Predict 454: Advanced Machine Learning

Assignment 2

Predictive Modeling in Regression

Jack Doig

Introduction

This paper outlines the results from an exercise in exploratory data analysis (EDA), statistical graphics, and predictive modeling exercise for a round-cut Diamond data set, which contains the commercial data about the sale of the Diamond as well as their characteristics.

The Data originate from a study of both on-line and bricks-and-mortar jewelers in the Wisconsin area by Brian A. Pope.

This paper firstly does a simple data quality check, followed by basic EDA, some more advanced model-driven EDA, and finally some detailed modeling. The goal is to first find interesting relationships and fully understand the structure of the data with a view to optimizing decisions during modeling.

The modeling exercise in question is a continuous regression problem, where the target variable is the Sale price of the Diamond.

Part 1: Data Quality Check

1.1 Basic Data understanding

The purpose of this section is to gain a basic statistical understanding of the data set. The table below describes the structure of the data set.

Metric	Value
Number of observations	425
Type of Target Variable	Continuous
Categorical Independent Variables	3
Continuous Independent Variables	3
Total Number of Variables	7
Size of Data File	17.2 kb

Figure 1.1: basic data set metrics

The table above shows that this data set is made up of both continuous and categorical variables. It is also a relatively small data set, with only 425 observations, so there should be no computational challenges in this study, however this may have a downstream impact on the decision of validation and training set size during modeling.

The table below describes each of the variables. It is important to invest time in gaining a real-world understanding of the data so as to understand the real-world impact of outliers, patterns and distributions, and possibly get clues so as to optimize the EDA and modeling process.

The target variable is the price, which is a continuous variable, representing the price at which the diamond is sold.

Variable	Type	Description
Carat	Categorical	Standard Unit of Weight For Gemstones, rounded to the nearest hundredth of a carat. Diamonds are measured in one-quarter increments.
Color	Continuous	Represents the color of the stone. Color varies from yellow to brown. Note: this variable could be considered categorical, although the current grading is correlated with price.
Cut	Categorical	An inconsistently applied measure of the quality of the cut of the diamond surface. Two categories: Ideal and not ideal
Clarity	Continuous	A measure of how many carbon imperfections the stone carries. Could also be treated as categorical
Channel	Categorical	The sales source of the Diamond: Internet, Mall or Independent
Store	Categorical	The specific store location where the Diamond is to be bought/sold
Price	Continuous	The sale price of the Diamond in USD. The target variable

Figure 1.2: basic description of variables

1.2 Basic Descriptive Statistics

The table below outlines the basic statistics for the continuous variables.

	df.carat	df.color	df.clarity	df.price
nobs	425.00	425.00	425.00	425.00
NAs	0.00	0.00	0.00	0.00
Minimum	0.20	1.00	2.00	497.00
Maximum	2.48	9.00	10.00	27575.00
1. Quartile	0.72	3.00	5.00	3430.00
3. Quartile	1.21	6.00	7.00	7792.00
Mean	1.04	4.31	6.13	6355.99
Median	1.02	4.00	6.00	5476.00
Stdev	0.42	1.86	1.60	4404.24
Skewness	0.70	-0.01	-0.30	1.71
Kurtosis	0.43	-0.77	-0.17	3.77

Figure 1.3: basic statistics of continuous variables

The following observations are notable and can be made about the basic statistics outlined in the above table:

1. There are no missing observations
2. The price distribution is both skewed and peaked, so may need some transformation
3. All of the continuous variables sit within reasonable boundaries given the information about the variable meaning above

1.3 Unconditional Distributions of continuous variables

The distributions of the continuous variables are shown in the image below.

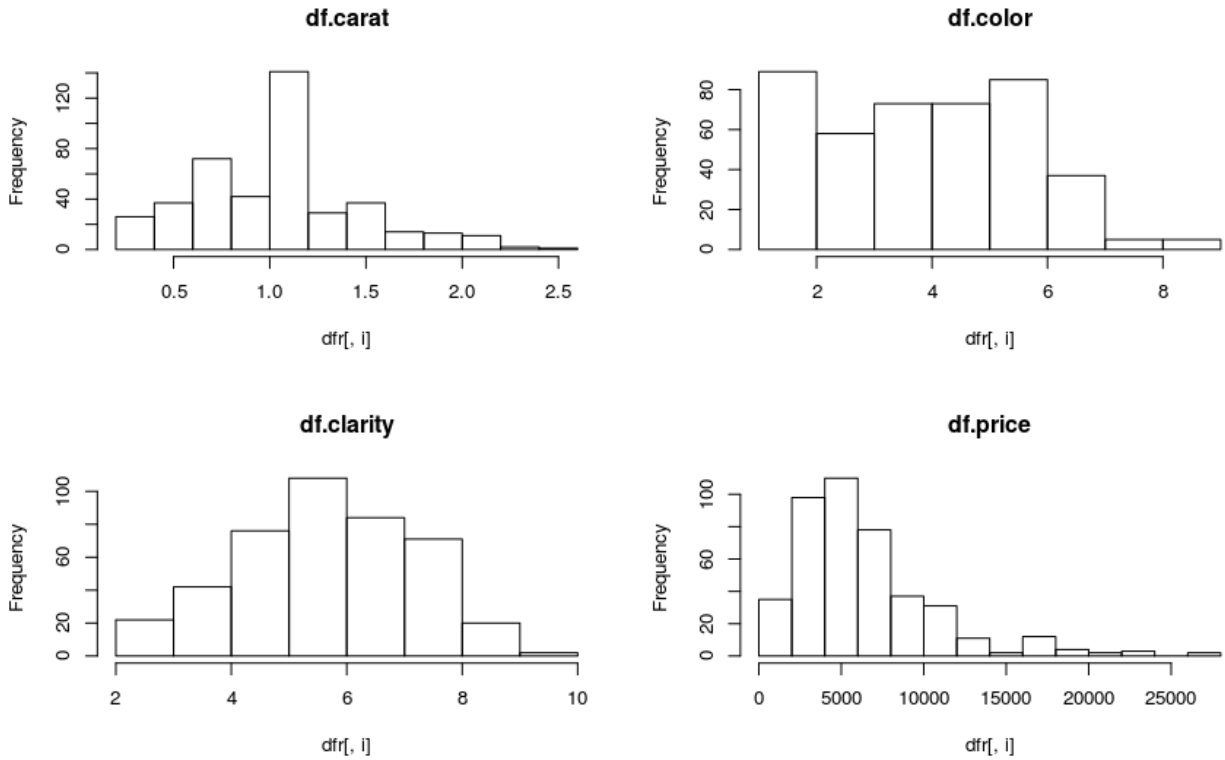


Figure 1.4: Continuous variable distributions

The observations of these distributions are:

1. The price distribution is indeed skewed and peaked so may need transformation.
2. Carat is not normally distributed so may need transformation, however clarity is.
3. Color appears to have an even distribution over the range.

1.4 Outlier detection – continuous variables

As the scaled boxplot shows below, there are a number of possible outliers in the Price and Carat variables.

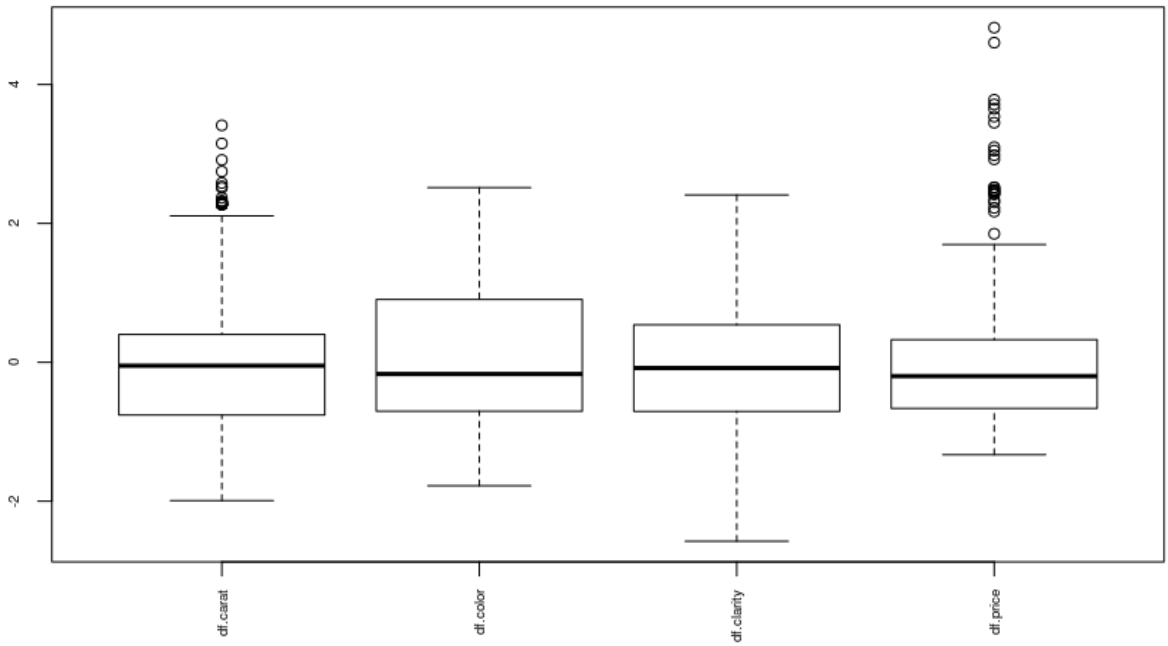


Figure 1.5: Box plot of all continuous variables

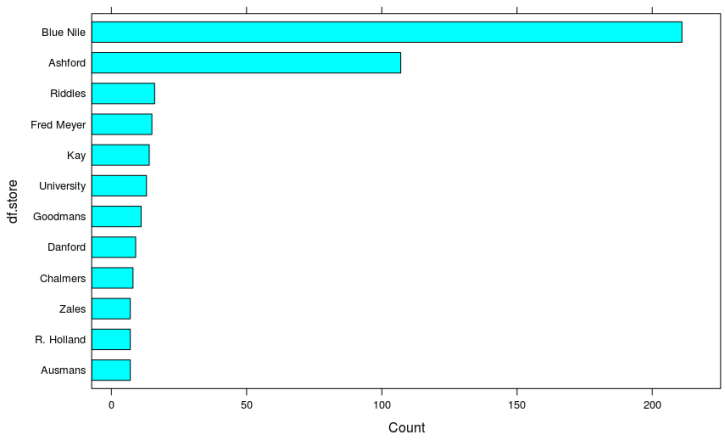
The table below shows the number of observations sitting outside 3 standard deviations. Price has 2 observations greater than 4 standard deviations from the mean. Both Price and Caret may need to be transformed.

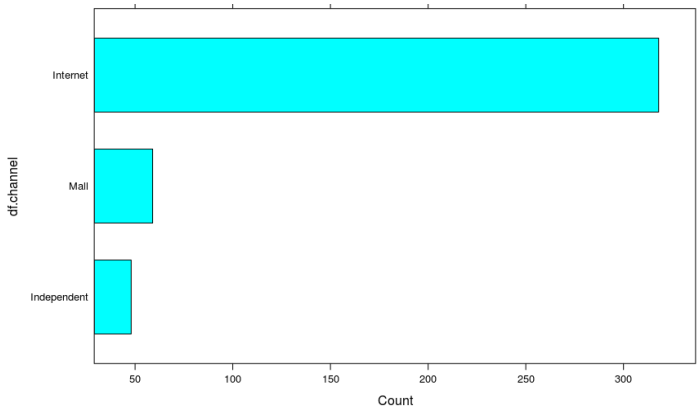
Standard Deviations from Mean	3+	3.5+	4+	5+
carat	2	0	0	0
color	0	0	0	0
clarity	0	0	0	0
price	9	6	2	0

Figure 1.6: Number of observations greater than n SDs from the mean by variable

1.5 Understanding the distribution of the categorical variables

The images below show the distributions of the categorical variables.





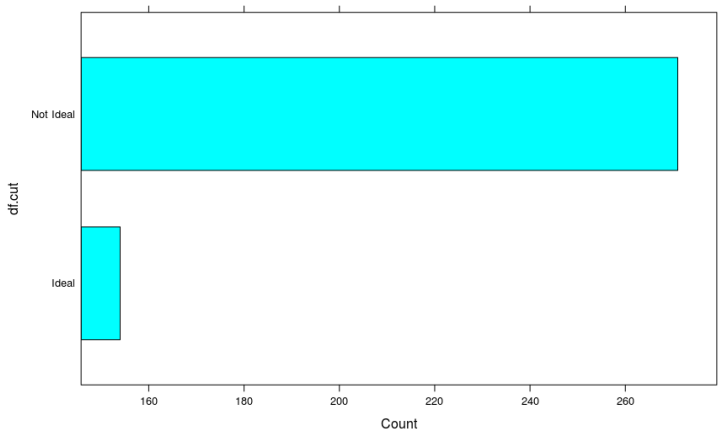


Figure 1.7: Distribution of the categorical variables

There are two stores which dominate the Store data, Blue Nile and Ashford. Similarly, the Channel predictor is dominated by Internet, and almost all stones fall into the “Not Ideal” category. Later it will be explored if these categories offers any separability with respect to the target variable.

Part 2: Exploratory Data Analysis

The next section is a natural continuation of the data quality check, however it extends the data analysis in that is begins to apply exploratory techniques to the relationships between the predictors themselves, and the relationships between the predictors and the target, rather than considering these variables in isolation.

2.1 Correlations

Highly correlated data within a multi-variate data set can impact the performance of some models, so it is useful to understand the correlations within the predictor data set. The diagram below shows the strength of correlation between the continuous predictors. It is also important to get a strong correlation between the predictors and the target variable, using transformations if need be.

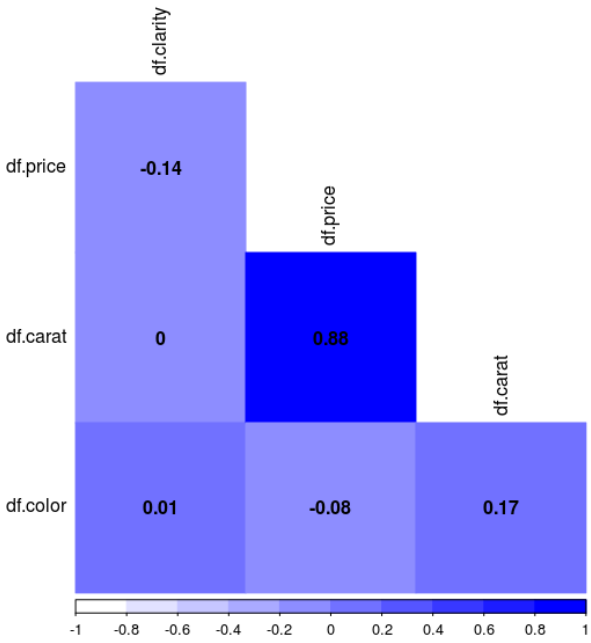


Figure 2.1: Correlations between continuous variables

This chart indicates there is a strong correlation between Carat and Price, which indicates this variable will is useful for modeling, while there is low levels of correlation among the remainder of the variables. This suggests that no dimensionality reduction will be needed to address multicolinearity issues.

2.2 Conditional Distributions against the target variable

There is not a large degree of apparent separability that the categorical variables offer the price. The class offering the most separability is the Store variable, and the boxplots below indicate this separability.

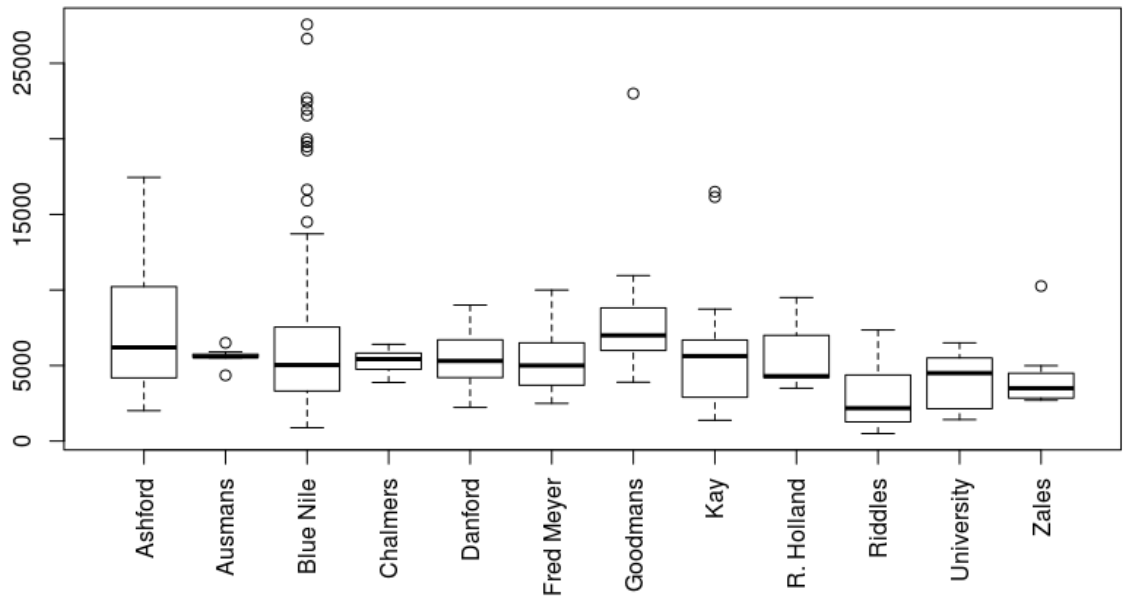


Figure 2.2: Distributions of Price by Store

2.3 Improving correlation of the Continuous Variables with the target

In order to improve the predictive accuracy of the final modeling for the linear regression based models, it is important to improve the linearity of the relationships between the predictors and the target. The most successful example of this is the transformation of both Caret and Price with log functions, which results in a much more linear relationship, and the outliers confirm to this linear relationship also.

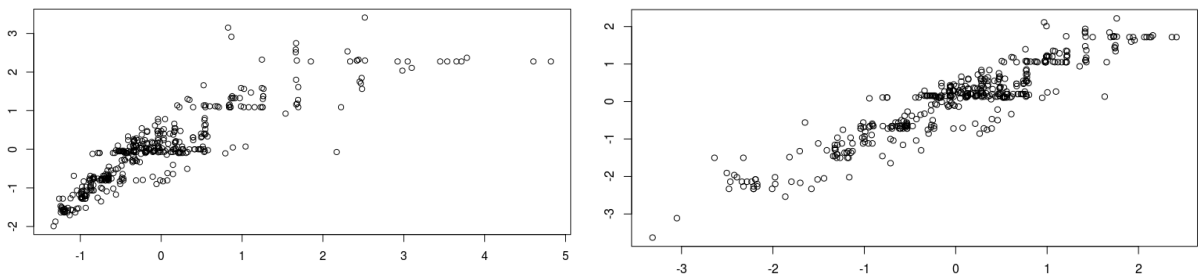


Figure 2.3: Untransformed Price-Caret plot (left) and log-log transformation (right)

Part 3: Model Based Exploratory Data Analysis

3.1 Fitting a tree for EDA insight

The diagram below shows the resulting of fitting a tree using a minimum split of 20:

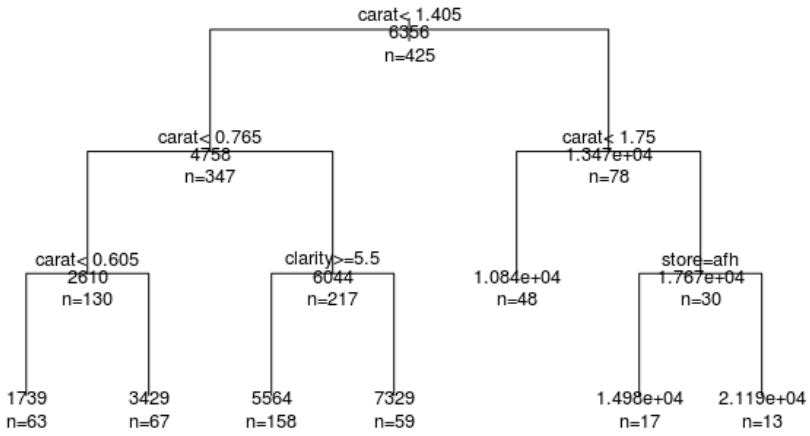


Figure 3.1: Visualization of a simple tree fit to the Data

Variable	Importance Metric
Carat	69
Store	5
Clarity	3
Color	2
Channel	1
Cut	0

Figure 3.2: Variable importance measures from the tree

This strongly indicates that the Caret is the most important predictor of the price variable, whereas Cut and Channels offer relatively little insight into the price.

3.2 Fitting a Random Forest to understand variable importance, transforms and interaction

I have also fit a Random Forest in order to understand the importance of variables. The Random Forest confirms the importance of the Caret variable, although not as pronounced, and while the top three variables are the same as the tree, the order is changed:

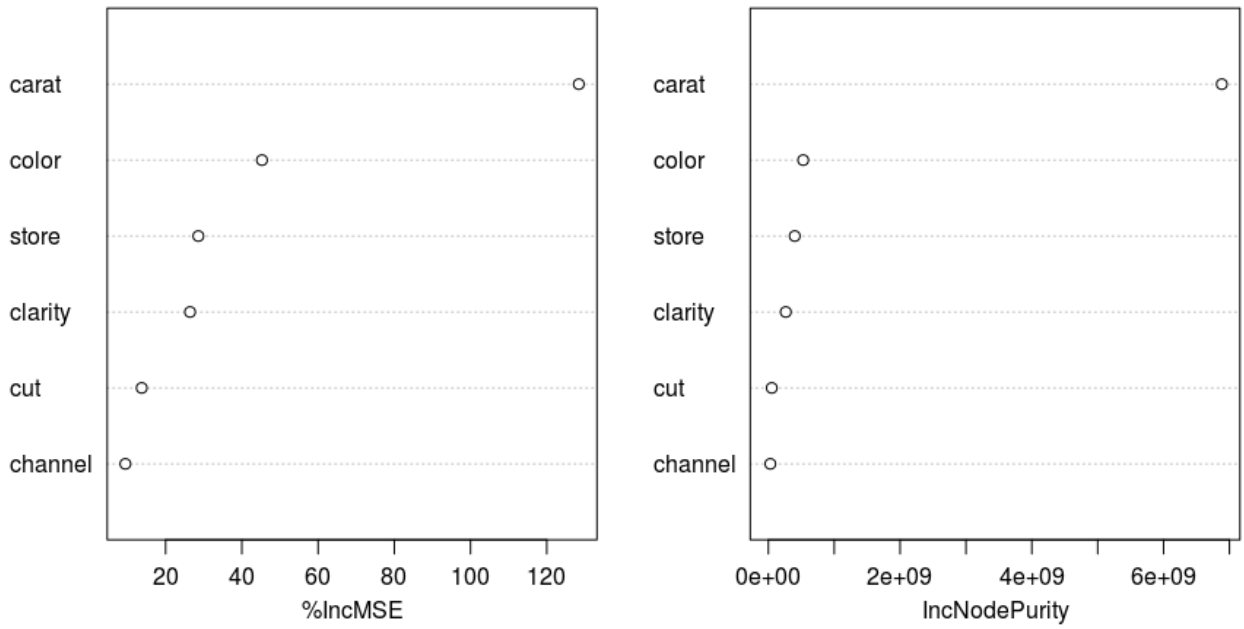


Figure 3.3: Variable importance according to the Random Forest

There appears to be no reason to transform Color and Clarity, as their relationship with the target variable is linear, as shown by these partial dependence plots for the variables Clarity and Color:

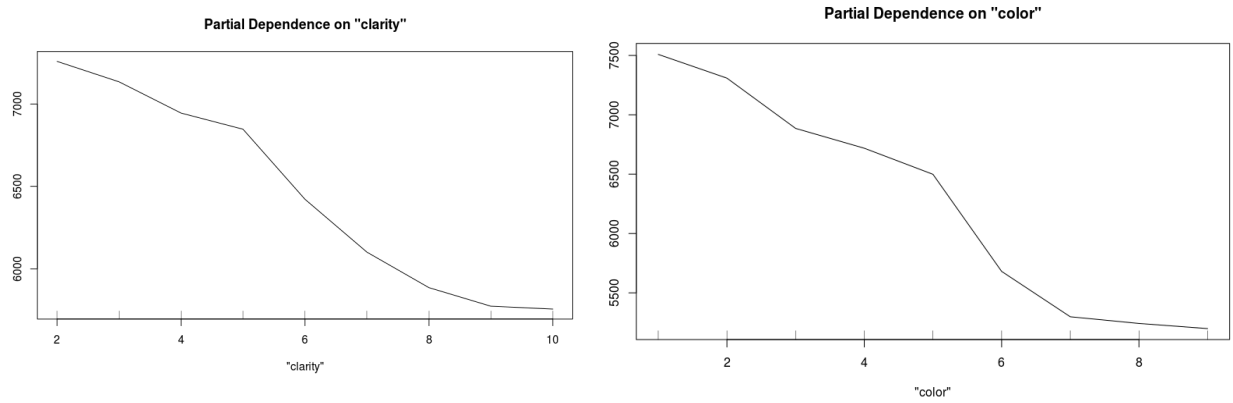


Figure 3.4: Variable partial dependence plots showing linearity

Having explored 3 dimensional partial dependency plots (pictured below), there is a possible interaction between color and carat.

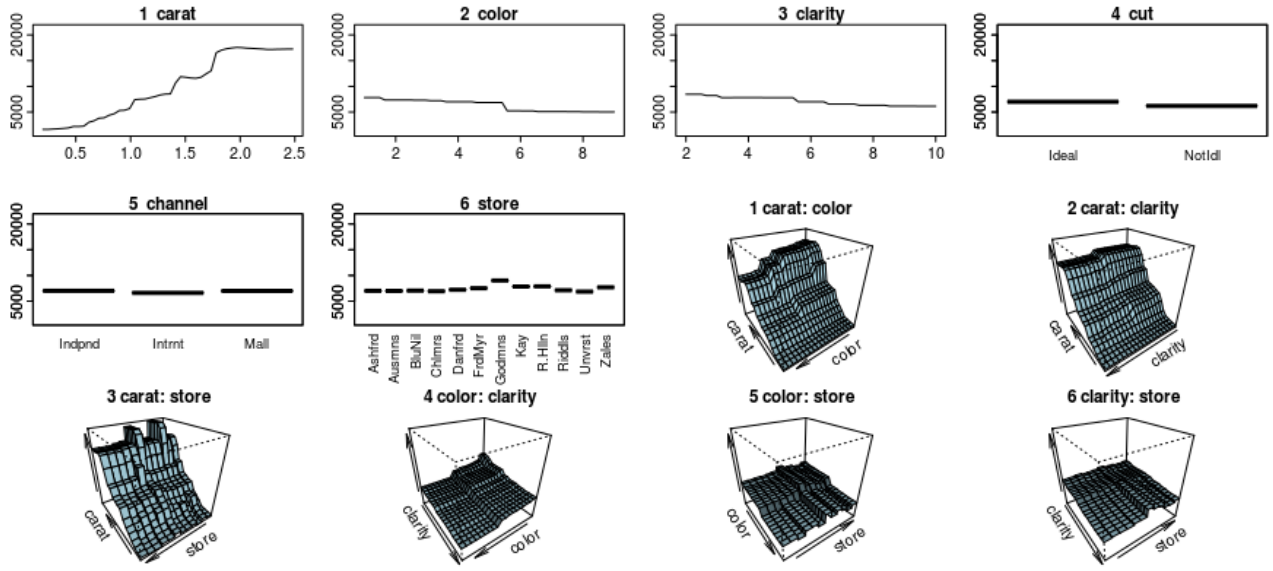


Figure 3.5: Variable partial dependence plots showing 3 dimensional dependencies

Part 4: Variable Transforms

In order to address the outliers in the Price and Caret variables, and the non-linear relationships between the Price variable and some of the predictors, I explored a number of different transformation types. The best transformation was the Log10 transformation of both the Price variable and the Caret variable.

There are no missing variables in the data set, and generally the outliers cannot be discounted as being unreasonable, so therefore there is no data imputation required.

There is possibly an interaction between color and carat which will be explored further during modeling with a separate model.

Part 5: Modeling

For this section, I have split the data 70% training and 30% test. Any model requiring validation will need to split the training set.

5.1 Baseline Model - Naive regression model using backwards variable selection and no transforms

In order to establish a baseline model, we first run a baseline model in the form of a backwards variable selection linear regression with no transformations and no interactions terms.

This resulted in the following baseline measures:

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Backwards Variable Selection, no transforms (baseline)	9	2,851,116	1688.525	5072.521	5113.152	0.9241	0.9264	No transforms No interaction terms

5.2 Regression model using forward variable selection

Observations from this initial model are that Carat is confirmed as very important as it was in every model generated by the variable selection process. The other important variables were color and clarity (as expected)

5.2 Further Linear Regression Modeling

The following models were then applied to the data:

- Regression model using backwards variable selection including variable transforms
- Regression model using stepwise variable selection including variable transforms
- Regression model using all subsets variable selection including variable transforms
- Regression model using LASSO including variable transforms

Below are the results and the findings from this exercise:

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Backwards Variable Selection, transforms	9	650,413	806.48	-311.087	-270.4586	0.9618	0.963	No interaction terms
Stepwise Variable Selection, transforms	10	644608	802.875	-312.71	-249.92	0.9646	0.9628	No interaction terms
Forwards Variable Selection, transforms	10	597,610	773.05	-315.52	-267.51	0.964	0.9626	No interaction terms
Best Subsets variable selection, transforms	9	650,413	806.48	-311.087	-270.4586	0.9618	0.963	No interaction terms
Lasso Regression, transforms	12	708,657	841.82					No interaction terms Lamba=0.1

The findings are that the log transform to both Price and Carat had a major impact, and that a 10-variable model chosen by forwards selection is the best performing on out-of-sample measure.

All models include the following continuous variables: log(Carat), Color, Clarity,

All models include the following categorical variables: Store=Riddles, Fred Mayer, Goodmans, the Channel=Mall, and Cut=Ideal.

Part 6: Linear Regression with interactions

Using the results from the Random Forest exploratory data analysis, which showed a possible interaction between Color and Carat, and Carat and Clarity I tried adding these interaction terms to a forwards variable selection algorithm

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Forwards Variable Selection, transforms, interactions	12	560,613	748.74	-350.79	-295.39	0.969	0.967	

The addition of the interaction terms has made a significant impact on the in-sample and out-of-sample measures of performance. Note that the forward variable selection model included both the interaction terms as well as the original variables in the best model.

Part 7: Tree models

Fitting a simple tree model on the raw data, the transformed variables and then adding the interaction terms (3 models) yields the following results:

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Simple Tree	20	2,279,583	1509.83					
Simple Tree with Transforms	20	3,038,645	1743.17					
Simple Tree with transforms, interaction	20	3,038,645	1743.17					

Simple tree models perform considerably worse than the linear regression models, and appear to be weakened by the transforms and interaction. The trees with derived variables actually performed worse than the baseline model.

Part 8: Ensemble Models

Fitting Random Forest and Bagging ensemble models to the data for the non-transformed, interaction terms and the transformed variables (3 models) yields the following results:

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Random Forest	20	760,899	872.29					500 trees Mtry=11
Random Forest, transforms	20	798,209	893.43					500 trees Mtry=13
Bagging	20	994,998	997.49					500 trees
Bagging, Transforms	20	1,008,213	1004.1					500 trees
Random Forest, transforms, interactions	20	891,524	944.21					500 trees Mtry=10

Even why optimizing the mtry argument, the ensemble models do not perform as well as the linear regressions on this data, and the transformations and interaction terms appeared to weaken the models.

Part 9: Model Comparison and Conclusion

9.1 Model Comparison

The final model comparison for all models developed is shown below, with the best model highlighted in red.

Model Type	Number of Variables	Out of Sample MSE	Out of Sample RMSE	AIC	BIC	AdjR2	R2	Notes
Backwards Variable Selection, no transforms (baseline)	9	2,851,116	1688.525	5072.521	5113.152	0.9241	0.9264	No transforms No interaction terms
Backwards Variable Selection, transforms	9	650,413	806.48	-311.087	-270.4586	0.9618	0.963	No interaction terms
Stepwise Variable Selection, transforms	10	644608	802.875	-312.71	-249.92	0.9646	0.9628	No interaction terms
Forwards Variable Selection, transforms	10	597,610	773.05	-315.52	-267.51	0.964	0.9626	No interaction terms
Best Subsets variable selection, transforms	9	650,413	806.48	-311.087	-270.4586	0.9618	0.963	No interaction terms
Lasso Regression, transforms	12	708,657	841.82					No interaction terms Lamba=0.1
Random Forest	20	760,899	872.29					500 trees Mtry=11
Random Forest, transforms	20	798,209	893.43					500 trees Mtry=13
Bagging	20	994,998	997.49					500 trees
Bagging, Transforms	20	1,008,213	1004.1					500 trees
Random Forest, transforms, interactions	20	891,524	944.21					500 trees Mtry=10
Forwards Variable Selection, transforms, interactions	**12**	**560,613**	**748.74**	**-350.79**	**-295.39**	**0.969**	**0.967**	**transforms Interactions**
Simple Tree	20	2,279,583	1509.83					No interaction terms
Simple Tree with Transforms	20	3,038,645	1743.17					No transforms No interaction terms
Simple Tree with transforms, interaction	20	3,038,645	1743.17					transforms Interactions

9.2 Conclusion

The impact of creating a strong linear relationship between the target variable and the most important predictor (Carat) had a major impact on the model accuracy. This lessened the impact of outliers, and improved the predictive accuracy of the modeling.

The impact of finding the interaction terms, and adding them to the model alongside the original terms was also significant.

Many of the additional categorical variables appeared to be of no value to predictive accuracy.

The ensemble and tree-based models appeared to be weakened by adding derived predictors, or by transforming predictors. The Random Forest and Tree models performed worse than the linear regression models.

Appendix: R-Code

R-code developed in the course of this project

```
#1 basic stats
library("fBasics")
BasicStatsDF <- my.Summary(df_cont)
write.csv(BasicStatsDF,"basicStats.csv")

#2 distributions (cont)
dev.off()
par(mfrow=c(2,2))
PlotHistograms(df_cont)

#transformed predictors
dev.off()
par(mfrow=c(2,3))
PlotTransformedHistograms(df_cont)

#plot pairs, big enought ot be visible
PlotPairsAgainstTarget(df_cont$df.price, df_cont[,-4])
PlotPairsAgainstTarget(log(df_cont$df.price), df_cont[,-4])
PlotTransformedPairsAgainstTarget(df_cont$df.price, df_cont[,1])
PlotTransformedPairsAgainstTarget(df_cont$df.price, df_cont[,2])
PlotTransformedPairsAgainstTarget(df_cont$df.price, df_cont[,3])

#target
dev.off()
PlotHistograms(df_target)

#conditional boxplots
densityplot(~ df[,7] , group=channel, data = df )
densityplot(~ df[,7] , group=store, data =  df , ylim=c(0,0.0005))
densityplot(~ df[,7] , group=cut, data = df )

boxplot(df$price~df$channel , cex.axis=1, las=3)
boxplot(df$price~df$store , cex.axis=1, las=3)
boxplot(df$price~df$cut , cex.axis=1, las=3)

#transformed target
dev.off()
par(mfrow=c(2,3))
PlotTransformedHistograms(df_target)

#3 distributions (cat)
#this is for when the binaries are split already
#barchart(colSums(df[,11:14]))

#not split
PlotCategoricalBarCharts(df_cat)

#4 correlations
cor <- cor(df_cont, method="pearson")
library(ggplot2)
library(reshape2)
qplot(x=Var1, y=Var2, data=melt(cor), fill=value, geom="tile")

library(corrplot)
col3 <- colorRampPalette(c("white", "blue"))
corrplot(cor, order="AOE",  method="color",  col=col3(10),addCoef.col="black",
tl.cex=1, tl.col="black",type = c("lower"),diag = FALSE)
corrplot(cor)
```

```
pairs(df_cont)

#5 outliers
dev.off()
boxplot(scale(df_cont), cex.axis=0.60, las=3)
x <- GetTotalSDsfromMean(df_cont)
write.csv(x,"sdsfromMean.csv")

#6 fit a tree for EDA (methood="class" or "anova" )
library(rpart)
dev.off()
EDA_tree <- rpart(price~., data=df, method="anova",control =
rpart.control(minsplit = 20))
plot(EDA_tree, uniform = TRUE,margin = 0.2)
text(EDA_tree, use.n=TRUE, all=TRUE, cex=0.8)
summary(EDA_tree)
round(EDA_tree$variable.importance/100000000)

#7 Model based EDA (RF)
library("randomForest")
rf = randomForest(price ~ ., data = df, mtry = 5, ntree = 500, importance = T)
rf.pred = predict(rf, df)
mean((df$price - rf.pred)^2)
importance(rf,type=1)
varImpPlot(rf)
partialPlot(rf, df, "carat" )
partialPlot(rf, df, "color" )
partialPlot(rf, df, "clarity" )

#trying to find interaction
library(plotmo)
plotmo(rf)

#8 Clustering for EDA (not used)
pairs(scale(df[,2:8])
      pairs(df[,9:14])
      pairs(df[,c(2,3,4,5)])

#9 PCA for EDA (not ued)
library(ggbiplot)
df.pca <- prcomp(df_cont,center = TRUE,scale. = TRUE)
g <- ggbiplot(df.pca, obs.scale = 1, var.scale = 1, circle = TRUE)
plot(g)
points(g, cex = .5, col = "dark red")

#### MODELLING
#recoding the categoricals
df2 <- df
cats <- data.frame((cbind((cbind(with(df2, model.matrix(~ cut + 0)), with(df2,
model.matrix(~ channel + 0)) , with(df2, model.matrix(~ store + 0)))))))
df2 <- data.frame(cats, df)
df2 <- df2[,-c(21,22,23)]

#set up training/test
temp_index <- sample(nrow(df2), round(nrow(df2)*0.3))
df_training <- df2[-temp_index,]
df_test <- df2[temp_index,]

#training/test for log variables
df_training_2 <- df_training
df_test_2 <- df_test
df_training_2$carat <- log(df_training$carat)
df_test_2$carat <- log(df_test$carat)
```

```
library(leaps)
#10 simple linear  model
model1 <- lm(price~. , data=df_training)
pred1 <- predict(model1, df_test)
MSE <- mean((pred1 - df_test$price)^2)
MSE

model1 <- lm(log(price)~. , data=df_training)
pred1 <- predict(model1, df_test)
MSE <- mean((exp(pred1) - df_test$price)^2)
MSE

#looking at the log-log correlation
model1 <- lm(log(price)~. , data=df_training_2)
pred1 <- predict(model1, df_test_2)
MSE <- mean((exp(pred1) - df_test$price)^2)
MSE

#best subset, tforms
model2<-regsubsets(price~.,data=df_training_2,nbest=1, method=c("exhaustive"))
summary(model2)
plot(model2,scale="adjr2")
summary.out2 <- summary(model2)
which.max(summary.out2$adjr2)
summary.out4$which[9,]

model_2 <- lm(log(price)~cutIdeal + channelMall +storeAshford + storeFred.Meyer
+ storeGoodmans + storeRiddles+ carat +color +clarity, data=df_training_2)
pred2 <- predict(model_2, df_test_2)
MSE <- sum((exp(pred2) - df_test$price)^2) /nrow(df_test)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model_2 )
BIC(model_2 )

#backward
model3<-regsubsets(price~.,data=df_training,nbest=1, method=c("backward"))
summary(model3)
plot(model3,scale="adjr2")
summary.out3 <- summary(model3)
which.max(summary.out3$adjr2)
summary.out4$which[9,]

model_3 <- lm(price~cutIdeal + channelMall +storeAshford + storeFred.Meyer +
storeGoodmans + storeRiddles+ carat +color +clarity, data=df_training)
pred3 <- predict(model_3, df_test)
MSE <- sum((pred3 - df_test$price)^2) /nrow(df_test)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model_3 )
BIC(model_3 )

#backward, transforms
model32<-regsubsets(price~.,data=df_training_2,nbest=1, method=c("backward"))
summary(model32)
summary.out32 <- summary(model32)
which.max(summary.out32$adjr2)
summary.out4$which[9,]

model_32 <- lm(log(price)~cutIdeal + channelMall +storeAshford +
```

```
storeFred.Meyer + storeGoodmans + storeRiddles+ carat +color +clarity,
data=df_training_2)
pred32 <- predict(model_32, df_test_2)
MSE <- sum((exp(pred32) - df_test_2$price)^2) /nrow(df_test_2)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model_32 )
BIC(model_32 )

#forward, transforms
model4<-regsubsets(price~.,data=df_training_2,nvmax = 20, method=c("forward"))
summary(model4)
model4$rss
dev.off
par(mfrow=c(1,1))
plot(model4,scale="r2")
plot(model4, scale="adjr2")
plot(model4, scale="bic")
summary.out4 <- summary(model4)
which.max(summary.out4$adjr2)
summary.out4$which[10,]

model_4 <- lm(log(price)~cutIdeal +  channelIndependent + channelMall +storeKay
+ storeChalmers +storeFred.Meyer + storeGoodmans +  storeRiddles+ carat +color
+clarity, data=df_training_2)
pred4 <- predict(model_4, df_test_2)
MSE <- sum((exp(pred4) - df_test$price)^2) /nrow(df_test)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model_4 )
BIC(model_4 )
summary(model_4)

#forward, transforms, interactions
df_training_3 <- df_training_2
df_test_3 <- df_test_2
df_training_3$CaretColor <- df_training_3$carat*df_training_3$color
df_test_3$CaretColor <- df_test_3$carat*df_test_3$color
df_training_3$CaretCla <- df_training_3$carat*df_training_3$clarity
df_test_3$CaretCla <- df_test_3$carat*df_test_3$clarity

model9<-regsubsets(log(price)~.,data=df_training_3,nvmax = 21,
method=c("forward"))
summary(model9)
summary.out9 <- summary(model9)
which.max(summary.out9$adjr2)
summary.out9$which[12,]

model_9 <- lm(log(price)~cutIdeal +  channelIndependent + channelMall +storeKay
+ storeChalmers +storeFred.Meyer + storeGoodmans +  storeRiddles+ carat +color
+clarity+CaretCla+CaretColor, data=df_training_3)
pred9 <- predict(model_9, df_test_3)
MSE <- sum((exp(pred9) - df_test$price)^2) /nrow(df_test)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model_9 )
BIC(model_9 )
summary(model_9)

#stepwise
```

```
library(MASS)
model5 <- lm(log(price)~.,data=df_training_2 )
slm.stepwise <- step(model5,direction="both")
step <- stepAIC(model5, direction="both")
step$anova # display results
summary(step)
pred5 <- predict(step, df_test_2)
MSE <- sum((exp(pred5) - df_test$price)^2) /nrow(df_test)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(model5  )
BIC(model5  )
summary(model5)

#RF
library("randomForest")
rf = randomForest(price ~ ., data = df_training, mtry = 20, ntree = 500,
importance = T)
rf.pred = predict(rf, df_test)
MSE = mean((df_test$price - rf.pred)^2)
RMSE <- sqrt(MSE)
MSE
RMSE

#RF
library("randomForest")
rf = randomForest(log(price) ~ ., data = df_training_2, mtry = 20, ntree = 500,
importance = T)
rf.pred = predict(rf, df_test_2)
MSE = mean((df_test_2$price - exp(rf.pred))^2)
RMSE <- sqrt(MSE)
MSE
RMSE

#RF
library("randomForest")
rf = randomForest(log(price) ~ ., data = df_training_3, mtry = 10, ntree = 500,
importance = T)
rf.pred = predict(rf, df_test_3)
MSE = mean((df_test_2$price - exp(rf.pred))^2)
RMSE <- sqrt(MSE)
MSE
RMSE

#Lasso
library(glmnet)
x<-model.matrix(log(price)~., df_training_2)
y<-log(df_training$price)
grid=10^seq(10,-2, length=100)
lasso <- glmnet(x,y,alpha=1,lambda=grid)
plot(lasso)
#cv
lasso.cv <- cv.glmnet(x,y,alpha=1,lambda=grid)
bestlam <- lasso.cv$lambda.min
#prediction
x_test <- model.matrix(log(price)~., df_test_2)
lasso.pred <- predict(lasso, s=bestlam, newx=x_test)
MSE <- mean((exp(lasso.pred) - df_test$price)^2)
RMSE <- sqrt(MSE)
MSE
RMSE
AIC(lasso   )
```

```
BIC(lasso   )
summary(model5)

#trees! (methood="class" or "anova" )
library(rpart)
dev.off()
EDA_tree <- rpart(price~., data=df_training, method="anova",control =
rpart.control(minsplit = 5))
pred_TREE <- predict(EDA_tree, df_test)
MSE <- mean((pred_TREE - df_test$price)^2)
RMSE <- sqrt(MSE)
MSE
RMSE

EDA_tree <- rpart(log(price)~., data=df_training_2, method="anova",control =
rpart.control(minsplit = 5))
pred_TREE <- predict(EDA_tree, df_test_2)
MSE <- mean((exp(pred_TREE) - df_test$price)^2)
RMSE <- sqrt(MSE)
MSE
RMSE

EDA_tree <- rpart(log(price)~., data=df_training_3, method="anova",control =
rpart.control(minsplit = 5))
pred_TREE <- predict(EDA_tree, df_test_3)
MSE <- mean((exp(pred_TREE) - df_test$price)^2)
RMSE <- sqrt(MSE)
MSE
RMSE
```