Ben Buzzee, Thanh Nguyen, Kellie McClernon
Team name: NotTooDeep_Learning
May 1, 2017

Current placement: 253

Kaggle Placement

Week	Model	CV Error	Test Error	Placement
May 1	Ensemble (PLS and xgboost)	-	0.11756	253
April 24	Ensemble (PLS and Lasso)	-	0.1198	478
April 10	Ensemble	0.116	0.12168	673
April 3	GBM	0.1307	0.13066	1161

Best Fitted Models

Model	CV Error	Test Error
Ensemble (simple average of PLS and Lasso)	-	.119
PLS	0.114	0.120
Lasso	0.112	0.120
Several models as features in GBM	0.111	0.125

April 24th

We finished data clean up early this week, dealing with NAs and removing features. We removed features that had little variation in the data, thus they did not add information about the sale price since they were the same for most of the houses. We want features that will allow us to parse the cases into distinct groups, but not so many groups that we run into the curse of dimensionality and have sparsity. With little or no variation in the predictor value for all cases, the predictor will be unlikely to contribute meaningfully to our final predictions.

Once the data was cleaned, we fit the same models that preformed well before the data clean - such as GBM, PLS, and Lasso. We noticed that just by handling the data processing well, we saW improvements in our CV errors by 0.01. Our CV errors are slightly better than our Kaggle test erros. Since we used `caret` to fit the model, we are not sure if this is because `caret` is training the model then on only 90% of the data and not the entire data set, resulting in poorer predictions. Additionally, our CV score could be too optimistic if part of our data clean process is actually leveraging the entire data before we do our CV splits. We plan to next fit xgboost to see how that perfoms.

We have been trying to fit a simple neural network. However, even though the CV errors appear good, the test errors are quite high. This suggests to us that the neural network is over-fitting on the training data and unable to be flexible enough to give good predictions on our hold-out test data.

April 17th

This week we focused on data processing. As a starting place, we read over the suggestions made by Tanner Carbonati on the forums. From the thread, we learned about converting ordinal categories, such as the quality and condition variables, to numerical ordinal values, i.e. 1 - 5. This seems to have value because it will help for models that depend on numerical inputs only while still maintaining the information in the original data set. Additionally methods such as random forest

can be improved by this categorization because then if the tree splits on these variables it is seeking the optimal cut-off point between low and high scores which intuitively seems desirable. Also, some categories, such as Lot Contour, are skewed because most cases fall into the category of "Level", but then "Non-Level" is further divided into "Banked", "Hill", and "Low". If we simplify the category to level/non-level then the groups are more even and we have more cases for the non-level category. Since the data set is not very deep, having categories with too much granularity can reduce our prediction power by increasing the curse of dimensionality.

We have tried to fit linear models and they individually have not been very successful. Therefore, we suspect that the response is not linearly related with predictors. So next we are planning to fit a non-linear model using neural networks.

April 10th

Our goal for this week was dimension reduction. To accomplish this, we performed a PCA to see if we could identify any predictors with high eigenvalues or very low values. However, most values were very high and fairly close to each other. We suspect this is because of the high correlation among our variables. A possible next step is to look into a cluster analysis of our predictors since clustering highly correlated variables could help us to reduce the number of predictors.

We also tried Lasso as a method of predictor selection by looking for near-zero coefficients. Originally the data set contains 80 predictors, many of which are related. We are convinced that not all predictors are necessary and if we reduce the "noisy" predictors our method of choice will be better able to select those that truly influence the response. Additionally, we would like to consider interactions, but interactions on 80 variables quickly explodes and our dataset is not very deep.

We had considered linear models but given the large number of categorical predictors we think that tree based models might have the most potential. We are noticing that so far, without much feature engineering, all our models have roughly the same CV, around 0.13. We need to find a way to improve our feature selection in order to get at least one standalone model with error rate closer to 0.10. We know we want to ensemble our models but if all the models we ensemble have an similar error rate, the ensembling does not give us a very significant lift. In fact, our first ensemble was with three models, all of which had similar error rates of 0.14 to 0.13, and combining them got us a CV error rate of about 0.12, which is not a very large improvement. We are currently considering group lasso as a way to deal with natural groupings of predictors in the data set.

For next week, we are planning to explore the predictors with near zero variance, i.e. columns that have essentially the same category for all fields in our training, and converting the sale month to season. Neighborhood has a lot of categories and thus is being over-emphasized in tree models. We would like the reduce the number of categories for this predictor; we are considering doing this by mean sale price.

April 3rd

This week our team focused on exploring and cleaning the Ames Housing data. We found that most of the NAs present had known values mentioned in the data description document provided by Kaggle. After replacing the NAs with known values, we still had a handful of missing values to deal with. To allow us to fit models we replaced those missing values with medians for quantitative variables or modes for categorical. More work will be needed to ensure replaced missing values are well chosen.

After fitting a handful of individual models, we decided to submit our best individual model to

provide us with a baseline RMSE we can strive to improve. The generalized boosted regression model ('gbm' in caret) provided the best repeated cross-validation RMSE at .1307 (for log transformed saleprice). Our submission results were surprisingly similar with a RMSE of .13066. Our current rank is 1161 out of 2245 teams. Now that we have a "clean" dataset, we can now focus on dimension reduction, feature engineering, and combining models.