Case 1: Iowa House Price

RSM338

1006581353 - Dawood Khokhar
1006776375 - Marjorie He
1006722101 - Jeffrey Chen
1006769613 - Carson Feng
1005990849 - Sherry You

The objective of creating this Iowa house price model is to choose the best prediction model, whether it be Linear, Lasso, or Ridge Regression. A good model can be identified once we compare the training and validation set's mean squared error (MSE). Whichever model has the lowest and most consistent MSE is neither under-fitted nor over-fitted. Additionally, we want to gauge each model's performance with the addition of four variables. Adding more terms to the multiple regression inherently improves the fit since it gives a new coefficient to force a better fit. So, we must be wary that adding more variables makes it more likely to overfit (making up trends that don't exist to connect existing points) the model to the training data. Lasso and Ridge regression may prove valuable in this scenario since both are regularization techniques that reduce overfitting.

To make our modified Iowa house price model, we added variables that would intuitively impact house prices such as the Lot Frontage, Lot Shape of a house, Kitchen Quality, and the square footage of low quality finished floors. Also, we wanted to avoid adding new variables that are highly correlated with existing variables as this would weaken the model as it becomes biased towards one direction.

The first variable we added, Lot Frontage, describes the linear feet of the street connected to the property. We'd expect a positive relationship between feet connected to the property and house price. Secondly, we added the Lot Shape variable, which describes the general shape of the property. Interestingly, the more irregular the lot shape, the higher the house price. Since this was categorical data, we changed the labels to correspond with the appropriate dummy variable. "Irregular" lot shapes were assigned a value of 3, "Moderately Irregular" was given a value of 2, "Slightly Irregular" was given a value of 1, and "Regular" was given a value of 0. We gave a higher value to the more irregular values because it will contribute to a higher house price once it's normalized. This is proven in figure 4 as lot shape and sale price are positively correlated i.e the increased dummy values for more irregular shaped lots was correct because more irregularly shaped lots contribute to higher prices.

For the third additional variable, we chose Kitchen Quality (KitchenQual). Kitchen quality is a fundamental factor that many people consider when buying a home, the better the kitchen quality the higher the price. Earning a characterization of "Excellent" should increase the house price, while "Poor" should decrease the house price. Since Kitchen Qual is a categorical variable, we had to transform it into a numerical format. "Excellent" is assigned a dummy variable of 4, "Good" is assigned a variable of 3, "Typical" is assigned a value of 2, "Fair" is assigned a value of 1, and "Poor" is assigned a value of 0. These value assignments created a positive relationship between the kitchen quality variable and sales price (Figure 4).

Finally, we decided to include the fourth variable LowQualFinSF, which describes the low-quality finished square feet on all floors. Intuitively, as the square footage of low-quality floors increases, the house price should decrease. One thing to note about this variable is that most of the data points have 0 LowQualFinSF, which means it should not significantly affect the model.

As seen in figure 4, our new variables were not too closely correlated with the existing variables, so we proceeded with them. While creating the revised model with 4 new variables, we had to split the data between the training, validation, and test set before imputing. The division of the data is an attempt to replicate the situation where the model is built on past information (training) which will be tested on

future (unknown) information (test set). This means all the pre-processing of data, such as imputing missing values, should be done on the training set alone. This is because the algorithm is being trained on the training set, and the best guess for data we haven't seen in the validation and test set, is the mean of data we've seen before. If test data was used to build the model, it ceases to be test data and the model risks overfitting.

We considered 2 imputation methods: mean imputation and linear regression. With both methods we first cleaned the data and replaced all the missing values with "Missing." Then, with mean imputation, we replaced all missing values with the mean of the training set (data points 1-1800). However, some limitations of mean imputation is that it reduces the variance of the imputed variables, shrinks standard errors (which invalidates most hypothesis tests and the calculation of confidence intervals) and does not preserve relationships between variables such as correlations. Conversely, our second imputation method (the one we used), Linear Regression imputation creates a linear function using the training set to predict missing values. The advantage of linear regression imputation is that it preserves the relationship between the variables and often leads to a more accurate estimate of the missing values. Therefore, we used linear regression as our imputation method. We then normalized the data of all 4 additional variables and started creating our new Iowa House price model to report on its performance.

After evaluating all the models, we decided to use Lasso as the model for prediction since it had the lowest MSE scores across its training and validation sets. The Lasso technique is a modification of linear regression, where the model is penalized for the sum of absolute values of the weights. It shrinks the overfitting model and changes unnecessary variables to 0 as they are not statistically significant. When the extra variables are removed (set to 0), the model will be more precise when predicting the housing prices. The reason why Lasso performed better is because there was a small number of significant parameters and while the others are close to zero which simplifies its predicting process. Examples of insignificant variables include neighbourhood, lot shape, LowQualFinSF, etc. Conversely, Ridge works well if there are many large parameters of about the same value (when most predictors impact the response), which was not the case here. Therefore, we decided to evaluate the Lasso model.

The first major difference between the Lasso regressions with the added 4 features was the value of the intercept. Before the addition of the 4 new variables, the intercept of the model was -0.0, meaning that the linear regression went through the origin. However, after the addition of the 4 variables, the intercept became -0.001159, meaning the line changed. The next major difference was that adding the 4 features reduced some new variable coefficients to 0. For example, the variables "WoodDeckSF" and ""OpenPorchSF" held coefficients of 0.001029 and 0.00215 respectively before the addition, but both became 0 after (Figure 1 & 2). This indicates that the model was incomplete; the variable coefficients that were reduced to 0 were previously falsely attributed to influencing housing prices. Once the more statistically significant variables were added, these non-significant variables became useless and the Lasso regularization was able to remove them from the model. This, in turn, reduced overfitting and made the model more accurate as there is no chance that these insignificant variables could fit incorrectly to data in the training set. Additionally, we noticed that 2 of the additional variables added, LotShape and LowQualFinSF, held coefficients of 0 once added into the model. This indicates that these variables are not statistically significant towards the model and held no impact. However, the other two variables

KitchenQual and LotFrontage proved to be significant, with coefficient values of 0.115333 and 0.016792 respectively.

When we compared the Training Lasso Regression MSE to the Validation Lasso regression MSE, we observed that the MSE of the validation model has a higher MSE than the training model. At alpha level 0.005, the MSE computed with the training data set is 0.116548, while the validation set MSE is 0.118786. We notice the same pattern throughout all the other regressions (regression with four features and randomized regression). The difference between the MSE is due to the fact that the training model was fitted with the validation data, therefore when we introduce new data in the testing set, there will be an increase in MSE. In addition, we noticed that as the alphas increased, the MSE for each of the regressions also increased because alpha is the significance level which reduces the confidence level for the model.

After the addition of the four features, the MSE for both the validation set using the training set model and the MSE for the testing set have both been lowered because the addition of more variables creates a more accurate model (there's more data in the training set to analyze). At alpha level 0.005, the MSE with the validation set using the training set is lowered from 0.116548 to 0.110267. The MSE with testing set is lowered from 0.118786 to 0.112384. This phenomenon shows these 4 variables were statistically significant to predicting houses, resulting in a lower MSE and a more accurate model.

Finally, we wanted to evaluate the effectiveness of our Iowa house price Lasso Regression model when the data was randomized. Randomization could happen in 2 different ways: either the data is spliced differently using different ratios for the training, validation and testing set (ex. 80/10/10) or to shuffle the data in a different order but maintain the same ratios for the division of data (60/20/20). We decided to shuffle the data. Shuffling the data as a key implication on the imputation of the missing values. Since, the training data no longer consists of the same points, the linear regression model created for the missing values would also be different. Given the results in Figure 7, the random lasso training MSE increased and did slightly worse compared to all previous regression models. However, in the random Lasso Validation set (Figure 7), the MSE did significantly worse indicating that randomization had a negative effect in the model across all different alphas. The variation that can be explained by the model is much better at smaller alpha levels and as the alpha increases, the variation that the model can explain decreases. The reason for the higher MSE can be attributed to the possibility the model could be inaccurate if the randomized data in the training set is distinctly different from the data in the validation and test sets. Additionally, there was little change in the coefficients of the model; those with non-zero coefficients stayed similar and those with 0 coefficients changed to 0. The only change was the variable NoRidge, which became 0. Overall, the randomization to the data improves the lasso regression significantly.

Overall, using the various methods of regression, we have determined that Lasso regression is an optimal model to predict a precise price target for the Iowa housing market. Furthermore, the addition of other features with low correlation and randomizing the data would improve the model even more and decrease the overall MSE and can further explain the variation of the regression model.

Figure 1: Lasso Coefficients Before Addition of Variables

intercept	-0.0
LotArea	0.044304
OverallQual	0.298079
OverallCond	0.0
YearBuilt	0.052091
YearRemodAdd	0.064471
BsmtFinSF1	0.115875
BsmtUnfSF	-0.0
TotalBsmtSF	0.10312
1stFlrSF	0.032295
2ndFlrSF	0.0
GrLivArea	0.297065
FullBath	0.0
HalfBath	0.0
BedroomAbvGr	-0.0
TotRmsAbvGrd	0.0
Fireplaces	0.020404
GarageCars	0.027512
GarageArea	0.06641
WoodDeckSF	0.001029
OpenPorchSF	0.00215
EnclosedPorch	-0.0
Blmngtn	-0.0
Blueste	-0.0
BrDale	-0.0
BrkSide	0.0
ClearCr	0.0
CollgCr	-0.0
Crawfor	0.0
Edwards	-0.0
Gilbert	0.0
IDOTRR	-0.0
MeadowV	-0.0
Mitchel	-0.0
Names	-0.0
NoRidge	0.013209
NPkVill	-0.0
NriddgHt	0.084299
NWAmes	-0.0
OLDTown	-0.0
SWISU	-0.0
Sawyer	-0.0
SawyerW	-0.0
Somerst	0.0
StoneBr	0.016815
Timber	0.0
Veenker	0.0
Bsmt Qual	0.020275

Figure 2: Lasso Coefficients After Addition of Variables

intercept	-0.001159
LotArea	0.04084
OverallQual	0.264567
OverallCond	0.0
YearBuilt	0.046394
YearRemodAdd	0.028091
BsmtFinSF1	0.113352
BsmtUnfSF	-0.0
TotalBsmtSF	0.098285
1stFlrSF	0.026765
2ndFlrSF	0.0
GrLivArea	0.291035
FullBath	0.0
HalfBath	0.0
BedroomAbvGr	-0.0
TotRmsAbvGrd	0.0
Fireplaces	0.019954
GarageCars	0.026216
GarageArea	0.05591
WoodDeckSF	0.0
OpenPorchSF	0.0
EnclosedPorch	-0.0
Blmngtn	-0.0
Blueste	-0.0
BrDale	-0.0
BrkSide	0.0
ClearCr	0.0
CollgCr	-0.0
Crawfor	0.0
Edwards	-0.0
Gilbert	0.0
IDOTRR	-0.0
MeadowV	-0.0
Mitchel	-0.0
Names	-0.0
NoRidge	0.013647
NPkVill	-0.0
NriddgHt	0.070927
NWAmes	-0.0
OLDTown	-0.0
SWISU	-0.0
Sawyer	-0.0
SawyerW	-0.0
Somerst	0.0
StoneBr	0.014073
Timber	0.0
Veenker	0.0
BsmtQual	0.020271
LotFrontage	0.016792
LotShape	0.0
LowQualFinSF	-0.0
KitchenQual	0.115333

Figure 3: Computing MSE

alpha: 0.005	Before the four features	After the four features
MSE with validation set using training set model	0.116548	0.110267
MSE with testing set using testing set model	0.118786	0.112384
alpha: 0.0100	Before the four features	After the four features
MSE with validation set using training set model	0.116827	0.110579
MSE with testing set using testing set model	0.11999	0.112945
alpha: 0.0150	Before the four features	After the four features
MSE with validation set using training set model	0.118033	0.112133
MSE with testing set using testing set model	0.122372	0.1
alpha: 0.0200	Before the four features	After the four features
MSE with validation set using training set model	0.120128	0.114842
MSE with testing set using testing set model	0.12542	0.1166
alpha: 0.0250	Before the four features	After the four features
MSE with validation set using training set model	0.123015	0.118476
MSE with testing set using testing set model	0.128881	0.119773
alpha: 0.0375	Before the four features	After the four features
MSE with validation set using training set model	0.131786	0.127749
MSE with testing set using testing set model	0.138769	0.128637
alpha: 0.05	Before the four features	After the four features
MSE with validation set using training set model	0.140172	0.136366
MSE with testing set using testing set model	0.147205	0.136472

Figure 4: Correlation between the four chosen features and Sale Price

	LotFrontage	LotShape	LowQualFinSF	KitchenQual	SalePrice
LotFrontage	1.000000	0.193488	0.178451	0.144791	0.362517
LotShape	0.193488	1.000000	0.066384	0.181144	0.295283
LowQualFinSF	0.178451	0.066384	1.000000	0.213640	0.166605
KitchenQual	0.144791	0.181144	0.213640	1.000000	0.676551
SalePrice	0.362517	0.295283	0.166605	0.676551	1.000000

Figure 5: Comparison of Training and Validation MSE of the Original Lasso Regression

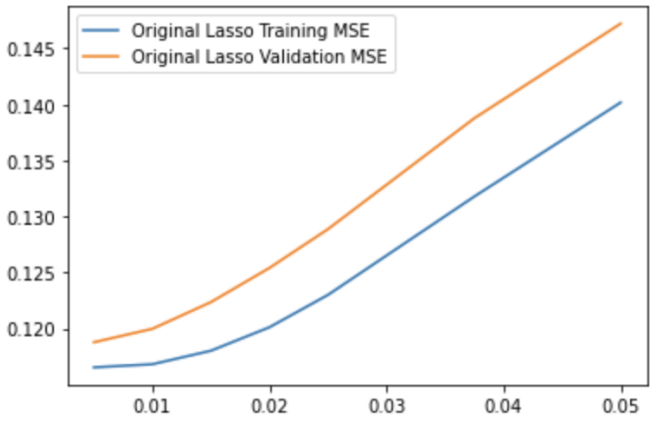

Figure 6: Comparison of Training and Validation MSE of the New and Random Lasso Regressions

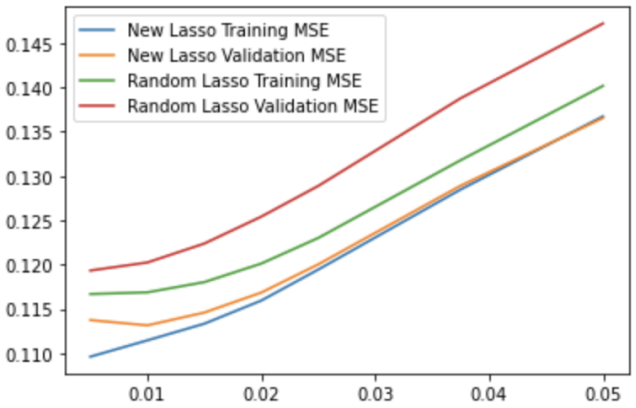

Figure 7: MSE Results of Multiple Lasso Regressions for Each Alpha Value

	Alphas	Original Lasso Training MSE	Original Lasso Validation MSE	New Lasso Training MSE	New Lasso Validation MSE	Random Lasso Training MSE	Random Lasso Validation MSE
0	0.0050	0.116548	0.118786	0.109627	0.113757	0.116698	0.119344
1	0.0100	0.116827	0.119990	0.111468	0.113159	0.116882	0.120249
2	0.0150	0.118033	0.122372	0.113345	0.114602	0.118035	0.122383
3	0.0200	0.120128	0.125420	0.115954	0.116856	0.120128	0.125420
4	0.0250	0.123015	0.128881	0.119451	0.120036	0.123015	0.128881
5	0.0375	0.131786	0.138769	0.128482	0.128913	0.131786	0.138769
6	0.0500	0.140172	0.147205	0.136729	0.136549	0.140172	0.147205