Predicting House Prices

Luca Parolo

L.Parolo@student.ru.nl

April 2018

1 Abstract

The house is one of the most important asset for families. For this reason is important to calculate the right price before engaging in a real estate purchase.In this paper we will describe four different approaches we used to tackle the Kaggle challenge of predicting house prices in Ames,Iowa,USA. The first approach is a Random Forest (RF) which has given a test error (RMSE) of 0.15924, the second is a Gradient Boosting Regressor with 0.1211,the third is a Elastic Net with an error of 0.11980. Then we will describe the stacking method we built from these three base models which achieved an error of 0.12 for a weighed averaged model and 0.1225 for a Ridge Regressor as meta-learner.

2 Introduction

One of the most important financial decision in the life of an average American family is to buy a house. Therefore is very important to be able to understand the price of the house in relation to the real estate market. The main goal of this paper is to solve this regression problems using different approaches. The problem and the dataset is taken from a Kaggle competition [1] where this problem and dataset is used for didactic purpose as an introduction to advanced regression techniques. The dataset consists of 2919 entries, 1460 of which as train set, and 1459 as test set, with 79 attributes each (23 nominal, 23 ordinal, 14 discrete, and 20 continuous). The performance measure used in this competition is the root-mean-squared-error(RMSE), where $RMSE = \sqrt{\sum^{n} \frac{(y_{pred} - y_{true})^2}{N}}$

3 Exploration, pre-processing and feature engineering

We did a preliminary exploration of the data set to understand how the variables were structured, if there were any missing value or outliers we could work on or any features we needed to modify. To gain a quick overview of the most important features, we computed the correlation matrix (Figure 1), which highlighted, not sur-

prisingly , how the "Overall Quality" and some other attributes all related to the dimensions of the house ,appeared to be the best predictors.

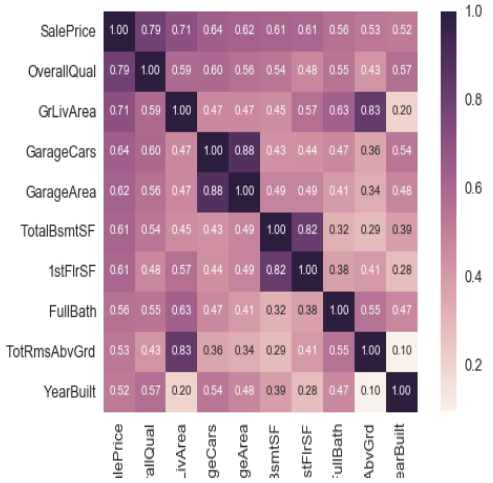

Figure 1: Most correlated attributes in the dataset

In the exploration phase of the dataset, it was possible to identify room to improve its quality and robustness. As some of the variables seemed skewed, we have proceed to take the logarithm to make the distribution symmetric; this helps, when calculating the error, to adjust the relative weight of each sample in the sum of the errors. I will briefly describe three aspects to take in to account in this specific case (as well as in general) in order to fit the models with an optimized training set: missing values, outliers and dummy variables.

3.1 Missing values

While exploring the dataset we identify many missing values (NaN). In total there were 6965 missing values in the train set over 18 variables and 7080 in the test set over 33 variables. This may seem a lot but if we take into account which variables scored the highest number of NaNs it seems reasonable. In fact most of the missing values refers to variables which are rare in average houses, such as the presence of the swimming pool or the fireplace. There are mainly three ways in which we can proceed: delete the column with the missing values, impute the missing values from the mean or mode of the attribute or leave it as it is. In our case we have proceeded to change the missing values of the variables where the NaN meant that the feature was not present, and to impute the values of the remaining NaNs with the mode, for non numeric variables, or the mean for the numeric ones.

3.2 Outliers

There were few outliers in the dataset and we proceeded to treat them as follow. In the case that the outlier was a clear typo we fixed it with what seemed the right data entry (for example a garage built in year 2208 was replace with 2008). When using boxplots to visualize outliers there were values over different attributes which could have been formally classified as outliers. However, they seemed to follow the linearity of the correlation with the Sale Price(the target variable). Therefore we proceeded to delete the datapoint only when the outlier value did not follow the linearity.

3.3 Augmentation

Data augmentation can often make the difference when trying to minimize the prediction error. In our case we built an extra feature (Total Area) adding the total square feet of the basement and of the living area above ground. We also made dummy variables from the categorical variables in order to be able to fit the regression models. This increased exponentially the number of features, but as the training set had only 1458 entries , it seemed still reasonable.

4 Approaches

In this section we describe the different approaches we used for the regression model.

1- Gradient Boosting Regressor

2- Random Forest Regressor

3- Elastic Net

4- Stacked regression

The first three are the base models we used to train the meta learner in the stacked regression. We have chosen those models as base models as they learn in different ways and this is exactly what we are interested in. The main strength of ensemble methods lies in fact in the ability to combine different base models that have learnt a specific feature of the training set.

4.1 Base models

4.1.1 Gradient Boosting Regressor

Boosting is a technique used to build a strong learner from multiples weak learners(that have a high bias , but a low variance), where a weak learner is defined as a model whose probability of outputting the right prediction is just above chance with an edge of γ. These in practice are usually decision trees with a branching factor of 2 and a max-depth of only 1 (and this is why they are called decision stumps rather than trees). The weak learners are integrated weighing their performance: the weak learner that performs worse than the others, will count less when integrated in the strong learner. Gradient boosting was initially proposed by Leo Breiman that observed that a differentiable loss function could have been used when miniminizing the error: this meant that the algorithm became an optimizaable problem that can could be solved using gradient descent.Therefore Gradient Boosting minimizes the risk

$R(f) = E_{p(x,y)}[L(f(x), y)]$

where L is the loss and is defined as:

$L(F(x), y) = (f(x) - y)^2$

4.1.2 Random Forest Regressor

Random forest also is an ensemble method and uses decision trees, building a "forest" of "weak learners" from them. However the random forest algorithm uses fully grown decisions trees. Each decision tree has, contrary to the gradient boosting re-

gressor, low bias and high variance.The trees are also grown to be as uncorrelated as possible from each other to maximally decrease variance.For our model we used a max-depth of 50.

4.1.3 Elastic Net

Elastic net is a regularized linear regression method originally proposed by Zou,Hui and Hastie [4].According to them "elastic net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together" .Even if it best suited when the dataset has a lot of dimensions and a few examples, it is often preferred over the Lasso (least absolute shrinkage and selection operator) method and over the Ridge regression as it usually overcomes the limitations of both, shape of the dataset aside. In fact it combines the "penalties" that the Lasso and Ridge use,which are respectively the L1 and L2, and this had proven to outperform the other methods in most cases.

4.2 Stacked regression

Stacking regression, or stacked generalization , was first proposed by Wolpert in 1992 [2] and can be defined as " a method for forming linear combinations of different predictors to give improved prediction accuracy"[3] . A very simply way of combining different models is to average the prediction of each model. A little more sophisticated way is to use a 2 level learning process(Figure 2). On the first level, each different models is fitted with the training data.On the second level another model learns from the prediction of the first level and combines the different base models. This means that a meta-learner is trained using the prediction of n different base learners and the final prediction (and error) is computed using the meta-learner.

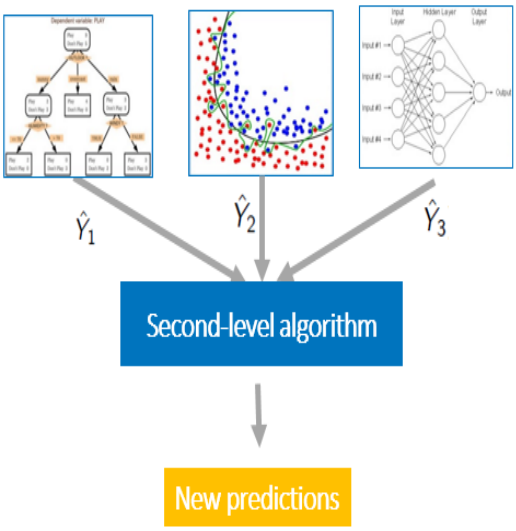

Figure 2: Stacking regression architecture.Each $\hat{Y_n}$ is the output of a model [5]

The rationale beyond this is that each base learner can make good predictions on some datapoint ,while performs poorly on some other kind of datapoint (which may have specific features) , and another overlapping model can be a good predictor on the datapoint where the first learner failed. The most challenging part therefore, is to find models that can "overlap" to each other, and that have good prediction where other performs poorly. For our meta learner we have used the three based models we described. We have first tried

just averaging the models and assigning a weight that informally followed the score of each predictor (0.4 for the Elastic Net, 0.4 for the GBoost and 0.2 for the Random Forest). Then we used Ridge Regressor as meta learner that was trained on the prediction of the base models.As we were not too satisfied woth the results we obtained so far for the meta-model, we tried first to train the meta model only on the EN adn GB which performed much better than the RFF. Then we tried using another EN as meta learner instead of the Ridge.

5 Results

The results were more or less as we were expecting (Figure 3). We first submitted on the Kaggle website the predictions of each base model separately. Then we made the stacking model. We tried with different combination of base models and meta learner. However , as we can see from the table, the best result was given by the simple Elastic Net model. As we see,just weighting the predictions of each base models, gave a fair result, just a little worse than the Elastic Net.

We expected however that the stacking model was going to score better than the other base models and this did not happen. This was probably due, on one hand, by the fact that the same weigh as the EN and Gboost predictions was given to the RFF predictions, that were the much less accurate compared to the other two base models. On the other hand, when training the base models we used only half of the entries as we used holdout, and this probably hampered the learning process, while the base models were trained on the whole training set. Overall, without too much hassle, only cleaning the dataset, making dummy variables and correcting the skewdness, together with a strong linear regressor, seems to be already enough to reach a decent score (we reached the 797^{th} position with the Elastic Net over 5,252 entries of the challenge).

Model	Score (RMSE)
Gboost	0.1211
RFF	0.1592
EN	0.1198
Weighted Avg	0.1200
Stacking/Ridge	0.1225
Stacking/EN	0.1224

Figure 3: RMSE for each model on the test set (computed by Kaggle). RFF is RandomForest EN is Elastic Net

6 Future directions

Even though we are fairly satisfied with the final score, we believe that there is still room for improvement.

1)The most important improvement could be in training the base models with more samples, also using bootstrapping if necessary.

2) More base models could have been used in the second level learning. We could have also checked what was the

most important predicting variable in each model and use this information to weigh the contribution of each base model. It would have been also interesting to experiment with Deep Neural Networks and Support Vector Machine.

3) Each base model could have been tuned better, especially the random forest, which performed weakly compared to the rest. We could have, for example, used bayesian optimization for the hyperparameters.

4) We could have built an extra feature based on the average of predictions of the base models to be add to the training set and train the meta learner with this augmented dataset.

5)We could have checked for skewdness of other variables and we could have used Box Cox transformation.

6) Multicollinearity between variables could have been reduced. In fact the best predictors are only few variables while most contribute very little.

7 Conclusion

In this project we explored different ways of performing multi variable advanced regression. We have also see how to approach some basic issues that are very frequent in data science. We developed a meta model based on three different base models that could be a good baseline model for future improvements. We also highlighted some possible future direction to develop further the model. We are confident that this, beyond being an excellent didactic exercise, would reach a better placing in the Kaggle competition final rank.

References

[1] *https://www.kaggle.com/c/house-prices-advanced-regression-techniques*

[2] *Wolpert, David H. "Stacked generalization." Neural networks 5.2 (1992): 241-259.*

[3] *Breiman, Leo. "Stacked regressions." Machine learning 24.1 (1996): 49-64.*

[4] *Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67.2 (2005): 301-320.*

[5] *from: "https://blogs.sas.com/content/subconsciousmusings/2017/05/18/ stacked-ensemble-models-win-data -science-competitions/"*