Stat 663 Project 2: Shrinkage and Selection

Jackson Gazin

2022-10-16

Abstract/Executive Summary

VRBO is a rental company which allows individuals to put their homes, apartments, or rooms up for short term rent. It is extremely popular for people looking for places to stay during vacation. Unsurprisingly, the price of these apartments often determines how frequently they are rented. In this report, we will discuss the process of predicting the price of a new rental unit based on characteristics of that unit. This will help the client determine prices for new rentals to make them attractive to customers. Data was collected from 1561 different rental units and included thirteen characteristics about each rental including its unit number, average satisfaction, number of reviews, and neighborhood along with the daily price of the rental. We used Ridge Regression, Lasso Regression, and Logistic Regression to predict the price of a new rental. We found that Ridge Regression was the most effective model at predicting rental price. More specifically, we found that on average the predicted price from our Ridge Regression model was around 62 dollars and 35 cents off from the actual price. We further discuss our findings and the limitations of the study in the following paper.

Part 1: Introduction

VRBO is a rental company which allows individuals to put their homes, apartments, or rooms up for short term rent. It is extremely popular for people looking for places to stay during vacation. Unsurprisingly, the price of these apartments often determines how frequently they are rented. Our client's goal is for us to build a model which can be used to predict the price of a new rental. This will help the client determine prices for new rentals to make them attractive to customers. In this report, we will discuss the process of predicting the price of a new rental unit based on characteristics of that unit. Data was collected from 1561 different rental units and included thirteen characteristics or "features" of each rental along with the rental's price. These thirteen features were the rental's unit number, average satisfaction, number of reviews, number of accommodates, number of bedrooms, minimum night stay, district, neighborhood, walk score, transit score, bike score, and percentage of rentals in the neighborhood. In the following sections, we will prepare our data so that it is adequate for model use, create a Ridge Regression, Lasso Regression, and Elastic Net Regression model to predictive price, and then choose the model which has the highest predictive accuracy.

Part 2: Data Cleaning

In preparation for building the model, my first step was to clean the data. One aspect of data cleaning is the process of dealing with incomplete data. Most models require our data set to be complete in the way variables are represented across different data points. Making predictions and building models are impossible when data sets have incomplete information. I first cleaned the data by checking if any of the data points were lacking values for any of the thirteen features.

Figure 2.1: Rows with NA values

dataset	NumberOfNAs
# Rows with NAs present in at least one column	138
# Rows with NAs present at least one column (not including minstay)	0
# Rows with NAs present in at least one colum (for transformed data)	0

Figure 2.2: Transformed Data

Description	Number
Number of Rows in Dataset	1561
Number of unique UnitNumber Values	1561

We can see in Figure 2.1 that there were 138 data points in our original data set with a missing value for at least one of the variables. However, we can also see that this only occurred for the minstay variable; for every row all variables besides minstay had a value. Minstay indicates the minimum stay for a particular rental. I then transformed my data by looking at all data points which lacked a value for minstay and replaced that value with 1. If someone is using a rental, we can assume that the minimum stay for that rental is 1 night unless otherwise indicated. Since I was able to replace all of the values that were missing with a real value, there were 1561 data points before and after we dealt with missing values.

In general, when we build a model, we want to make sure that our features have the potential to be used for prediction. In this case, a feature is only helpful if it contains information that could be useful in predicting the price of a new rental. For the feature UnitNumber, each data point has its own unique value, and the feature only indicates that each row is a unique rental. This idea is confirmed in Figure 2.2 as the number of rows in the data set is equal to the number of unique UnitNumber values. The fact that each data point is a different rental is not a feature that will be useful for prediction, and I therefore removed UnitNumber as a feature in the data set. We are trying to help the client "determine appropriate prices for new rental properties." Therefore, we also want our features to be useful for **new** rental properties. If a property is new, we will have no information on the customer feedback yet. Therefore, I am going to remove the features overall_satisfaction and reviews because they will not be available to a rental property that has not been on VRBO yet. Note, however, that I am going to include WalkScore, TransitScore, and BikeScore because these scores are independent of how many people have used the rental and will be determined before the rental is placed on VRBO.

In Figure 2.3, we can see a table which displays the name, class, and description of each feature we have. Class tells us how these data points are currently formatted.

Another aspect of data cleaning is ensuring that the format of the data is consistent with its intended

Figure 2.3: Variables and their classes

variables	class	description
Price	integer	The price in US dollars of a one-night rental
room_type	character	Is the rental for an entire house/apt, a private room, or a shared room?
accommodates	integer	How many people the rental can accommodate
bedrooms	integer	the number of bedrooms included in the rental price
minstay	numeric	the minimum number of nights an individual must book at the property
neighborhood	character	the name of the neighborhood in which the rental is located.
district	character	the name of the district in which the rental is located.
WalkScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by walking
TransitScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by public transit
BikeScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by biking
PctRentals	numeric	the percent properties in the neighborhood that are rental properties

Figure 2.4: Room Type and their Frequency

Room Type	Frequency
Entire home/apt	826
Private room	685
Shared room	50

Figure 2.5: Room Type and their Frequency

Room Type	Frequency
Entire home/apt	826
Private room	735

representation. Thus, we want the class of each feature to be consistent with its description.

There are three variables which I do not believe are formatted consistently with their intended representation. These include room_type, neighborhood, and district. Each of these are formatted as characters which means that each value is a letter or a collection of letters. However, consider that the description for room_type is, "Is the rental for an entire house/apt, a private room, or a shared room?". The variable is defined to only take on one of three categories at each data point. Therefore, it would make more sense if this variable was mapped as a categorical variable which means that the variable can take on a finite amount of values which often have no numerical meaning. Likewise, both neighborhood and district take on a finite amount of categories and would be better off formatted as categorical features.

We call the different categories of a particular categorical variable the levels of that categorical variable. Note, that the three models we will eventually be testing on this data set are all linear models. We will explain what that means more specifically later, but for now, it is important to note that if a categorical feature has k levels, our linear model will contain $k-1$ parameters for that feature. Each parameter quantifies how the presence of one level translates to a change in price. Consequently, we want each level to be present in at least 5% of the data points, so our model has enough information to build reasonable parameters for each level. That being said, in practice we would prefer that our levels account for as close as possible to 10% or more of all data points.

In Figure 2.4, however, we can see that the level "shared room" in room_type only accounts for $\frac{50}{1561} \approx 3.2\%$ of the data points. Therefore, I am going to change room_type to have two features: "Entire Home/Apt" and "Single Room" where the latter includes both private rooms and shared rooms. We can see in Figure 2.5 that both levels are now relatively evenly distributed! We can now transform room_type into a categorical variable.

When we look at neighborhood, we can see in Figure 2.6 that many of the levels contain a very few number of data points. I am going to transform our levels so that all levels with frequencies that account for less than 7 percent of the data are collapseed into a level called "Other", and the rest remain unchanged. The neighborhoods which do account for more than 7 percent of the data are Humboldt Park, Logan Square, Rogers Park, and West Town.

Figure 2.6: Neighborhood Names and their Frequency

Neighborhood	Frequency
Albany Park	51
Archer Heights	5
Avondale	60
Beverly	2
Bridgeport	50
Brighton Park	3
Burnside	2
Calumet Heights	2
East Garfield Park	37
Edgewater	35
Edison Park	1
Englewood	1
Gage Park	6
Hegewisch	2
Hermosa	2
Humboldt Park	127
Hyde Park	73
Irving Park	22
Jefferson Park	20
Kenwood	9
Lincoln Park	57
Lincoln Square	15
Logan Square	330
McKinley Park	15
Montclare	5
Morgan Park	4
Near West Side	42
North Center	37
North Park	7
O'Hare	5
Portage Park	20
Pullman	5
Rogers Park	123
South Chicago	10
South Shore	10
The Loop	101
Uptown	79
Washington Park	4
West Elsdon	4
West Englewood	1
West Lawn	3
West Town	154
Woodlawn	20

Figure 2.7: Neighborhood Names and their Frequency

Neighborhood	Frequency
Humboldt Park	127
Logan Square	330
Other	827
Rogers Park	123
West Town	154

Figure 2.8: District and their Frequency

District	Frequency
Central	101
Far North	336
Far Southeast	21
Far Southwest	6
North	484
Northwest	49
South	166
Southwest	38
West	360

We can now see in Figure 2.7 that each level accounts for at least 7% of the data. Therefore, we can be much more confident about the estimates that our model will build for the categorical variable. We can now format neighborhood as a categorical feature. Note, we would prefer each level to contain at least 10% of the data. Nevertheless, we will just need to be a little more careful about the conclusions we make regarding any parameters of this categorical variable.

For the feature district, we can see in Figure 2.8, that "Far Southeast", "Far Southwest", "Northwest", and "Southwest" have few data points. I am therefore going to consider the districts "Far South East", "Far Southwest", and "Southwest" to be in the South. I am also going to consider Northwest to be in the North. Note, that "Central" does not have any logical level to collapse into, but only accounts for around $\frac{100}{1561} = 6\%$ of the data points. Again, this is not quite ideal, and we will have to proceed with more caution in any of the parameters we derive for Central.

We can see in Figure 2.9, that the levels for district are much more balanced now. We can now transform district into a categorical feature.

Figure 2.9: District and their Frequency

District	Frequency
Central	101
Far North	336
North	533
South	231
West	360

Figure 2.10: Classes of Features for Clean Data

variables	class	description
room_type	factor	Is the rental for an entire house/apt or a shared room?
accommodates	integer	How many people the rental can accommodate
bedrooms	integer	the number of bedrooms included in the rental price
minstay	numeric	the minimum number of nights an individual must book at the property
Neighborhood	factor	the name of the neighborhood in which the rental is located.
District	factor	the name of the district in which the rental is located.
WalkScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by walking
TransitScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by public transit
BikeScore	integer	a score indicating how easy and safe it is to get to areas of need/interest by biking
PctRentals	numeric	the percent properties in the neighborhood that are rental properties

We can see in Figure 2.10 that each variable's format is consistent with its description. Note, that 'factor' is what R refers to as a categorical variable.

Data cleaning is now complete!

Part 3: Ridge Regression

Ridge Regression is a type of linear model. Given a random sample of data with $(\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p)$ features and a numeric response variable $\boldsymbol{Y}$, a linear model describes the relationship between the features and the response value for each point (i) as

$$Y_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} + \epsilon_i$$

where each ϵ_i is a random value representing the random error that is embedded in the world around us. We can use linear models to make predictions, $\hat{Y}_i$, where

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \cdots + \hat{\beta}_p X_{ip}$$

Linear models make predictions by using a set of data to estimate parameters $\beta_1, \cdots, \beta_p$ along with an intercept parameter β_0. We represent estimates for $\beta_0, \beta_1, \cdots, \beta_p$ as a vector $\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$.

However, not all linear models estimate these parameters using the same method. The most famous linear model is Least Squares Linear Regression (LSLR) whose $\hat{\boldsymbol{\beta}}$ is the $\hat{\boldsymbol{\beta}}$ which minimizes the Residual Sum of Squares (RSS). RSS is the sum of the squared difference between each estimate $\hat{Y}_i$ and the corresponding actual value Y_i. The $\hat{\boldsymbol{\beta}}$ for LSLR is called an "Unbiased Estimator" because if we were to imagine repeating the process of finding $\hat{\boldsymbol{\beta}}$ a large amount of times for many different sets of random data, the average of each $\hat{\beta}_j$ would be equal to the true β_j.

However, LSLR is often not as useful when our data has variables that are correlated with each other. When this happens, choosing the $\hat{\boldsymbol{\beta}}$ which minimizes RSS will often produce estimates which are extremely large and unrealistic given our set of data. In turn, the range of possible values which we believe the true parameters are contained in also become large.

This is when methods like Ridge Regression can become useful. Ridge Regression is a linear model and has the same form as LSLR, but rather than choosing the $\hat{\boldsymbol{\beta}}$ which minimizes the RSS, we choose the $\hat{\boldsymbol{\beta}}$ which minimizes the value of $RSS + \lambda \sum_{j=1}^{j=p} \hat{\beta}_j^2$. This decreases the value of each $\hat{\beta}_j$ and also decreases the range

of possible values which we estimate the true parameters to be contained in. We call $\lambda \sum_{j=1}^{j=p} \hat{\beta}_j^2$ our penalty term where λ is a tuning parameter. The existence of $\sum_{j=1}^{j=p} \hat{\beta}_j^2$ ensures that the value of each $\hat{\beta}_j$ will shrink to become more reasonable. Hence, Ridge Regression is referred to as a shrinkage technique. The tuning parameter λ controls how much shrinkage we want to occur.

One thing to be aware of for Ridge Regression is that it is not an unbiased estimator. If we were to take many samples of data and each time produce a $\hat{\boldsymbol{\beta}}$, the average of each $\hat{\beta}_j$ would **not** be equal to the true β_j. The amount of bias would be the difference between average of the estimates and their true value. This bias increases as λ increases. However, the spread of our estimates decreases as compared to LSLR and so does the range of possible values which we believe the true parameters are contained in. When conducting Ridge Regression, it is our job to choose a value of λ which adequately decreases this range but also does not let the bias become too large. If the bias is too large, our estimate will be unreasonable.

In practice, we choose the value of λ by first choosing a range of values for λ. For each λ, we choose the $\hat{\boldsymbol{\beta}}$ which minimizes $RSS + \lambda \sum_{j=1}^{j=p} \hat{\beta}_j^2$. We then use 10-Fold Cross-Validation to estimate the Test RMSE for each λ. When we are done, we save the model which contains the λ with the lowest test RMSE.

Before we do this with our data set, let's explain 10-Fold Cross Validation (10-Fold CV). In 10-Fold CV we randomly put our data points into 10 sets or "folds". For each fold, we make predictions on the points inside that fold by using a model trained with data from all the other folds. Thus, for each λ, 10-Fold CV will give us a prediction $\hat{Y}_i$ for each data point along with a test RMSE for that λ. MSE is known as the mean squared error and is equal to RSS divided by the amount of data points. RMSE or root mean squared error is equal to the square root of MSE. Simply put, RMSE tells us on average how far off we expect our prediction to be from the actual value. The model with the lowest Test RMSE can therefore be viewed as the most accurate model because it has the smallest average distance between the predicted price and the actual price.

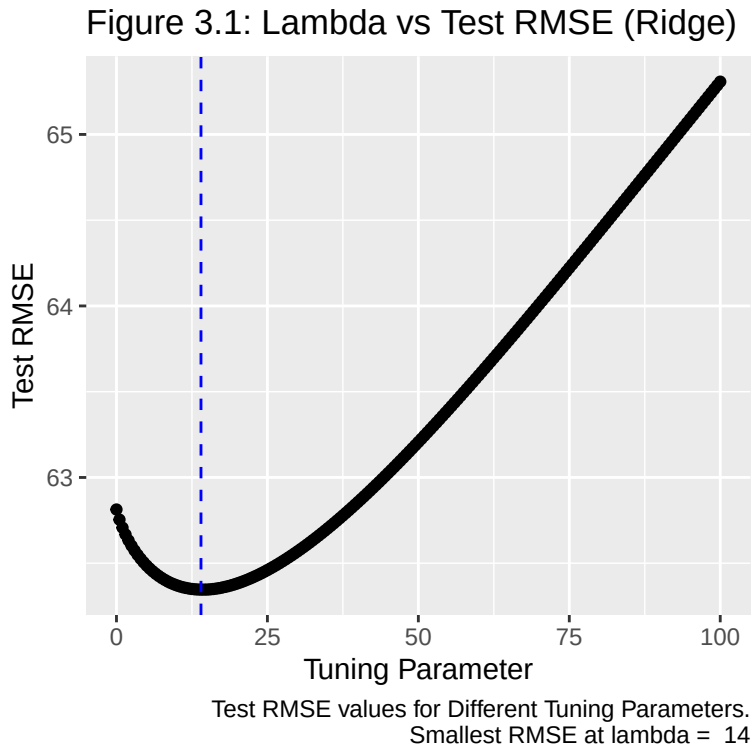


Figure 3.1 shows displays how the test RMSE changes with values of λ. We can see that $\lambda = 14$ had the lowest value for test RMSE. This means that if our client wants to predict the price of a new rental using Ridge Regression, they should use the tuning parameter $\lambda = 14$ if they want the highest predictive accuracy.

Figure 3.2: Lambda with lowest Test RMSE

lambda	RMSE
14	62.34613

Figure 3.3: Coefficients for Ridge Regression with Lambda = 14

	Coefficients
(Intercept)	-141.3415542
room_typePrivate room	-29.3742525
accommodates	10.1826886
bedrooms	28.2452943
minstay	-1.4112674
neighborhoodLogan Square	-7.5223519
neighborhoodOther	9.6666153
neighborhoodRogers Park	-9.2484641
neighborhoodWest Town	8.4660885
districtFar North	-6.6587872
districtNorth	-0.8321938
districtSouth	-10.8614588
districtWest	-4.3298302
WalkScore	0.4695494
TransitScore	1.6668020
BikeScore	0.4470325
PctRentals	-18.1634361

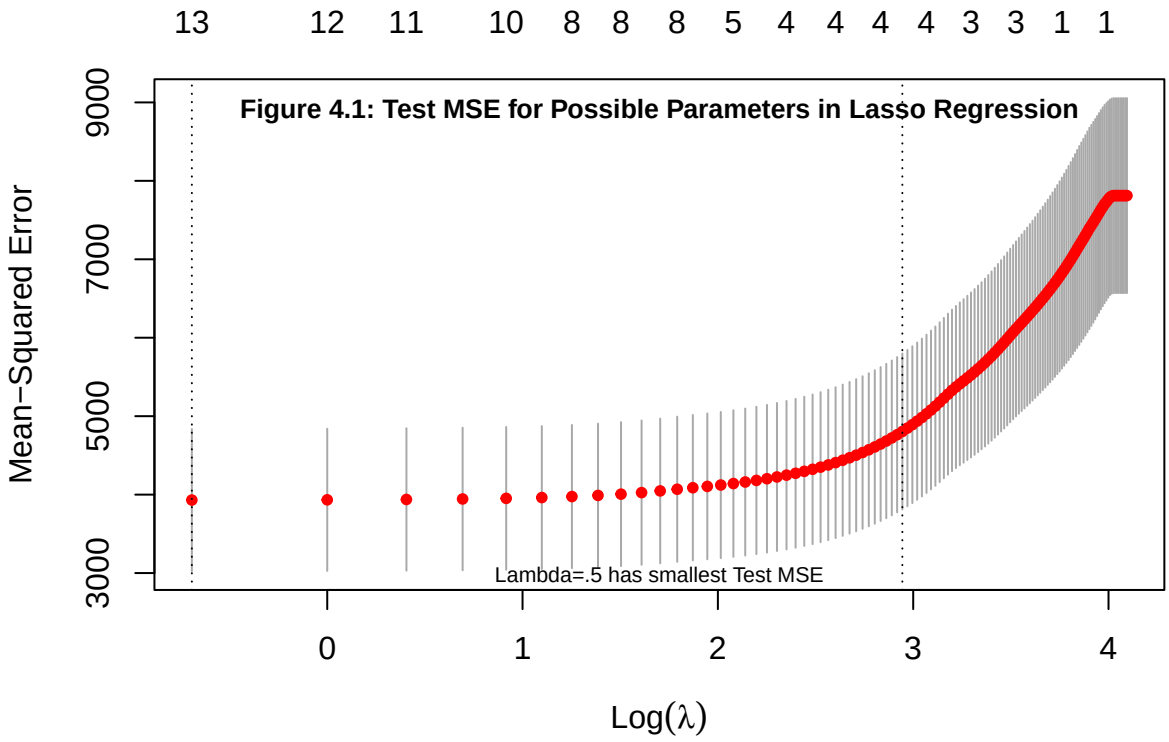
Figure 3.2 shows that $\lambda = 14$ had a corresponding test RMSE of approximately 62.35. Figure 3.3 shows the corresponding estimates of our parameters which yielded this test RMSE. Note, that test RMSE tells us on average how off our predicted price was from the actual price. Thus, Figure 3.2 and Figure 3.3 show that when using room_type, accommodates, number of bedrooms, minimum stay, neighborhood, district, WalkScore, TransitScore, BikeScore, and PctRentals as features for to predict price of rentals with Ridge Regression where $\lambda = 14$, on average we expected our estimated price for a new rental to be off from its true price by around 62.35 dollars.

Note, that the coefficients for the numeric features quantifies on average how a one unit increase in the feature translates to a change in price. Further note some categorical variables have more than one associated estimate. In general, if a categorical variable has x levels, there will be $x - 1$ coefficients for that categorical variable. Furthermore, each parameter tells us how the response variable should change on average if that level is observed. For example, room_typePrivate Room having a coefficient of -29.37 means that the rental being a private room on average will lead to around a 29 dollar drop in price.

Part 4: Lasso

Lasso Regression, like Ridge Regression and LSLR is a linear model. However, rather than choosing the $\hat{\boldsymbol{\beta}}$ which minimizes RSS or $\text{RSS} + \lambda \sum_{j=1}^{j=p} \hat{\beta}_j^2$, it chooses the $\hat{\boldsymbol{\beta}}$ which minimizes the value of $\text{RSS} + \sum_{j=1}^{j=p} \lambda |\hat{\beta}_J|$. Like Ridge Regression, Lasso Regression will also shrink the values of our estimates. However, unlike Ridge Regression, it allows for the possibility that some of our $\hat{\beta}_j$ will shrink to 0. Thus, Lasso Regression is both a shrinkage technique and a selection technique. It allows us to potentially shrink some estimates to 0 so that they are "selected" out of our model. As we increase our tuning parameter λ, both the shrinkage for each estimate increases, along with the probability that some estimates will shrink to 0 and be selected out. Lasso Regression is preferable to Ridge Regression when we want shrinkage and selection in our model. This can occur when we have a large number of features or value simplicity in our model and would prefer some features to be selected out.

The practice of building a model in Lasso Regression is extremely similar to Ridge Regression. Again, we choose a range of values for λ. However, now for each of these values, we estimate $\hat{\beta}$ which minimize $RSS + \lambda \sum_{j=1}^{j=p} |\hat{\beta}_j|$. We then use 10-Fold CV to estimate the Test RMSE for each λ. When we are done, we save the model which contains the λ with the lowest test RMSE.



In Figure 4.1, the first vertical line indicates that that $\lambda = .5$ was the tuning parameter which had the lowest MSE and therefore the lowest RMSE. The number 13 above it tells us that the model with $\lambda = .5$ used thirteen coefficients. This means that even at $\lambda = .5$ our model is still selecting out three coefficients to use in the model! This is what is so powerful about Lasso Regression. As we will see later, our predictive accuracy barely changed compared to Ridge Regression, but we were able to do so using three less coefficients!

Further note, that since our sole goal is predictive accuracy, and we want to minimize test RMSE, we should choose $\lambda = .5$ as our tuning parameter. Note, however, that depending on the goal of the client this may not always be true. Our client currently just wants us to predict price of a rental. Therefore, we only care about building the most accurate possible model. However, if our client also wanted us to be able to explain **how** different features can be used to predict price, we may want to choose a model that selects out even more features. In fact, the second vertical line of the graph shows us the highest tuning parameter which is one standard deviation away in terms of Test MSE from the tuning parameter which minimizes test MSE. This may be a more appropriate tuning parameter if we wanted to make our model interpretable as well as accurate.

Figure 4.3: Lambda with lowest Test RMSE for Lasso

lambda_lasso	RMSE_lasso
0.5	62.69738

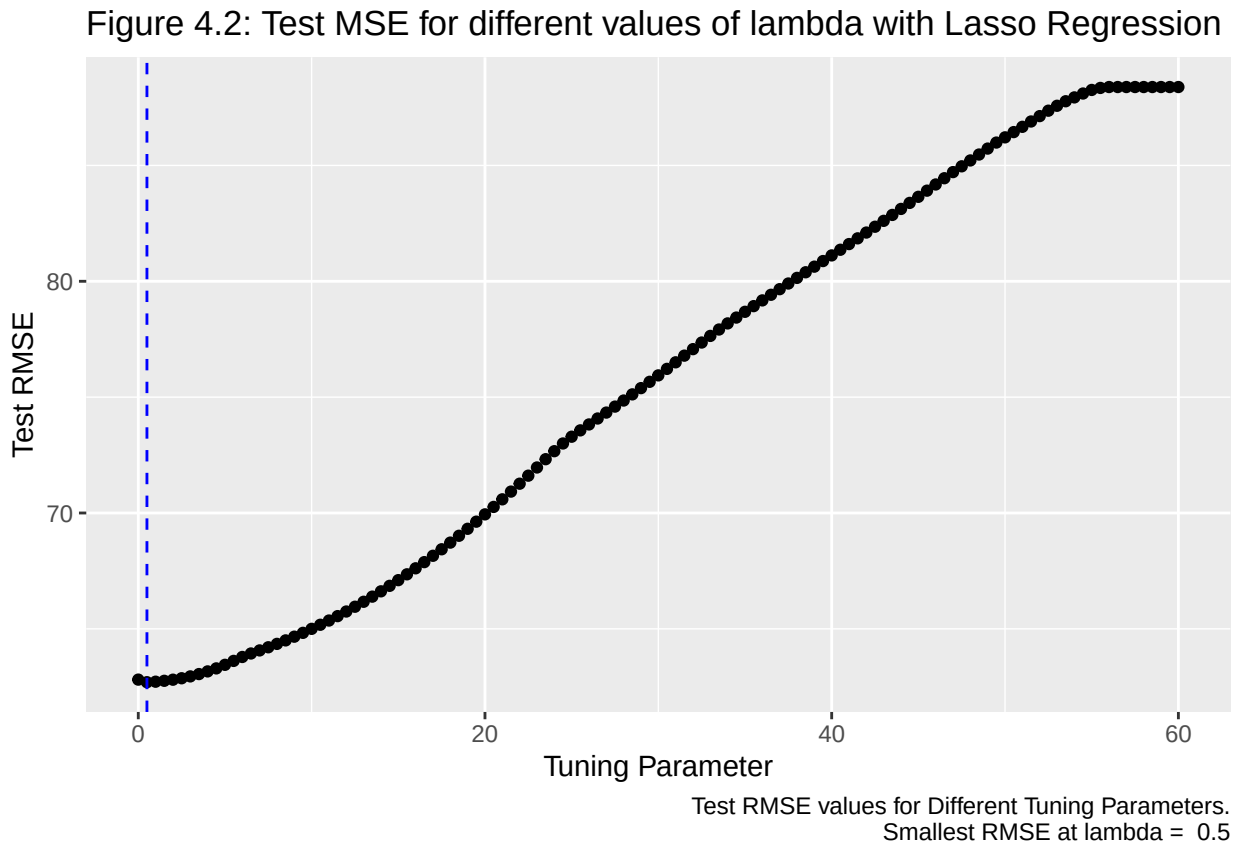


Figure 4.2 also shows the fact that $\lambda = .5$ was the tuning parameter with the smallest Test RMSE. Figure 4.3 displays the associated test RMSE with the tuning parameter of $\lambda = .5$. Further note that our test RMSE was approximately 62.7. Test RMSE tells us on average how far we expect our predicted price to be off from the actual price. On average our predicted price using this model was around 62.7 dollars off from the actual price. This is only slightly higher than our Test RMSE in Ridge Regression which was 62.35. On average, our estimates for price are only going to be around 30 cents less accurate when we use Lasso Regression to predict price rather than Ridge Regression. However, our Lasso Regression model selected some features out. Thus, we were able to simplify our model without letting our predictive accuracy suffer significantly.

Figure 4.4: Coefficients for Lasso Regression with Lambda = .5

	Coefficients
(Intercept)	-149.766
room_typePrivate room	-28.610
accommodates	11.084
bedrooms	30.518
minstay	-1.291
neighborhoodLogan Square	-7.533
neighborhoodOther	5.461
neighborhoodRogers Park	-13.594
neighborhoodWest Town	5.044
districtFar North	0.000
districtNorth	3.294
districtSouth	-1.995
districtWest	0.000
WalkScore	0.000
TransitScore	2.264
BikeScore	0.428
PctRentals	-27.005

Figure 4.4 displays the values we assigned for the coefficients of the model. We can see that "district FarNorth", "districtWest", and "WalkScore" all were selected out of our model.

Part 5: Elastic Net

The downsides of Ridge Regression and Lasso Regression are basically opposite from each other. Ridge Regression is not able to select many features so it often keeps features in the model that would be better off selected out. Conversely, Lasso Regression often shrinks out features that would be better off left in the model. Elastic Net tries to address this problem by choosing the model which balances the tendency for Ridge Regression to not shrink estimates to 0 and for Lasso Regression to shrink too many estimates 0. It does this by minimizing the following equation:

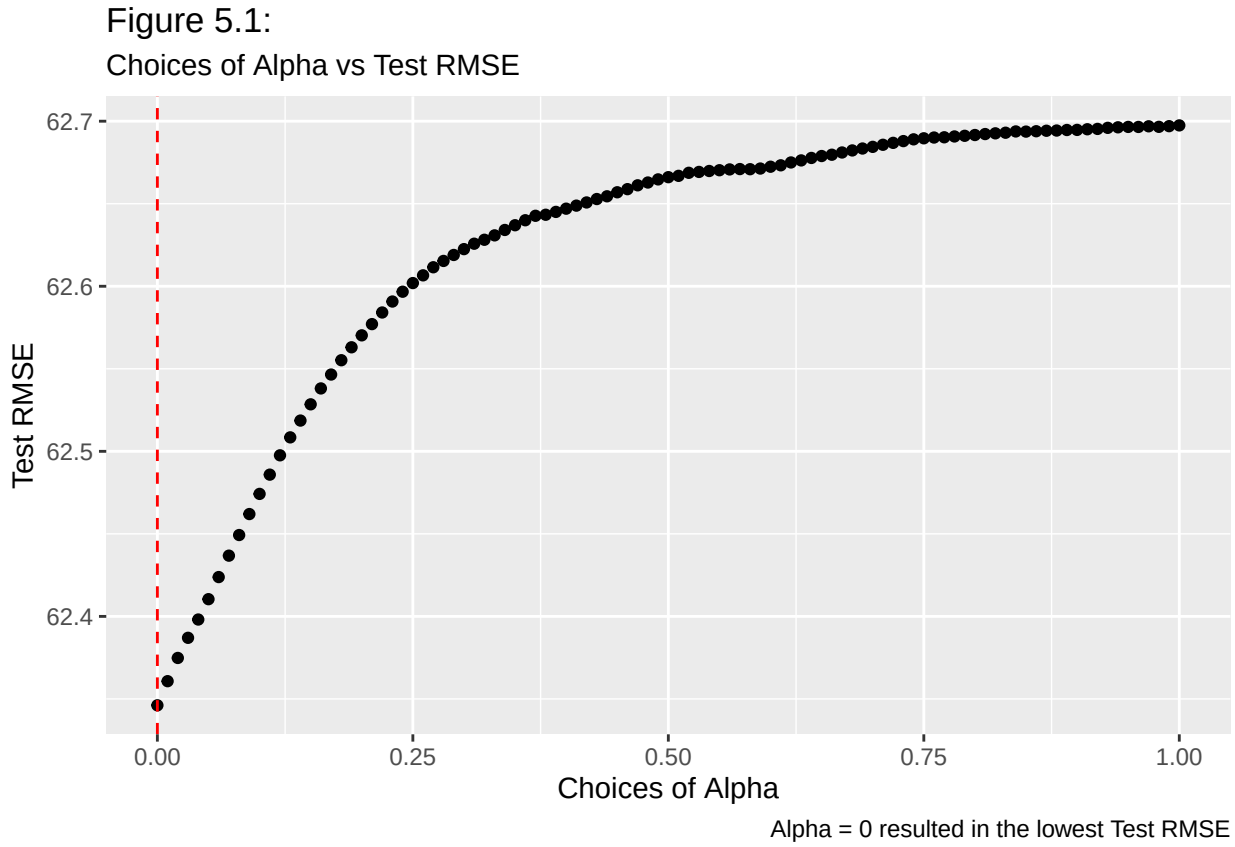
$$\text{RSS} + \lambda \sum((1-\alpha)\hat{\beta}_j^2 + (\alpha)|\hat{\beta}_j|) \text{ where } 0 \leq \alpha \leq 1$$

Now, we have two tuning parameters for our model: α and λ. α controls the balance between Ridge and Lasso. We can see that if $\alpha = 1$, we are just performing Lasso Regression, while if $\alpha = 0$ we are just performing Ridge Regression. Any α between these two values involves a estimate that strikes a balance between Ridge and Lasso. The higher our α value is the more likely we will shrink some estimates to 0.

How do we pick our α value? We first we choose a range of values from 0 to 1 which are typically in increments of .01 or smaller. Then for each value of α, we run 10-Fold CV on a range of λs and save the λ with the smallest Test RMSE. Finally, once we have done this for every α, we will choose the α with the smallest test RMSE.

Figure 5.2: Test RMSE for Elastic Net with Lambda = .5, Alpha =1

Lambda	RMSE	Alpha
14	62.34613	0



We can see in Figure 5.1 that $\alpha = 0$ was the tuning parameter which had the smallest Test RMSE for Elastic Net Regression. Since $\alpha = 0$, this means that we were doing Ridge Regression! We see that in Figure 5.2 we have a λ value of 14 and a Test RMSE of 62.34613 which matches what we got in Part Three! Further examining Figure 5.1 helps elicit why Ridge Regression with $\alpha = 0$ was preferable to any other iteration of Elastic Net Regression. We can see that the graph is almost completely non-decreasing, and thus the α with the smallest test RMSE was the smallest α. As α increases and the possibility that some features will be selected out increases, the test RMSE increases. This means that our model is most accurate when there is no possibility for any features to be selected out, and consequently all features are used.

Just like in Part Three, a test RMSE of approximately 62.35 means that on average we expected our estimated price for a new rental to be off from its true price by around 62.35 dollars.

Figure 5.3: Coefficients for Elastic Net Regression with Lambda = .5, Alpha =1

	Coefficients
(Intercept)	-141.342
room_typePrivate room	-29.374
accommodates	10.183
bedrooms	28.245
minstay	-1.411
neighborhoodLogan Square	-7.522
neighborhoodOther	9.667
neighborhoodRogers Park	-9.248
neighborhoodWest Town	8.466
districtFar North	-6.659
districtNorth	-0.832
districtSouth	-10.861
districtWest	-4.330
WalkScore	0.470
TransitScore	1.667
BikeScore	0.447
PctRentals	-18.163

Again, note that Figure 5.3 is equivalent to the table of coefficients we had for Part 3.

Part 6: Conclusions and Comparisons

In Figure 6.1, we can see the methods we used in this project with their associated test RMSE. We can see each value is extremely close. This is because in general, Ridge Regression, Lasso Regression, and Elastic Net Regression do not lead to signficant differences in predictive accuracy, but rather different methods of shrinking the coefficients. Thus, it is no surprise that their test RMSE metrics are extremely close to each other. That being said, if the clients wants the absolute best value of test RMSE, which is the predictive metric we have chosen, we would choose the Ridge Regression model. This is because it had the lowest Test RMSE, and consequently on average its prediction would be the closest to the actual predictions. In fact, the Ridge Regression price estimate is going to be around 30 cents more accurate on average than our Lasso Regression price estimate. Note, that our model for Elastic Net was equivalent to the Ridge Regression model because it resulted in an α of 0.

Ridge Regression resulted in a test RMSE of 62.34613, which means that on average our Ridge Regression model's predicted price would be off from the actual price by around 62.35 dollars. Is this a good measure? This depends on the context and scale of the values of the response variable. We can ascertain this both by considering what average prices are and by seeing a graph which shows our predicted values of price against actual values of price.

Figure 6.1: Regression Models and their test RMSE

Method	RMSE
Ridge Regression	62.34613
Lasso Regression	62.69738
Elastic Net	62.34613

Figure 6.2:

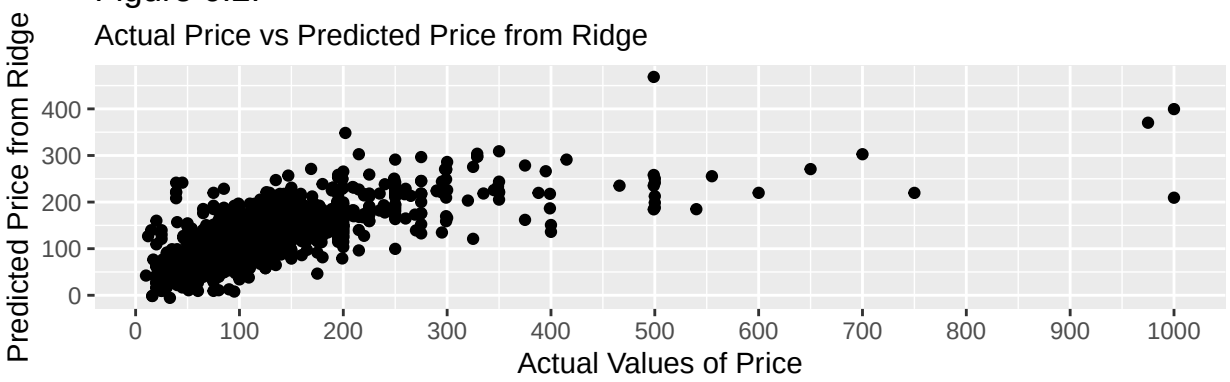


Figure 6.3:

Figure 6.2 displays how the predicted values of price for Ridge change with the actual value of price. We can see that our predictions are closer to the actual values at smaller prices. Figure 6.3 gives us a better idea of this concept as it shows the actual rental price vs the error which is the difference between the actual price and the predicted price. Again, we can see that in general as the price increases the error of our prediction also increases.

We can see in Figure 6.4 a summary of some metrics regarding our data and our model. The average price of the rental was around 109 dollars, while our model on average was around 62.35 dollars off from the actual price. Furthermore, our R^2 value for our model was 52 percent. R^2 tells what percent of variation in the data that our model explains. Thus, our model explained 52 percent of the variation in price.

Does this seem effective? Well, let's consider again the context of the problem we are trying to solve. We are trying to find a model which will estimate the price of a new rental. Since this is a new rental, we will not have any information on how the satisfied the customers are or how popular the rental is. We are just building a model that takes into account some properties of the rental itself that would be available before we put it on VRBO including the the neighborhood it is in, the district it is in, the amount of bedrooms, and the accommodates.

Consider that in real life these prices would be dictated by the market and the customer. More specifically, on average we can expect that rentals with higher prices reflect a higher renting appeal to customers and thus a willingness to pay a higher price. It is fair to assume that the average customers would utilize much

Figure 6.4: Ridge Regression Models with Metrics

Metric	Value
Average Price	109.50
R^2 for Ridge Regression Model	0.52
Test RMSE	62.35

Figure 6.5: Average Number of Reviews for Rentals

Dataset	Average.Number.of.Reviews
VRBO Data	27

of the features we had in our model to determine their appeal and willingness to pay for a particular rental. However, it is unrealistic to assume that the features we used would account for all of the features customers would use to quantify their willingness to pay. Features such as overall satisfaction and number of reviews which are not available to new rentals would most likely be used by customers to determine how appealing a property is. Features about the rental that take in to account its aesthetic appeal would also certainly be used by customers to determine their willingness to pay. However, features like these would also be extremely subjective, vary from person to person, be hard to quantify, and likely hard to gather.

Therefore, we can imagine many other features of rentals that would be used to set its market price which are not contained in our current data set and would likely be extremely difficult or impossible to gather for a new rental. Furthermore, it is important to highlight that this model is estimating a starting price for a new rental. We are not precluding the price from being updated as more customer information is received. Most of the data we have includes prices which were also most likely adjusted with customer input. Consider Figure 6.5. Each rental in our data set has on average 27 reviews. This means that on average rentals had at least 27 customer reviews to update their prices.

With this context in consideration, it is my opinion that this is a significantly accurate model at estimating a starting price for a our rental. Our estimated price is on average on 62.35 dollars off from the actual price. This actual price can be considered a "market price" as it was likely determined and adjusted with the response of the "market" in the form of significant customer feedback. As said before, our prediction is for a new rental that cannot take into account customer feedback, and therefore should be seen as a starting price which will be adjusted with continued customer input. Having our starting price point therefore only be on average 62.35 dollars off from "market" price point is therefore in my opinion significantly accurate. Being able to explain 52% of the variation in market prices with estimated prices that are effectively acting as starting prices seems similarly impressive. I would argue, therefore, that this model can be used as an effective tool to set a reasonable initial price for a rental which will only need to be adjusted slightly with customer feedback.