Niki Adelaja, Joelle Strom, Bao Van
Final Project
STAT 408
December 10, 2020

Feature Selection in Multivariate Regression

Abstract

Feature selection is an important step in multivariate regression, especially when working with complex datasets featuring a large number of predictors. There are multiple methods to rank and/or select independent variables with higher influence on the dependent variable. We compared four of these methods on the Boston Housing dataset: comparison of multiple models by ANOVA, MARS, Boruta, and random forest. The dependent variable was medv, the median value of the property. The ANOVA results indicated a significant change in the model when variables lstat, b, ptratio, and tax were each dropped, with an accompanying rise in residual sum of squares. MARS, Boruta, and random forest returned similar rankings in variables, with lstat, nox, ptratio, and rm among the most important and zn, b, chas, and rad among the least important. In this case we would drop zn and chas from the model, but future investigations may need to take computational cost and method limitations into account.

Introduction

To explore feature selection the Boston Housing dataset was selected. This dataset is widely available and was obtained from the R dataset package. The dataset was originally collected by the United States Census Services. The focus of the data is housing in the Boston, Massachusetts area. The dataset consists of 506 observations and 14 variables. The explanatory variables are as follows:

1. CRIM - per capita crime rate by town
2. ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS - proportion of non-retail business acres per town.
4. CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. NOX - nitric oxides concentration (parts per 10 million)
6. RM - average number of rooms per dwelling
7. AGE - proportion of owner-occupied units built prior to 1940
8. DIS - weighted distances to five Boston employment centres
9. RAD - index of accessibility to radial highways
10. TAX - full-value property-tax rate per $10,000
11. PTRATIO - pupil-teacher ratio by town
12. B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT - % lower status of the population

Comparison of Multiple Models Using ANOVA

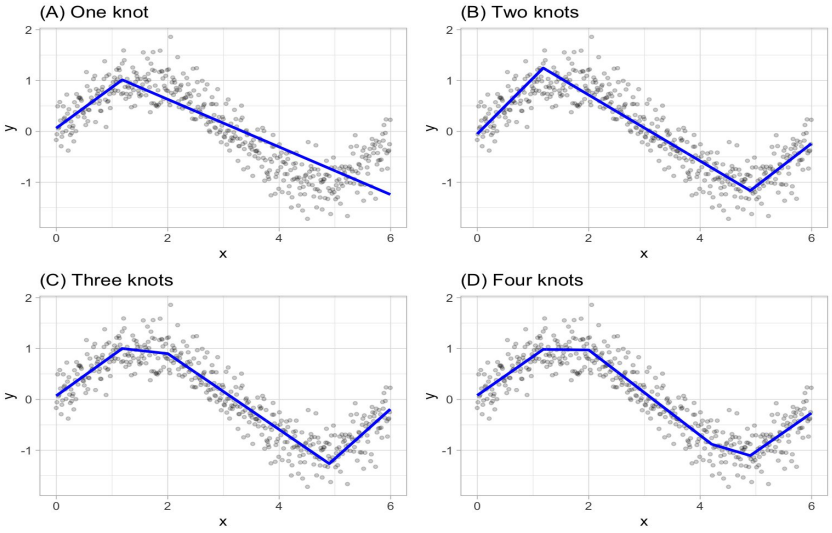
In this method of feature selection multiple models are compared using ANOVA to test whether the additional variables create a better model. To compare the fit of multiple models the anova() function output tests whether the more complex model is better than the model with the

lesser variable. If the p value is statistically significant the more complex model with the additional variable is more favorable (Prabakharan, 2017).To support this method a stepwise regression can be done to be highly selective in the process of discarding variables that do not contribute to the model and build the multiple models on the response variable. The process of stepwise regression is done by adding or removing variables based on the t-test.

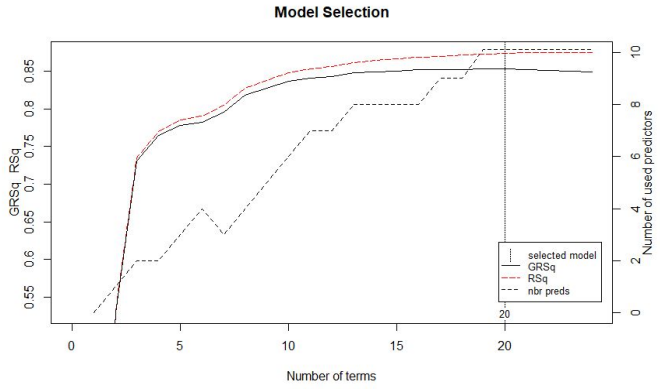
The ANOVA comparison that was done on the multiple models of the boston housing data showed that the model with all of the variables would be the better model given the variables contribution to the model. There were 5 models that were compared with the base model having all variables and the subsequent models dropping lsat, b, pratio, and tax.

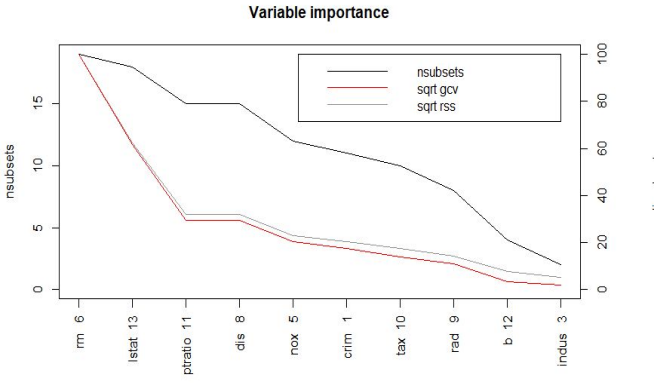
MARS

Multivariate adaptive regression splines (MARS) provide a convenient approach to capture the nonlinearity aspect of polynomial regression by assessing cutpoints (or knots) similar to step function.The algorithm assesses each data point for each predictor as a knot and creates a linear regression model with the candidate features. The MARS procedure will first look for the single point across the range of independent values where two different linear relationships between dependent variable and independent variables achieve the smallest SSE. What results is known as a hinge function. Once the first knot has been found, the search continues for a second knot.



This algorithm can continue until many knots are found, producing a highly non-linear pattern. At the end of the algorithm, once the full set of knots have been created, we can remove knots that do not contribute significantly to predictive accuracy. This process is called pruning (Friedman, 1991).



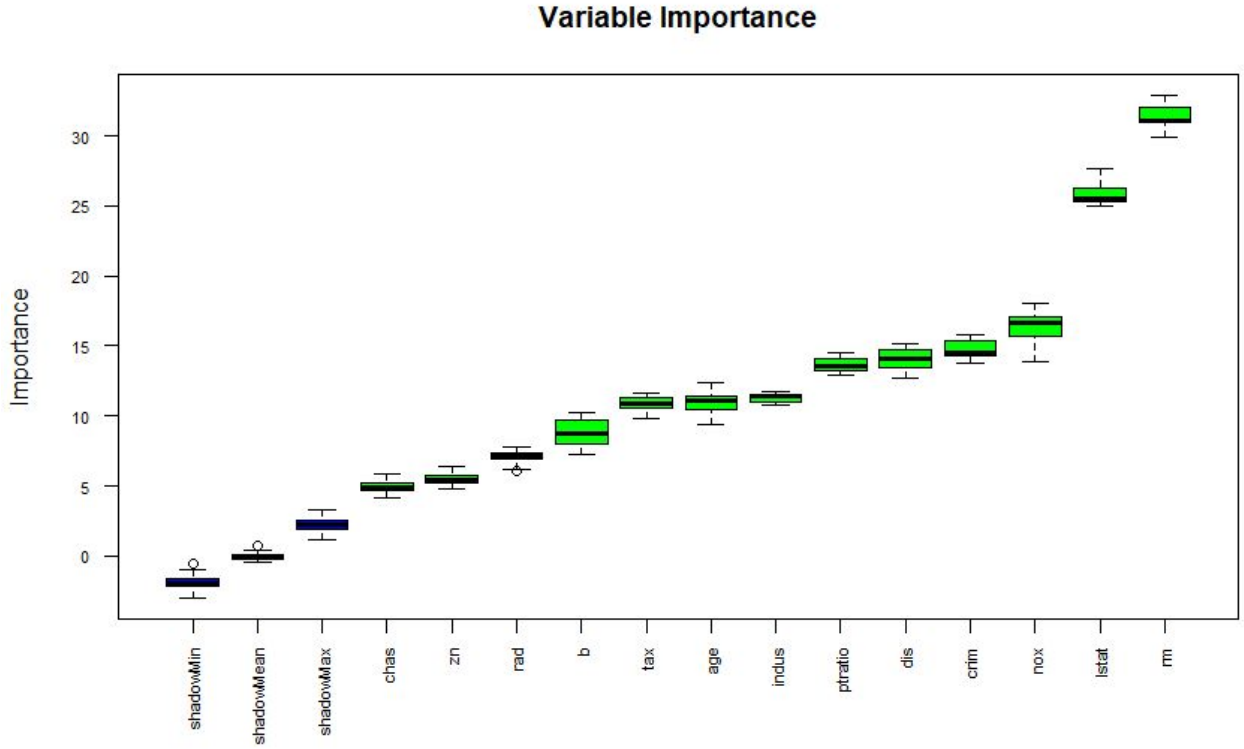


As we can see that the model found there are more than 20 terms by applying MARS onto our dataset. Those terms which are after 20th-term have exceeded our requirements or they do not contribute significantly to predictive accuracy. The result, we have 20 terms of linear lines and 10 variables that are important/significant to analyze. Of these variables, rm, lstat, and ptratio are the most important and rad, b , and indus are the least important. Zn, chas, and age have been dropped from the model because they were deemed insignificant to the model's accuracy.

Boruta

The Boruta algorithm is a wrapper built around a random forest classification algorithm. It tries to capture all the important, interesting features that are necessary for the dependent variables that we want to find. The procedure's process works in this way:

1. It duplicates the dataset, and shuffles the values in each column. These values are called shadow features.
2. It trains a random forest classifier on the extended dataset and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.
3. The algorithm checks for whether each of the real features have higher importance. That is, whether the feature has a higher Z-score than the maximum Z-score of its shadow features. If they do, it records this in a vector. If not, it removes features.
4. The algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs (Kursa & Rudnicki, 2010).



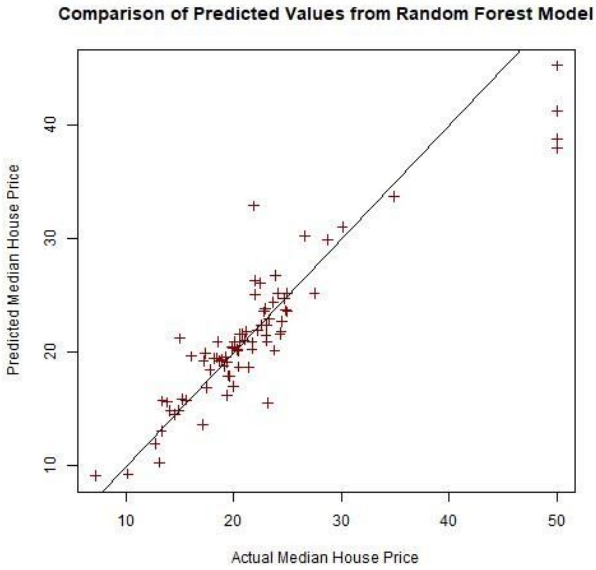
With our dataset, when the Boruta algorithm is applied to find the important variables for the dependent medv variable, the result is shown as the figure above. As we can see they all have a

Z-score higher than the Z-score of its shadow feature. Therefore, by doing Boruta, we accept all variables. Besides that, we can also see that rm is the most important variable and the least important one is chas.

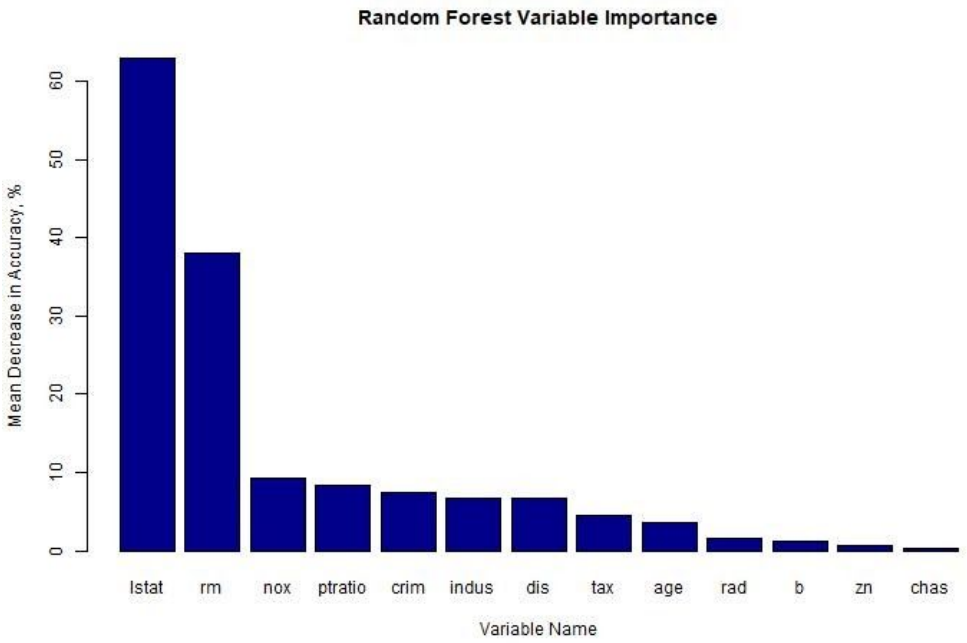
Random Forest

The random forest learning algorithm runs parallel decision trees and consolidates all of them to generate a predicted output. For random forest regression, this means averaging the predicted values of each decision tree to get the overall forest prediction. For each decision tree, a random sample of the dataset is chosen and at each node, a random subset of variables is chosen to evaluate for the split. The algorithm will try each of the variables and select the one that separates the data into groups in a way that minimizes mean squared error (MSE). These splits continue until MSE can not be reduced anymore. At this point the decision tree has numerical outputs for the dependent variable depending on how each predictor splits at each node for a given observation. At the end, the predictors for all the trees are averaged to obtain a random forest prediction.

The model is evaluated by finding the MSE for "out-of-bag" observations. These are the observations that were not included in the random sample for each tree. The function will run out-of-bag observations through their tree, obtain predicted values, then average the predicted values for a given observation across all trees where this observation was out-of-bag. This is $\hat{y}_{OOB}$ for the training dataset the model is based on. To calculate MSE, the formula $MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y}_{OOB})^2$ is used (Liaw & Weiner, 2002). MSE for the random forest model generated with medv as the dependent variable and all other variables as potential predictors was 10.92. To further evaluate the model, a subsection of the original dataset was run through the model to generate predicted values. This test dataset was not part of the initial training set, so the algorithm had never seen this data before. A comparison of predicted values and actual values is shown below. From the MSE and the strong correlation between actual values and predicted values in new data, the random forest model appears to be a good fit.



To evaluate variable importance, the model will calculate the percentage increase of MSE when the predictor is removed from the model. Random forest does not make decisions about which variables are important enough to keep in the model and which should be discarded the same way the MARS and Boruta methods do. The only output from random forest is a list of the variables and their importance, which is shown below.



From this chart, variables lstat and rm are the most important by far, and zn and chas are the least important. This is in line with output from MARS and Boruta. Random forest is a robust algorithm; the random sampling method at each tree and random subset of predictors at each node captures the full range of the dataset and the aggregation of trees prevents overfitting (Chakur, 2019). Because of this, it is more useful to leave all the variables in the model development stage than for a classic regression. The main drawback is that random forest can have a high computational cost, with run times of hours or even days for large datasets. This dataset is small enough that run time is not an issue, so the recommendation would be to leave all the variables in the random forest regression for the Boston Housing dataset.

Conclusion

Four methods of feature selection have been presented, each with similar, though not identical results. MARS, Boruta, and random forest all provide output with variables ranked by importance. The top variables common among these methods are: lstat, rm, nox, and ptratio. The least important variables common among the methods are: zn, chas, b, and rad. Between all three methods, there were slight variations in the order of the ranking and the relative importance of one variable compared against the others. However the general ranking was similar enough between the methods that there is no clear recommendation on which method to use based on variable importance ranking alone.

There are other considerations when performing variable selection. MARS and Boruta each have a criteria by which variables are either deemed important enough to keep in a model or unimportant and fit to be discarded. Random forest does not have this feature, so it should not be used for this type of decision. The MARS algorithm generated a piecewise regression model without variables zn and chas. Boruta does not generate a regression model but it does output variables deemed important enough to keep in further analysis, in this case all variables were deemed

important. If the user wants to generate a model and perform feature selection simultaneously, MARS might be a good method to start with. If the user already has another method in mind for regression, Boruta is a robust feature selection method which could better inform the set-up of the regression model.

Runtime and computational cost are major factors in these algorithms. Random forest has a high cost because of the parallel computation of each tree (Boruta also has this high cost because it depends on the random forest algorithm). With a large dataset, it might be helpful to narrow down variables of interest and only include those in the algorithm to reduce runtime. Model comparison with ANOVA can assist with this part of the decision-making process because the lm() function is less costly and the ANOVA tables allows the user to see how the residual sum of squares changes with each dropped variable, and whether the model is significantly different. Variables which increase error or do not cause significant changes can be dropped from random forest or Boruta consideration. Multicollinearity tests can also help with this step.

Bibliography

Chakur, A. (2019). Random Forest Regression. The Startup. medium.com/swlh

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics,* 19(1):1-67.

Kursa, M. B. & Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software,* 36(11).

Liaw, A. & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, ⅔:18-22.

Prabhakaran, S. (2017). Model Selection. R-statistics.co. http://r-statistics.co/Model-Selection-in-R.html

Credit

The decision of how to structure our project was made during a group meeting with all three of us, then we each chose one or two methods and ran the analysis for our methods. The dataset and dependent variable were also chosen during the group meetings. Each of us then created the powerpoint slides and wrote the sections of the paper for our methods, with Niki and Joelle taking the introduction and conclusion since Bao had two methods to write/present instead of one.

Appendix

```
#Initialize dataset
library(mlbench)
data(BostonHousing)

#Comparison of multiple models using ANOVA
baseMod <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + b + lstat,
data=BostonHousing)
mod1 <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio + b, data=BostonHousing)
mod2 <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax + ptratio, data=BostonHousing)
mod3 <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + tax, data=BostonHousing)
mod4 <- lm(medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad, data=BostonHousing)
anova(baseMod, mod1, mod2, mod3, mod4)
```

```
#MARS
library(earth)
marsModel <- earth(medv~., data=BostonHousing)
ev <- evimp(marsModel)
plot(marsModel, which=1)
plot(ev)

#Boruta
library(Boruta)
boruta_output <- Boruta(medv~., data=BostonHousing, doTrace=2)
boruta_signif <- names(boruta_output$finalDecision[boruta_output$finalDecision %in% c("Confirmed","Tentative")])
print(boruta_signif)
plot(boruta_output, cex.axis=.7, las=2, xlab="", main="Variable Importance")

#Split dataset into training and testing samples
sample = sample.split(BostonHousing$medv, SplitRatio = 0.75)

train = subset(BostonHousing, sample==TRUE)
test = subset(BostonHousing, sample==FALSE)

dim(train)
dim(test)

#Perform random forest regression with training dataset, median house value is dependent variable
ptm <- proc.time()
rf <- randomForest(medv ~ ., data=train, importance=TRUE, ntree=500)
proc.time()-ptm
rf

#Predict new values and examine variable importance
yhat <- predict(rf,newdata=test)
imp <- as.data.frame(rf$importance)
imp <- imp[order(-imp$`%IncMSE`),]
write.table(imp, file="HousingRFvarimportance.txt")

jpeg(filename="rFvarimpgraph.jpeg", width=720, heigh=480)
barplot(imp$`%IncMSE`,
      main="Random Forest Variable Importance",
      xlab="Variable Name", ylab="Mean Decrease in Accuracy, %", col="darkblue",
      names.arg=c("lstat","rm","nox","ptratio","crim","indus","dis","tax","age","rad","b","zn","chas"))
dev.off()


#Evaluation metrics
jpeg(filename="rFeval.jpeg")
plot(x=test$medv, y=yhat,
```

```
    main="Comparison of Predicted Values from Random Forest Model",
    xlab="Actual Median House Price", ylab="Predicted Median House Price",
    col="darkred", pch=3)
abline(0,1)
dev.off()

res <- test$medv-yhat
jpeg(filename="rFresiduals.jpeg")
qqnorm(res)
qqline(res)
dev.off()
```