Evaluating Regression Models

Summary

Understanding the value of machine learning alone, has given your firm an advantage over those who are still skeptical or do not have the resources to implement it. While using traditional conventional methods for assessing housing markets has worked well in the past, incorporating machine learning methods has proven to give a competitive advantage over its competitors. With the influx of data being gathered, it is not effective to use a "one model fits all" kind of approach. With data changing continuously and new parameters being added to define a target variable, the best approach it to let the machine decide which model works best in the most effective way.

Research Design

In this study we will be using the data from the Boston Housing study to determine which regression model would best help us determine the median value of homes. The four models used were, Linear, Ridge, Lasso and Elastic. The main difference between linear and the later models are that the linear models are not regularized, which is typically achieved by adding weights. The whole purpose of regularization is to prevent overfitting if the model is to complex.

Data Analysis

The data obtained was well structured and relatively clean. Considering we are doing a regression analysis, the neighborhood field had to be dropped as it contained categorical values. The only other alteration to the data was converting the tax and rad fields from integer to float. The chas field was left as an integer as it was a binary field.

Regularized models can be sensitive to the scale of input features, therefore it was important that we scaled the data. Results of this can be seen in *Appendix A – Figure 1*

Model Evaluation

In order to differentiate the performance of the regressions models we had to understand the cost function. As the title implies, the cost function is basically how costly the predicted value was, or in other words, what was the error between the predicted value and the actual value. When it comes to regression models it is understood that the best way to determine this error is by calculating the root mean square or the R-squared value, also known as the coefficient gradient. In this study we would have to get the mean of 506 different values as we implemented a “leave one out” cross validation, which is where the number of folds used is equal to the number of observations. The shape of the input data for a fold is shown in Figure 2 in Appendix A.

Figure 3 in Appendix A shows the root mean squared error results obtained for the different models.

The lower the error the better the model, and in this case it seems the Ridge Regressions model best fits the data with an error value of 0.3730. There are twelve different parameters to work with therefore it seems that regularization is a good option to simplify the model and avoid overfitting.

Recommendation

Although in this study Ridge Regression seemed to be the best model for machine learning, when possible we would want to use all the variables and data in hand. Ideally we would want to understand how the neighborhood variable factors the median home value as well.

If we wanted to stick to regression models, we could possibly create dummy values for the different neighborhoods or create a hybrid model that combines both classification and regression models.

Appendix A

```
StandardScaler(copy=True, with_mean=True, with_std=True)
[  22.52885375    3.61352356   11.36363636 ...,  408.23715415   18.4555336
   12.65306324]
[   9.1730981     8.59304135   23.29939569 ...,  168.37049504    2.16280519
    7.13400164]
```

Figure 1 – Scaled Data using StandardScaler

```
Fold index: 497 ----------------------------------------

Shape of input data for this fold:
Data Set: (Observations, Variables)
X_train: (505, 12)
X_test: (1, 12)
y_train: (505,)
y_test: (1,)
```

Figure 2 - Fold Index

```
Average results from 506-fold cross-validation
in standardized units (mean 0, standard deviation 1)

Method               Root mean-squared error
Linear_Regression        0.373554
Ridge_Regression         0.373057
Lasso_Regression         0.405951
ElasticNet_Regression    0.392428
dtype: float64
```

Figure 3 – Root mean squared error