Applied Machine Learning HW3: Housing Price Predictions

Aiden Shaevitz
Shaevita@oregonstate.edu

1 Understanding the Evaluation Metric

1. The equation for the RMSLE is as below:

$$\textit{Root Mean Squared Logarithmic Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(y_i+1)-log(\hat{y}_i+1))^2}$$

2. The difference between RMSE and RMSLE is that the latter metric looks at the difference between the logarithm of the predicted value and the logarithm of the actual value as opposed to just the difference between the predicted and actual values.

3. This contest adopts RMSLE because it makes errors in predicting the values of inexpensive houses and expensive houses affect the final result equally. It does this by producing error that is more relative to the sample as opposed to taking absolute error.

4. This 0.11 intuitively means that on average the actual/predicted price is 1.116 (e^{0}.11) times greater than the predicted/actual price (could go either way).

5. Submitting the sample csv netted me a score of 0.40613.

6. My team name on Kaggle is AidenShaevitz.

2 Naive data processing: binarizing all fields

1. I found that there were 7227 features when performing the naive binarization.

2. (15, 5, 108, 989, 2, 3, 4, 4, 2, 5, 3, 25, 9, 8, 5, 8, 10, 9, 110, 61, 6, 8, 15, 16, 5, 305, 4, 5, 6, 5, 5, 5, 7, 601, 7, 131, 730, 686, 6, 4, 2, 6, 721, 390, 21, 810, 4, 3, 4, 3, 8, 4, 4, 12, 7, 4, 6, 7, 97, 4, 5, 422, 6, 6, 3, 253, 193, 116, 17, 72, 8, 4, 5, 5, 21, 12, 5, 9, 6)

3. It is not necessary to add the bias dimension as the sklearn regression model conveniently adds it automatically.

4. My RMSLE is 0.152.

5. These fields make sense. Some of the positive traits include some things I would expect to be very positively correlated with price such as the overall quality. The same goes for the most negative traits.

```
Col, Field,  Weight
(48, 'FullBath', 0.13788508271345906)
(16, 'OverallQual', 0.1373861900234647)
(11, 'Neighborhood', 0.12506866127916277)
(43, '2ndFlrSF', 0.11459160439582075)
(16, 'OverallQual', 0.10272474402066556)
(21, 'RoofMatl', 0.09321732652345455)
(45, 'GrLivArea', 0.0909257665599269)
(3, 'LotArea', 0.08730592235726188)
(60, 'GarageCars', 0.08708499414324003)
(11, 'Neighborhood', 0.08413985812963574)
```

```
Col, Field,  Weight
(1, 'MSZoning', -0.19636108259923807)
(45, 'GrLivArea', -0.12847483017916952)
(67, 'EnclosedPorch', -0.12121140550155647)
(16, 'OverallQual', -0.11986660467818158)
(35, 'BsmtFinSF2', -0.10718677376491481)
(3, 'LotArea', -0.10718677376491481)
(17, 'OverallCond', -0.10106071888960845)
(60, 'GarageCars', -0.09245671790571539)
(16, 'OverallQual', -0.08970765563015565)
(53, 'TotRmsAbvGrd', -0.0882450202459617)
```

6. My weight for the feature bias is 12.18. This seems fairly reasonable as it indicates that the starting price is skewed from the origin by approximately $162,000 which seems like the price of a cheap house.

7. My score was 0.15740 placing me at rank 2624.

3 Smarter binarization: Only binarizing categorical features

1. The first drawback is that some of the numerical fields are directly related to the sale price (OverallQual, OverallCond, YearBuilt, LotArea, etc..). Binarizing these features abstracts that correlation into one hot encoding, basically rendering those natural correlations useless. The second drawback is that binarizing these features will greatly increase the number of sparse features (mostly zero valued features), adding little to the quality of the model, as well as increasing computation time.

2. For the mixed features having a NA instead of a numeric value typically indicated the lack of that feature. I filled those with zeroed values because they should all be positively correlated to the price and making them zero represents their unfavorable condition.

3. After using the ”smarter” binarization my RSMLE on the dev set decreased to 0.129. My test score decreased to 0.151 which is better than before, but not as much as I expected. This gave me a ranking of 2930. As for the other questions previous questions, here are the lowest and highest coefficients:

```
Col, Field,  Weight
(21, 'RoofMatl', 0.6292490760793249)
(71, 'PoolQC', 0.5020554986386175)
(13, 'Condition2', 0.474109221209482)
(21, 'RoofMatl', 0.47157404687538895)
(20, 'RoofStyle', 0.4055044860568987)
(15, 'HouseStyle', 0.3526393782815829)
(73, 'MiscFeature', 0.29041084984126164)
(62, 'GarageQual', 0.2841413783940419)
(21, 'RoofMatl', 0.2671754136021055)
(21, 'RoofMatl', 0.2615517049253798)

(21, 'RoofMatl', -2.2546290195790726)
(13, 'Condition2', -0.7045319645939263)
(13, 'Condition2', -0.6286592789770766)
(0, 'MSSubClass', -0.36600110417649745)
(1, 'MSZoning', -0.34686490138145565)
(71, 'PoolQC', -0.2750790103593811)
(22, 'Exterior1st', -0.2364509899781388)
(54, 'Functional', -0.21584779871935383)
```

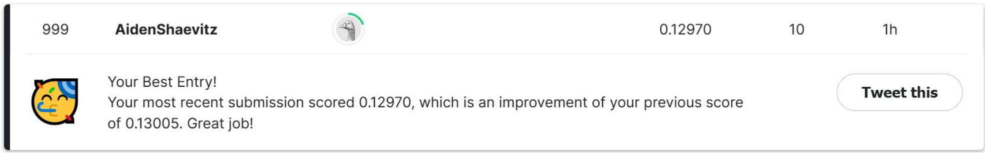
```
(54, 'Functional', -0.19835789118754946)
(63, 'GarageCond', -0.1978965303637472)
```

My bias term was 10.47.

4 Experimentation

1. Using regularized linear regression I was able to get a dev score of 0.125, a test score of 0.13005, and a ranking of 1022. My best alpha across a pretty large range was somewhere around an alpha of 20.
2. I played around with changing the scaling of several terms by making them quadratic and I found there to be little improvement in terms of the dev set except for changing the LotArea and the OverallQuality. My dev score was 0.1253 and my test score was 0.13259. This put me at ranking 1022. I then tried a logarithmic term of the LotArea and found that this helped my score a pretty good amount. This brought my dev score down to 0.123, but my test score remained higher than my best at 0.13184.
3. This feature doesn't really make sense because it encourages large differences in remodeling and year build. Really old homes that have been remodeled recently are not at all guaranteed to be worth more than a newer home that has only had a recent remodeling. I ended up trying adding a difference term between the YearBuilt and YearSold, as well as a multiplicative term between OverallQual and overallCond. This gave me a dev set score of 0.1252 which is the same as before. My test score did not improve, nor did my rank. I believe that it is possible that there are simply too many other features for the couple I added to carry enough influence to change the end result. The features I chose to combine may also have been fairly useless to combine.
4. Maybe one relationship that can be drawn between the two is that in adding non-linear features there is the addition of dimensionality that could better capture the nonlinearities present in the features.
5. I went through the various different regression models provided through the sklearn package and found that nothing outperformed the Ridge regression model.

   My best dev error was 0.123, my best test error was 0.12970, and my best rank was 999. This was achieved with the ridge regression model, a nonlinear log(lotArea) feature, and doing the feature combinations mentioned above.



999 AidenShaevitz 0.12970 10 1h

Your Best Entry!
Your most recent submission scored 0.12970, which is an improvement of your previous score of 0.13005. Great job!

Debriefing

1. I spent maybe 10 hours on the assignment (1 of those were report writing and formatting).
2. I'd say that this assignment felt pretty easy, although I definitely could have gone a lot further exploring methods to try and improve my score.
3. I worked on it completely alone.
4. I would say that I feel 90% comfortable with the material (however I say this very conservatively knowing that it's hard to know exactly what you know and what you don't).
5. No additional comments.