Assignment #2

James R. Herbick

Introduction

Overview / Purpose

This assignment involves analyzing characteristics of diamonds in order to develop a model that will predict the price of a diamond. This assignment also establishes a predictive modeling framework for a linear regression problem. One of the very first steps of any modeling project is to perform a data quality check. In the data quality check, I am familiarized myself with the data, looked for missing values, and tried to identify any potential outliers. Next, I performed exploratory data analysis (EDA) which investigated the relationships between the independent and dependent variables in the data. The EDA process also included model-based approaches, which helped identify the most relevant subset of attributes in the data.

After performing EDA but before beginning the model-building process, I applied a series of variable-selection techniques. These techniques further identified those independent variables that had the strongest relationship with the dependent variable, the price of a diamond. Finally, I explored multiple modeling methods and determined the best model for predicting the price of a diamond. The modeling techniques that I explored are: multiple linear regression, multiple linear regression with interaction terms, a regression tree, and a random forest.

Modeling Problem Statement

The modeling problem in this assignment is a pure prediction problem. The dependent variable in this assignment is the price of a diamond, and it is a continuous variable. Therefore, linear regression techniques for predicting the price of a diamond were explored. The predictive modeling techniques used in this assignment were: multiple linear regression, decision trees, and random forests. Finally, completion of this assignment will provide a predictive modeling framework for working through a linear regression modeling problem.

Data

General Description

The dataset for this assignment includes 425 observations and 7 variables. Figure 1 is a table of all the variables in the dataset with summary information about each. Of the 7 variables, price is the dependent variable; it is a continuous variable. All of the categorical variables were treated as factors throughout the model-building process. In addition to the 7 original variables, 2 derived variables were created. The first, internet, isolates whether or not a sale occurred over the internet distribution channel. The second derived variable, is the logarithm of the sale price, called logprice. This allowed me to investigate the effects of dependent variable transformations during the model-building process.

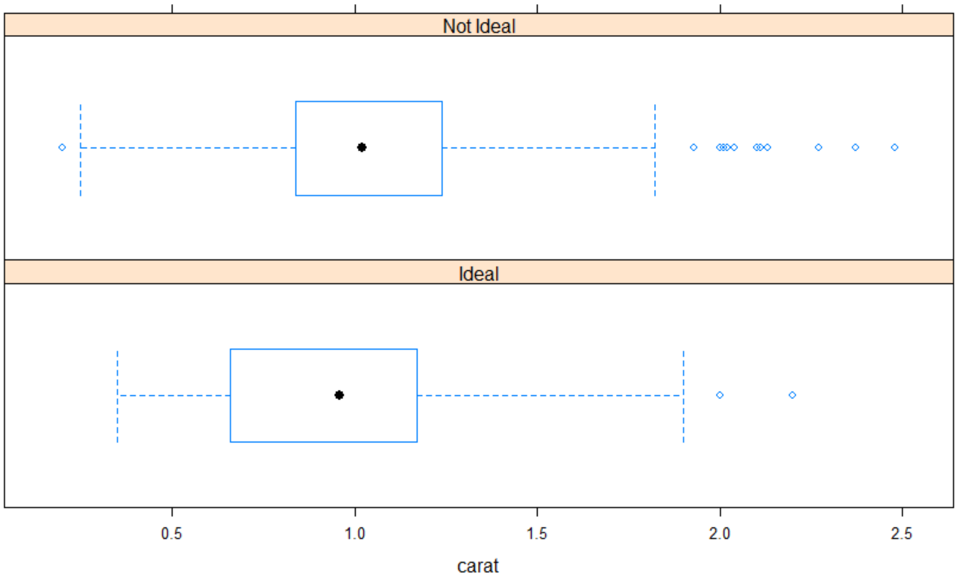
FIGURE 1: Data elements

	Variable Name	Data Type	Original / Derived	Description
1	carat	NUMBER	Original	This is a continuous independent variable. Carat represents the weight of a diamond. One carat equals 0.2 grams.
2	color	FACTOR	Original	The colors present in a particular diamond. Colors range from D through M, and are stored as 1-10. Color is a categorical variable with 9 levels present in the dataset.
3	clarity	FACTOR	Original	A measure of the purity of the stone. Clarity can take values between 1-11. Clarity is a categorical variable with 9 levels present in the dataset.
4	cut	FACTOR	Original	Categorical variable with 2 levels: Ideal, Not Ideal. The cut represents a jewler's assessment of the diamond in a simplified, 2-level categorical variable.
5	channel	FACTOR	Original	Categorical variable with 3 levels: Independent, Internet, Mall. This variable represents the sales channel through which a given diamond sale took place.
6	store	FACTOR	Original	Categorical variable with 12 levels. This variable represents the store name where the diamond sale took place.
7	price	INTEGER	Original	Sale price of the diamond. This is the dependent variable. It is a continuous variable.
8	logprice	NUMBER	Derived	The logarithm of the sale price variable. This is a transformed version of the dependent variable.
9	Internet	FACTOR	Derived	A categorical variable with 2 levels: yes, no. This variable represents whether or not the sale took place through the internet sales channel.

Data Quality Check

As mentioned in the introduction, the first step in any modeling project is to perform a data quality check. The intent in this step of the model-building process is to identify any missing values or outliers in the data. An additional objective of the data quality check is to identify any potential issues in the data that could affect the model-building process. There are no missing values in the data. Most of the independent variables are categorical variables, however, carat is a continuous variable. The carat variable does have some outliers. Figure 2 shows box plots for the carat variable, grouped by cut. As you can see in figure 2, the outliers begin around a carat value of 2. Therefore, I capped all carat values to have a maximum of 2.

FIGURE 2: Carat outliers



Another important aspect of the data quality check is investigating the distributions of the data. Figure 3 shows histograms of the observations with regard to price, cut, channel, and whether or not the sale took place through the internet channel. It appears that diamonds that are not of an ideal cut have a larger range of prices. This is a bit contrary to my initial thoughts. I would have thought that diamonds with an ideal cut would fetch higher prices. Looking at the channels, it appears that the internet channel has a wider range of prices than the other channels. It also appears that if the sale occurs at an independent retailer, there is a small grouping of sales with a high price near $25,000. Finally, the internet channel confirms the earlier statement that there appears to be a broader range of prices for those diamonds sold through the internet channel.

Figure 4 shows histograms of the diamonds' prices by store. Not every unique value for the store variable is shown in figure 4. However, one can see that Zale tends to sell a great frequency of diamonds at a price of about $5,000 and a smaller number of diamonds at a higher price, near $10,000. Once can also see from figure 4 that Goodmans has a majority of its sales between $4,000 and $10,000. But, they also do have sales at the $25,000 price point.

Figure 3: Independent variable histograms (cut, channel, and internet)

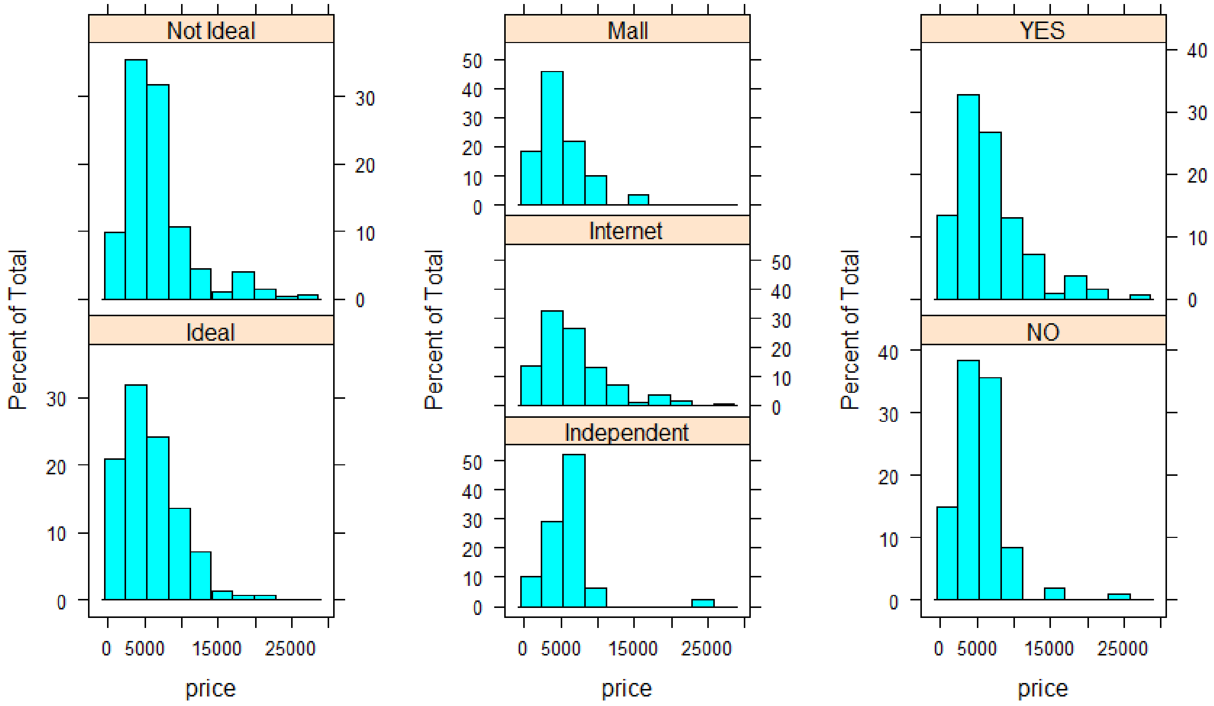
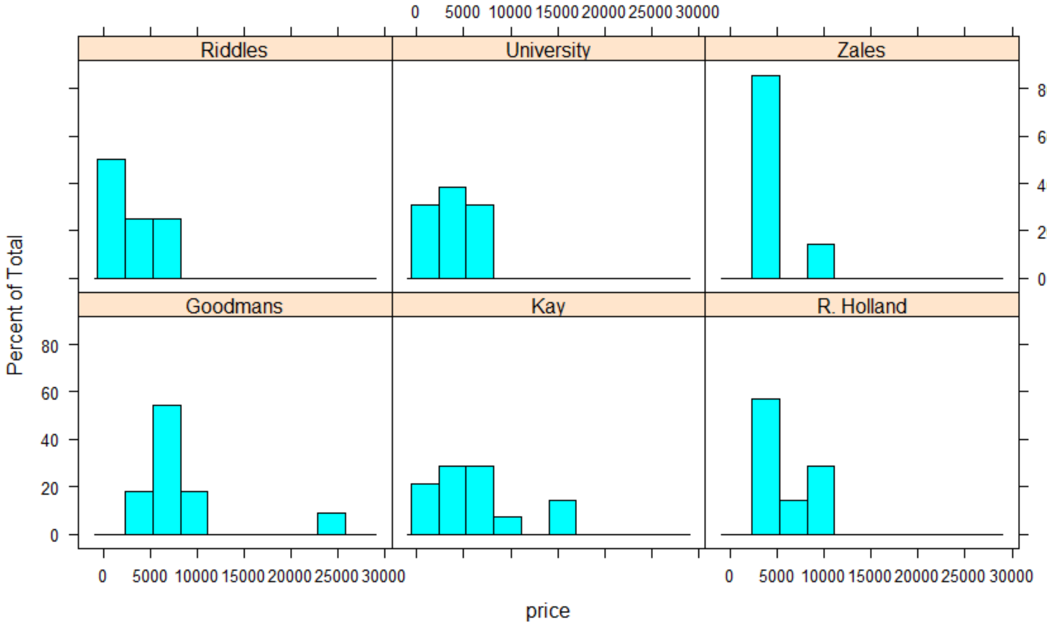


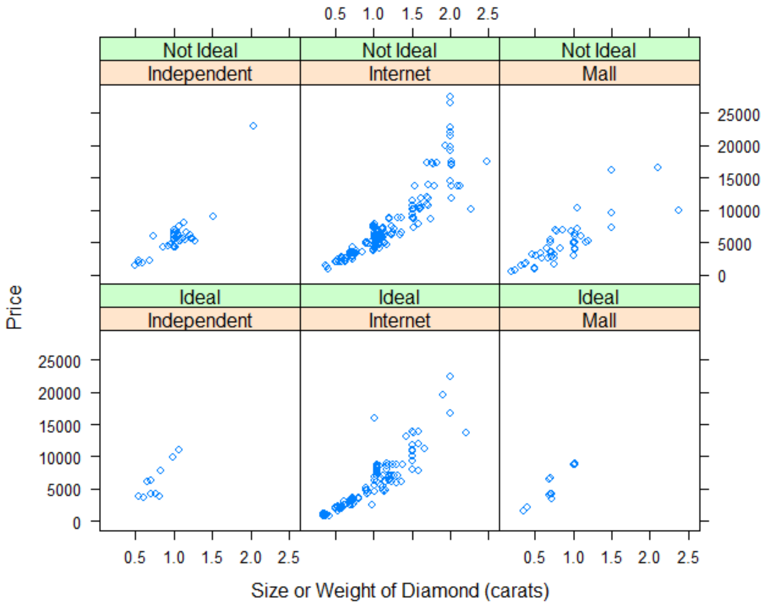
Figure 4: Price by store



Exploratory Data Analysis (EDA)

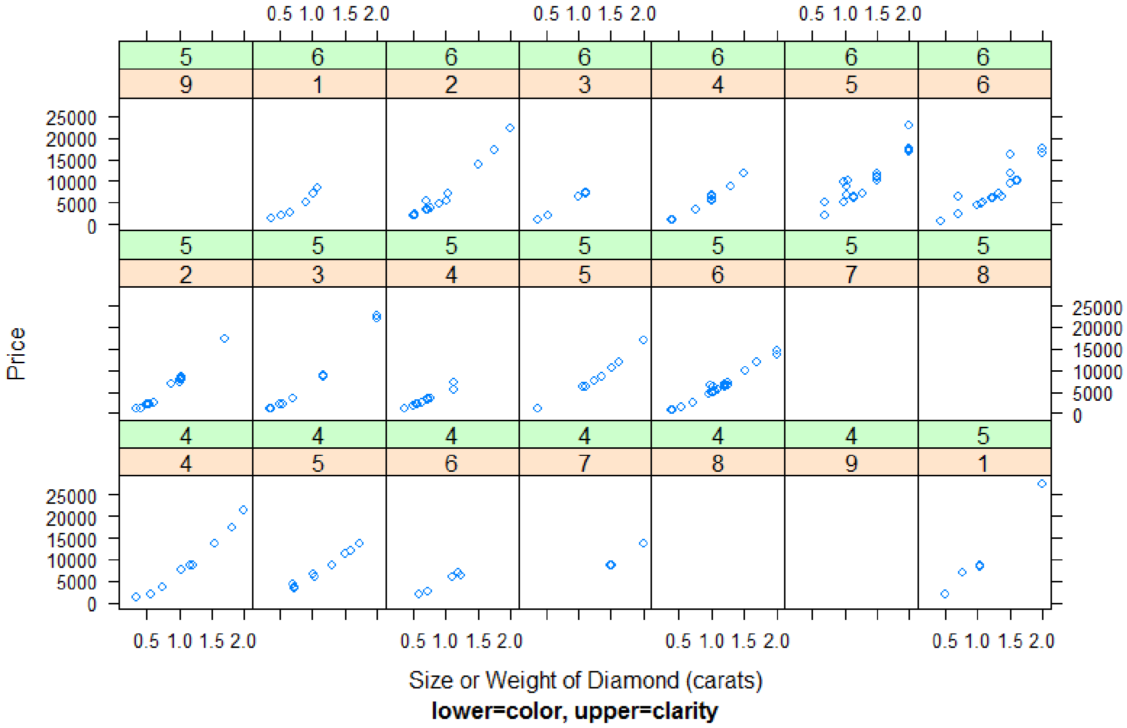
After performing the data quality check, I proceeded to explore the relationships between the dependent and independent variables. First, I examined various relationships between the independent variables and price, the dependent variable. Figure 5 shows scatterplots of price versus carat, grouped by cut and channel. In nearly each section within the scatterplot, there is a positive correlation between carat size and price, regardless of the cut or channel.

FIGURE 5: Carat by price



Taking a slightly different look at the data, I examined price versus carat size grouped by color and clarity. This breakdown is shown in Figure 6. Figure 6 does not include every unique combination. However, here again nearly every subsection of the scatterplot indicates a positive correlation between carat size and price.

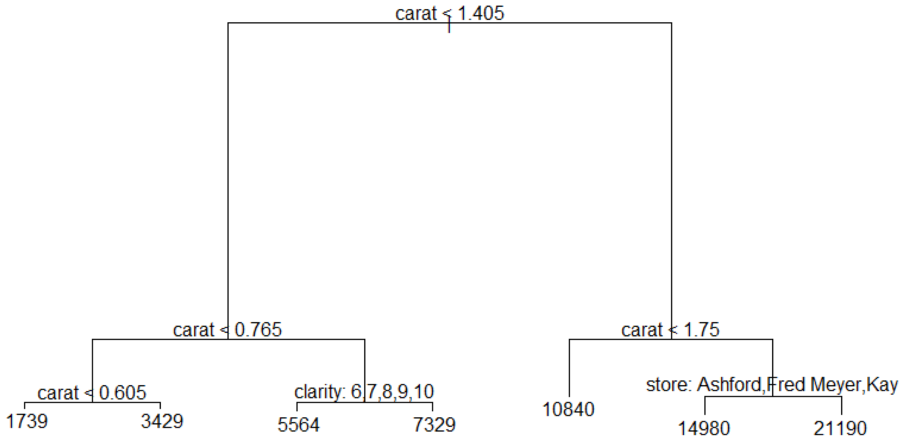
FIGURE 6: Price by carat (grouped by color and clarity)



Model-Based EDA

As a final step in my EDA, I utilized a regression tree to gain additional insights into those independent variables that contribute significantly to the price of a diamond. Figure 7 shows the graphical depiction of the regression tree. As suspected from earlier EDA work, carat size plays a particularly significant role in determining the price of a diamond. In addition, the clarity and store variables appear to have an effect as well. This is consistent with earlier discoveries in the EDA process.

FIGURE 7: EDA Regression tree



Model Building

Naïve Model

To begin the model-building process, I examined a naïve model after using the backward variable selection technique. From the backward variable selection, the selected variables were: carat, color, clarity, cut, and store. With carat being the most significant predictor of diamond price from all previous EDA, I fit a simple linear regression model with just carat as the independent variable. Figure 8 shows a scatterplot of the residuals vs. fitted values. Figure 8 also shows a normal Q-Q plot of the residuals. The scatterplot of the residuals vs. the fitted values starts narrowly to the left and spreads out to the right as the fitted values increase. This creates a bit of an open horn pattern, which is not desirable. The desired result is to have homoscedastic residuals that show no visible pattern. The normal Q-Q plot also shows a deviation from the diagonal line to the far right of the plot. Both of these patterns indicate that a transformation of the dependent variable may be in order.

Therefore, I reran the simple linear regression with the logprice as the dependent variable. Figure 9 shows the same residual and Q-Q plots as a result of using logprice as the dependent variable. In figure 9 we see that the scatterplot of the residuals vs. the fitted values are more homoscedastic. There is no visible pattern. In addition, the normal Q-Q plot shows a less pronounced upward curve in the far right of the Q-Q plot. Therefore, I used the logprice as the dependent variable for the rest of my model-building process.

Figure 8: Residual and Q-Q plot for price dependent variable

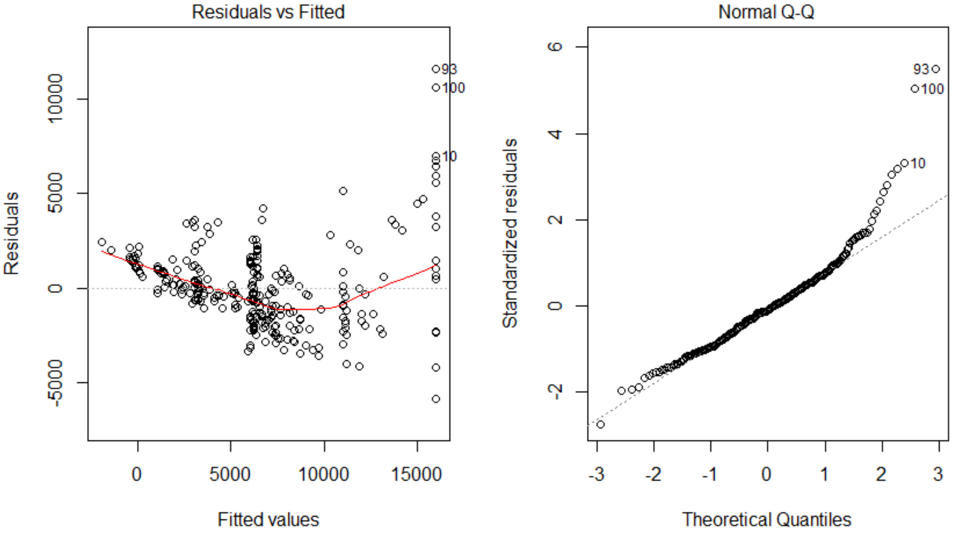
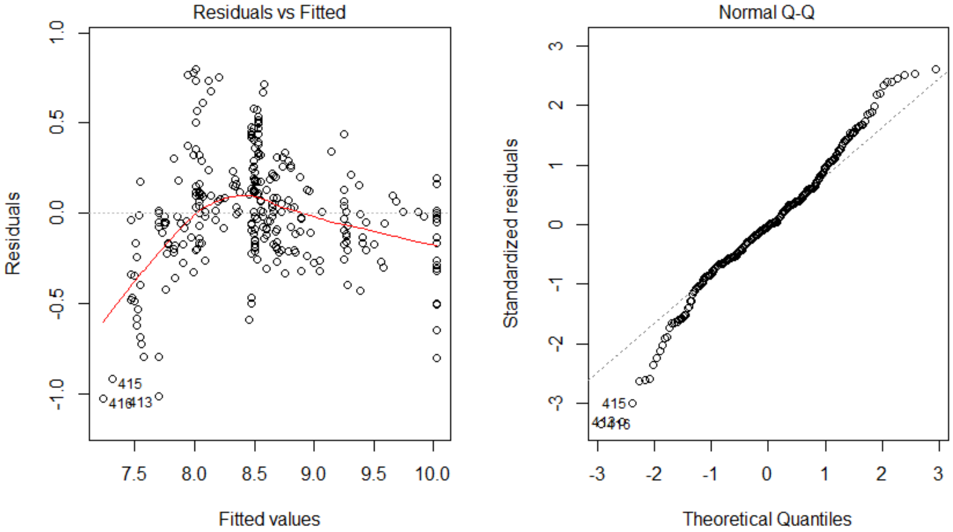


Figure 9: Residual and Q-Q plot for logprice dependent variable



Variable Selection Techniques

After examining a naïve model, I utilized a series of variable selection techniques. These techniques included: forward, backward, stepwise, and all subsets regression. I also utilized the Lasso, which is a coefficient shrinkage technique where some variable coefficients are set to zero. Figure 10 shows the results of the variable-selection techniques. As you can see in figure 10, the forward, backward, stepwise, and all subsets methods selected the same variables. The lasso method chose the cut and internet channel, while the other methods did not. Also, the lasso method did not utilize the store variable.

Figure 10: Variable selection techniques

Independent Variable	Forward	Backward	Stepwise	All Subsets	LASSO
carat	YES	YES	YES	YES	YES
color	YES	YES	YES	YES	YES
clarity	YES	YES	YES	YES	YES
cut	NO	NO	NO	NO	YES
channel	YES	YES	YES	YES	NO
store	YES	YES	YES	YES	NO
internet	NO	NO	NO	NO	YES

Model Suite

The next step in the model-building process was to build various models suitable to a linear regression situation. The model suite included: linear regression models, a regression tree, and a random forest.

Linear Regression

For the linear regression models, I ran a model that had no interaction terms and a model that did include an interaction term. While revisiting the regression tree from the EDA, I chose to include an interaction between clarity and carat. Figure 11 shows the resulting fit statistics from the training data. We can see that the interaction term slightly improves the model, and increases the adjusted R squared slightly.

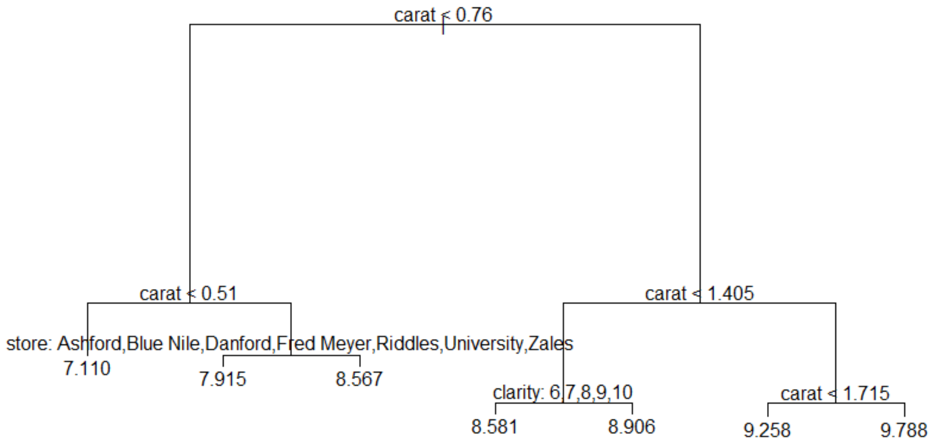
Figure 11: Linear regression results

Model	Adj R Squared
No interaction term	0.8889
With interaction term	0.8952

Regression Tree

The regression tree shown in figure 12, again identifies the carat variable as a significant predictor in the model. In addition, clarity and store values are used to determine splits in the tree. These results are consistent with many of the variable selection techniques.

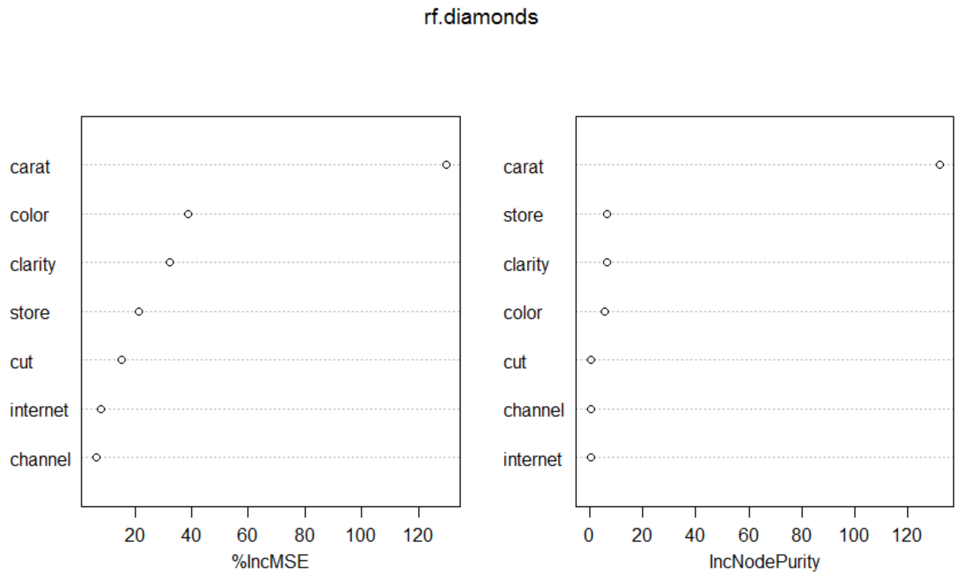
Figure 12: Regression Tree



Random Forest

Figure 13 shows the importance of variables as determined by the random forest. Variable importance is displayed by the percent increase in MSE when a particular variable is removed from the model. Variable importance is also shown as the increase in node purity provided by the variable. Again, we see that carat tops the list of variables as far as importance goes. Color and clarity are the next most important variables from an MSE perspective.

Figure 13: Random forest variables



Conclusions

I calculated the mean squared error (MSE) on both the training and test datasets for each model to compare their performance. The lower the MSE, the better a model's predictive accuracy. I compared the MSE between models both in sample (on the training data) and out of sample (on the test data). For a pure prediction problem, the best determination of a model's performance is on the test data. Figure 14 shows that the random forest performed the best on the test data and had the lowest MSE.

Figure 14: Model Results

Model	Training MSE	Test MSE
Linear Regression (no interactions)	0.05238363	0.06104900
Linear Regression (with interactions)	0.0482154	0.06044488
Regression Tree	0.04539	0.07211875
Random Forest	0.0374877	0.03916958

Appendix

R Code

```
# ------------------------
# data quality check / EDA
# ------------------------

summary(diamonds)  # no missing values

# Boxplots of continuous variables.
a <- bwplot( ~ price | cut, data = diamonds, layout=c(1,2))
b <- bwplot( ~ price | channel, data = diamonds, layout=c(1,3))
c <- bwplot( ~ price | internet, data = diamonds, layout=c(1,2))

c.2 <- bwplot( ~ price | store, data = diamonds, layout=c(1,2))

bwplot( ~ carat | cut, data = diamonds, layout=c(1,2))


# Print multiple trellis objects to the same page, this works
plot(a, split = c(1, 1, 3, 1), aspect="xy")
plot(b, split = c(2, 1, 3, 1), newpage = FALSE, aspect="xy", layout=c(0,3))
plot(c, split = c(3, 1, 3, 1), newpage = FALSE, aspect="xy", layout=c(0,3))


# Histograms of price by the categorical variables.
d <- histogram( ~ price | cut, data = diamonds, layout=c(1,2))
e <- histogram( ~ price | channel, data = diamonds, layout=c(1,3))
f <- histogram( ~ price | internet, data = diamonds, layout=c(1,2))

histogram( ~ price | store, data = diamonds, layout=c(3,2))
histogram( ~ price | color, data = diamonds, layout=c(3,2))
histogram( ~ price | clarity, data = diamonds, layout=c(3,2))

# Print multiple trellis objects to the same page, this works
plot(d, split = c(1, 1, 3, 1), aspect="xy")
plot(e, split = c(2, 1, 3, 1), newpage = FALSE, aspect="xy", layout=c(0,3))
plot(f, split = c(3, 1, 3, 1), newpage = FALSE, aspect="xy", layout=c(0,3))






# install the lattice graphics package prior to using the library() function



# let's prepare a graphical summary of the diamonds data
# we note that price and carat are numeric variables with a strong relationship
# also cut and channel are factor variables related to price
# showing the relationship between price and carat, while conditioning
# on cut and channel provides a convenient view of the diamonds data
```

```r
# in addition, we jitter to show all points in the data frame

xyplot(jitter(price) ~ jitter(carat) | channel + cut,
       data = diamonds,
       aspect = 1,
       layout = c(3, 2),
       strip=function(...) strip.default(..., style=1),
       xlab = "Size or Weight of Diamond (carats)",
       ylab = "Price")


# This has tons of combinations
xyplot(jitter(price) ~ jitter(carat) | color + clarity,
       data = diamonds,
       aspect = 1,
       layout = c(7, 3),
       strip=function(...) strip.default(..., style=1),
       xlab = "Size or Weight of Diamond (carats)",
       ylab = "Price",
       sub = "lower=color, upper=clarity")


# Redo, just on clarity, not color... after looking at the decision tree
xyplot(jitter(price) ~ jitter(carat) | clarity,
       data = diamonds,
       aspect = 1,
       layout = c(5,2),
       strip=function(...) strip.default(..., style=1),
       xlab = "Size or Weight of Diamond (carats)",
       ylab = "Price",
       sub = "by clarity")




# ------------------------
# Fitting Regression Trees
# ------------------------
library(tree)
library(MASS)
set.seed(1)

par(mfrow=c(1,1))

# Create training set, create regression tree
# train = sample(1:nrow(Boston), nrow(Boston)/2)

tree.diamonds=tree(logprice~.-price,train)
summary(tree.diamonds)

# Plot the tree
plot(tree.diamonds)
text(tree.diamonds,pretty=0)

# See if pruning will help performance
cv.diamonds=cv.tree(tree.diamonds)
plot(cv.diamonds$size,cv.diamonds$dev,type='b')
```

```
# Alternate way to prune the tree
prune.diamonds=prune.tree(tree.diamonds,best=7)
plot(prune.diamonds)
text(prune.diamonds,pretty=0)

# Make predictions on the test set with the unpruned tree
yhat=predict(tree.diamonds,newdata=test)
# boston.test=Boston[-train,"medv"]
plot(yhat,test$logprice)
abline(0,1)
mean((test$logprice-yhat)^2)





# -------------
# Random Forest
# -------------

library(randomForest)

# Random forest using 6 random variables
set.seed(1)
rf.diamonds=randomForest(logprice~.-price, data=train, mtry=6,importance=TRUE)

mean(rf.diamonds$mse)

yhat.rf = predict(rf.diamonds,newdata=test)
mean((test$logprice-yhat.rf)^2)

# View the importance of each variable
importance(rf.diamonds)
varImpPlot(rf.diamonds)  # Plot the importance measures
```