Stat 627 Project Report

Aaron Niecestro

December 8, 2019

There have been lots of models, reports, and ideas of how people think what should determine the price of a house. This report is similar to many written and well-established articles already published, but the difference I this article is that the pricing property models are only focused in the District of Columbia. I have read lots of literature about property pricing models that cover all over the United States and worldwide, and have even run some models in linear, logistic, Bayesian regression (see appendix table 1) to see if I can gain more insights into what determines the price of a property and whether a property is worth the price the sellers offer. I am a new statistician, so making models and looking over data to gain insights into it is in my job description. However, I yet to use the new machine learning techniques and tools I have recently learned to help answer my research questions and gain more insight into the data I downloaded. That is my purpose behind this paper, I wish to explore and use the new material I learned to better inform myself.

In the Bayesian model I created (see appendix table 1), I learned that having a grade grouping intercept was important, since it allowed explain some of variance in the Bayesian model. In the logistic model I created I learned which variables best explain the odds of what makes a property in DC qualified to sell. This now brings to some of questions of interest. The questions I interested in answering are as follows: what are the best variables to help explain a property pricing model, does analyzing classification pricing models help better explain the data then the linear, logistic, and Bayesian models did, and how good are the classification pricing models, namely can I trust the results?

The data for the following analysis and models were collected from Kaggle DC Residential Property[1]. Following the basic statistical guidelines, after I downloaded the data, I started to assess what each column in the dataset was trying to represent. This was a rigorous task since one of the columns in the dataset, for example qualified, had no description to them. I had no idea what exactly this column meant or was trying to explain since there was no data key or description in on the Kaggle website. To fix this problem I tried to reach out to the people who put up the dataset on Kaggle and I got answer a couple of months later explaining that the qualified column meant whether a property in DC was worthy of selling or not. Although I was happy the dataset providers answered my question, I did hold reservations on fully trusting the provided answered. So, I did my own research and concluded that I should assume the qualification determination was decided by the DC housing guidelines and the property passing all the inspections. I have kept this assumption constant throughout the entire analysis.

The dataset from Kaggle had originally 158957 rows and 49 columns, but I do not wish to look at all that data, so I filtered the columns and observations to be more believable and easier to work with. When I say I made this dataset more believable, I mean that I filtered the data so that property would not be sold for less than $10,000 and the number of rooms the property had was less than or equal to forty rooms. I know that this was the best way to make the

[1] https://www.kaggle.com/christophercorrea/dc-residential-properties

data seem more believable, and it was not the only modification I made, but it was the only way I think of during these past couple of months.

The final smaller dataset I ended up working with, which I later called the final dataset, was the dataset I used for all of my analysis and model creation. This final dataset had 57610 rows and 21 columns, but I did not stop there. Since most of my models were about predictions, I broke the dataset in half to create training and testing datasets. I felt splitting the final dataset at random into 50% portions for the training and testing datasets was the best because I believed that this would ensure that my predictions would be the most accurate and trustworthy. If the data was broken up into any other kind of portion, I believe that my results might be biased because one of the datasets, training or testing, has more observations then the other.

Unfortunately, when I was looking through the original dataset from missing dataset there was a lot of missing data. So much that if I tried to case-wise delete on only the variables I have above then I would be left with only 20% of the original dataset. For Example, each column besides the ID column had missing rows between at least 1 observation to over nearly seventy percent missing. I tried to fix and remedy so missing data entries by substituting 0 into the missing columns. I understand that this is not the best solution, it was one of the few remedies I could think of when I started my analysis.

Since the dataset had 49 columns, and this is too many variables to explain, this report will describe only the variables used in the models and analysis. The models used the following response and predictor variables:

1. Price_10K – the price of most recent sale in 10 thousand dollars. The average price of property in DC in ten thousand dollars was $57.725, and the range was from $1.027 to $910.000 (in $10,000).
    a. This is the response variable in every one of my models.
2. Bathrooms – the number of full bathrooms the property has. The average number of bathrooms was 2.522, and the number of bathrooms ranged 0 to 11.
3. Rooms, the number of rooms the property has. The average number of rooms was 7.438, and the number of bathrooms ranged 0 to 30.
4. Bedrooms, the number of bedrooms the property has. The average number of bedrooms was 3.422, and the number of bathrooms ranged 0 to 15.
5. Kitchens, the number of kitchens. The average number of kitchens was 1.247, and the number of kitchens ranged 0 to 6.
6. Fireplaces, the number of fireplaces. The average number of fireplaces was 0.6422, and the number of fireplaces ranged 0 to 9.
7. AYB.age, the number of years since the earliest time the main portion of the building was built. The average number of AYB.age was 84.58 years, and the number of AYB.age ranged 1 to 254.00 years.
8. EYB.age, the number of years since an improvement was built more recent than actual year built. The average number of EYB.age was 49.24 years, and the number of EYB.age ranged 1 to 86 years.
9. Remodel.age, the numbers of years since the property was last remodeled. The average number of remodel.age was 8.398, and the number of remodel.age ranged 0 to 139.
10. Condition, a categorical rating of the condition of the property.
11. AC, a categorical rating of whether a property has AC or not.
12. Grade, a categorical rating of the grade of the property.

Now that the data cleaning was completed, I had to choose which models would best answer my research questions. The model and statistical tools I believed would help explanation my research question and give me more insights into my data were as follows:

1. Linear regression along with hypothesis testing, and variable selection,
2. Polynomial regression based off of the final linear regression model,
3. K Nearest Neighbor (KNN),
4. Linear discriminant analysis (LDA),
5. Quadratic discriminant analysis (QDA),
6. Ridge regression,
7. Lasso regression,
8. PLS
9. PCR
10. Regression Trees
11. Support Vector Machines

It should be noted that throughout these models and statistical tools I made diagnostic plots and used cross-validation to make sure that my results were valid and calculate the mean squared error.

The first type of model I created was linear regression. I choose to use linear regression because I wished to model the predictions of price in 10 thousand dollars for properties in DC. I wished to get a basic idea of which variables are best used to price the price of a property in DC and linear regression is what first came to mind. I thought that if I could make a good linear regression model then I could use this model as a foundation for the rests of my models. To find the final multiple linear regression I used f-tests, partial f-tests, lack of fit tests and variable selection (stepwise AIC, stepwise, BIC, forward selection, and backward selection). For example, one of the partial f-test hypothesis tests I used was as follows:

$$H_0: Beta_{Excellent} = Beta_{Fair} = Beta_{Good} = Beta_{Very\ Good}\ 0$$

$$H_A: \text{At least one Beta} \neq 0$$

The result I obtained from this partial f-test was that the p-value less than $2.2x10^{16}$. With my p-value being less than 0.05, then I could conclude that I am able to reject the null hypothesis, so I should keep the multiple categorical dummy variables for condition. It should be noted that this is just one of many the hypothesis tests I used. When using variable selection, I found that the lowest AIC was 744905.4, and lowest BIC was 32182.

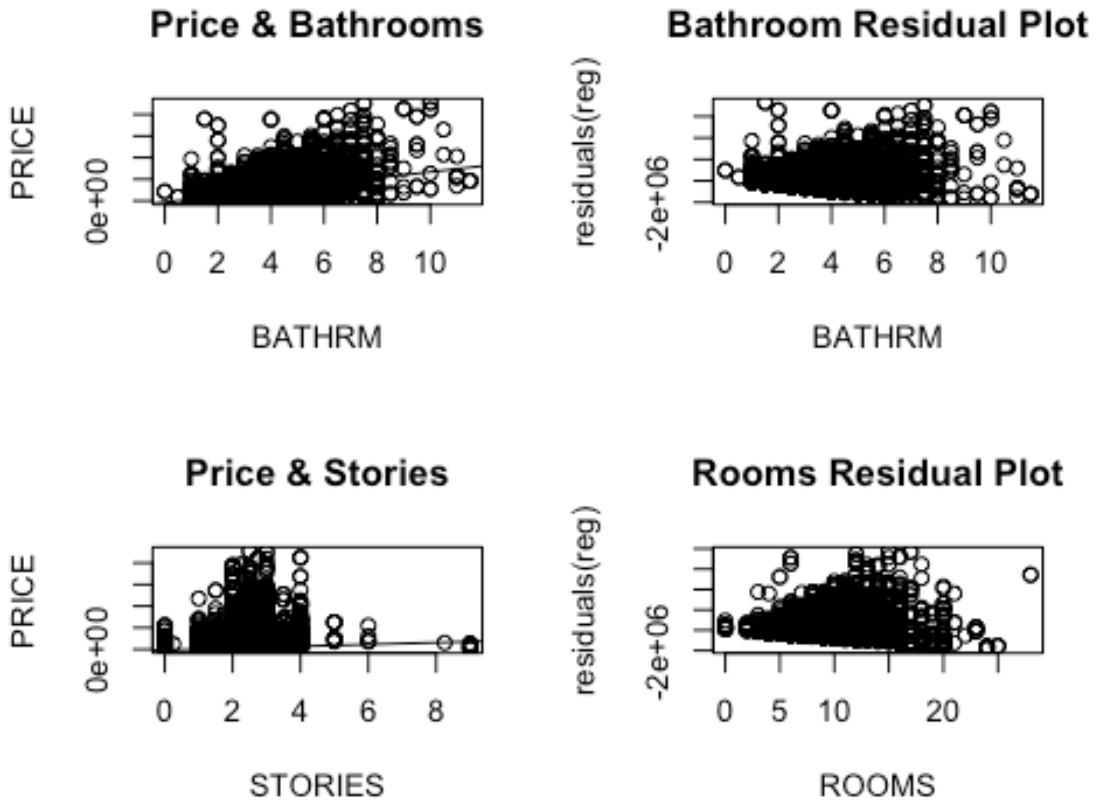
The best model I was able to obtain from the multiple hypothesis tests and variable selection was as follows:

$Price_{10K}$ = -11000 + 10.34*Bathroom + 1.087*Rooms + 3.289*Bedrooms – 0.2069*Stories -
9.1112*Qualified=Unqualified + 3.472*Grade=Average + 21.99*Grade=Excellent
+ 140.1*Grade=Exceptional + 32.60*Grade=Fair + 1.476*Grade=Good +
42.73*Grade=Superior + 8.358*Grade=Very Good – 4.406*Kitchens +
9.271*Fireplaces + 13.62*Ward 2 + 10.46*Ward 3 – 4.872*Ward 4 –
4.266*Ward 5 + 7.233*Ward 6 – 2.612*Ward 7 – 17.6*Ward 8 -
33.34*Latitude – 159.7*Longitude + 0.2911*AYB.age – 0.3354*EYB.age +
43.34*Condition=Excellent + 8.248*Condition=Fair + 7.026*Condition=Good

+ 24.81*Condition=Very Good

Overall the model was very good, but the model still has room for improvement. What I mean is that the residual standard error was 33.49, but it could be lower. The lower the residual standard error the better the model. However, it should be noted that getting this low of a residual standard error on 28774 degrees of freedom is very good, it is just that I believe the lower the error is the better the model. Also, the adjusted R^2, which is at 64.99%, could be a lot higher. Personally, I would have liked an adjusted R^2 of 90% or higher, but considering the degrees of freedom is large , I will let this slide for now. This model could have some interaction terms and transformation to the variables to make it better, but I had a reason behind leaving this model as it was before exploring more. The reason I left this model as a basic linear regression model was because I wish to create a foundation and base for the other models and tools, I will be using later in my analysis. The silver lining with this model was that the only not statistically significant variable in the above model was Ward 7, and Stories. I am very thrilled that considering the number of variables I have in this model that nearly all my predictors are statistically significant, especially bearing in mind that getting a model with all statistically significant variables is near practically impossible.

Now that my basic linear regression model foundation was completed, I was able to more onto building more complicated models such as a polynomial model. The reason I thought of creating a polynomial model after making a basic linear model was because the linear diagnostics model created above was not normally distributed, form the residual and diagnostics, it seems that the variables would need transformations, if I wished to improve the model to make the model better. The way I went about creating this polynomial model was by finding and comparing the best powers for the quantitative variables. I compared the adjusted R^2, Mallow's CP and Bayesian Information Criteria (BIC) powers with one another and analyzed which power would be best to use. I do not believe that just because a model with a higher adjusted R^2 can explain the model better, the adjusted R^2 would be the best to use. It is better to test everything and double check your results then just assuming something without further testing. this would make the best. For example, I found that for AYB.age the best power was 5. I found this power by using the maximum adjusted R2, minimum mallows cp and minimum BIC, which all showed the power to be 5.



In the above top left plot, we can see that a normal linear line does not work very well, so maybe I should think of using transformation. Some of the transformations I was thinking of using were logging the variables, square roots, fractional powers, and powers. It seems that as the number of bathrooms increases the price of a property in DC will increase as well if everything as remains constant. In the top right plot, we can see in the residual plot above that the residuals are not normally distributed around 0. It looks the residuals are in a side-ways cone shape. Further testing for outliers will need to be implanted since the residuals are not so bunched together or following a uniform pattern. In the bottom left plot above we can see that a normal linear just like the top left plot above it. It seems that transforming the variables has become a more concrete idea I will need to implement in my later analysis. Again, some of the transformations I was thinking of using were logging the variables, square roots, fractional powers, and powers. It should be noted that I could of filter the data so that not stories were greater than 5 now because most residential properties are not selling a building with more than 5 stories. But I felt that I already did enough filtering to the data, anymore might increase bias, and it might just be possible in DC, so better to be cautious and note the abnormality. It seems that as the number of stories goes up the price increases until about 4, where prices tend to drop as long as all the other variables remain constant. In the bottom right graph, we can see that the residuals for rooms are not constant around 0 and almost in a side-ways cone shape just like the plot above it.

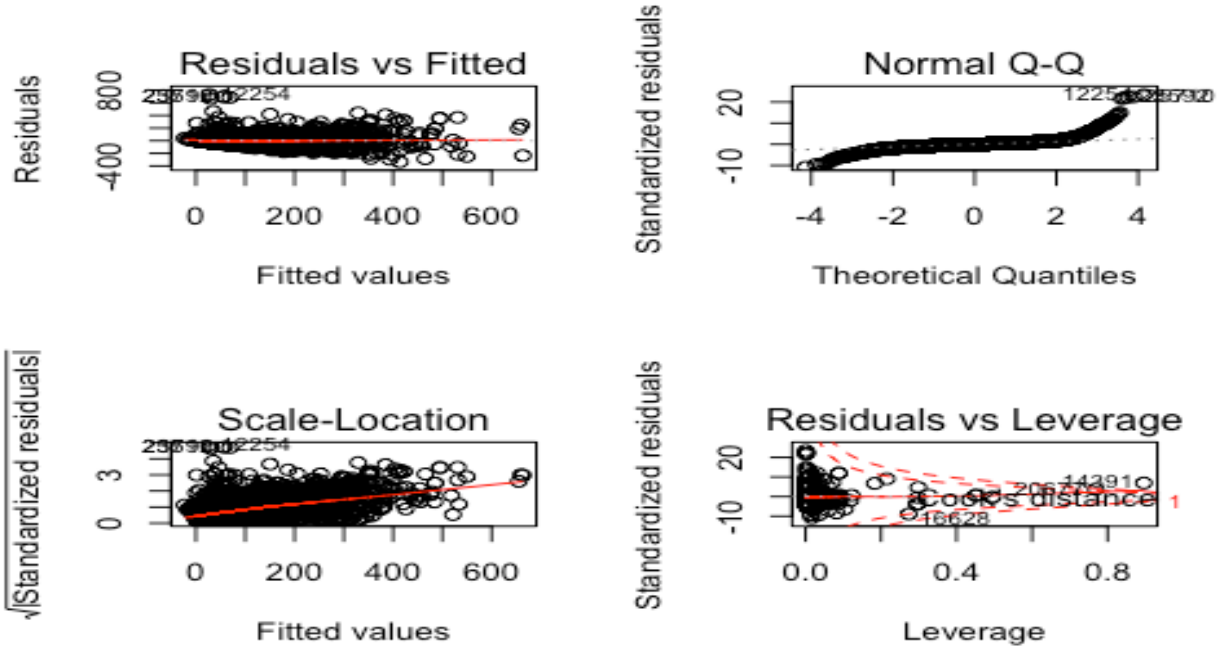
Below is the final polynomial model I created. The final polynomial model I created was as follows:

$Price_{10K}$ = -10630 + 1107*Bathroom + 1094*$Bathroom^2$ + 225.5*$Bathroom^3$ – 389.1*$Bathroom^4$ - 231.2*$Bathroom^5$ – 160.4*$Bathroom^6$ – 159.5*$Bathroom^7$ + 31.89*$Bathroom^8$ + 278.2*Rooms + 501.1*Bedroom – 234.1*$Bedroom^2$ – 395.3*$Bedroom^3$ – 74.66*$Bedroom^4$ – 13.13*$Bedroom^5$ – 13.33*$Bedroom^6$ – 143.2*$Bedroom^7$ + 3.925*$Bedroom^8$ - 9.335*Qualified=Unqualified - 1.257*Grade=Average + 14.75*Grade=Excellent + 82.41*Grade=Exceptional + 23.65*Grade=Fair + 3.827*Grade=Good + 23.41*Grade=Superior + 9.918*Grade=Very Good – 127.4*Kitchens + 996.5*Fireplaces + 275.9*$Fireplaces^2$ – 82.18*$Fireplaces^3$ – 3.944*$Fireplaces^4$ + 18.71*Ward 2 + 7.069*Ward 3 – 6.353*Ward 4 – 3.714*Ward 5 + 8.107*Ward 6 – 5.006*Ward 7 – 15,88*Ward 8 - 17.96*Latitude – 147.7*Longitude + 5031*AYB.age – 3019*$AYB.age^2$ + 1731*$AYB.age^3$ - 409.7*$AYB.age^4$ - 5288*$EYB.age^1$ + 2399*$EYB.age^2$ – 595.4*$EYB.age^3$ + 207.9*$EYB.age^4$ – 308.2*Remodel.age – 99.86*$Remodel.age^2$ + 111.9*$Remodel.age^3$ – 44.05*$Remodel.age^4$ - 52*Condition=Excellent – 1.942*Condition=Fair – 0.7768*Condition=Good – 17.18*Condition=Very Good + 28.18*Bathrooms*Condition=Excellent – 4.799*Bathrooms *Condition=Fair – 3.635*Bathrooms *Condition=Good – 13.96*Bathrooms *Condition=Very Good

```
## Residual standard error: 31.12 on 28745 degrees of freedom
## Multiple R-squared:  0.691,  Adjusted R-squared:  0.6904
## F-statistic:  1089 on 59 and 28745 DF,  p-value: < 2.2e-16
```

This model above is a little better than the basic linear regression model I had earlier. The adjusted R^2 in this model 69.17% and the residual standard error is slightly lower at 31.12. this model does not have statistically not significant variables, $Bedroom^4$, $Bedroom^8$, $Fireplaces^3$, Condition=Fair, Condition=Good, and the interaction term of Condition=Good and Bathroom but considering that having more variables usually means increasing the number of statistically not significant, this is all normal. Again, although I would have liked a R^2 that was greater than 90%, I will take what I can get.

Although my final model was completed, and shown above, that does not mean I am done. The next thing I did was check the model diagnostics and see if I have outliers in this polynomial model like I did earlier in my analysis with the linear regression model. One can see some of the diagnostics plots I made for the polynomial model below.



In the top left plot (Residuals vs Fitted) we can see that we have outliers and the residuals are like a side cone shape. One can also see that there are some outliers in this plot in the upper left-hand corner, but further testing by the outlier test will not to be done to see if they are real outliers or not. In the top right plot (Normal Q-Q), we can see that the model is not normally distributed at the tails and that we have outliers. One can also see that there are some outliers in this plot in the upper right-hand corner, but further testing by the outlier test will not to be done to see if they are real outliers or not. In the bottom left plot (Scale Location), we can see that we have the outliers repeating here as well, and that we do not have homoscedastic. The red line is not straight and is in an upper line, and that we have outliers in the upper left corner of this plot. In the bottom right plot (Residuals vs Leverage), we can see that we definitely have outliers like the earlier plots show and that we do not have homoscedastic. It seems that most of the standardized residuals follows the dotted red line but the ones that do not are outliers.

It should be noted that I was unable to test the model for normality using the Shapiro tests since the training dataset had more than 5,000 observations. It seems that R will not use the shapiro test when the number of observations exceeds 5,00 observations, so I tested for normality by the graphs above. The result for normality was that the data and model are not normally distributed. After checking the model for homoscedastic, normality, constant variance, and checking the residuals I tried to see if there were outliers in my data. I found that were outliers in my training set data when I used the outlier test. Some of these outliers were at observation number 12254, 23592, 25710, and there were several others as well. It was surprising to find out that the outlier test in R that there were only 20 outliers in the data. I thought there would be a lot more considering the training data alone had more than 27,000 observations in it. I usually assume that more observations would lead to more outliers, but I guess I got lucky in this model, and analysis. I took these outliers out of the data and continued with the rest of my analysis.

One of my concerns when making the linear and polynomial models was multicollinearity, but it seems that was not an issue since the variance inflation factor (vif) for

the variables showed all the results that were less than 5. This now concludes my analysis for the polynomial model I created.

The next statistical algorithm I used was K nearest neighbor (KNN). I used KNN because to split my data into several classes and predict the classification of these new classifications. I wished to use the KNN algorithm is predict the classification for overpriced and moderately priced properties in DC. I did this because I needed to know how well I was the classifications and are the above linear and polynomials models I created any good. If the classification mean square error was below then I know that more work and analysis had to done to my models and research questions.

Now I am going to talk about some of the ways I used KNN to predict the 2 classifications. However, before I began discussing my analysis results, it should be noted now that this is only one of many ways, I used KNN in my analysis project. To begin my KNN analysis I had to come up with classifications to predict. So, I choose to use the mean of the training set observation for price to break up the two classification. So, if price in ten thousand dollars was smaller than the mean of price in 10 thousand was smaller, this as classified as reasonable. Whenever the price was greater or equal to the mean of price in 10 thousand was smaller, this as classified as overpriced. Some of the categories I used for KNN was K =10, K=20, and K=1000 along with a bunch of other K's. I also used KNN with 3 categoreis based off of the 1st quartile, mean, and 3rd quartile and tried K =10, K=20, and K=1000 along with a few others K's before attempting the KNN for loop for k=1 to k=100. I did not submit the code based on the instructions of the project.

The following results are based off of the two categories (reasonable and overpriced) are based off the value close to the mean of PRICE_10K. When the PRICE_10K < 57.420 it is labeled as Reasonable, and PRICE_10K > 57.420 is labeled as expensive. The KNN mean squared error for k=5 is 0.5459469. We can se that the minimum class rate for the KNN for loop for K from 1 to 100 is K=1. The mean squared error for k=1 form the KNN for loop is 0.5277209. It seems that the mean squared error fluctuates up and down as K increases but overall the mean squared error goes up as K increases.

After KNN, I decided to use linear discriminant analysis (LDA). The reason I chose to use LDA is because I needed to find a new feature space to project the data in order to maximize classes separability. LDA is another classification tool I wished to use to help me explain my analysis better. I found that the best LDA model is with priors at 0.5 for each of the 2 classes.

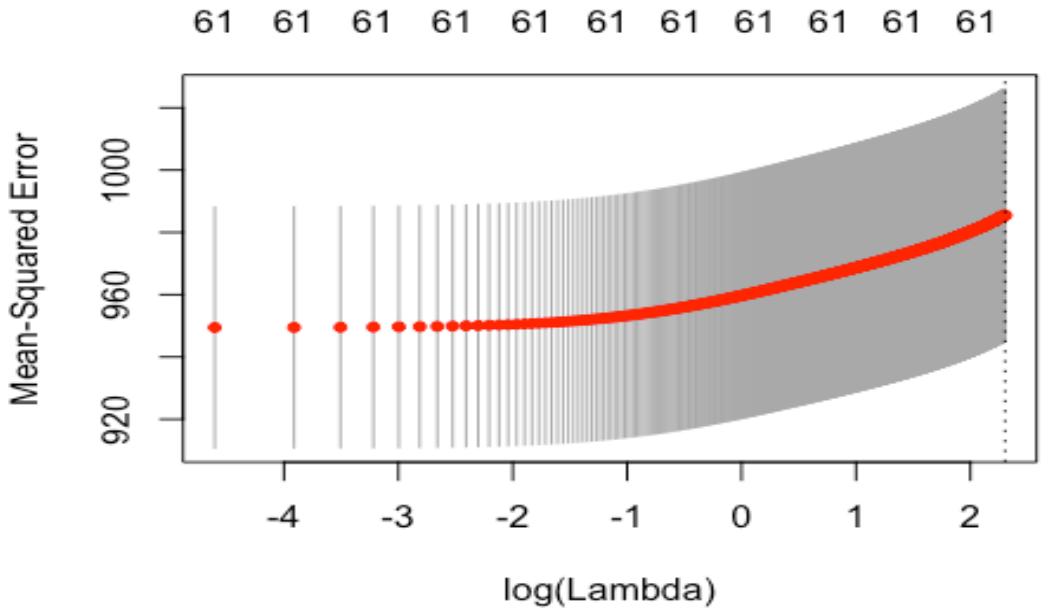
The two categories are based off the value close to the mean of PRICE_10K. When the PRICE_10K < 57.725, it is labeled as Reasonable, and PRICE_10K > 57.725, it is labeled as expensive. My results show me that LDA without priors' correct classification rate is 0.8093387. My results show me that LDA with priors' correct classification rate is 0.8050338. It seems taht with priors has slightly lowered the correct classification rate.

Of course, after making the LDA models and finding the correct classification rate, I need to use QDA. I believe that if one wishes to use LDA then one should use QDA as well because of the bias variance tradeoff between these two tools. Also, QDA is a little more flexible than LDA, so I thought that my results would show me something interesting, even though LDA has lower variance.

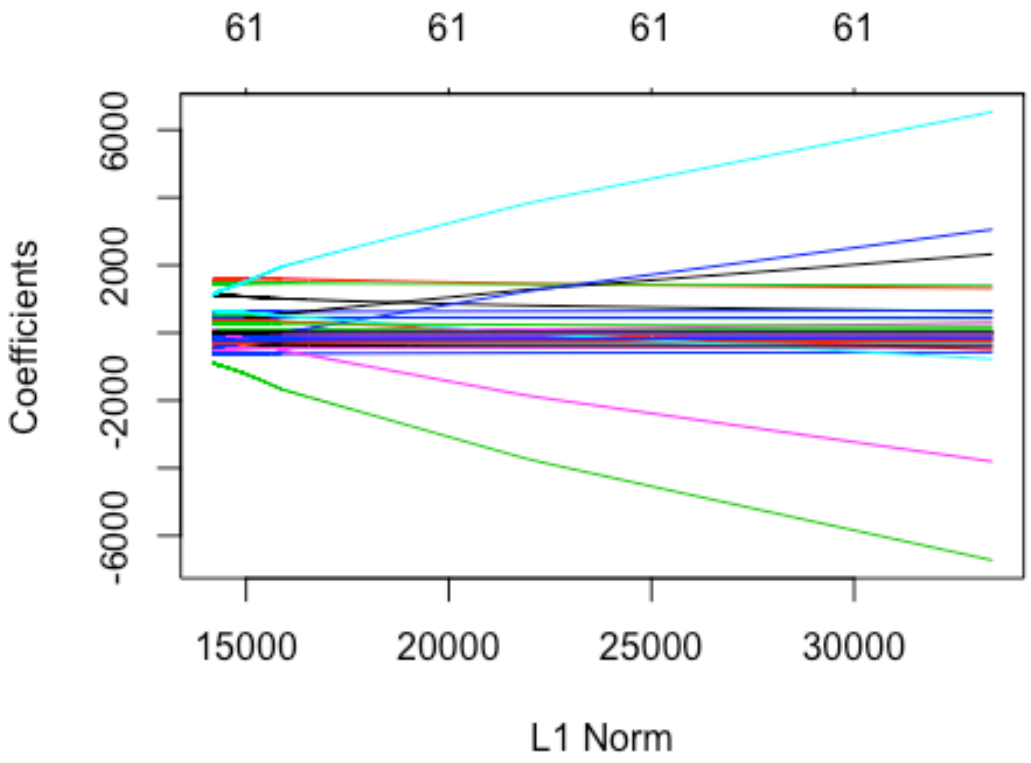
The two categories are based off the value close to the mean of PRICE_10K. When the PRICE_10K < 57.725 it is labeled as Reasonable, and PRICE_10K > 57.725 is labeled as expensive. My results show me that QDA without priors' correct classification rate is 0.8060753. My results show me that QDA with priors' correct classification rate is 0.7967367. the best

priors I could find that would get me the highest classification rate was 0.5, 0.5. It seems that with priors has slightly lowered the correct classification rate.

After using LDA and QDA, I decided to use ridge regression. It seemed to me the most logical thing to do because I wished to know how my final polynomial would react. I used ridge regression because I wished to see which variables coefficients ridge regression thinks should shrunk and go to 0. I thought if I knew which variables ridge regression believes should be eliminated then I might be able to get a better linear and polynomial model. So, I used ridge regression with the polynomial model I created earlier.

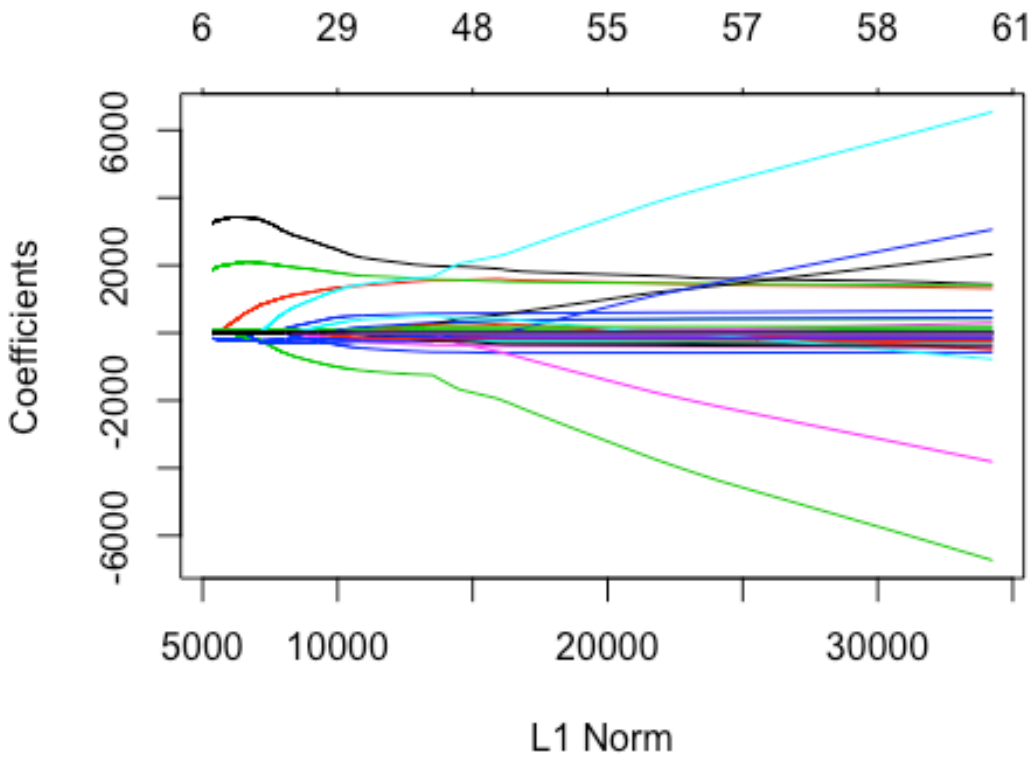


It seems that for the log(lambda) plot that the mean squared error increases as the log(lambda) increases. The minimum cross validation lambda is 0. So, it seems that ridge regression based off the minimum cross validation lambda would like most of the variables to go to 0. It is interesting that the standard deviation lines (gray lines) are very large in the above plot. Each one of these standard deviation lines take up about half of the plot. I would have thought they would have been smaller. More analysis will need to be completed to look into the reasons behind this.

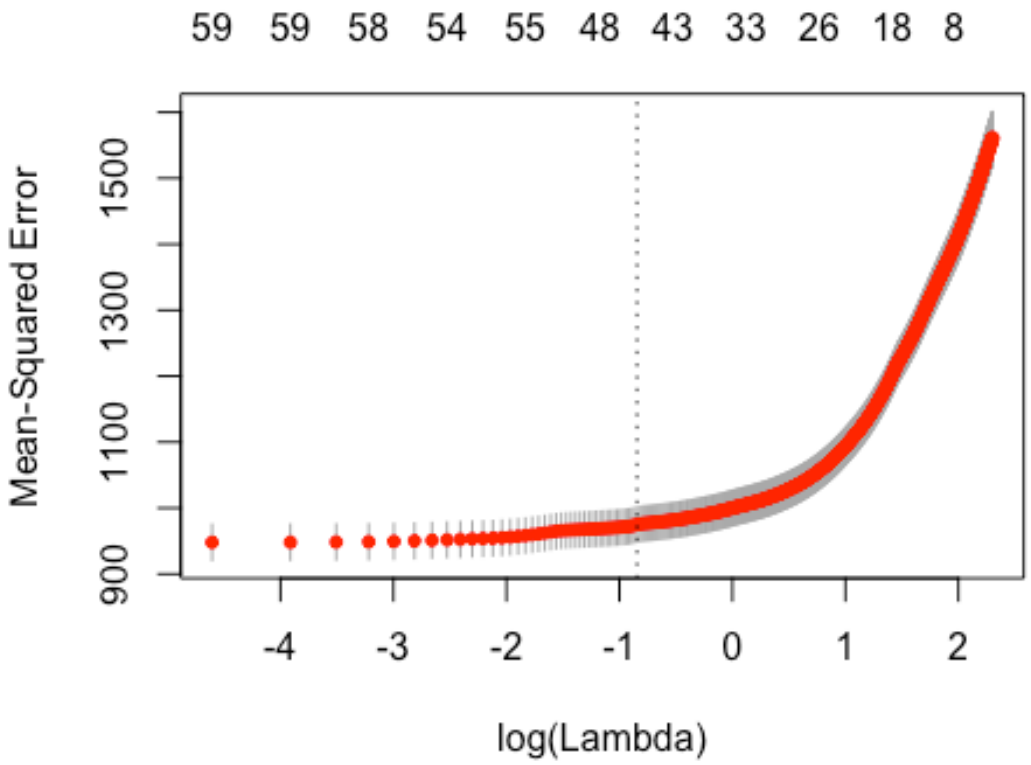


The L1 norm plot is very interesting. We can see that as the L1 Norm decreases and the Coefficients get closer to 0, rigdge regression will try to get a lot of the predictor's variables to 0. Some predcitors will get to close to 0 much faster than other as we can see form the light blue, light gree and pink outer lines. This is a very colorful graph. In the future I have to figure out if there is a way to set the colors to certain variables because then I could tell from this graph which variables ridge regression believes goes to 0.

After Ridge regression I decided to use LASSO regression. I was looking into shrinkage operators and one cannot use ridge regression without using LASSO regression. The reason I used Lasso regression was because I wanted to analyze which variable coefficients lasso regression believes should shrink to 0. Unlike with ridge regression, which shrinks the coefficients close to 0, lasso will shrink the coefficients to 0. I need to observe the difference between ridge and lasso because these differences would be very important to my analysis. I also wished to compare and contrast the results from these two shrinkage operators.

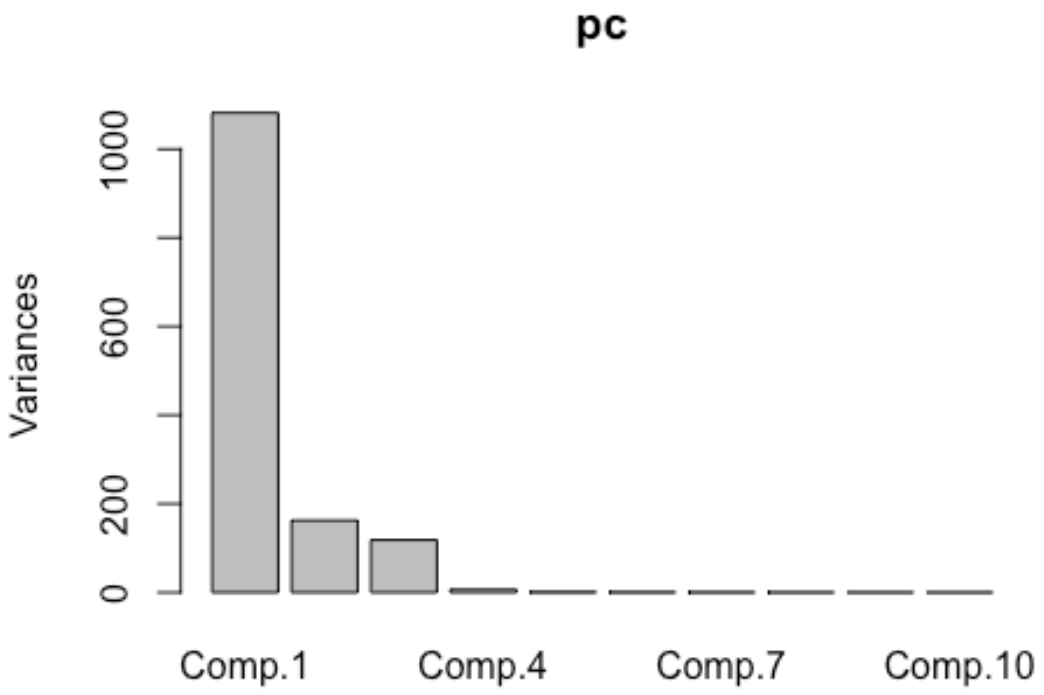


This first plot for lasso is interesting. It seems that as the L1 Norm decreases and the Coefficients get closer to 0, the coefficients will try to get to 0. However, it is really interesting that some of the lines in the L1 norm go up and then decrease right before the L1 Norm right before the coefficients should shrink to 0. This plot above is a little different to the ridge regression plot since most of the coefficients in the ridge regression plot would just shrink to 0, but not increase then decrease before 0.

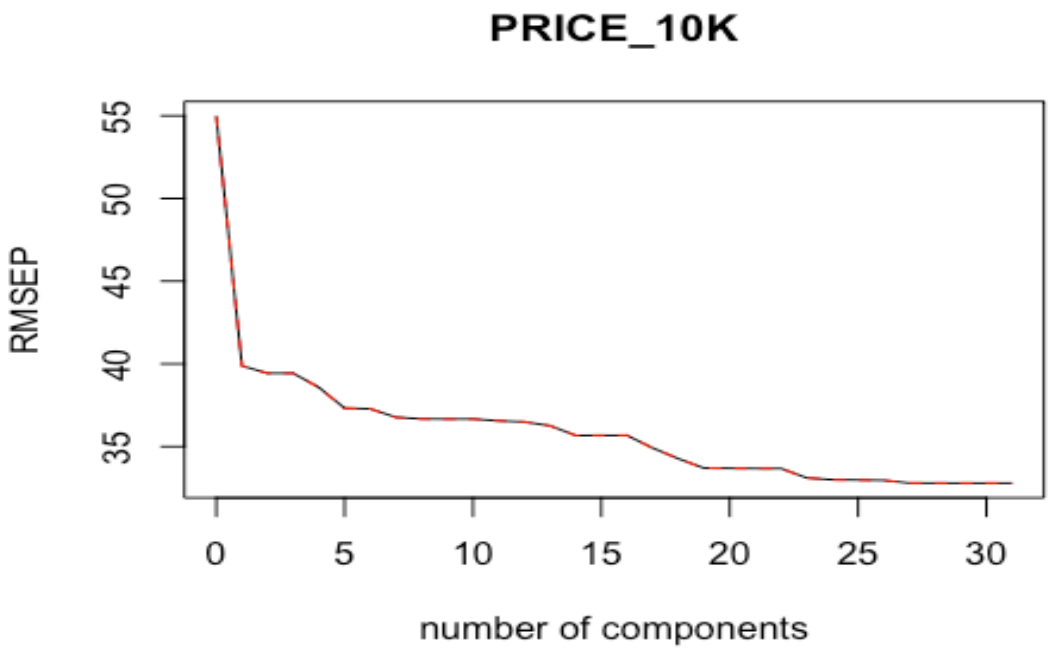


It seems that as the mean-squared error increases as the log(lambda) increases, but unlike with ridge regression the standard deviavation (gray lines from red dots) is a lot smaller. The minimum cross validation lambda is 0. The lasso the test mean squared error, estimated by the validation set approach is 624946878110, which is very large, considering how big the model this does make some sense. I like the lasso plot above a little better than the ridge regression plot earlier. I was expecting the standard deviations lines (grey lines) to be similar to the ones above for the ridge regression plot. It is interesting though that as the log(lambda) increase the grey lines decrease, but I guess that should be expected.

The next type of machine learning tools I decided to use is Principal component regression (PCR). Principal component regression is a regression analysis technique that is based on principal component analysis (PCA). Typically, it considers regressing the outcome on a set of predictors, or explanatory variables based on a standard linear regression model, but PCR uses PCA for estimating the unknown regression coefficients in the model. I wanted to make sure that there is no multicollinearity issue, so I used this model. I also wanted to see the number of components PCR would think be best to use.

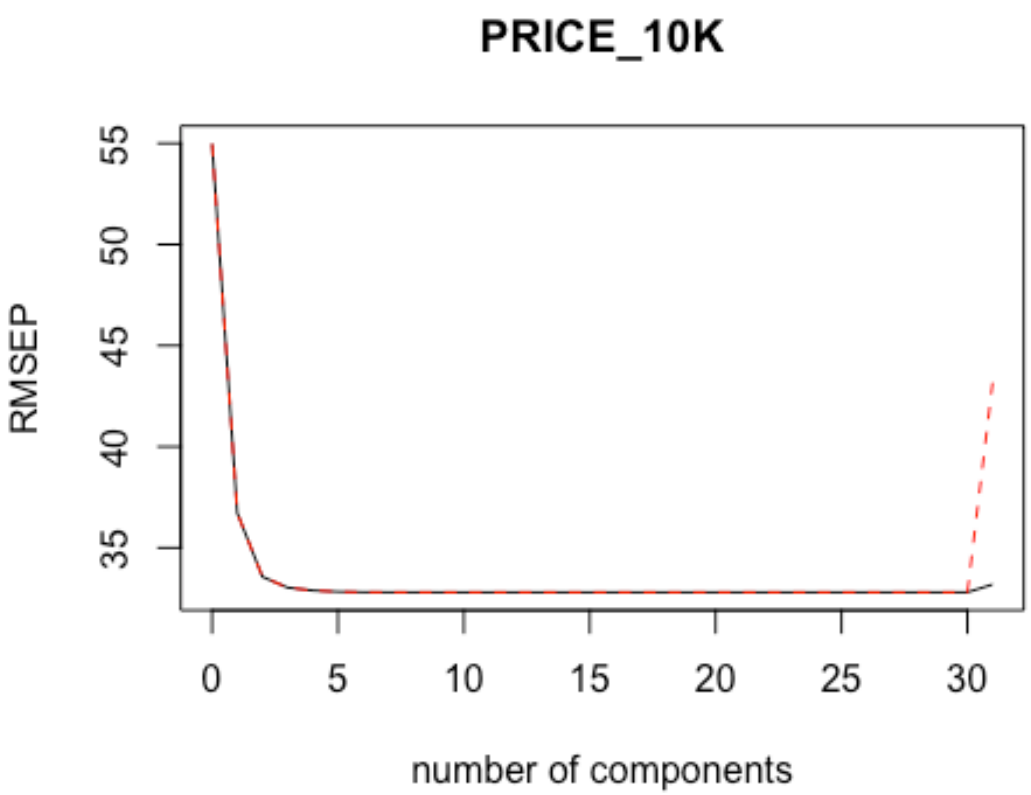


We can see in the plot above that the first component variance is very large and decrease sharply by the second component. It is interesting that after the 3rd composition there is very little change to almost no visible change in the variance.



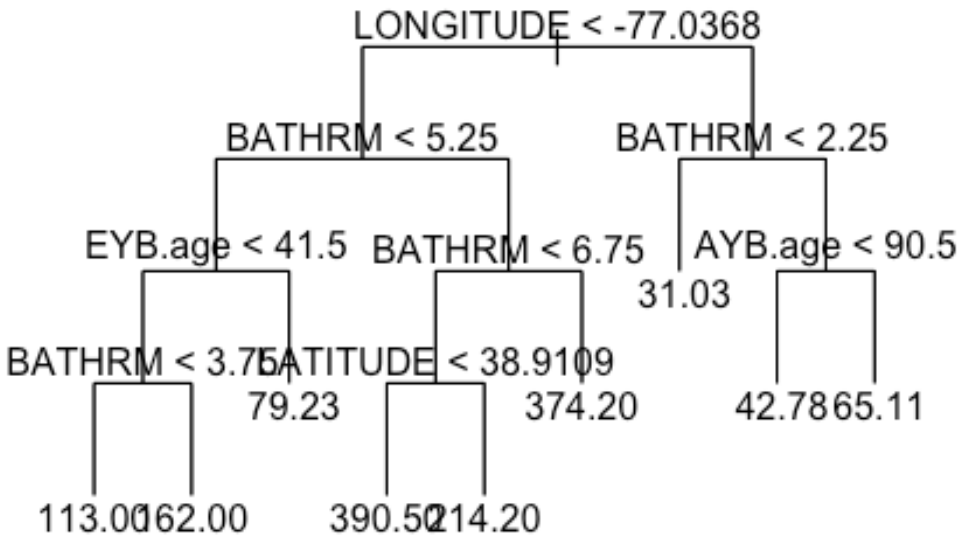
It seems as the number of compoenents increase the Cross validation and adjusted cross validation decreases. The smallest Cross validation and adjusted cross validation is at 29 compoenents and stays that way unitl the last 31st compoenent. We can see in the pc plot that by the second component the variance decreases by five times, and by the 5th computation we can see that the variance is very small and does not change much. In the second plot of RMSEP and number of compoenents, it seems that by the 27th component the RMSEP does not decrease too much, or at least what I can see.

Following principal component regression (PCR) I decided to use Partial least squares regression. Partial least squares regression (PLS) is a statistical method that bears some relation to principal components regression; instead of finding hyperplanes of maximum variance between the response and independent variables, it finds a regression model by projecting the predicted variables and the observable variables to a new space. Because both the X and Y data are projected onto these new spaces. PLS regression is particularly suited when the matrix of predictors has more variables than observations, and when there is multicollinearity among X values. However, I used PLS not because I have more variables then observations but because I was worried that there might be multicollinearity among the X variables.



It seems as the number of compoenents increase the Cross validation and adjusted cross validation decreases. The smallest Cross validation and adjusted cross validation is at 29 compoenents and stays that way unitl the last 31st compoenent. A lot mor eof the variance is explined as the number of components increase the the training dataset. The largest variance explained in the training dataset is at the 31st component with X having 102.3 variance explained, and Price_10K having 63.0 variance explained. In the plot of RMSEP and number of compoenents, it seems that by the 3rd component the RMSEP does not decrease anymore, or at least what I can see. It is interesting that at the 30th component the black line for RMSEP increases and the red dotted line also increases but a less than the black line.

Once I completed the partial least squares analysis, I moved onto making regression trees. Regression Trees are used when the decision tree has a continuous target variable. I used regression trees mainly to visualize each step of tree, which can help with making rational decisions on variable importance and give priority to a decision criterion. I liked to make regression trees because making a decision based on regression is much easier than most other methods. Since most of the undesired data will be filtered outlier each step, you have to work on less data as you go further in the tree. It is easy to prepare a regression tree. Another reason I like regression trees is because it can be represented on a simple chart or diagram.

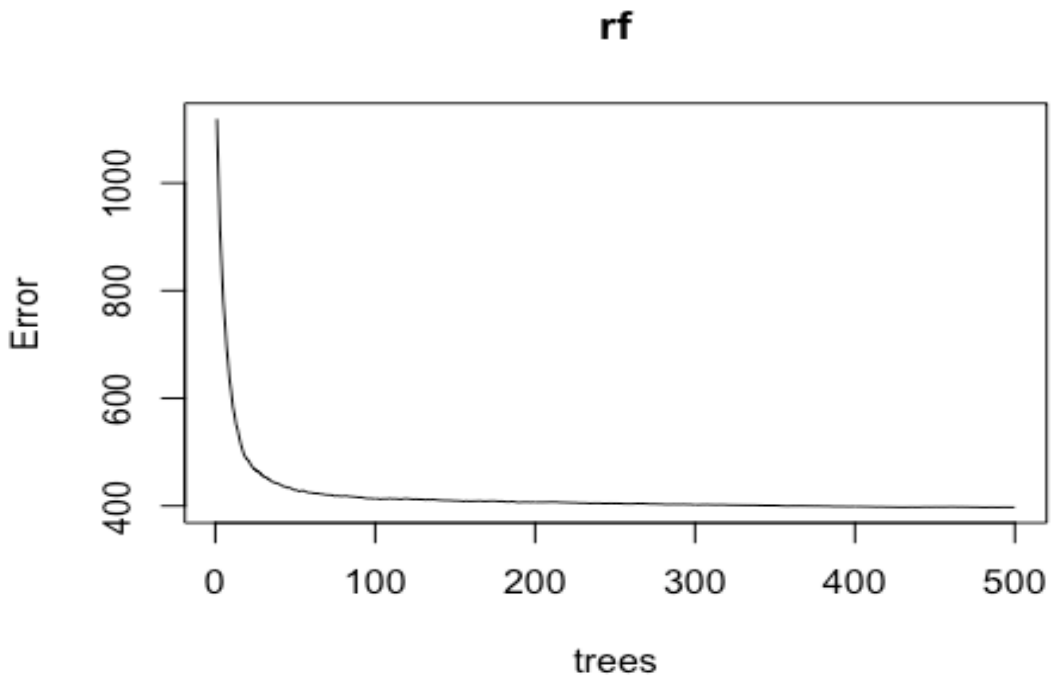


In this tree regression tree plot we can see that we have 11 terminal nodes and the graph is a little messy. However, there was nothing I could do to make this graph any neater, so I am sorry for the long variable names and output. I will look into making this regression tree more visualizing pleasing and when I come up with a better-looking tree, I show that new tree, so until then please bear with what I have above. The variables used in this plot were Longitude, Bathrooms, EYB.age, Fireplaces, AYB.age, and Latitude. I will not provide an example of how this plot works. For example, if the longitude as greater than -77.0391 and bathrooms were greater than 2.25, and AYB.age was greter than 95.5 then this tree predicts that the property will be worth 67.33 ten thousand dollars, or 673,300 dollars.

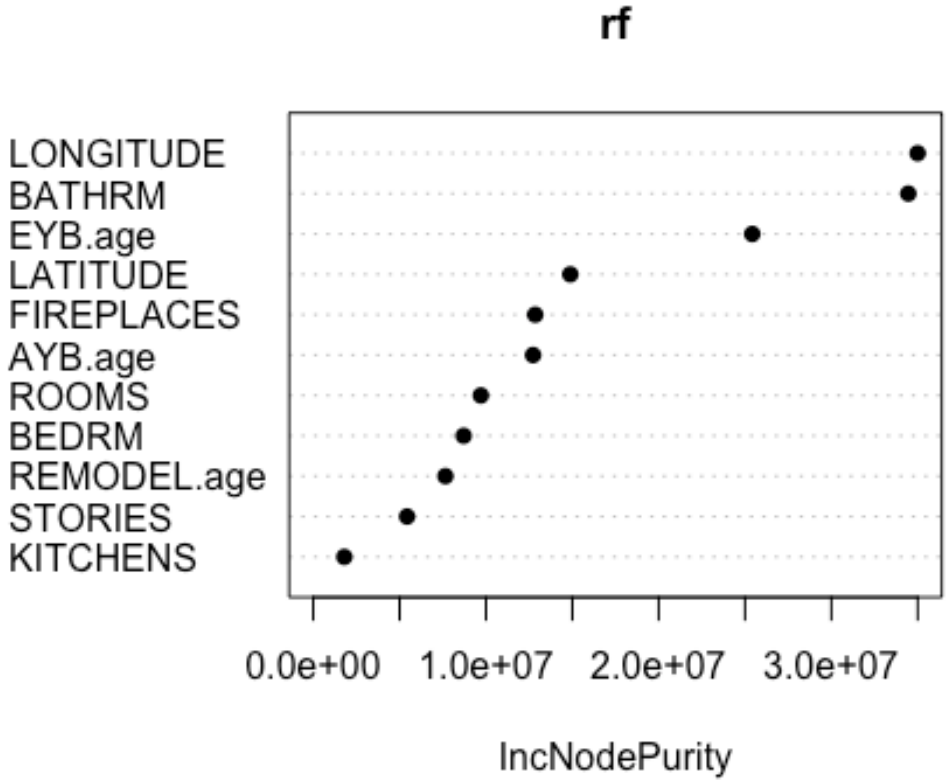
After making this initial regression tree comes pruning the tree. The reason I used pruning is because it is a technique in machine learning algorithms that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier and improves the predictive accuracy by the reduction of overfitting. It is just like gardening, if one does not prune and take care of their garden will outgrow its bound and one will not be able to see their once beautiful creation anymore.

After pruning the regression tree that one sees above, I decided to make a random forest model. The reason I choose to make random forests models is because it was the next step after making the regression tree. Different kinds of models have different advantages. The random forest model is very good at handling tabular data with numerical features, or categorical features with fewer than hundreds of categories. Unlike linear models, random forests are able to capture non-linear interaction between the features and the target. As one read earlier then

data and models, I created were non-linear so making random forests I felt that would capture the non-linear interaction between the response and the predictors. One important note is that tree-based models are not designed to work with very sparse features. When dealing with sparse input data (e.g. categorical features with large dimension), we can either pre-process the sparse features to generate numerical statistics, or switch to a linear model, which is better suited for such scenarios. I have taken this information into account and will make sure to keep this in my mind when I analyze the random forest results.



In the random forest plot, we can see that the error starts to slowly decrease and stay steady around 400 trees. Between 0 trees and 200 trees the error does decrease a lot but afterwards between 200 trees and 400 trees it seems to decrease but at a much slower pace. I can not really see a difference between the tree sizes of 450 and 500.

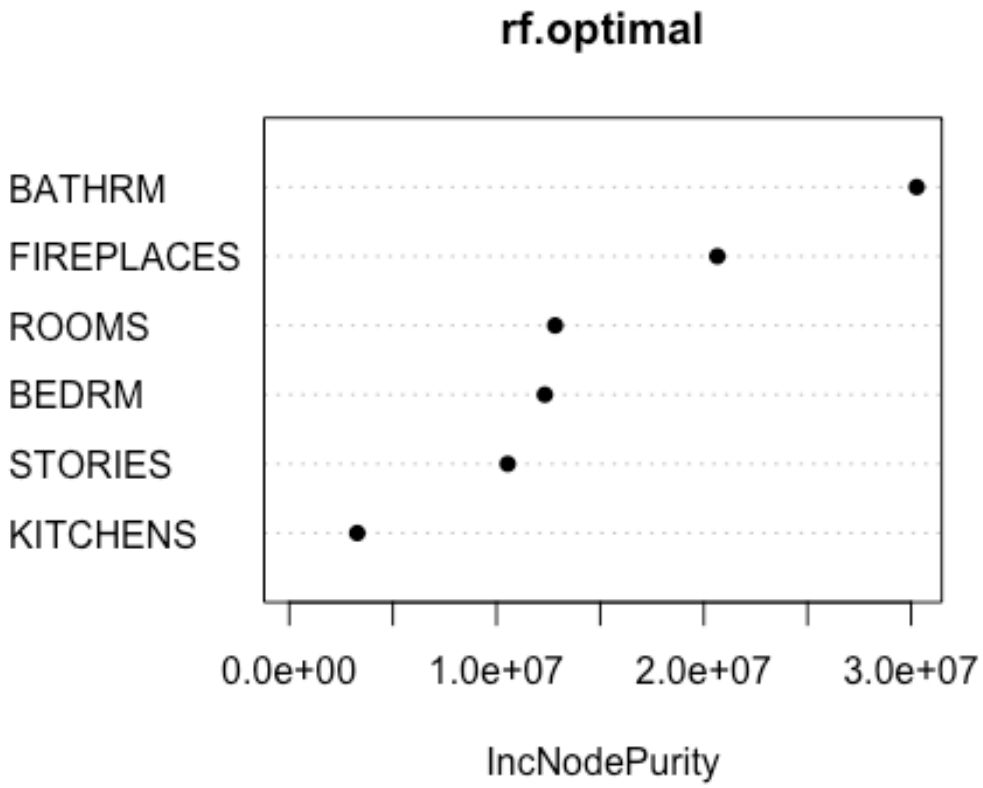


The importance plot for the random forest model shows me that the most important variable in this model is Longitude followed by bathrooms with a small amount of difference. The least important variable is kitchens. It is interesting to note that from is the large gap between EYB.age and Bathroms, and EYB.age and Latitude. Before making tis plot I originally thought that longitude and latitude importance will we right next to one another, but it seems from this plot that I was wrong in that assumption.

After using random forest, I needed to check my results by cross validation and bagging. Bootstrap Aggregation or Bagging, is a simple and very powerful ensemble method. This method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model random forests could create. Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithms that have high variance. Since decision trees are sensitive to the specific data on the training dataset then if the training data is changed, the resulting decision tree can be quite different and in turn the predictions can be quite different. Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees. I was once thinking of changing the training set to 60% and the testing set to 40% just to see what the results look like, but bagging helps me take of this issue of mine.

After using bagging and cross validation approach, I tried to find the optimal random forest model with predictor, bathrooms, rooms, bedrooms, stories, kitchens, fireplaces. The results showed me that the cross validation mean-square error of prediction, estimated by the validation set cross-validation is 678.1715, wich is very large but still smaller than the other

prediction MSE before, namely ridge and lasso MSE. The optimal random forest had 143 trees, and one spilt for each of the variables (mtry=1). The variance explained was 54.81%, which is good, but I still would have liked the variance explained percentage to be higher. The more the optimal random forest model variance can explain the better the model is.

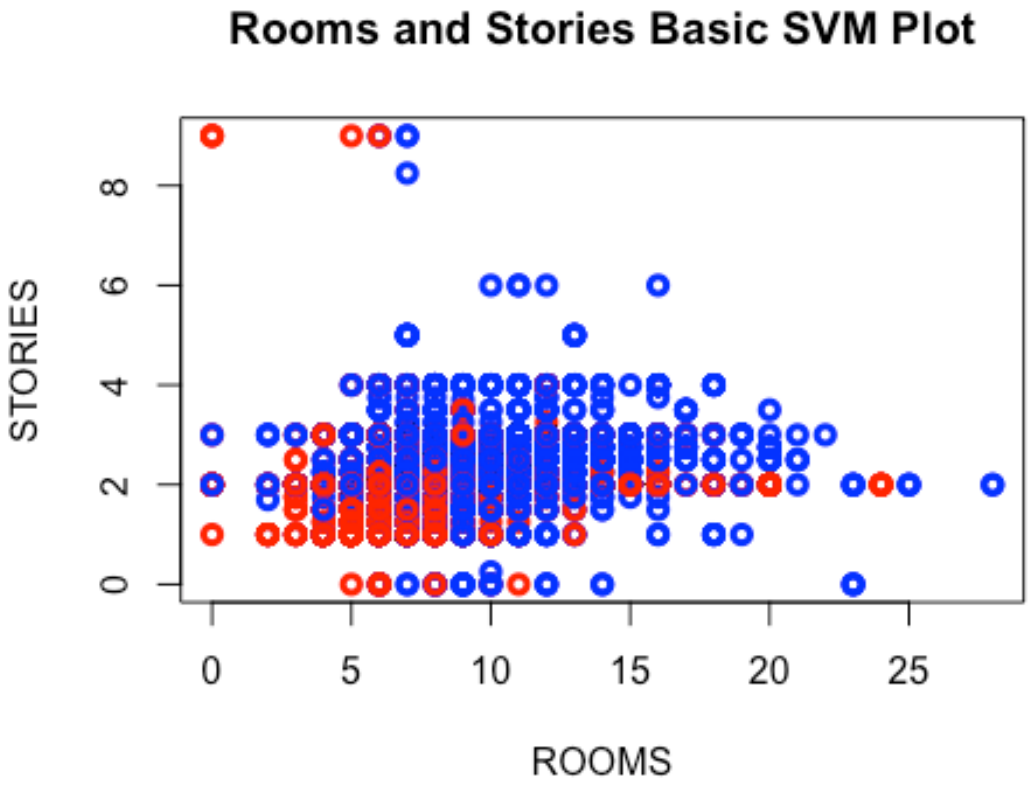


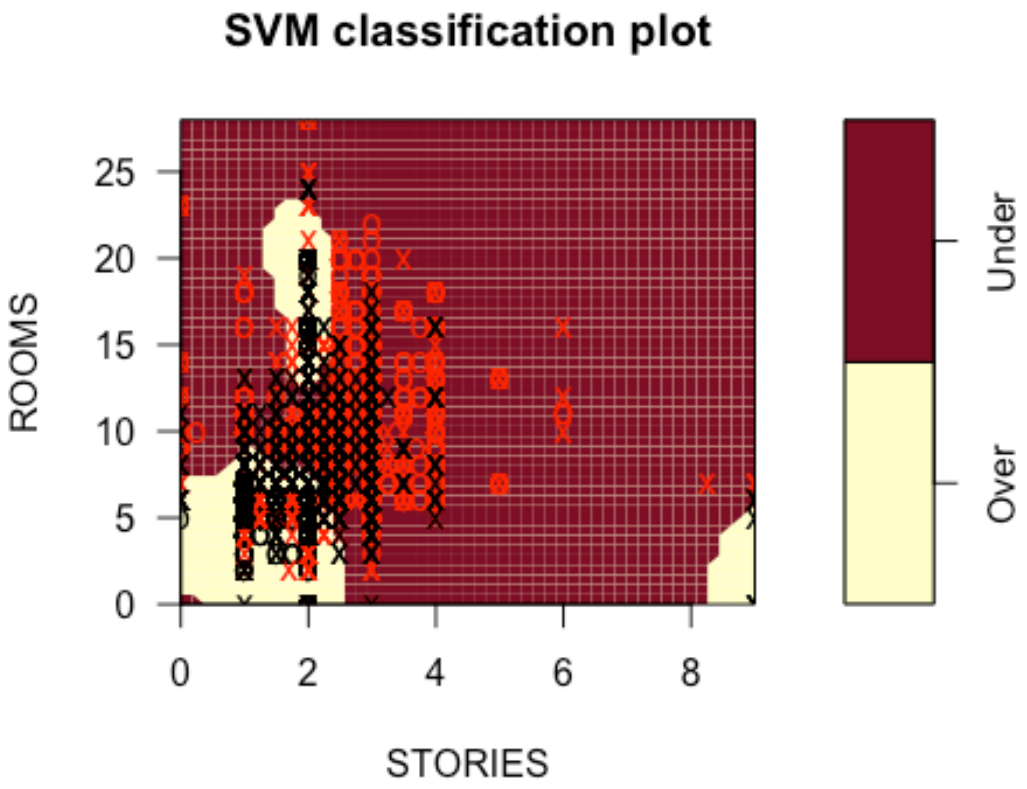
It seems the random forest optimal model did its job. Now when I look at the importance plot above, I can see that bathrooms are the most important variable in the random forest model. While kitchens seem to be the least most important variable in the model. Although kitchens maybe the least important variable in this model, I do not think that it is still pretty important since I would approximate that the node purity is around 6,500,000, and the bathrooms node purity is around 31,000,000. What is till interesting is that we still have a rooms, bedrooms, and stories node purity importance almost right up top of one another. They are almost clustered together like they were previous but this time we can see the break and difference a little more.

The last machine learning technique I used was support vector machines. A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors. I used support vector machines is because one of the advantages to using this technique is it uses regression to avoid the difficulties of using linear functions in the high-dimensional feature space, and the optimizes the problem is transformed into dual convex quadratic programs.

Below is basic plot of two variables I used in my models classified into two categories of the median of price. If the price was greater than the median of price, which was equal to 44.365,

then is was labeled as over (colored as blue), and if the price was less than the median price, which was equal to 44.365, then is was labeled as under (colored as red). We can see from the plot below that we might be able to fit a hyperplane in the plot, but it will not be that great. This is because the 2 classifications are all over the place and we cannot see any clear distinction on where the blue and red dots are.





The plot above has a prediction mean squared error is at 0.6385176. The plot above has a kernel with radial. I choose this plot to be shown since this had the highest predicted MSE out of the four kernels I tested. The hyperplane is not the best I guessed in my prediction since a lot of the observations are not on the right side of the hyperplane and within hyperplane margins. More work and analysis will have to be run to get the support vector machine plot analysis perfected.

I can conclude the final model with machine learning techniques is good but there is still room for improvement. This is a good start to making something even better. The final model I created has an adjusted R^2 of 69.04%, and a residual standard error of 31.12. We can even see in the residual plots, data diagnostics, and model diagnostics that this model and data still lack normally distributed data, especially at the tails. If the tails in the qq-plot were closer to the dotted line I would not worry so much, but since the tails are way off the dotted line, I would say there is a concern. I have learned a lot from this project and taken away a lot of useful and insightful information, so now I can try to create something better. For example, maybe re-running the model again at different set seed numbers and see if my results vary drastically. If they do, I will make sure to comment that in future work. Also, I took our stories in the model based on the variable selection, and hypothesis tests results. However, I wish to see if I changed the seed and I keep the stories variable in my model since seems to be somewhat important based on the optimal random forest model, how my results will change. It seems that the mean squared errors are all over the place, but the range of the man squared errors for all the techniques and models I used above, except for ridge and lasso regression are between .5 and .65. This is not great, and I would have preferred to have a higher classification rate, for example a prediction

MSE or classification rate around .85 or higher, but there was so much data, and work put into this project that I will take what I can get.

Now I can answer my previous research questions. The previous questions I had were as follows: what are the best variables to help explain a property pricing model, does analyzing classification pricing models help better explain the data then the linear, logistic, and Bayesian models did, and how good are the classification pricing models, namely can I trust the results? The answers to these questions are as follows: 1) the best variables to use are rooms, longitude, latitude, bathrooms, fireplaces, bedrooms, and stories. 2) analyzing classification pricing models does indeed help better explain the data, and is slightly better than using linear, logistic and polynomial model. More research will be needed to be applied to see how the classification pricing models compare to a Bayesian model, if they can even be compared at all. 3) The classifications models are good, but could be better, more analysis will need to be done to make them better.

This has been a good learning experience and I will take what I learned and move forward. Next time I might think about combing my machine learning Bayesian tools to see if I can come up with a better model than the one, I created in this project and the Bayesian project. Thank you for reading this very long report.

Model Comparison Table

Simple Linear Model	*Price_10K* = -7863 + 9.988*Bathroom + 0.7407*Rooms + 2.633*Bedrooms - 1.172*Stories - 1.003*Unqualified + 2.877*Grade=Average + 23.85*Grade=Excellent + 155.2*Grade=Exceptional + 15.33*Grade=Fair + 0.8134*Grade=Good + 46.50*Grade=Superior + 8.616*Grade=Very Good - 3.858*Kitchens + 9.305*Fireplaces + 8.452*Ward 2 + 12.01*Ward 3 - 4.366*Ward 4 - 5.517*Ward 5 + 3.526*Ward 6 -6.267*Ward 7 – 18.40*Ward 8 - 43.39*Latitude – 124*Latitude + 0.2879*AYB.age - 0.2937*EYB.age – 0.1789*Remodel.age + 41.17*Condition=Excellent + 6.386*Condition=Fair + 8.411*Condition=Good + 2.722*Condition=Very Good
Logistic Regression (Price was the response variable in this model)	Log(π/1-π) = - 3.158 - 0.000004374*Price + 0.007901*√Price - 0.2907*(AC=Yes) + 0.1821*Rooms + 2.221*(Rooms^0.2) - 0.04816*√BEDRM + 0.1069*(CNDTN=Excellent) - 1.196*(CNDTN=Fair) + 0.2849*(CNDTN=Good) – 1.373*(CNDTN=Poor) + 0.6739(CNDTN=Very Good) – 0.4306*(Ward=2) – 0.2858*(Ward=3) – 0.0515*(Ward=4) + 0.2901(Ward=5) + 0.000001085PRICE*(AC=Yes) + 0.00000002.698*(PRICE*ROOMS) + 0.0000003.897(Price*Ward=2) + 0.0000005486*(Price*Ward=3) + 0.000001052*(Price*Ward=4) – 0.0000003.313*(Price*Ward=5)
Bayesian Model	Price_10K = - 0.920544 + 21.137957*Bathrooms + 0.575739*Bedrooms + 0.220931*Rooms - 4.707369*Kitchens + 14.905539*Fireplaces + 0.466959*AYB.age - 0.849885*EYB.age - 0.123803*Remodel.age - 5.168444*AC=Yes - 64.649561*Condition=Excellent + 0.851290*Condition=Fair - 1.928609*Condition=Good - 43.564921*Condition=Very Good + 6.387614*REMODEL.age*CONDITION=Excellent - 0.180788*REMODEL.age*CONDITION=Fair + 0.128282*REMODEL.age*CONDITION=Good + 1.319853*REMODEL.age*CONDITION=Very Good + 0.005704*ROOMS*AYB.age + 0.937902*ROOMS*AC=Yes + 28.054394*BATHRM*CONDITION=Excellent + 2.305573*BATHRM*CONDITION=Fair + 3.230037*BATHRM*CONDITION=Good + 18.855757*BATHRM*CONDITION=Very Good