Forecasting Bike Rental Demand

Max Liu

February 9, 2018

1. Introduction

Given a data set from the Ames Assessor's Office, we want to predict home selling prices. This data is useful because it is similar to what a typical home buyer may look at or want to know before making a purchase on a home. We will try to use the features within the data set to create a model that is able to best predict the selling price of a home.

2. Dataset

The data set we are looking at contains 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables. Due to the large amount of nominal and ordinal values, intuitively we would believe that some sort of tree would best be able to handle the decisions for the nominal and ordinal data.

2.1 Data Encoding

Because so many features are either nominal or ordinal, this means we need to encode the categories of these features into numerical values that can be interpreted by our model. For the given data, we have encoded the categorical values by assigning each category within a feature a numerical value. A tree should not have difficulty dealing with this type of encoding because at every step it should be able to decide what to do next based on the numerical value of the category.

An additional issue that we ran into while cleaning and encoding the data is that the training and testing data contain a different number of categorical descriptors within certain features. An example of this is that the descriptor "Roll" within the Roof Material feature appears only in the testing data and not in the training data. We can continue to encode these labels with a new numerical value but this may be an issue for the model when trying to predict the sale price of the testing data.

2.2 Log Transformation

The RMSE for our predictions will be tested on the log of the sale prices given. There are 2 ways to approach this, we could predict the sale price itself and then transform the predicted price into log and compare against the known log transformed sale prices, or we can start off my transforming all of the sale prices to log and predicting the log sale price value. We have found that our models perform much better when we first transform the sale price into log and then try to predict this log value. The log value can then be transformed back into the actual sale price value by exponentiating it.

3. Models

For each of the models we hold out 20% of the training data to use as a testing set after the remaining training set has been used to train the model. This will give us an unbiased review of the model performance. Each model itself is cross validated with 5 folds and the best parameters are used to train the final model that predicts the testing sale prices.

3.1 Tree Regression

We started with a simple tree regression which we expect to have decent but mediocre results with respect to the more advanced models we run later. It was found that the optimal depth for our tree regression was 6 which resulted in an R^2 of 0.73 and a RMSE of 0.204. These results are not too significant but it seems like we are headed in the right direction with the tree based regressions.

3.2 AdaBoost Regression

With the AdaBoost regression we find that the best learning rate is 0.2, the best loss function is exponential, and the best n-estimators is 200. Using these parameters, we find that the R^2 is 0.808 and the RMSE is 0.172.

3.3 Random Forest Regression

We expect the random forest regression to perform much better than the tree regression as the random forest should be able to better distinguish between noisy and useful features as well as have a better method of deciding which branches to traverse. With the random forest regression we find that the best n-estimators is 600 and it is best to use the square root of our total n-features as the predictors used in the tree. With this we achieve a R^2 of 0.865 and a RMSE of 0.144.

3.4 Support Vector Regression

Surprisingly, the results from our support vector regression are quite poor. For the kernel type we used a rbf kernel, this is primarily because it was extremely computationally expensive to test the poly and linear kernels and the run time would have been unreasonable (after multiple hours the model was still not done training). We found the optimal C and epsilon were 0.25 and 0.1 respectively. Using these parameters we find almost no correlation between our data with the sale price. The R^2 was 0.001 and the RMSE was 0.392. Even when we increased the value of C to greater than 1000, we continue to get very poor results. This inaccuracy is most likely due to the random label encoding of the features. It would probably be extremely difficult for the support vector machine to distinguish between these effectively.

4. Conclusion

Model	MSE	RMSE	R^2
Tree	0.042	0.204	0.730
AdaBoost	0.029	0.172	0.808
Random Forest	0.021	0.144	0.865
Support Vector	0.154	0.392	0.001

As we can observe, the best model for this problem is a random forest regression. We then run the random forest regression over all of our training data and find that the best parameters are 600 n-estimators, and uses square root n-features. These parameters were then used to predict the log price of each home. The final csv file with the sale price predictions are transformed back from the log value into the actual sale value.