Introduction

We compared lasso, ridge and Linear regression in this study.

Assumptions:

Total number of features after creating dummy variables = 232

We did not drop multi-correlated features:

1. Lasso performs feature selection
2. Ridge will drive down values of bad features towards zeo
3. We wanted to see how badly Linear regression performs with
    a. Many features some of which are multi-correlated

We use **Robust scaling**, because some variables are skewed and Robust Scaling has less effect of outliers.

We performed log transformation on SalePrice because it is skewed.

We used Stratified sampling with a dummy variable to create train/test sets, because even after log transform there was a slight skew.

Derived variables for year column were created:

df['Age'] = 2022 - df['YearBuilt']

df['RemodAge'] = 2022 - df['YearRemodAdd']

df['GarageAge'] = 2022 - df['GarageYrBlt']

Question 1

What is the optimal value of alpha for ridge and lasso regression? What will be the changes in the model if you choose double the value of alpha for both ridge and lasso? What will be the most important predictor variables after the change is implemented?

Answer 1

	alpha	features	r2_train	r2_test	mse_train	mse_test	mae_train	mae_test	mse_test_to_train
Linear	-	230	0.9523	0.8415	0.0070	0.0217	0.0612	0.0920	3.0977
Ridge	15.0000	228	0.9340	0.9034	0.0097	0.0132	0.0699	0.0780	1.3622
Lasso	0.0010	69	0.9237	0.9004	0.0112	0.0136	0.0748	0.0782	1.2156

Best values are chosen by visual inspection instead of using model.best_params_.

Value which had minimum std deviation for train, and lower difference between test and train score was chosen.

Ridge: 15

Lasso: 0.001

These values get rid of the overfitting that happened in Linear Regression model.

They have lower test errors but higher train errors, which is expected. This means that the models are more robust than Linear regression.

Multiply best values by 2

	alpha	features	r2_train	r2_test	mse_train	mse_test	mae_train	mae_test	mse_test_to_train
Linear	-	230	0.9523	0.8415	0.0070	0.0217	0.0612	0.0920	3.0977
Ridge	15.0000	228	0.9340	0.9034	0.0097	0.0132	0.0699	0.0780	1.3622
m2Ridge	30.0000	228	0.9288	0.9029	0.0104	0.0133	0.0725	0.0784	1.2710
Lasso	0.0010	69	0.9237	0.9004	0.0112	0.0136	0.0748	0.0782	1.2156
m2Lasso	0.0020	49	0.9143	0.8932	0.0126	0.0146	0.0793	0.0815	1.1615

When we double values optimal alpha for lasso and ridge:

1. Train and test erros increased – This is expected because increasing alpha increases bias in the model.
2. Train and test Adjusted R-squared decreased – This was also expected because as bias increases and variability decreases, model becomes less accurate
3. Mse_test_to_train decreases – This is ratio of trian error / Test Error – This decreased, which indicates, that although the model is

a. less accurate,
b. it is overfitting less – variability decreased and bias increased
c. Model is more Robust

4. For lasso – Number of features decreased from 69 to 49
5. Number of features for Ridge is same

Top 10 important predictor variables with coefficients (**negative coeff means variable is negatively correlated**):

Original Best alpha		Multiply alpha by 2	
	Ridge		**m2Ridge**
OverallQual	0.1070	**MSZoning_RM**	0.1073
GrLivArea	0.0967	**Utilities_NoSeWa**	0.0904
Neighborhood_Crawfor	0.0864	**Exterior1st_HdBoard**	0.0652
MSZoning_RL	0.0684	**Exterior2nd_AsphShn**	-0.0582
Neighborhood_IDOTRR	-0.0661	**LandSlope_Sev**	0.0540
Age	-0.0652	**Condition1_RRNe**	0.0499
SaleCondition_Normal	0.0608	**Condition2_RRAe**	0.0499
CentralAir_Y	0.0581	**GarageType_CarPort**	0.0493
Neighborhood_StoneBr	0.0563	**Neighborhood_Somerst**	-0.0481
TotalBsmtSF	0.0532	**Exterior1st_BrkComm**	0.0468

Original Best alpha - Lasso		Multiply alpha by 2 - Lasso	
	Lasso		**m2Lasso**
GrLivArea	0.1668	**SaleType_New**	0.1631
OverallQual	0.1284	**MSZoning_RL**	0.1412
SaleType_New	0.1013	**Neighborhood_Edwards**	-0.0941
Neighborhood_Crawfor	0.0989	**3SsnPorch**	0.0740
Age	-0.0858	**KitchenQual_Gd**	0.0647
Neighborhood_Somerst	0.0697	**HeatingQC_TA**	0.0554
MSZoning_RL	0.0609	**SaleCondition_Normal**	0.0551
SaleCondition_Normal	0.0520	**TotalBsmtSF**	0.0545
BsmtFinSF1	0.0520	**WoodDeckSF**	0.0513
TotalBsmtSF	0.0520	**Neighborhood_IDOTRR**	0.0450

We can see that top 10 predictors are different for both, Ridge and Lasso when alpha is increased.

Question 2

You have determined the optimal value of lambda for ridge and lasso regression during the assignment. Now, which one will you choose to apply and why?

Answer 2

	alpha	features	r2_train	r2_test	mse_train	mse_test	mae_train	mae_test	mse_test_to_train
Linear	-	230	0.9523	0.8415	0.0070	0.0217	0.0612	0.0920	3.0977
Ridge	15.0000	228	0.9340	0.9034	0.0097	0.0132	0.0699	0.0780	1.3622
Lasso	0.0010	69	0.9237	0.9004	0.0112	0.0136	0.0748	0.0782	1.2156

We will choose to apply Lasso regression because:

1. It has only 69 features.
    a. This will make it easy to explain and interpret.
    b. This also makes the model more robust.
2. Adjusted R2-square is comparable to ridge. But ridge uses 228 features. So ridge model is more complex than Lasso.
3. Linear regression is overfitting – Adjusted R2-sqaured are 0.95 (train) and 0.84 (test)

Residual plots and plots for coefficients are given in the notebook.

Question 3

After building the model, you realised that the five most important predictor variables in the lasso model are not available in the incoming data. You will now have to create another model excluding the five most important predictor variables. Which are the five most important predictor variables now?

Answer 3

	alpha	features	r2_train	r2_test	mse_train	mse_test	mae_train	mae_test	mse_test_to_train
Linear	-	230	0.9523	0.8415	0.0070	0.0217	0.0612	0.0920	3.0977
Ridge	15.0000	228	0.9340	0.9034	0.0097	0.0132	0.0699	0.0780	1.3622
Lasso	0.0010	69	0.9237	0.9004	0.0112	0.0136	0.0748	0.0782	1.2156
LassoDrop	0.0010	82	0.9114	0.8868	0.0130	0.0155	0.0811	0.0857	1.1901

Dropping top 5 features results in:

1. Higher train/test errors.
2. Lower Adjusted R-squared, thus lower accuracy.
3. Higer number of features with **SAME alpha.**

Following are the top 5 most important features with coefficients:

	LassoDropped
2ndFlrSF	0.1858
1stFlrSF	0.1456
SaleCondition_Partial	0.1033
Neighborhood_Somerst	0.0949
Functional_Typ	0.0776

Question 4

How can you make sure that a model is robust and generalisable? What are the implications of the same for the accuracy of the model and why?

Answer 4

	alpha	features	r2_train	r2_test	mse_train	mse_test	mae_train	mae_test	mse_test_to_train
Linear	-	230	0.9523	0.8415	0.0070	0.0217	0.0612	0.0920	3.0977
Ridge	15.0000	228	0.9340	0.9034	0.0097	0.0132	0.0699	0.0780	1.3622
m2Ridge	30.0000	228	0.9288	0.9029	0.0104	0.0133	0.0725	0.0784	1.2710
Lasso	0.0010	69	0.9237	0.9004	0.0112	0.0136	0.0748	0.0782	1.2156
m2Lasso	0.0020	49	0.9143	0.8932	0.0126	0.0146	0.0793	0.0815	1.1615

To make a model robust and generalizable:

1. Make sure to use regularization, to increase bias and decrease variability
2. Use RFE/feature elimination to get rid of multi-colinear features
3. Make sure that

   abs(Train Adjusted R-Squared - Train Adjusted R-Squared) < 5

4. Make sure that Error terms are randomly distributed with mean zero
5. The fewer features the better

All of the above my imply that we may have to choose a less accurate model to get a robust model.

This happens because we are purposely using a model with:

1. higher bias
2. Fewer features
3. Easier to explain and use in the real world