Manhattan Real Estate

Jordan Berninger
Kelsey Blackstone

1. Abstract

In this paper, we analyze a data set of Manhattan Real Estate that includes property price in USD. We are interested in determining which models are the most effective at predicting real estate price, as well as which variables are significant predictors. In the *Exploratory Data Analysis* section, we introduce the most important variables in our data set and provide several insightful visualizations. We also justify a log-transformation to our response variable. In *The Data* section, we walk readers through the data cleaning process, which includes re-factoring categorical variables and performing Principal Component Analysis on several highly correlated variables. In the *Modeling* section, we introduce the Multiple Linear Regression, Lasso Regression, Ridge Regression and XGBoost models that we fit. The final model resulted in 10 significant variables for the prediction of property value in Manhattan. In the *Conclusions* section, the reader will find a summary table with the final models' training set fit metric as well as the out-of sample test set performance. Here, one may find residual plots from the final model as well as shortcomings and potential further work for this data. At the end of the *Conclusions* section, we provide a link to the data and all relevant code.

2. Introduction

There are many variables that impact the price of real estate, including location, property size, and access to social services. In recent decades, America's biggest urban environments are becoming increasingly dense and expensive. Manhattan provides a vibrant example of this urban transition and market trend. Using a data set of Manhattan real estate prices, we aim to discover which variables are significant predictors of real estate price. This will give us some idea of what is truly valued in real estate. We expect that if similar analyses were applied longitudinally, we would be able to discern market trends. Additionally, we are interested in developing a model that predicts TotalValue with the most accuracy. An accurate model would provide great utility to real estate brokers or anyone who is considering moving to Manhattan.

We have a continuous response variable and a mix of continuous and categorical predictor variables. Accordingly, we are interested in fitting Multiple Linear Regression, Lasso Regression, and Ridge Regression models in this project. We will fit each model on a training set, determine which variables are deemed significant by the model, predict the response variable on a held-out test set, evaluate the predictive performance, and analyze the residual variables. Additionally, we will highlight which variables are deemed significant by which models. In addition to our proposed regression models, we also fit and tested an XGBoost model in this project as a benchmark comparison.

2.1. Prior Analyses

This data set was presented at the Open Data Science Conference West 2019 by Jared Lander, a lecturer in the Columbia University Statistics Department, during a workshop on XGBoost. In the workshop, Lander fit XGBoost models on the data set and inspected the impact of different hyperparameters on model training time and model performance. This was restricted to binary classification models on the HistoricDistrict variable, so our analysis has some fundamental differences from what we have already seen.

2.2. Goals

Our goals for this analysis are both inferential and predictive. We are interested in finding the best model in order to predict the TotalValue of

properties. As more information on our predictors comes to light through exploratory data analysis and model fitting, we would like to discover which variables are deemed the most significant factors in order to make these predictions. We use Lasso Regression and the Benjamini-Hochberg correction for multiple testing to determine which variables are most significant. Additionally, note that our best model should be user-friendly to any non-statistician. We keep in mind the need for a simplistic model that gives interpretable conclusions, however, there is great value in developing a highly accurate prediction model, even if the model is opaque.

3. The Data

The data set is a CSV with 40,371 rows and 47 columns of data on property characteristics for real estate in Manhattan. Our chosen response variable is the total property value in USD, captured in the 'TotalValue' variable. We have several numeric fields related to the property dimensions, including Building Area (total area of property), Lot Area, Number of Stories, Retail Area and Factory Area.

There are several categorical variables related to the social service districts of the property, such as the type of Fire Service provided to that property, Police Precinct, Health Area, and School District.

In addition, there are a number of columns related to the zoning category of the real estate. This includes Land Use, Class, Zoning District, Lot Type, and Historic District. We also know the century in which the structure was built. The data set also contains several miscellaneous variables.

3.1. Response Variable Transformation

A histogram of the response variable, TotalValue, shown in Figure 1 shows that the data has a long right-tail. In model fitting, the quality of the model tends to improve if the response variable's distribution looks approximately normal; thus, we chose to perform transformations on the response in order to detect normality of the data. Disregarding a short right-tail, transformations revealed that the TotalValue is approximately log-normal, so our analyses will continue using the log-TotalValue as

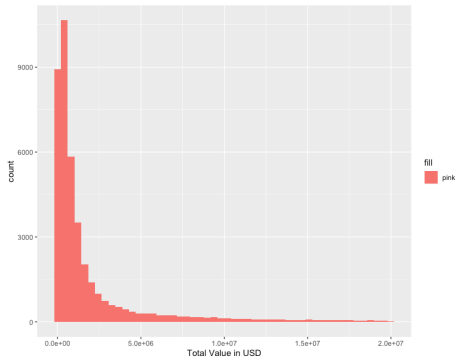

Figure 1: Total Value

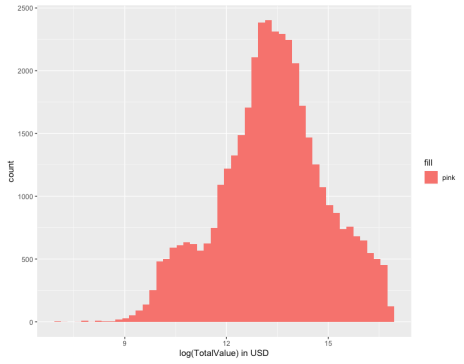

Figure 2: Log Total Value

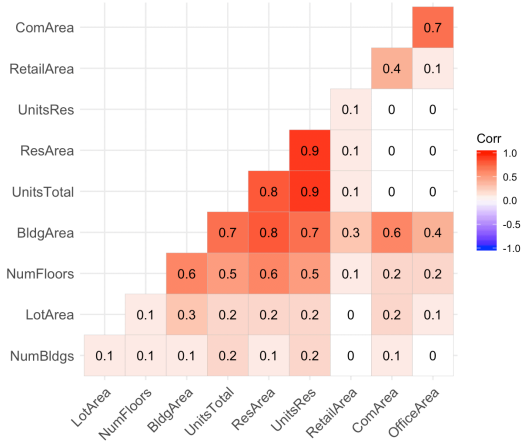

Figure 3: Continuous Variable Correlation

the response variable (shown in Figure 2). However, other transformations could be investigated through the Box-Cox process - we leave this as an area of further study.

3.2. Reducing Number of Parameters

Through exploratory data analysis, we noticed that several of our categorical variables have many, many unique factor levels. We decided to condense some of these factor levels into meaningful groups in order to reduce the number of parameters in our following models.

The "Class" variable, which describes the property function, originally had 25 levels. We refactored this into four new categories: "apartment/home", "business", "misc", and "utility". Additionally, the "Built" column describes which century the property was built. Beginning with ten factor levels, we reduced this to three levels: 18th century, 20th century, and unknown. It is important to note that this data set does not include properties built in the 19th Century; however, we did not take actions (like sub-setting the data set) because of this fact.

We checked for and removed rows that contained NA values; this was a negligible proportion of our data. There are also a number of variables in the data set that we either do not understand or do not have information on. These variables include: "CommFAR", "ResidFAR", "BuiltFAR", "FacilFAR", and "High". We emailed Jared Lander (the data source) regarding these variables, but did not receive a response, and decided to drop these columns from the data frame.

3.2.1. Principal Component Analysis

Our data frame contains ten continuous predictor variables. To consider the problem of multicollinearity, we examined correlation plots exhibited in Figure 3. The figure shows that many of the continuous variables are highly correlated. In order to reduce the dimensionality of our model and account for multicollinearity, we used Principal Component Analysis to identify the five most important area variables: LotArea, BldgArea, ComArea, ResArea, and OfficeArea. The results can be found in Table 1. We determined these five variables were the most important because they have substantial loadings in the first two principal components, which account for 86% of the collective variance. We implemented both *princomp*() and *prcomp*() for this principal component analysis, and used *prcomp*() because it captured more variance within two and three principal components. The remaining continuous variables were then dropped from the training and test sets.

Table 1: PCA Loadings for Continuous Variables

	PC1	PC2
LotArea	0.13	0.04
BldgArea	0.79	0.09
ComArea	0.26	0.69
ResArea	0.52	-0.60
OfficeArea	0.12	0.39
RetailArea	0.04	0.06
NumBldgs	0.00	-0.00
NumFloors	0.00	-0.00
UnitsRes	0.00	-0.00
UnitsTotal	0.00	-0.00

3.3. Training & Test Sets

We then split the data randomly into training and test sets. The training set contained approximately 75% of the observations, 30,278 rows, and the test set contained approximately 25% of the data, 10,093 rows. This 3:15 ratio was chosen arbitrarily, and we leave trying different training:test ratios as an area of further study.

4. Models

For this data, we chose to fit Multiple Linear Regression, Lasso Regression, and Ridge Regression models in order to predict TotalValue. Lasso and Ridge Regression introduce a penalty term on the coefficients, facilitating parameter shrinkage, based on the value of a tuning parameter λ. Lasso can assign a value of 0 to the coefficients of variables it decides are insignificant. These two model shrinkage methods help to reduce variance but also introduce bias in the parameter estimates. Accordingly, we use Lasso as a method for variable selection. In Table 4, we see that a linear model fit with the variables selected by Lasso has higher test set

predictive performance than the analogous Lasso Model.

Throughout the modeling process, we must consider the problem of multiple testing since we have a large number of predictor variables. We employ both the False Discovery Rate and Benjamini-Hochberg p-value corrections on the full linear model to determine which variables are truly significant. Another consideration is the interpretability of the final model. As stated in the *Goals* section, we aim to provide an inferential model for the prediction of TotalValue; this includes providing a comprehensive, user-friendly model.

4.1. Lasso Regression

We used the $cv.glmnet()$ function to fit cross-validated Lasso Regression models with $k = 10$ folds. This trained model returned two different λ values that were derived by different error minimization processes. These values are $\lambda_{min} = 0.003652$ and $\lambda_{1se} = 0.071706$ and can be seen in Figure 4. We also computed the mean of these two values, which gave us $\lambda_{mean} = 0.0376$. These three λ values were then used as the parameter value for the $glmnet()$ models. We track the impact of λ on the models coefficients in Figure 5. Since we are using Lasso as a method for variable selection, we used the largest λ as our penalty term for variable selection. The variables selected when using $\lambda_{1se} = 0.071706$ can be found in Table 2. We predicted our Lasso models on our test set and record the predictive performance in Table 4.

In order to reduce the variance of the coefficients, and thereby let some coefficients go to zero, Lasso Regression introduces bias into the parameter estimates. Accordingly, we fit a Multiple Regression Model using only the variables selected by Lasso. We see in Table 4 that this Lasso Linear Model has a substantially higher out-of-sample R^2 than the original Lasso model, which is a significant finding from this analysis. Using λ_{min} from the cross-validation process, the Lasso model assigns 8 out of 18 variables to have coefficients equal to 0. This leaves us with 10 predictor variables selected as significant by the Lasso Regression Model. We fit a linear model on this subset of variables, whose ANOVA table can be seen in Table 2 and whose co-

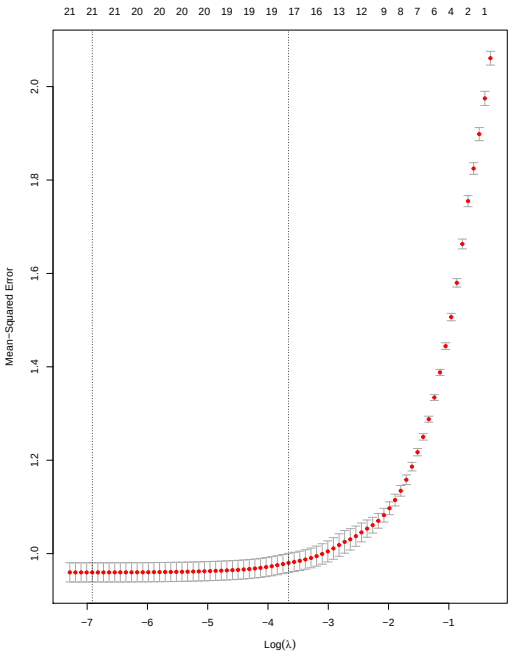

Figure 4: Cross Validated Lasso λ's

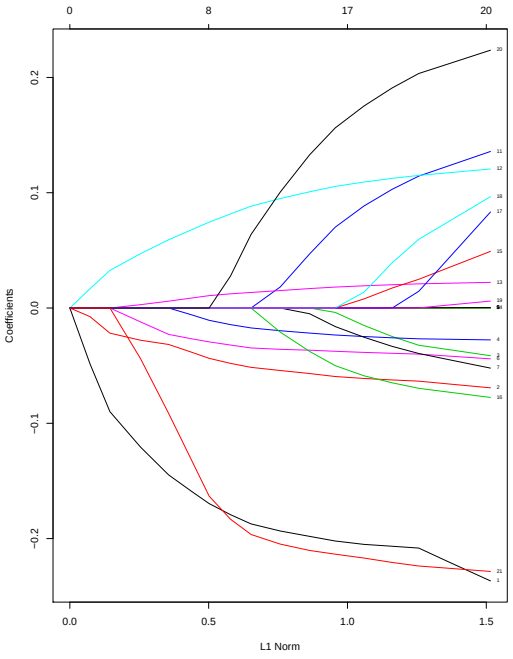

Figure 5: Cross Validated Lasso Coefficients

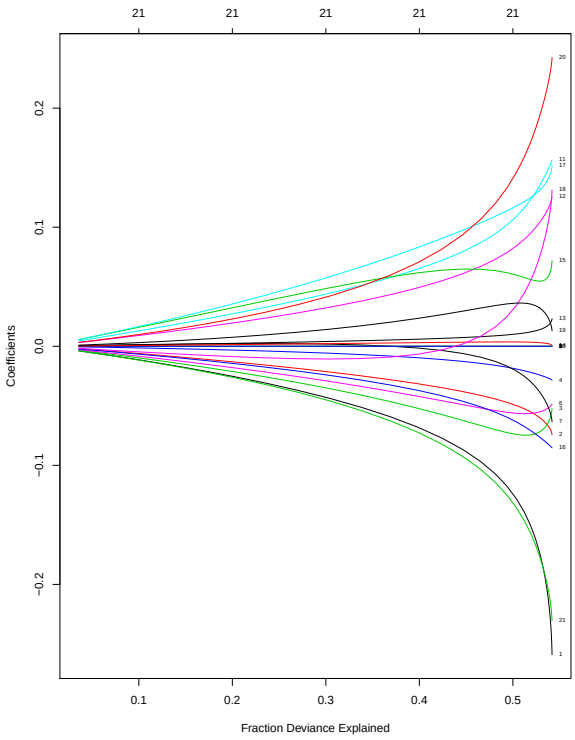

Figure 6: Cross Validated Ridge λ's

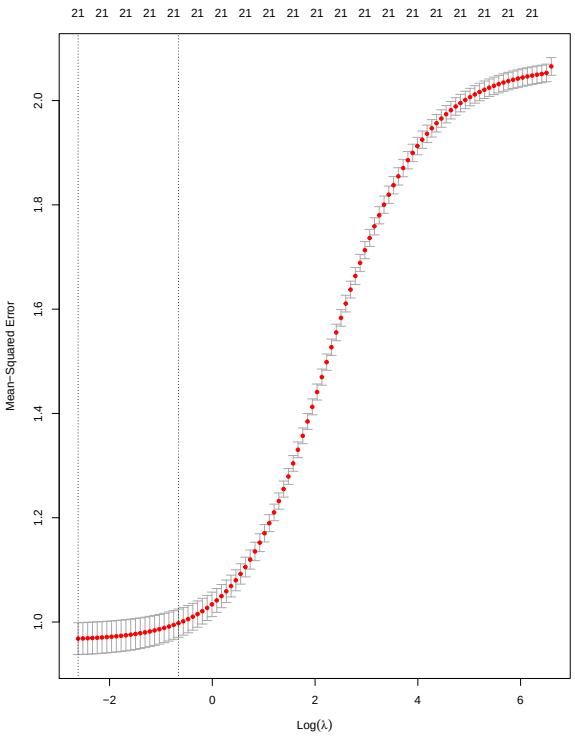

Figure 7: Cross Validated Ridge Coefficients

efficients can be found in Table 3. We note that all of the Lasso Regression models have similar performance metrics, which are lower than that of subsequent linear models. Lasso proves effective as a variable selection method, not a predictive model.

4.2. Ridge Regression

Ridge Regression is similar to Lasso Regression; the main differences are that Ridge assigns a penalty to the sum of the squared magnitude of coefficients and does not set coefficients to zero (rather, they asymptotically decrease towards zero). We followed the same process to fit Ridge models as we did with Lasso, using $cv.glmnet()$ with $k = 10$ folds to compute to different minimizing λ's, and again, computed the mean *lambda*. These values are $\lambda_{min} = 0.08840251$ and $\lambda_{1se} = 0.3568825$ and can be seen in Figure 7. Then we fit $glmnet()$ models with these λ's and computed the out-of-sample predictive performance of the model. In Figure 6 we see the shrinkage of the coefficients across λ values and in Figure 7 we see the impact of λ on the training set MSE.

Since Ridge Regression does not perform variable selection, we did not fit a linear model from the output of the Ridge models. We also note that in the Table 4, there is very little difference in predictive performance from our various Lasso and Ridge models.

5. Full Linear Model with Corrections for Multiple Testing

We first fit a full linear model with our 18 predictor variables using the lm() function. Then, we employed the Benjamini-Hochberg method with $\alpha = 0.05$ to correct the p-values for multiple testing. With this B-H adjustment, we determined three variables were insignificant and dropped them from the model (HealthArea, OfficeArea, and Landmark).

Next, we fit a linear model with only the remaining variables and call this our B-H Linear Model (or LM-BH) from this point forward. This model's ANOVA table is provided in Table 5 and we note that all variables have high F-statistics.

We also investigated the FDR correction method

to correct for multiple testing, but at $\alpha = 0.05$, this adjustment method did not assist in removing variables. The variables in the B-H Linear Model and their F-statistics can be found in Table 2.

5.1. Fitting Linear Model using Lasso Selected Variables

Next, we fit a model using the variables deemed significant by the Lasso Model with $\lambda_{1se} = 0.071706$ into a Linear Regression Model. This was the largest λ returned by the $cv.glmnet()$ function and thus had the highest penalty term on the parameters. We compare the coefficients and the response variance explained by each variable for the full linear model and the Lasso variable linear model in Table 2 and see that the two models are very similar. All variables selected by the Lasso model were also selected by the Benjamini-Hochberg method. Table 2 shows that BldgArea has the highest F-statistic, by far, in both the B-H Linear Model and Lasso Linear Model ANOVA Tables. Additionally, we see very similar orders of variables when we order them by F-statistic. The most noteworthy difference between the two models in Table 2 is that the B-H Linear Model assigns a relatively high F-statistic to Class2, a variable which was not selected by the Lasso model. We also note that Table 2 shows us that Lasso dropped 5 additional variables that were deemed significant after B-H correction (Class2, HistoricDistrict, OwnerType, ComArea, IrregularLot).

We also want to compare the coefficients across our B-H Linear Model and our Lasso Linear Model. If we see that a variable is assigned a positive coefficient by one model and a negative coefficient by the other models, this means the models disagree on the impact of this variable. In Table 3, we see the subset of variables with $|\beta_i| > 1$ and note that no variables' coefficient flips signs across models. This indicates that the models identify similar relationships between the predictors and response, allowing us to have more confidence in their parameter estimates.

5.2. Neighborhood Linear Model

We have three categorical variables that are related to social service districts: SchoolDistrict, Council, and PolicePrct. Since real estate price is heavily influenced by neighborhood, we decided to concatenate the three variables into one interaction term, which is essentially a granular neighborhood variable. We then fit a linear model using only an intercept term and this neighborhood variable. We see in Table 4 that this model accounts for less than one third of the response variance. This was an interesting find, but we did not include this neighborhood variable in any other models.

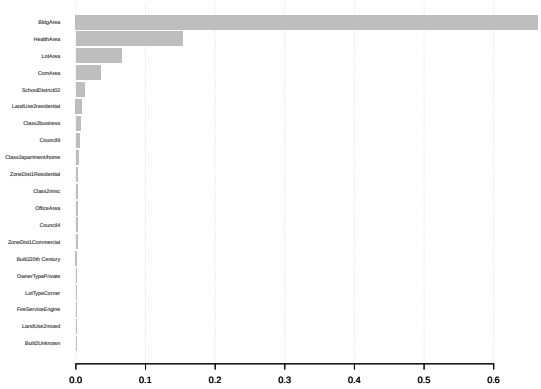

Figure 8: XGBoost Variable Importance

6. XGBoost

XGBoost is a powerful open-source library for machine learning that uses gradient boosting methods that fit an ensemble of decision trees, where each tree is itself a weak classifier and is trained on the errors of the previous tree. The resulting model has very high predictive performance across a wide variety of situations, but it is difficult to ascertain the importance and impact of the predictor variables. We fit an XGBoost model on this data to obtain a benchmark for how well a model can make predictions. In Table 4 we see that our XGBoost model has the best predictive performance. We also see in Figure 8 that it assigns great importance to BldgArea. There are many hyper-parameters we could have tuned in this model, but we did not decide to spend time on this.

Table 2: Comparing B-H and Lasso Linear Models : ANOVA

	Predictor	LM-BH F-Stat	LM-BH p-value	LM-Lasso F Stat	LM-Lasso p-value
1	BldgArea	17520.53	0.00	15986.02	0.00
2	LotArea	3674.33	0.00	4083.60	0.00
3	SchoolDistrict	2575.86	0.00	2308.48	0.00
4	LandUse2	831.17	0.00	435.38	0.00
5	Built2	658.32	0.00	867.15	0.00
6	Class2	608.19	0.00		
7	ZoneDist1	575.63	0.00	515.88	0.00
8	Council	519.65	0.00	465.71	0.00
9	HistoricDistrict	444.41	0.00		
10	OwnerType	439.82	0.00		
11	ComArea	348.10	0.00		
12	LotType	247.54	0.00	241.74	0.00
13	FireService	172.61	0.00	154.69	0.00
14	IrregularLot	115.12	0.00		
15	PolicePrct	97.17	0.00	87.09	0.00

7. Conclusions

Which model is our favorite? Which variables are significant? Table 4 shows the test-set R^2 for each model. The top 4 models are XGBoost, LM-BH, LM-Full (no variables dropped) and LM-Lasso. Clearly, if our main priority is predictive performance, the XGBoost model is best, and Figure 8 tells us that BldgArea is the most important variable by far, in the eyes of XGBoost.

However, if our main goal is inferential - determining which variables are the most important in predicting TotalValue - then we believe the LM-Lasso model is the best. This model is represented in the following form:

$$log(TotalValue)_i = \beta_0 + \beta X_i + \epsilon_i$$

where $log(TotalValue)_i$ is the estimated logTotalValue for observation, i. β_0 is the intercept term, β is the vector of least square estimates for the predictor variables, and X is the vector of values for the predictors.

The LM-Lasso model has a slightly lower test set R^2 compared to LM-BH, but LM-Lasso does so using five fewer predictors. We think this is significantly better - negligible loss in predictive performance with significantly less variables. The LM-Lasso model tells us that BldgArea, LotArea, SchoolDistrict, LandUse2, ZoneDist1, Council, LotType, FireService and PolicePrct are significant covariates. This model captures nearly 60% of the response variance. We note that all three categorical variables related to neighborhood (PolicePrct, Council, SchoolDistrict) are considered significant, in addition to BldgArea. We also note that OwnerType, Class2, HistoricDistrict, ComArea, and IrregularLot (amongst others) are not selected as significant by our Lasso model. In the following section, we inspect the residual plots of the LM-Lasso model to see if there is significant evidence to doubt this model's conclusions. We also provide the ANOVA table for this model in Table 5 at the end of the paper.

7.1. Checking Model Assumptions

Recall that for regression models, we assume that the residuals, ϵ, are independent, identically distributed Normal, with mean zero and equal variance, i.e.:

$$\epsilon \overset{\text{iid}}{\sim} N(0, \sigma^2)$$

for every ϵ. We check these assumptions by looking at the model's residual plots and Normal Q-Q plot. In Figure 10, the Q-Q Plot looks approximately normal for residuals from the -3.5th theoretical quantile to the 4th quantile, but shows a tail for residuals between the -4th and -3.5 quan-

Table 3: Comparing Full and Lasso Linear Models : Coefficients

	Variable	LM-BH Coef	LM-BH p-value	LM-Lasso Coef	LM-Lasso p-value
1	(Intercept)	12.60	0.00	10.93	0.00
2	SchoolDistrict10	1.16	0.01	1.17	0.01
3	Council9	-1.29	0.00	-1.29	0.00
4	PolicePrct50	-2.86	0.00	-2.95	0.00
5	LotTypeBlock Assemblage	2.71	0.00	2.96	0.00
6	LotTypeConnecting	2.94	0.00	3.17	0.00
7	LotTypeCorner	2.82	0.00	3.12	0.00
8	LotTypeInside	2.37	0.00	2.58	0.00
9	LotTypeInterior Lot	0.97	0.00	1.19	0.00
10	LotTypeIsland Lot	3.73	0.00	4.14	0.00
11	LotTypeMixed or Unknown	2.81	0.00	3.15	0.00
12	LotTypeSubmerged Land Lot	1.97	0.00	2.32	0.00
13	LotTypeWaterfront	1.93	0.00	2.36	0.00
14	Class2misc	-1.03	0.00		
15	LandUse2residential	-1.55	0.00	-0.48	0.00

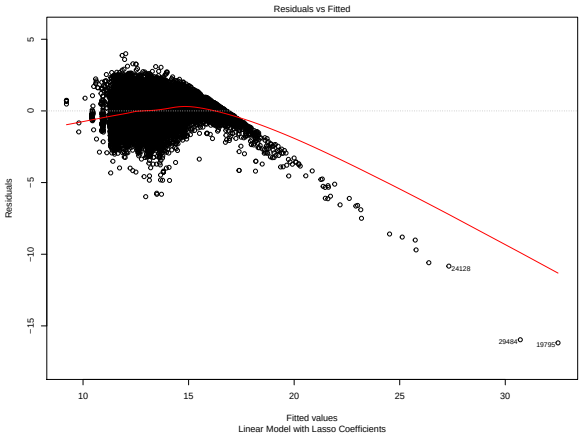

Figure 9: Residuals vs. Fitted Values

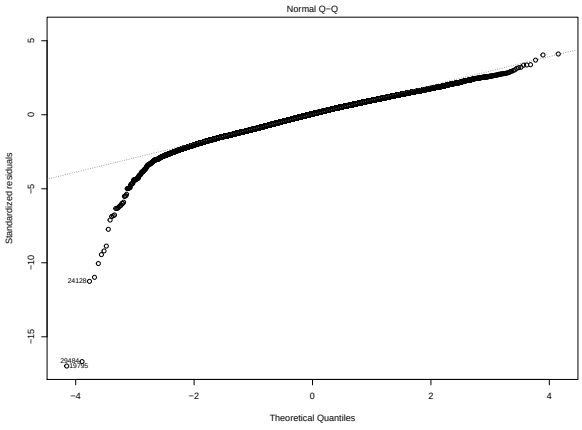

Figure 10: Normal Q-Q Residual Plot

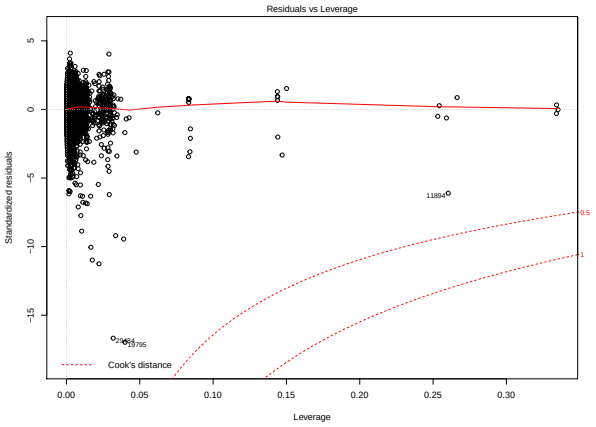

Figure 11: Residuals vs. Leverage Plot

tiles. This very distinctive tail indicates that there is some pattern in a subset of the response variable that we fail to accurately model.

Figure 9 shows that the residuals look approximately symmetric about the zero line for fitted values smaller 15. Fitted values greater than 15 exhibit a clear pattern of these residuals; this means that a subset of our residuals clearly violate our model assumptions. For large fitted values, we have mostly negative residuals, indicating that the predicted points are much higher than our actual values. It follows that our model is over predicting for large values of logTotalValue. Our initial reaction was that there were likely a high number of repeat values for large logTotalValues; this may cause a pattern in the residual plot. However, we did not find a problem with repeated values. We re-evaluated the histogram of logTotalValue and saw that the distribution is not perfectly symmetric - it is skewed with a left tail. It's possible that this distorts the fit of our LM-Lasso model and is one possible explanation for the residual pattern.

Figure 11 shows that outliers may be identified using Cook's distance as a threshold. Cook ' s distance measures the influence that each observation has on the predicted outcome, and can be computed for observation i using:

$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{y}_j - \hat{y}_{j(i)}\right)^2}{pMSE}$$

Where $\hat{y}_{j(i)}$ is the fitted response value for the jth observation not including i, $\hat{y}_j$ is the jth fitted response value, MSE is the mean squared error, and p is the total number of coefficients in the regression model. As determined by Cook's Distance, we do not identify any outliers from this model.

7.2. Further Work

There are several areas for further study. Considering different transformations to TotalValue using the Box-Cox process is a primary area of further study. This may address the issues that we are currently over-fitting large TotalValue properties. Changing the ratio of training to test set size and evaluating the impact would be interesting. We could also subset the data to only buildings from the 20th Century - the lack of data points from the 19th Century is a strange feature of the data set and models might perform better on a specific segment. Since the LM-Lasso was our preferred model, it would be interesting to evaluate how different λ values from $cv.glmnet()$ select variables, and how linear models fit on those variables perform on the test set - this would be like optimizing λ in order to optimize the linear model error, and not the Lasso model error. Additionally, repeating this analysis longitudinally could give us insight into trends in what the real estate market values.

7.3. Reproducing This Work

A CSV of the data and all relevant **R** code can be found at:

1. https://github.com/jberninger/ucsc_stats204_fall2019
2. https://github.com/kelseyblackstone/STAT204_Final

Table 4: Results Table : Models' Training and Test Performance

	Model	Training Metric	Training Metric Value	Test R^2
1	LM - NULL	AIC	27063.099	-0.0001
2	LM - FULL	AIC	-4772.25	0.6297
3	LM - B.H.	AIC	-4774.746	0.6298
4	LM - LASSO	AIC	-1466.45	0.5954
5	LM - Hood	AIC	17662.056	0.2660
6	LASSO.cv.lambda.1se	Dev Ratio	0.505	0.4735
7	LASSO.cv.lambda.min	Dev Ratio	0.531	0.4961
8	LASSO.cv.lambda.mean	Dev Ratio	0.531	0.4986
9	RIDGE.cv.lambda.1se	Dev Ratio	0.512	0.4949
10	RIDGE.cv.lambda.min	Dev Ratio	0.529	0.5004
11	RIDGE.cv.lambda.mean	Dev Ratio	0.521	0.4995
12	XGBOOST	?	?	0.8885

Table 5: LM-Lasso (Best Model) ANOVA Table

	term	df	sumsq	meansq	statistic	p.value
1	SchoolDistrict	6	13169.92	2194.99	2308.48	0.00
2	Council	9	3985.33	442.81	465.71	0.00
3	FireService	2	294.17	147.08	154.69	0.00
4	PolicePrct	22	1821.73	82.81	87.09	0.00
5	ZoneDist1	4	1962.08	490.52	515.88	0.00
6	LotArea	1	3882.84	3882.84	4083.60	0.00
7	BldgArea	1	15200.10	15200.10	15986.02	0.00
8	LotType	9	2068.68	229.85	241.74	0.00
9	Built2	2	1649.03	824.52	867.15	0.00
10	LandUse2	3	1241.92	413.97	435.38	0.00
11	Residuals	30218	28732.40	0.95		