
Kaggle Predictive Model Report

STAT361

Ryan Middleton & Umar Khan

December 8th, 2024

ABSTRACT

This report will analyze the “Boston” housing dataset. The model seeks to predict the median value of owner-occupied homes in \$1000s (MEDV). The best predictive model we found combines a nonlinear Lasso regression and a random forest, averaging their predictions. This model outperforms stepwise AIC regression and nonlinear ridge regression, achieving a lower root mean square prediction error of 2.395.

INTRODUCTION

The Kaggle dataset “Boston” consists of 13 predictors listed below.

Title	Description
CRIM	Per-capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25'000 sq. ft.
INDUS	Proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centres
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per \$10,000
PTRATIO	Pupil-teacher ratio by town
B	$1000(B_k - 0.63)^2$ where B_k is the proportion of blacks by town
LSTAT	Percentage lower status of the population

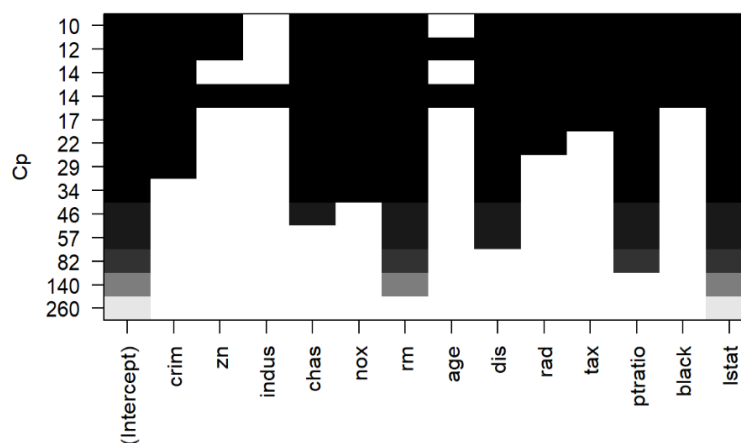
The goal of this project is to enhance the prediction model so that it performs better than the trivial linear model involving all the predictors. This will allow us to accurately predict the median house value (MEDV). The accuracy of the model is evaluated based on the metric root mean squared prediction error (RMSPE). We seek to have the lowest RMSPE possible, which implies that we have little error in our predictions. We will explore three models that are both simple and can optimize this metric.

DATA ANALYSIS

Before implementing our models, we will analyze the data to extract the necessary preliminary information we need to create successful models. This step is essential for understanding the relationships within the data, evaluating the importance of each predictor, and identifying patterns that may influence the model's structure. Specifically, we can determine which variables to include in the model, and an appropriate order that represents the data. We can tailor our models accordingly given the data analysis.

Maslow's Cp Statistic

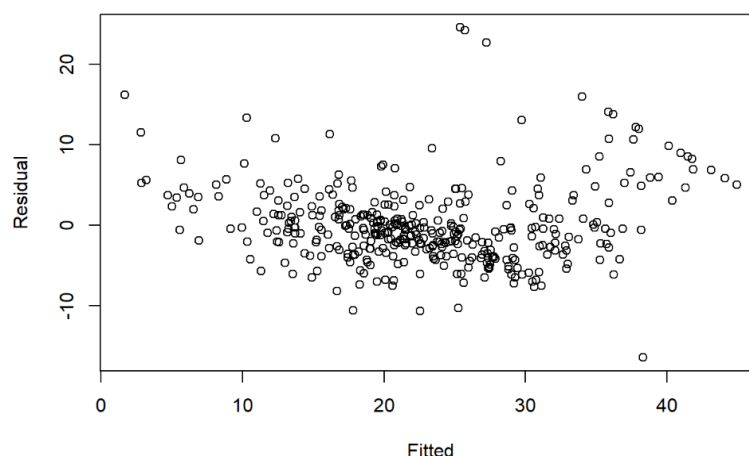
First, we will look at the Maslow's Cp Statistic which compares the precision and bias of the full model to models with a subset of the predictors.



This graph displays varying degrees of importance for each predictor. The more black squares are a general indicator of importance. LSTAT and RM emerge as the strongest predictors. Meanwhile, INDUS, AGE, and ZN are among the weakest since they contain a lot more white squares than black. This means that they may not have a significant impact on the data and can be removed.

Fitted Versus Residual

Next, we will analyze the Fitted vs. Residual graph. This plot will help us identify non-linearity, unequal error variances, and outliers.



The Fitted vs. Residual graph shows a decent fit and reveals a few key insights. The plot shows a slight funnel shape where we see the residuals being more spread out as the fitted values increase. In fact, between the fitted values 10 and 30, the residuals tend to be more tightly clustered. Similarly, there is a set of points with residual values above 20. These appear to be potential outliers that may be disproportionately influencing the model. This information suggests that our model will likely need to be nonlinear considering the plot forms a slight quadratic curve, and the presence of heteroscedasticity.

Influence Statistics

We will use ‘influence.measures(model)’ which goes through the training data and gives multiple influence measurements and an indication of which indexes are significant.

Output:

```
[(*) : 94-98, 109, 110, 112, 115, 118, 119, 122, 135, 157, 190, 270, 274, 275,
      277-282, 287, 309, 314, 316, 320, 372-374 ]
```

This information gives us the indexes of the influential points which were indicated to exist in the previous test. These can be removed when needed for a better fit, and more accurate training data.

Summary Information

We will use ‘summary(model)’ which will tell us key information about each predictor and the performance of the linear model.

```
##
## Call:
## lm(formula = medv ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4449  -2.9231  -0.6063   2.2156  24.5896
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.148911    6.210430   6.626 1.21e-10 ***
## crim        -0.132999    0.036329  -3.661 0.000288 ***
## zn           0.039915    0.016965   2.353 0.019154 *
## indus        0.012679    0.074582   0.170 0.865107
## chas         2.967850    1.080076   2.748 0.006291 **
## nox         -16.362308    5.058274  -3.235 0.001326 **
## rm           3.533316    0.490719   7.200 3.35e-12 ***
## age          0.003979    0.015979   0.249 0.803478
## dis         -1.469930    0.239584  -6.135 2.18e-09 ***
## rad          0.348479    0.081878   4.256 2.64e-05 ***
## tax         -0.012953    0.004602  -2.815 0.005140 **
## ptratio     -1.074000    0.163417  -6.572 1.68e-10 ***
## black        0.007668    0.003256   2.355 0.019021 *
## lstat       -0.586815    0.060211  -9.746 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.032 on 372 degrees of freedom
## Multiple R-squared:  0.7328, Adjusted R-squared:  0.7235
## F-statistic: 78.49 on 13 and 372 DF, p-value: < 2.2e-16
```

The result shows that 11 out of the 13 predictors have p-values less than 0.05, indicating that they are significant to the response variable, MEDV. Specifically, many variables such as RM, LSTAT, and PRATIO have p-values less than 0.001, proving they are highly significant predictors. In contrast, as seen before, AGE and INDUS do not contribute anything meaningful to model and should be removed. Finally, the Adjusted R-squared result of 0.7235 tells us that the predictors are strong in explaining the variation in the dependent variable.

PROPOSED PREDICTIVE MODELS

After the preliminary data analysis, we will explore two main ideas when proposing our models. The first is, it may be beneficial to remove insignificant predictors, namely AGE and INDUS, and second, there may need to be a higher order term present, namely a quadratic one.

Stepwise Akaike Information Criterion (AIC)

The first model we will try is given by the Stepwise AIC approach.

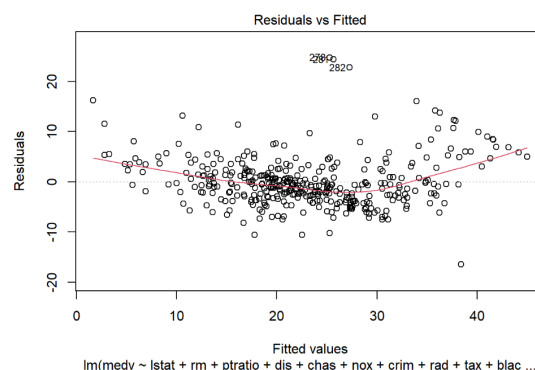
$$AIC(p + 1) = n\ln(SS(Res)_p) + 2(p + 1) - n\ln(n)$$

Stepwise AIC helps balance accuracy with simplicity, removing and adding predictors using the AIC to measure a predictor's importance to the model. At the very least, the criterion tells us which predictors to remove from the model. Additionally, we will explore whether a higher order term is necessary for the data.

We found the model with the code:

```
sfit <- step(null, scope = list(lower = null, upper = full), direction = 'both')
```

This gave the model that includes all the predictors except for AGE and INDUS (as expected).



This model didn't provide a satisfactory fit. We had a root mean squared prediction error of 3.88528 which is worse than the full model. Evidently we needed a much more robust model to make meaningful improvement. In hindsight, AIC works to simplify the model, not actually

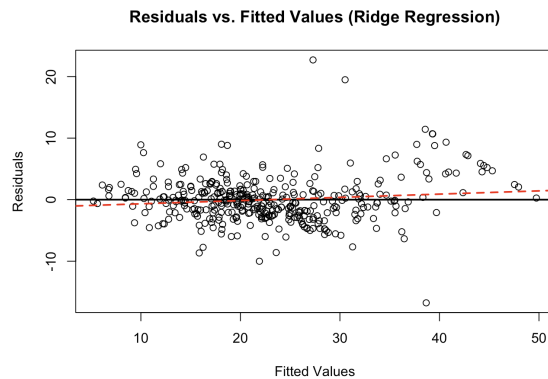
improve upon the fit, so this should be an obvious result. However this test still is useful for further identifying strong and weak predictors. For our future models, we now know for certain that a polynomial regression model is needed.

Polynomial Ridge Regression

Next, we explored a nonlinear ridge regression.

$$\hat{\beta}_{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j)^2$$
$$\text{subject to } \sum_{j=1}^p \beta_j^2 \leq t$$

Ridge regression shrinks the regression coefficients by imposing a penalty on the size. This penalty discourages large coefficients which improves generalization to new data, which is perfect for the goal of this project. However, this method shrinks the coefficients towards zero, but rarely sets them exactly to zero. We will explore the effect that not actually completely removing AGE and INDUS has on the model (as suggested by the previous one).



We see that this is a notable improvement from the full linear model, and the AIC model. The residuals spread is relatively consistent and are randomly scattered around zero as fitted values increase. This suggests that a polynomial regression captures the data well. Specifically, using a quadratic term is optimal.

However, this model has an RMSPE of 3.18537. This score is an improvement from the AIC model indicating that using a quadratic regression is a positive choice. However, this score suggests that this model is also not very accurate in its forecasts. Given that ridge regression is one of the strongest modelling techniques, the low performance suggests we may want to consider implementing a machine learning approach for more accurate predictions.

Polynomial Lasso Regression & Random Forest

Learning from the previous model attempts, we realize that we must consider a polynomial (quadratic) model stronger than one that just removes the insignificant predictors. The candidate we found that satisfies this is an average between two models, one being a nonlinear Lasso regression, and the other, a random forest.

Lasso regression works in a similar fashion to ridge regression, but can shrink some coefficients all the way to zero, practically allowing for selection of predictors. This means that we will accomplish the removal of the unnecessary predictors just as we outlined in our data analysis. This also fits our use case since we have a large number of predictors.

$$\hat{\beta}^{\text{Lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2$$
$$\text{subject to } \sum_{j=1}^p |\beta_j| \leq t$$

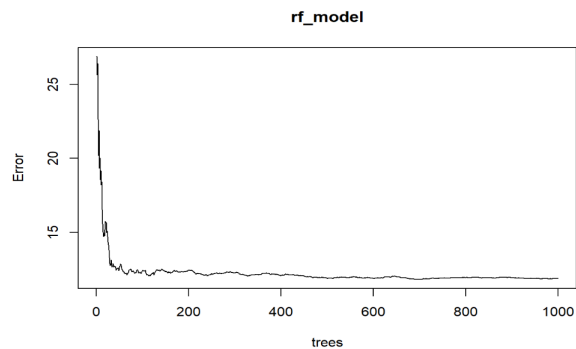
A random forest is a strong machine learning algorithm which employs multiple decision trees to make accurate predictions. The general idea is to create a “forest” of decision trees which are each given a different random subset of the data. These subsets are sampled with replacement to ensure only moderate and controlled differences between trees. In the summarization step, the predictions from all the trees are combined to produce a strong final result. Using this method in our model is suitable because the relationship between predictors and MEDV is complex and nonlinear, there are potentially irrelevant predictors, and heteroscedasticity and outliers are present given the data.

Although averaging two statistical model approaches is uncommon, we found that this approach worked best for this particular dataset. Combining the results gives us the best of both models by helping remove weaknesses, and achieving the goals we outlined previously. Specifically, it reduces model-specific bias and allows for better generalization in the unseen data. The polynomial Lasso regression captures the relationship we see in the data, while the random forest handles the complex, nonlinear and irregular patterns that the Lasso regression might overlook. By averaging the outputs, we allow for flexibility, minimize overfitting risks, and enhance the robustness of the predictions. This leads to a model that captures the unique characteristics of the dataset.

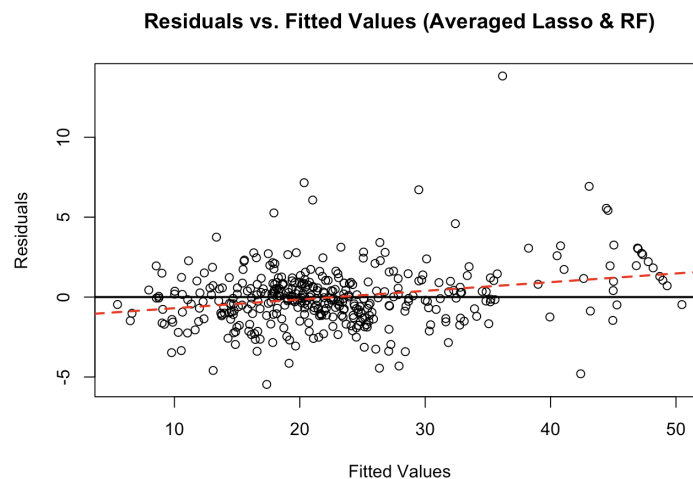
We found the model with the code:

```
lasso_model <- cv.glmnet(as.matrix(x_train_poly), y_train, alpha = 1)
lasso_predictions <- as.vector(predict(lasso_model, s = lasso_model$lambda.min, newx =
as.matrix(x_test_poly)))
```

```
rf_model <- randomForest(x_train, y_train, ntree = 1000, mtry = 4, importance = TRUE)
rf_predictions <- predict(rf_model, newdata = x_test)
final_predictions <- (lasso_predictions + rf_predictions)/2
```



This graph shows the error rate steeply dropping as the number of trees increases. We can conclude we used enough trees by lack of improvement and plateau in the error rate with the addition of more trees once we reach 1000 trees.




This plot shows the Residuals vs. Fitted Values for the averaged Lasso Regression and Random Forest model. It shows a notable improvement from the Ridge regression and AIC plots. Specifically, the model captures the underlying trend of the data without significant errors, the spread of residuals appears relatively constant around zero across various fitted values, and there is a reduced presence of outliers in the plot. Comparing the graphs, this one clearly outperforms the other two, which tells us this model can forecast predictions better than the others.

This aligns with our root mean square prediction error of just 2.395. This is a far better Kaggle score than the scores of our other models. This makes it the preferred choice for accurate predictions and our best model.

CONCLUSION

The best model we found that captures the data and produces a small root mean square prediction error is one that averages a nonlinear Lasso regression and a random forest. Out of the other proposed models, this one performed the best with a RMSPE of 2.395. We can conclude that this model is strong and can accurately predict the median value of owner-occupied homes in \$1000s (MEDV) from the “Boston” dataset.

MODEL	RMSPE
Stepwise AIC	3.88528
Polynomial Ridge Regression	3.18537
Lasso + Random Forest	2.395

	submission.csv Complete · Umar Khan · 9d ago	2.39500	<input type="checkbox"/>
---	--	----------------	--------------------------

References

Devon Lin. STAT 361 Final Project --- Fall 2024.

<https://kaggle.com/competitions/stat-361-final-project-fall-2024>, 2024. Kaggle.

Devon Lin. STAT 361. Queen’s University, Ontario, Canada. September 3 2024 - December 3

2024. Chapters 1-11.

IBM. (2024, October 25). What is Random Forest? <https://www.ibm.com/topics/random-forest>.