

Whenever attempting to build a model, easy re-implementation should always be a goal upfront. As a project unfolds it will become necessary to add or remove features, re-manipulate, or change the general model processes to attempt to improve the current working model. Keeping good checks in place (like functions and well catalogued scripts) can make it easy to transition from one model construction to the next.

It is also noteworthy that a lot can be done with a little amount of data alterations. Through six iterations, there was not significant change in the input being used to generate the model.