Data Mining: Project 3

Akash Shanmugam

10/28/24

Dataset link:

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/

Data Introduction:

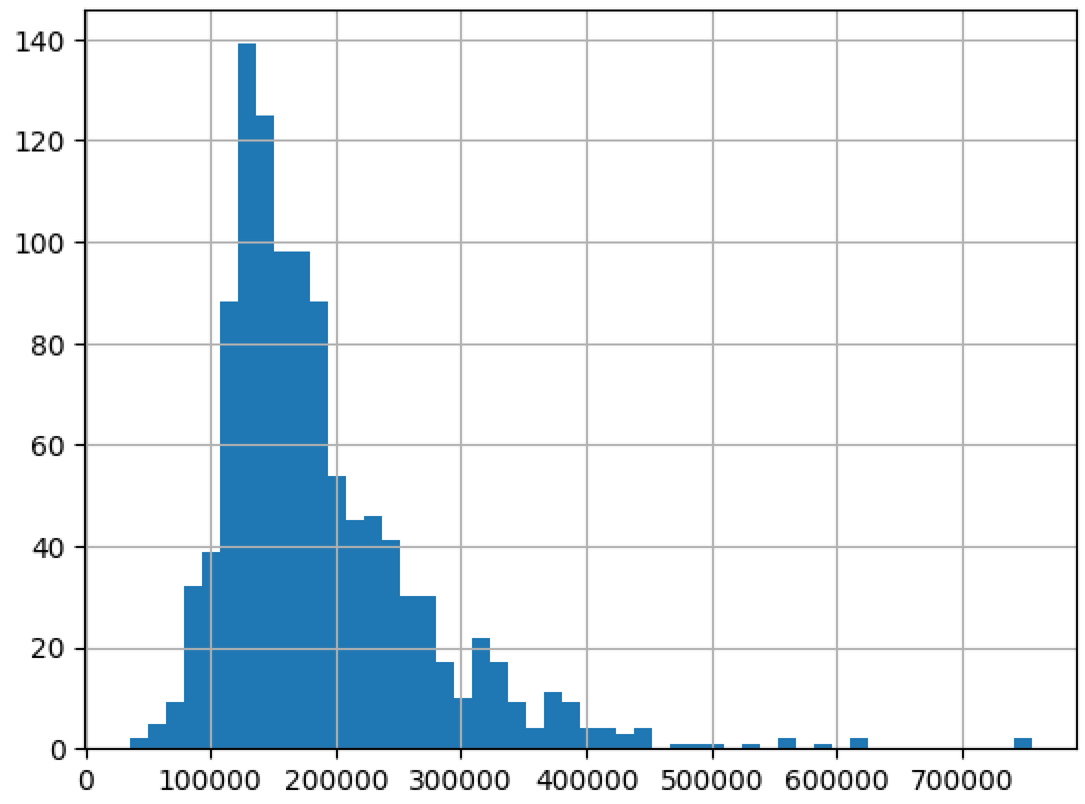
The dataset I decided to use for this data mining project is the provided “House Prices - Advanced Regression Techniques” dataset and more specifically the train.csv part of the dataset as I wanted to see if I could split the data from that to create a training and testing csv and I didn’t want to wait to get the test.csv file after submitting my code. The training dataset by itself contained 1460 rows of data with 80 whole features. As intimidating as this was, I knew I had to do preprocessing anyway to reduce most of them for my model. The goal of the project/dataset itself is to be able to accurately predict final sales prices of houses based on all the features. The dataset itself is part of a larger competition on Kaggle to be able to create different types of regression models to handle the data more and more accurately. This project will mostly be focusing on a few regression models in comparison to each other rather than from the competition as a whole.

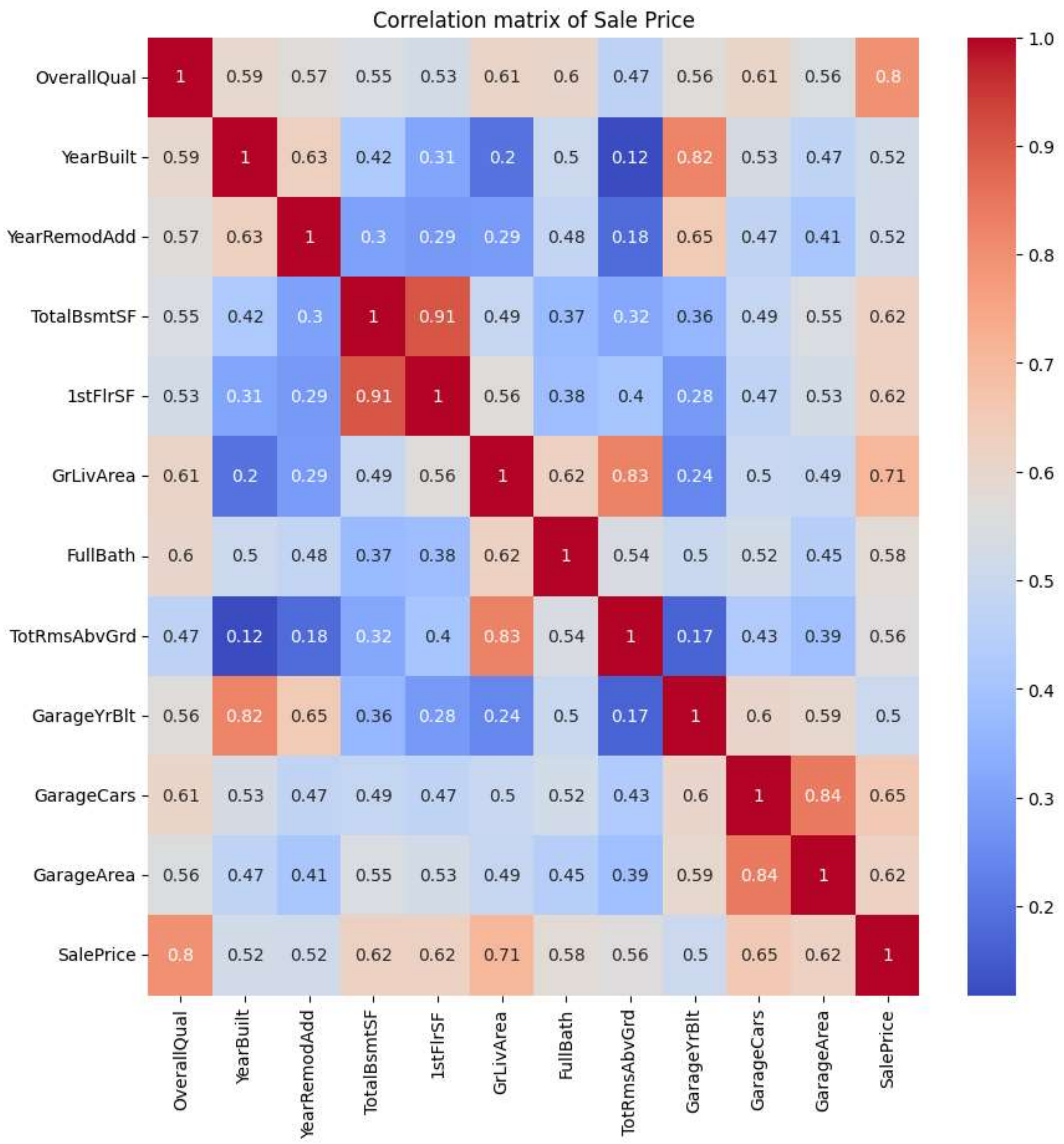
Pre-Processing the Data:

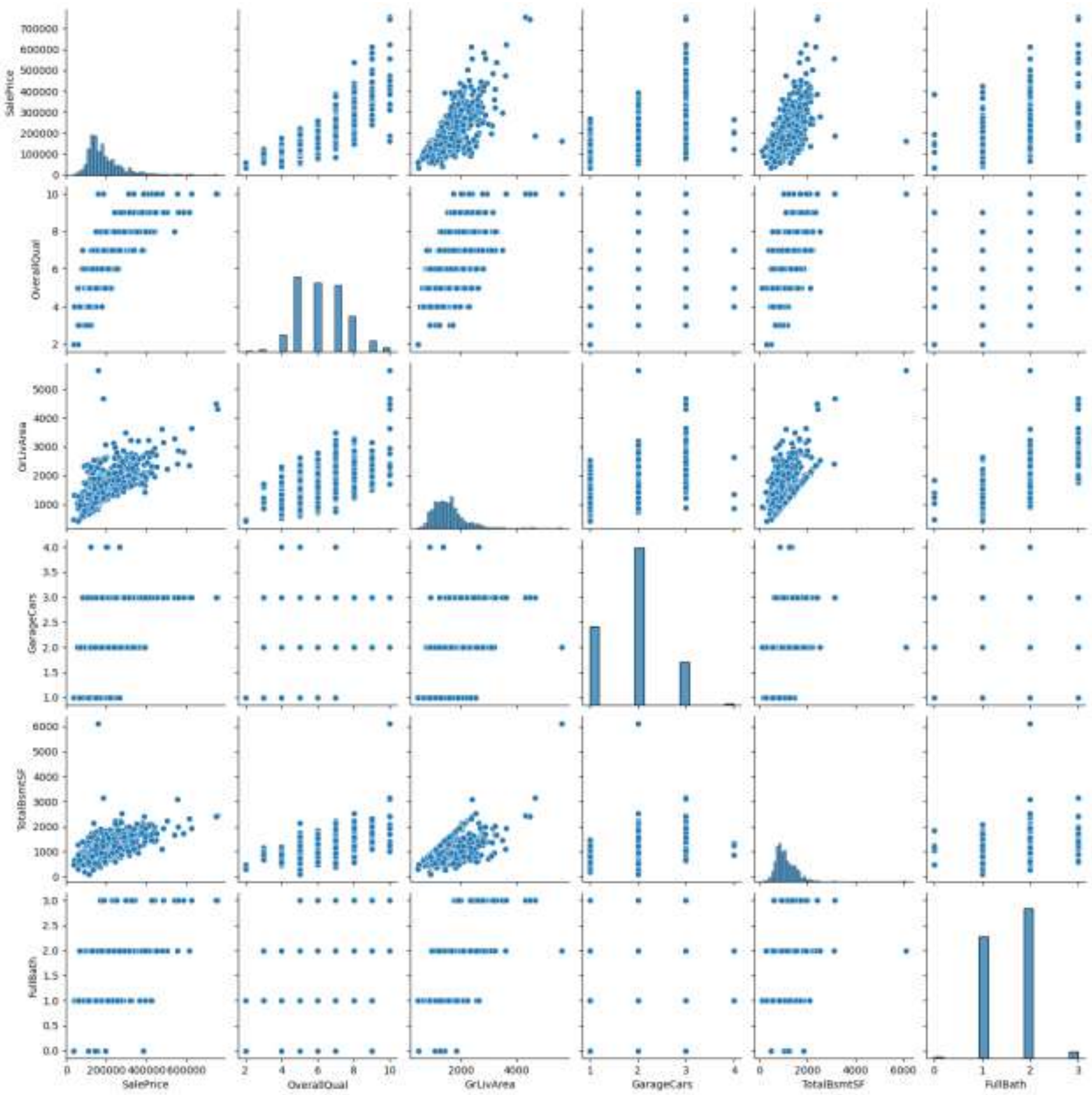
The data in the dataset immediately stood out as having way too many features to feasibly make a model with, therefore the first thing I needed to do was understand the data enough to thoroughly examine what features needed to be cut. Looking at the data types, I could tell that the data types varied a lot from integers to objects. I needed to first look at the features with the most null values given that the amount of features that existed could lead to some having much more than others. During this, it was clear that a few columns here skewed heavily into being mostly null: PoolQC, MiscFeature, Alley, Fence, MasVnrType, FireplaceQu. All of these attributes had more null values than half of the rows in the dataset and clearly had to be dropped. I looked at the remaining mean of the null value rows to make sure there were no outliers and then dropped the rows that I could. This left me with 1094 rows and 75 columns.

Data Understanding:

With the current type of data in the dataset and the types of regressions I wanted to make, I excluded the object type features to understand the numerical correlation between the features and the main objective that is needed for this project: the Sales Price. First I checked the outliers in the Sales Price data to make sure the values wouldn’t be massively skewed and with a histogram saw that the data only had very few outliers that would likely not affect the model making. Using this, I made a correlation matrix in seaborn using a heatmap to see what features correlated with the sales price metric. Looking at the graph, it's pretty clear that some features specifically are more correlating than others with visible ones being “OverallQuality”, “GrLiveArea”, or “GarageCars”. This was the initial look into what columns I needed to narrow in on. Next I used the correlation feature to look at the same data but purely numerically sorted and found the top couple of features that were independent of each other to create a pairplot of. The pairplot as a whole showed some odd visualizations due to the unscaled nature of the data, there were still clear correlations between other features an example being “GrLivArea” and “TotalBsmtSF”. At this point I made the decision to use the version of the dataset that only contained the numerical features for the regression models to be able to see the data clearly. I split the dataset in two for training and testing while making sure the “Id” and “SalePrice” columns were dropped for the dataset.

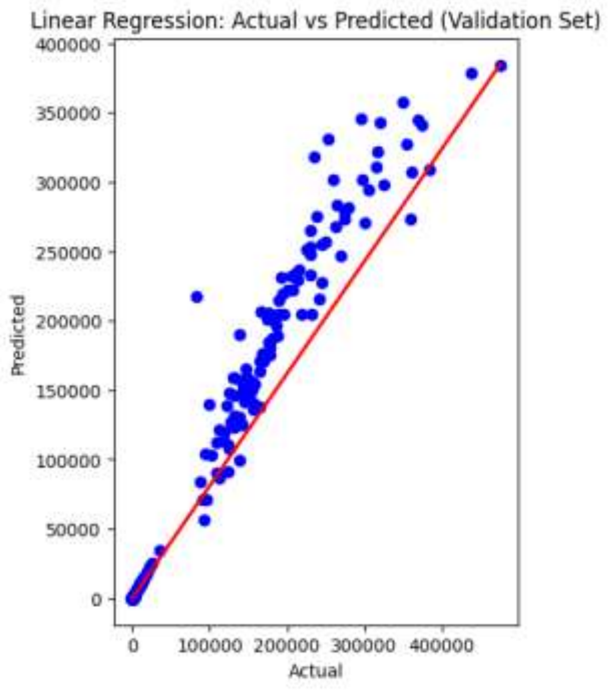


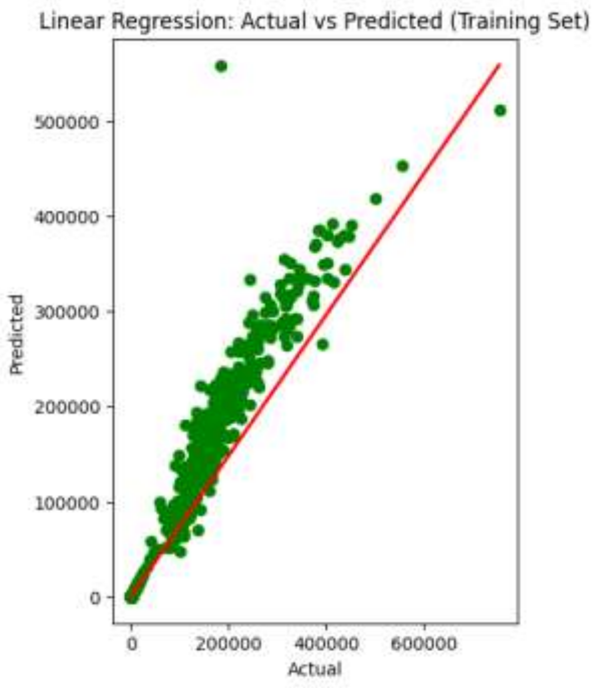

Correlation matrix of Sale Price
OverallQual
YearBuilt
YearRemodAdd
TotalBsmtSF
1stFlrSF
GrLivArea
FullBath
TotRmsAbvGrd
GarageYrBlt
GarageCars
GarageArea
SalePrice
1 0.59 0.57 0.55 0.53 0.61 0.6 0.47 0.56 0.61 0.56 0.8
0.59 1 0.63 0.42 0.31 0.2 0.5 0.12 0.82 0.53 0.47 0.52
0.57 0.63 1 0.3 0.29 0.29 0.48 0.18 0.65 0.47 0.41 0.52
0.55 0.42 0.3 1 0.91 0.49 0.37 0.32 0.36 0.49 0.55 0.62
0.53 0.31 0.29 0.91 1 0.56 0.38 0.4 0.28 0.47 0.53 0.62
0.61 0.2 0.29 0.49 0.56 1 0.62 0.83 0.24 0.5 0.49 0.71
0.6 0.5 0.48 0.37 0.38 0.62 1 0.54 0.5 0.52 0.45 0.58
0.47 0.12 0.18 0.32 0.4 0.83 0.54 1 0.17 0.43 0.39 0.56
0.56 0.82 0.65 0.36 0.28 0.24 0.5 0.17 1 0.6 0.59 0.5
0.61 0.53 0.47 0.49 0.47 0.5 0.52 0.43 0.6 1 0.84 0.65
0.56 0.47 0.41 0.55 0.53 0.49 0.45 0.39 0.59 0.84 1 0.62
0.8 0.52 0.52 0.62 0.62 0.71 0.58 0.56 0.5 0.65 0.62 1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

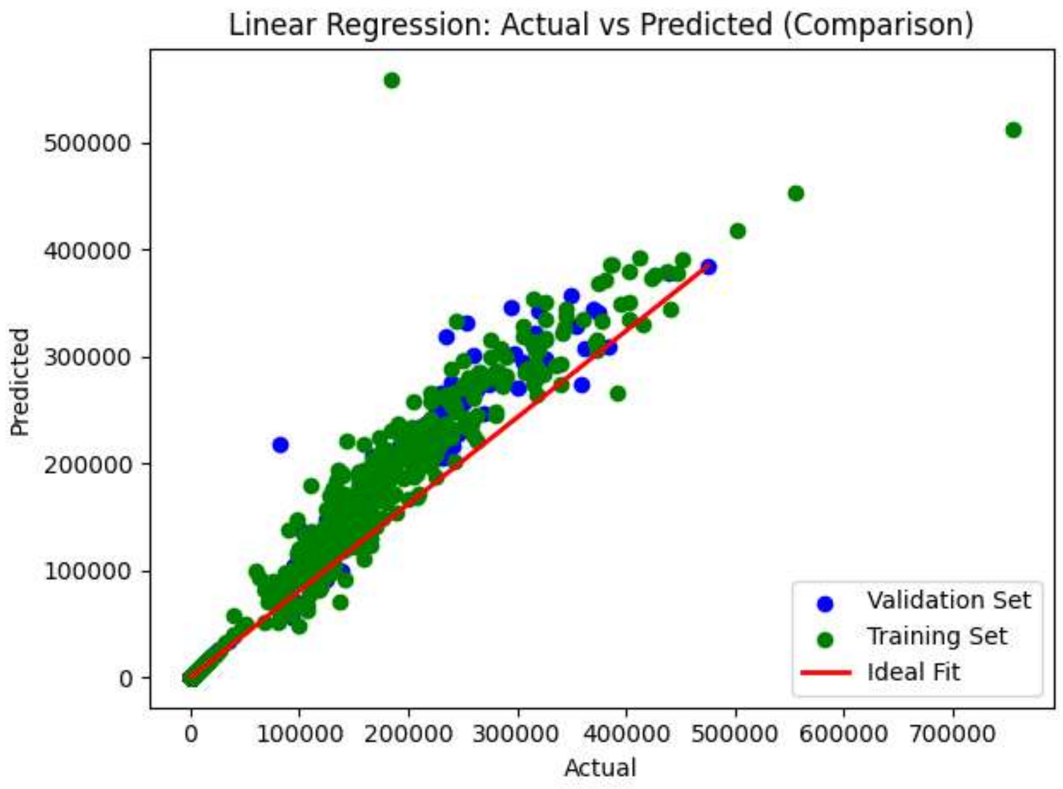

SalePrice
OverallQual
GrLivArea
GarageCars
TotalBsmtSF
FullBath

Experiment 1: Pairplot of the overall relationship of features

In terms of regression model creation, I first utilized the train_test_split function from scikit-learn to create the validation dataset. Set some default testing sizes and random states and got started on my first linear regression model. Using some advice from code submissions on the kaggle competition, I decided to implement GridSearchCV for cross-validation of the data features. With this extremely simple linear regression, I was able to reach an r2 score of 0.943 from the data. The model has a training and testing RMSE in the same ballpark as well with 5531 and 4940 respectively. I made a graph of the projected linear model as well as a comparison for clarity. To further evaluate the RMSE score of the model I tried to adjust the variables but didn’t achieve a drastically different result. I decided to use the next model to achieve a more accurate prediction.

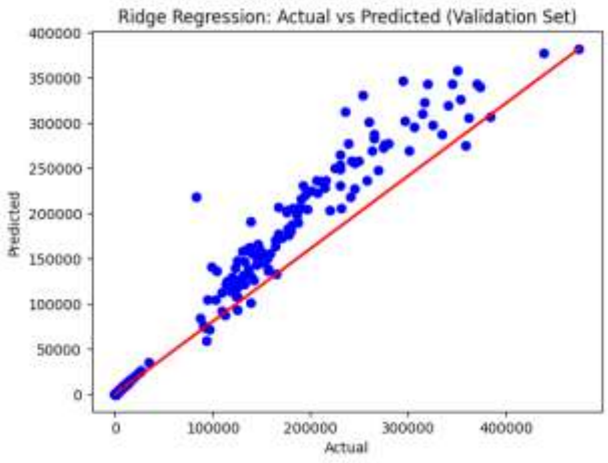


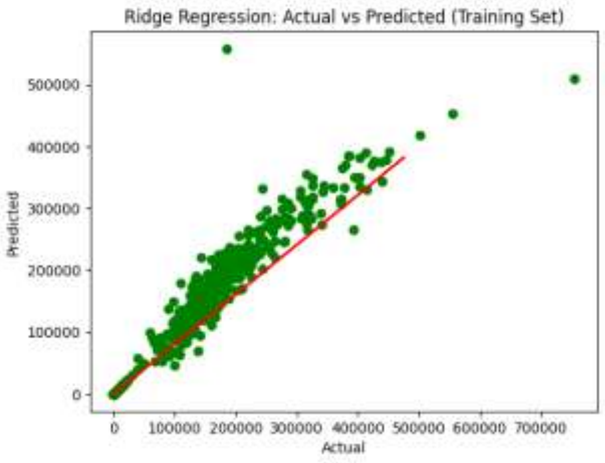




Experiment 2: Ridge Regression Model

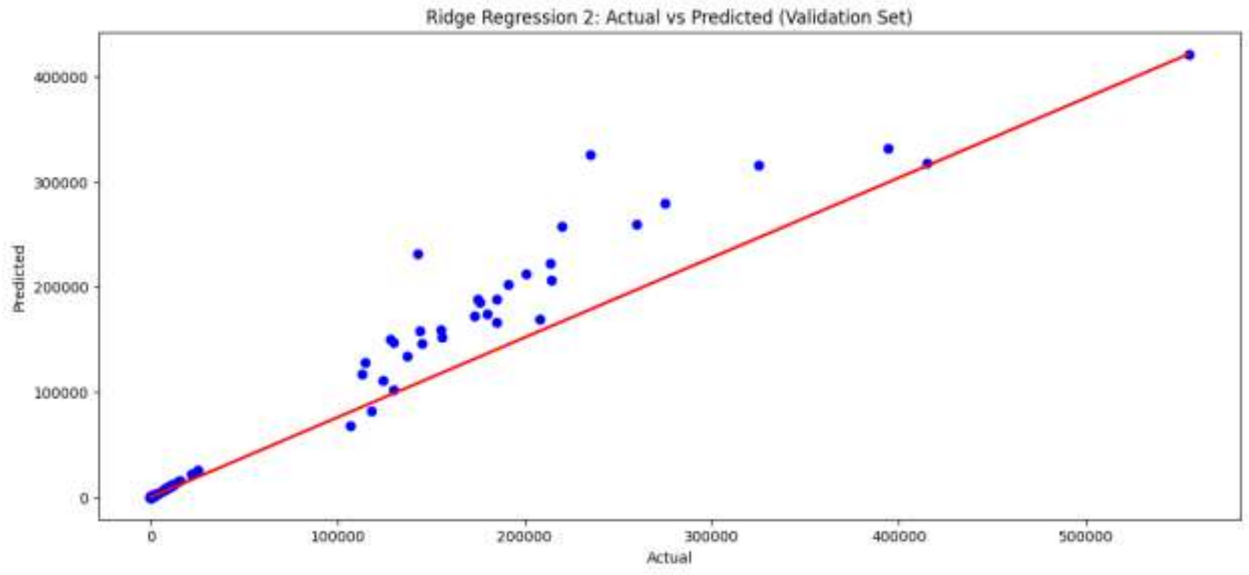
The next model I wanted to use for this data was a ridge regression model due to the type of data I was working with and already having relative success with the linear regression for sales price. I used recommendations from the competition code on how specifically to scale ridge regression data, the kinds of alpha to use, as well as the best way to visualize the model. Here I was able to show that the ridge regression model immediately showed up as having a more accurate r2 score with 0.947 from very minor tuning. The RMSE also was reasonable in this case which led to my visualizations provided. Here there isn't much to actually say other than the fact that the training set being more clustered could cause some issues with the previous models in terms of overfitting the data.

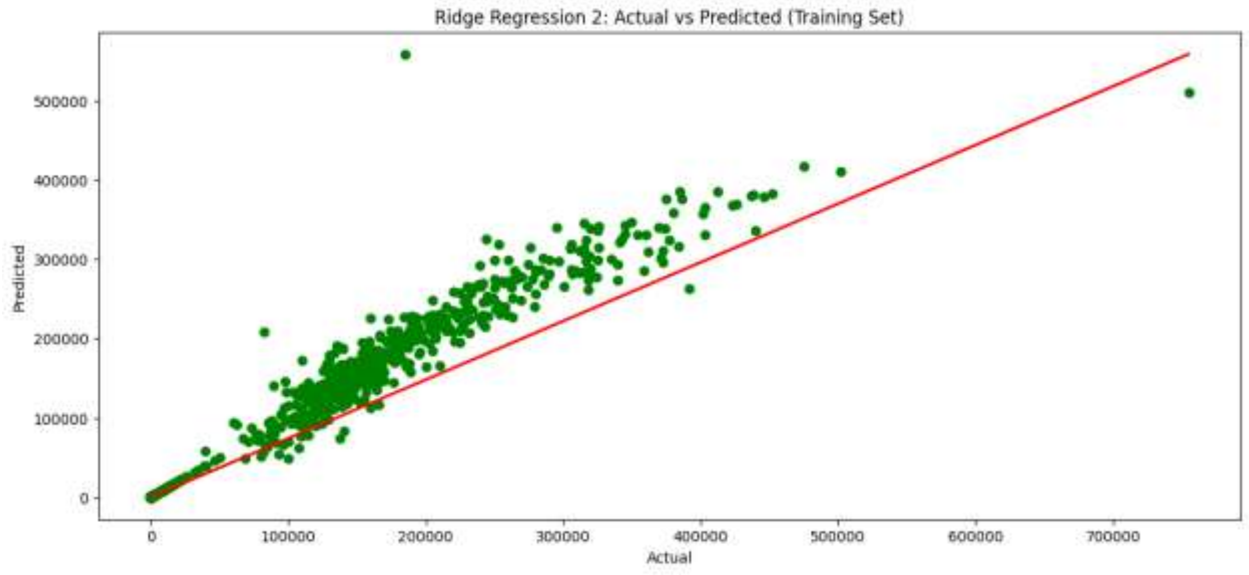


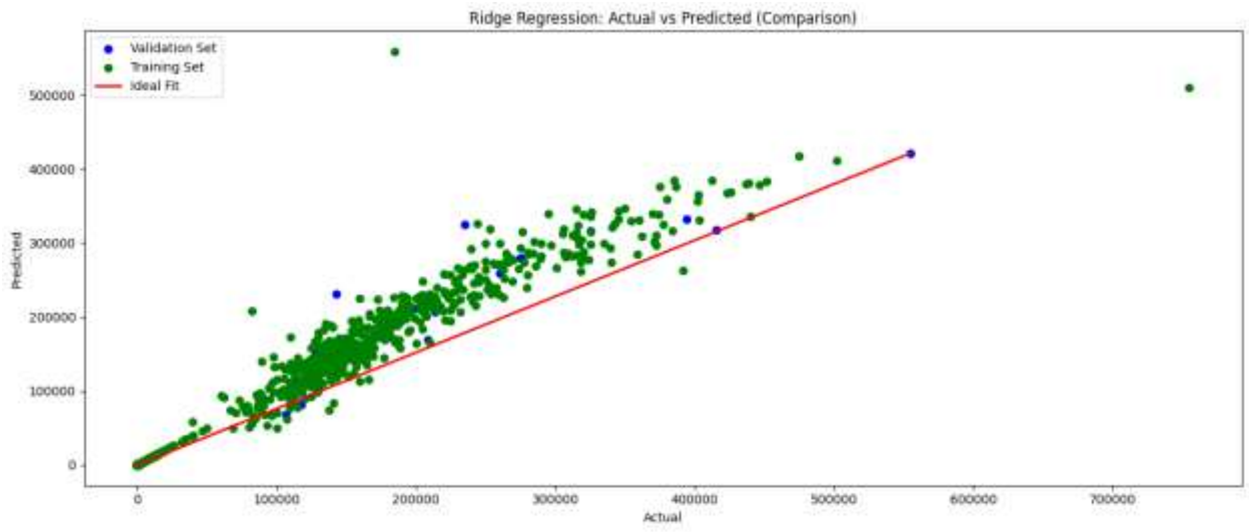


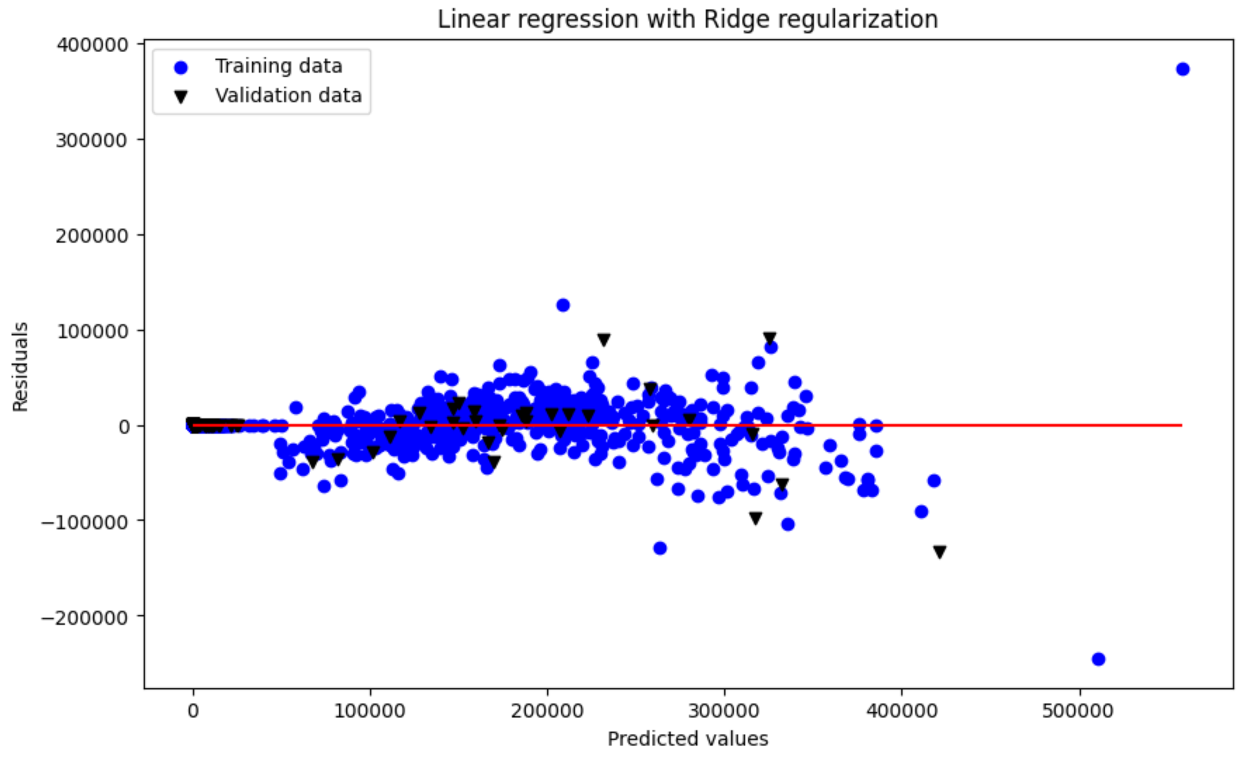
Experiment 3: Ridge Regression Model 2

For my third regression model I initially wanted to use a lasso regression model however with my coding of the model as well as the data, it was unfeasible to make the model converge as the model wasn't able to despite the iterations. Due to this, I returned to the ridge regression model with some notable alterations. While tuning this model, I wanted to show the difference between some variables as shown in the code and was able to get this iteration up to a 9.5 r2 score with limited iterations. The RMSE score in this model was the most different as well with the validation at 6710 and training at 5314. The models here show more clarity than the previous ridge regression as well. There is also a comparison plot and regularized comparison. Overall a better result than the previous experiment.









Impact:

The impact of this project can be summarized as having the potential to show different types of predicted sales prices for each of the possible numerical features included in the final dataset. By inputting those attributes manually into a model like this, you would be able to calculate a house’s sales price for your own means whether you’re selling or are trying to find a deal that is reasonable. A negative impact could be that in this case by viewing each individual metric of factors that affect housing price, the real estate market can strategically increase prices of houses easier without adding any real value.

Conclusion:

From this project I learned how to meaningfully improve on models while also preprocessing much more thoroughly than my previous projects. The advice from the kaggle competition helped in that I learned what features were important to include and keep track of for the final model. I managed to learn a lot from this project overall and tweaked a lot in terms of my final models.

References:

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/

Kaggle competition ref 1: https://www.kaggle.com/code/agleev/only-data-pre-processing

Kaggle competition ref 2: https://www.kaggle.com/code/rbyron/simple-linear-regression-models