E-Z-Houses Machine Learning Housing Model

Introduction

Problem Statement

Buying a house is no small feat. There are so many factors involved from the original listing to the time escrow is closed. Not to mention the different amount of features that a house can have. No two houses are exactly identical either . That being said,what factors contribute the most to a house price ? And can we predict future housing prices?

Background

E-Z- Houses is a real estate company, located in Ames,Iowa. They have been in business for over 20 years and are interested in leveraging technology to identify homes that they would be able to sell for the most profit. They are also extremely interested in the most important features(variables) as they relate to the sale price.

Goal

Our goal is to create and implement a machine learning model that can take previous data and tell us the future price of a home,given the various features in the data set.We will also do an in depth EDA to evaluate the relationship between the sale price and the other 79 features.

Data

The data comes from Ames Housing Study.Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.[2]

Data Cleaning and Data Wrangling

Fixing missing data

As far as cleaning the data was concerned, there was not much to do. Since it is a kaggle competition. The data came separated in test and train files. That being said, the chart below shows the total of missing features along with the percentage missing for each file.

[11]:

	total	Missing_Ratio
PoolQC	1456	99.8
MiscFeature	1408	96.5
Alley	1352	92.7
Fence	1169	80.1
FireplaceQu	730	50.0
LotFrontage	227	15.6
GarageCond	78	5.3
GarageQual	78	5.3
GarageYrBlt	78	5.3
GarageFinish	78	5.3
GarageType	76	5.2
BsmtCond	45	3.1
BsmtQual	44	3.0
BsmtExposure	44	3.0
BsmtFinType1	42	2.9
BsmtFinType2	42	2.9
MasVnrType	16	1.1
MasVnrArea	15	1.0
MSZoning	4	0.3
BsmtHalfBath	2	0.1

[10]:

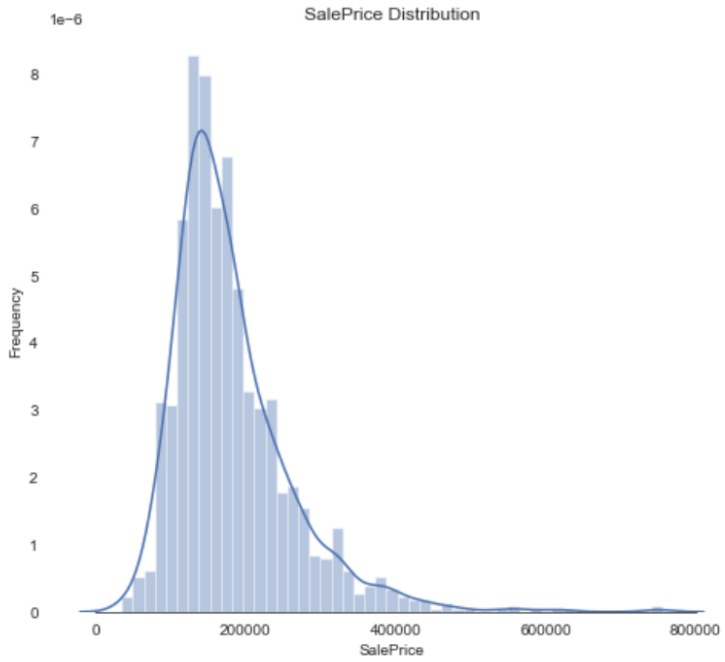
	total	Missing_Ratio
PoolQC	1451	99.7
MiscFeature	1402	96.3
Alley	1365	93.8
Fence	1176	80.8
FireplaceQu	690	47.4
LotFrontage	259	17.8
GarageCond	81	5.6
GarageType	81	5.6
GarageYrBlt	81	5.6
GarageFinish	81	5.6
GarageQual	81	5.6
BsmtExposure	38	2.6
BsmtFinType2	38	2.6
BsmtFinType1	37	2.5
BsmtCond	37	2.5
BsmtQual	37	2.5
MasVnrType	8	0.5
MasVnrArea	8	0.5
Electrical	1	0.1
RoofStyle	0	0.0

,

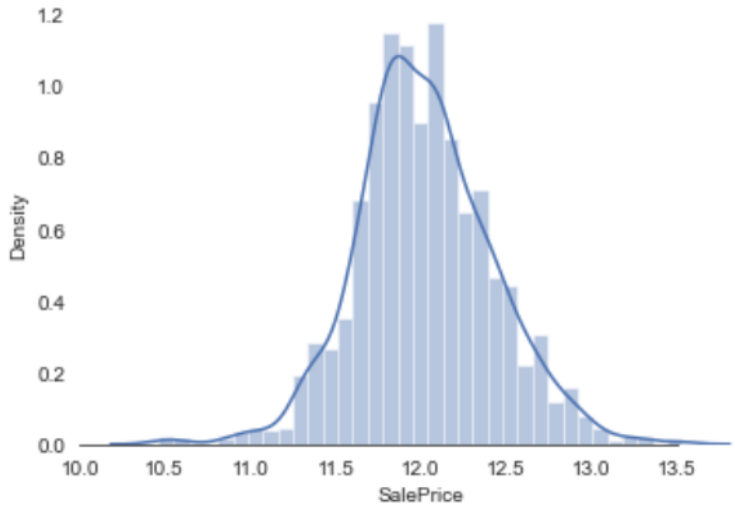
Since we are dealing with both categorical and numerical data, we will be using different forms of imputation. For example, to fill in ‘LotFrontage’ we decided to use the mean value. For other features, such as ‘SaleType’,’Exterior1st’,ect, we decided to use mode which makes sense as it is the most common. In addition,we also used a for loop to fill in categorical features with ‘none’. Some of the categorical features were also stored as numbers, in turn, we then had to convert back into strings.

Fixing numerical skewness

When we originally plotted the sale price, we could see that there was a skew of +1 with kurtosis >6.



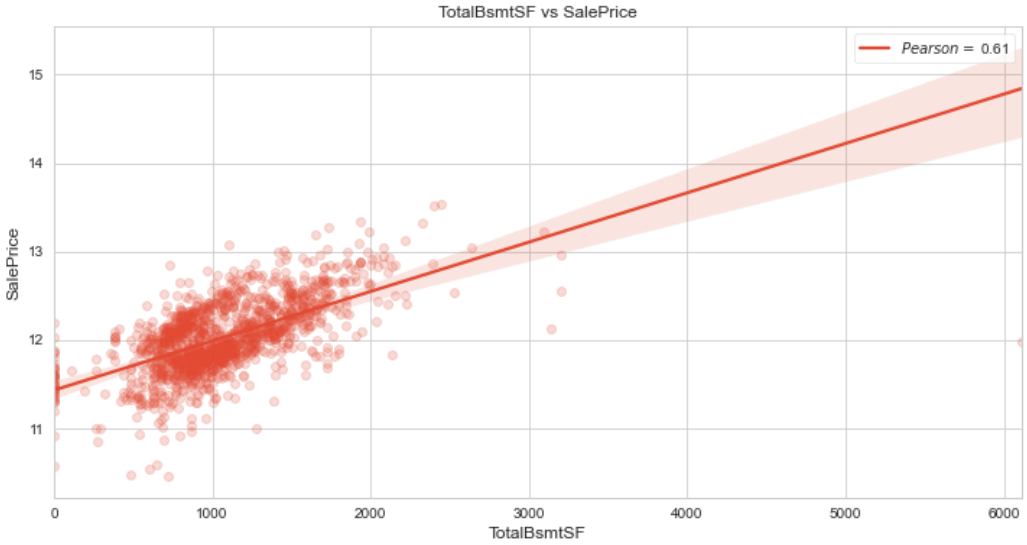
To fix this we used the log1p, which applies log(1+x) to all elements of the column. Below is the transformed SalePrice Distribution.



We then used log transformation to take care of the remaining skew on our features.

Removing Outlier

SalePrice v TotalBsmntSF



The outliers have been removed, but we can see how most homes are between 2000-5000 sq ft.

Fixing categorical data:

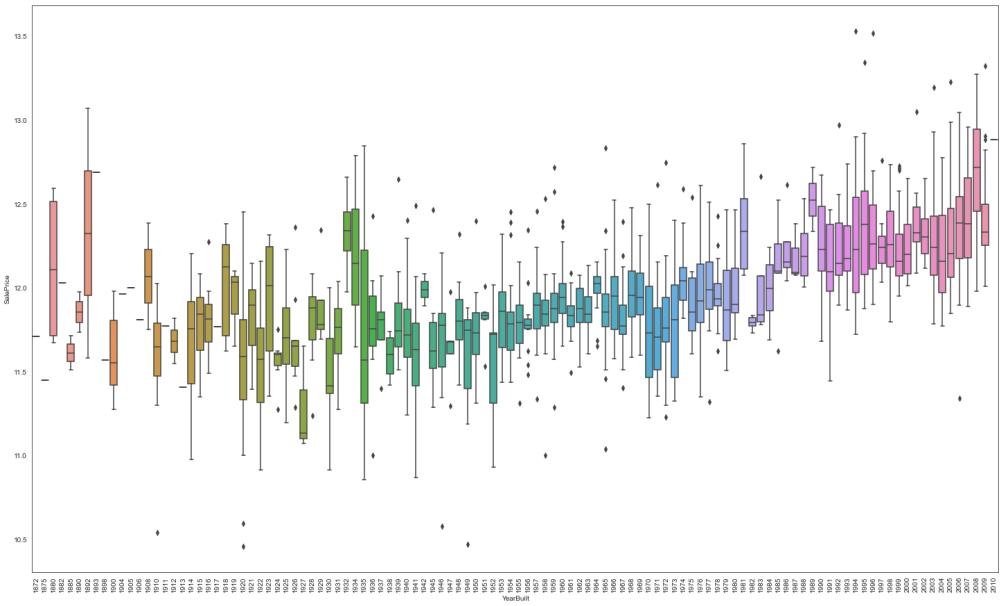
When it comes to categorical encoding, there are a couple methods that can be used, such as one hot encoding,get_dummies, and label encoding. Because we are going to be using regression, we would not want to increase the likelihood of multicollinearity, therefore we will be going with pd.get_dummies. We also want to reduce redundancy, which is the reason why we're using get_dummies vs label encoding.

Adding a feature

We added the total SQ feature in addition, so that way we can have a much clearer example of the total square footage of the house and how it relates to sale price.

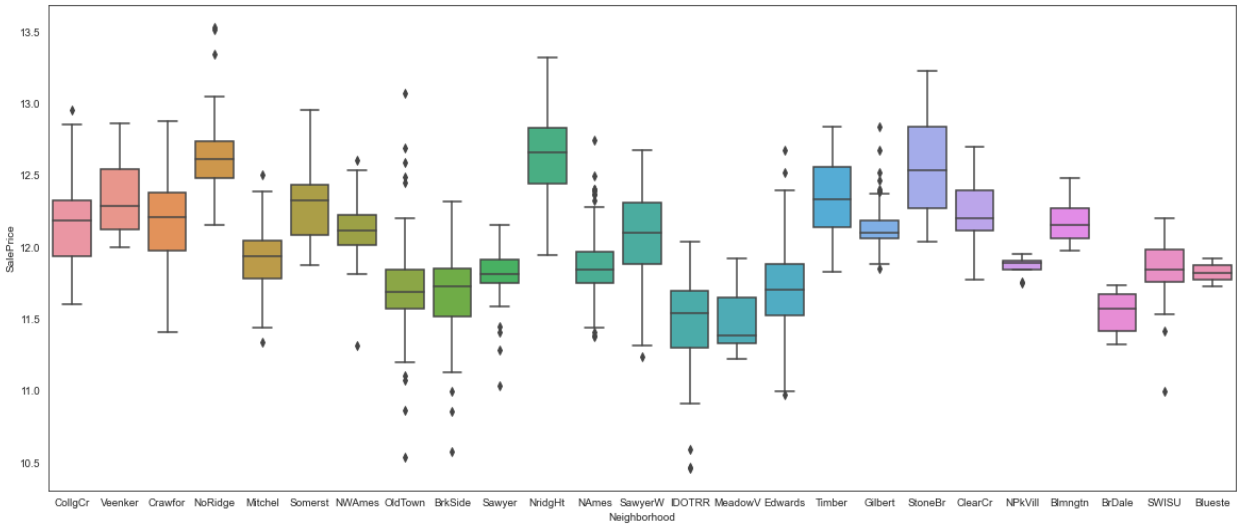
EDA & initial insights

SalePrice v YearBuilt



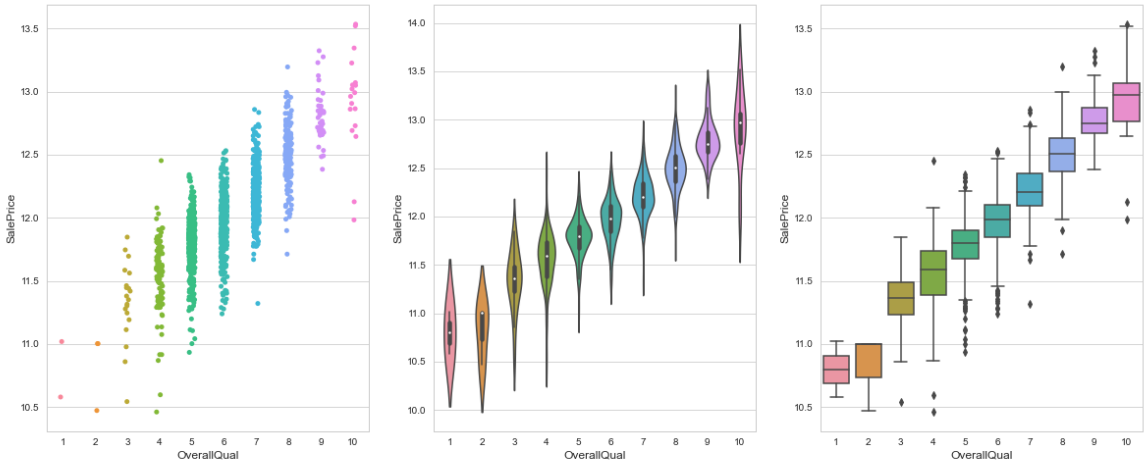
The first relationship that we explored was SalePrice vs YearBuilt. What was interesting was that there were more outliers than expected in the data. Also, as the years increased, the median price for homes increased as well. In addition, we would be remiss to add that inflation most likely had something to do with increased housing prices but does not answer the entire question.

SalePrice v Neighborhoods



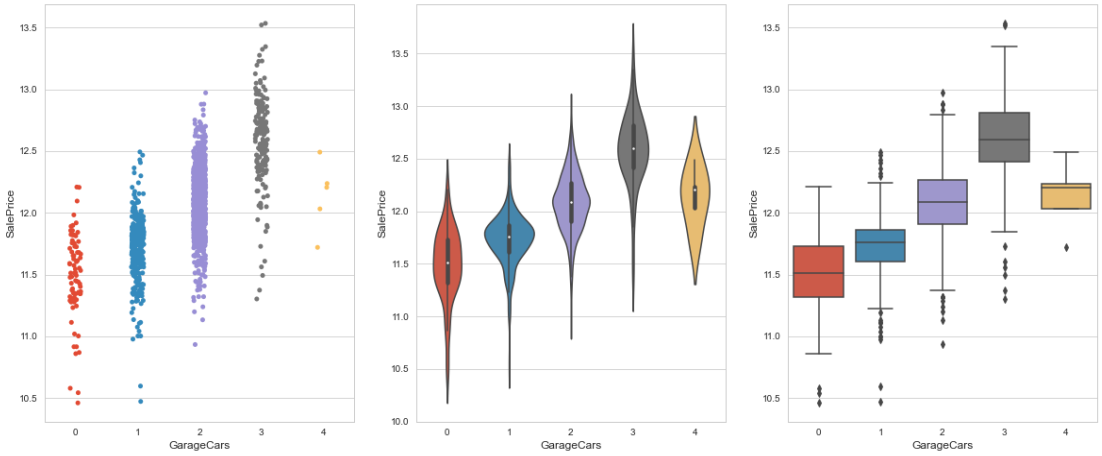
We can see from the graph above that there are some outliers, especially in old town. In addition, NAmes,CollgCr,OldTown,Edwards have the highest number of houses.That being said, we can see how some communities are more expensive, whereas other communities have a much wider range of prices. Also,StoneBr,NoRidge and NridgHT are the most expensive neighborhoods.

SalePrice v OverallQuall



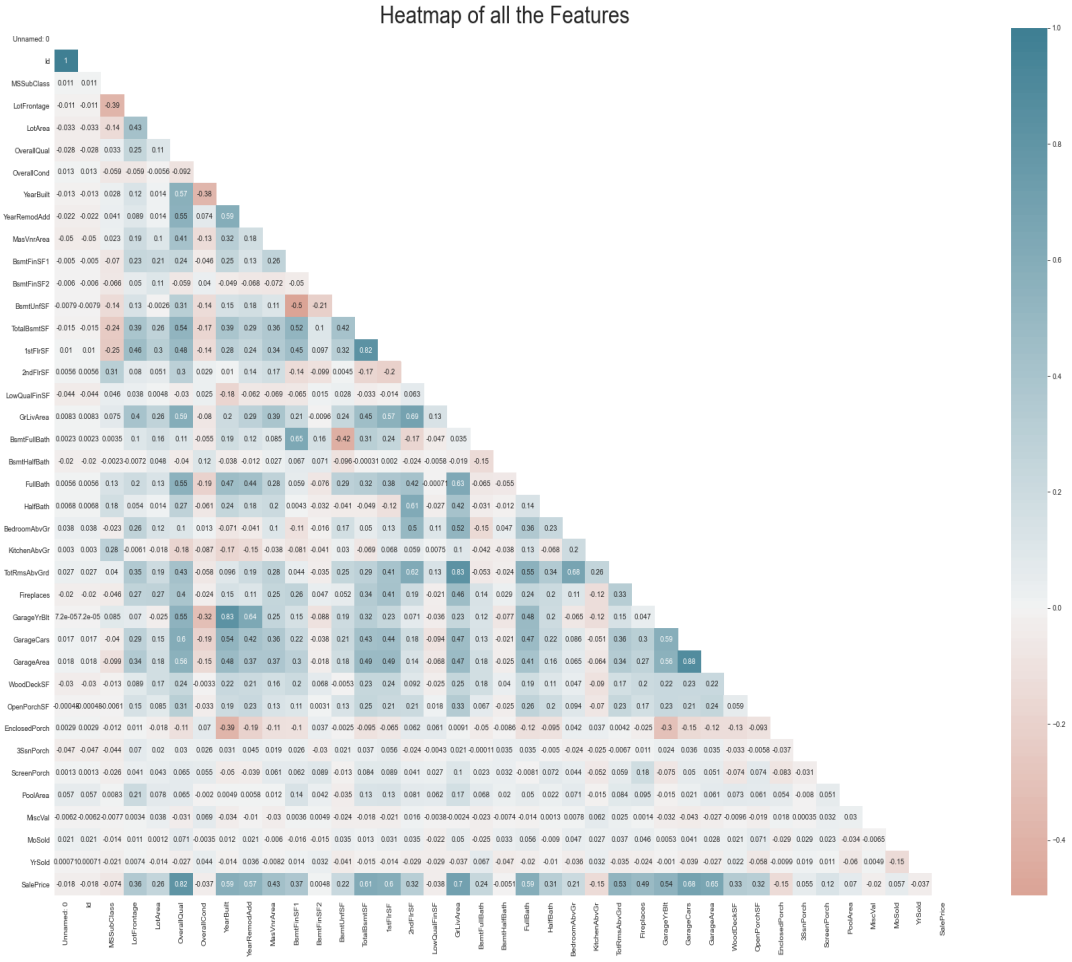
There is a positive correlation between the two features. The price of the houses increase with the overall quality. From violinplot, the #4 and #10 qualities showed the most range as far as the sale price is concerned.

SalePrice v GarageCars



There is also a positive correlation here as well, with the exception of a few outliers. It does make sense that the price increases with the increased square footage to house cars. An interesting find is that homes with 4 car garages are all priced at or greater than the mean sale price. On the other side, homes with 1 or less GarageCars are less than the mean sale price.

Heatmap



This heatmap shows us the correlation between all the features in the data set. The highest correlations found were : 82% between the 1stflr and TotalBsmtSf,83% between YearBuilt and GarageYrBlt,88% between GarageCars and GarageArea
That being said, we have removed GarageArea,TotRmsABVGr,1stFlrSF in order to reduce collinearity that we detected when we first ran linear models.

There was a lot of useful information gathered in this part of the project. One of those being the positive correlation between SalePrice and GarageCar. What was interesting is that the 3 car garage homes demanded the highest sale price vs 4 car garage. One factor to help explain that is the neighborhood is priced differently, which would lead to the variability in price.

From our correlation chart in the EDA, the values with the highest correlation to Saleprice are OverallQual,GrLivArea,GarageCars,GarageArea,TotalBsmtSF. Because quality is hard to measure and often based on subjection rather than objection, is not as reliable a statistic that we would like. That being said, the main takeaway from the correlating features is that square footage and yearbuilt, in general, have the biggest impact on housing price which comes as no surprise.

Another discovery was the most expensive homes in the area were not in the same communities that have the most homes. Homes made in the 1930's had the greatest price fluctuations. The communities by Saleprice graphic is a great starting point for our clients real estate agent team. They can use the graphic to help guide their clients based on what price point they can afford.

Modeling

The goal is to predict the housing price. Since we will be using supervised learning methods, we took the data and performed a test_train_split(25%test,75%train). The target variable is the 'SalePrice'.

Next, we used the standard scaler to fit and transform X_train while just transforming X_test. Then we proceeded to use Linear Regression as our first model because the simplest methods often work best and are a great starting point. That being said, we came uncross something troubling. The R2 accuracy was negative by a lot, which means that multicollinearity must be present in the data. Further investigation and second opinions did conclude that there is an element of multicollinearity present. In addition, the Variance Inflation Factor, which means the greater the value of R-squared, greater is the VIF. Hence, greater VIF denotes greater correlation. This is in agreement with the fact that a higher R-squared value denotes a stronger collinearity. Generally, a VIF above 5 indicates a high multicollinearity.[6]

We also tried using Linear Regression via OLS in stats models. However, the outcome did not change. It has to be something to do with the features and how they interact with the model. The biggest problem is the number of how many features that are involved

that can be creating the error. Good news is that our other models were not affected by this oddity.

That being said, we wanted to push on and try other models that enforce penalties such as Lasso, Ridge and Elastic Net regression. The difference between ridge and lasso regression is that it tends to make coefficients to absolute zero as compared to Ridge which never sets the value of coefficient to absolute zero.[5] Sometimes, the lasso regression can cause a small bias in the model where the prediction is too dependent upon a particular variable. In these cases, Elastic Net is proved to better combine the regularization of both lasso and Ridge. The advantage of that it does not easily eliminate the high collinearity coefficient.[5]

The next model that we used was Ridge regression. To get the best alpha of 1, we used GridSearchCV. After that, we fit, transformed and predicted the model. The scores are displayed below.

```
---------- Ridge ----------
R square Accuracy:  0.8867355562906313
Mean Absolute Error Accuracy:  0.08857800403658181
Root Mean Square Error test = 0.13268582623755076
```

```
---------- Lasso ----------
R square Accuracy:  0.9084294674746836
Mean Absolute Error Accuracy:  0.08084319662299135
Root Mean Square Error = 0.11930416483696503
```

```
---------- ElasticNet ----------
R square Accuracy:  0.9060302424411506
Mean Absolute Error Accuracy:  0.08403557362565785
Root Mean Square Error  = 0.12085699401753719
```

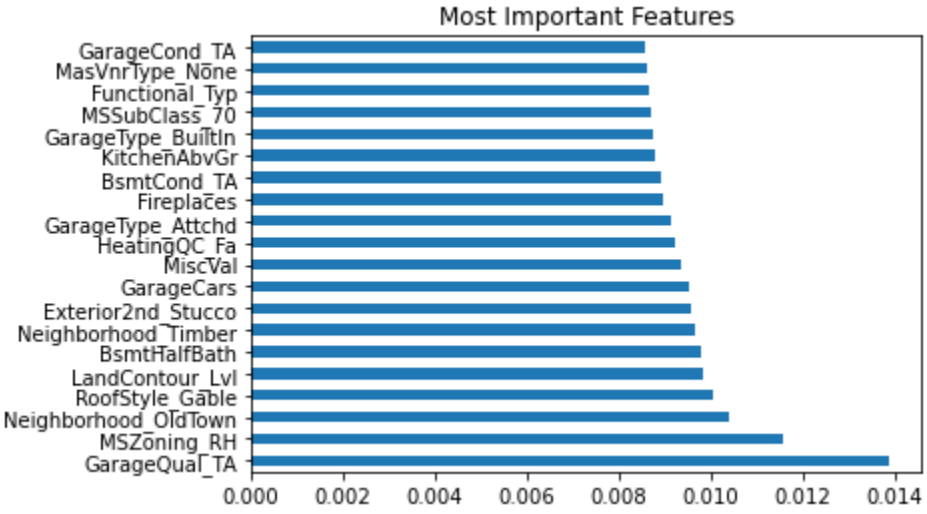
Then used lgb and gradient boosting regressors to see how decision trees would compare to using regression models. The reason why we chose LGBM was because of how fast the model would compute as well as its emphasis on accuracy.

```
----------LGBM----------
R square Accuracy:  0.9025395460718486
Mean Absolute Error Accuracy:  0.08708156623539755
Root Mean Square Error = 0.1230812646354938
```

```
----------GBR----------
R square Accuracy:  0.9125990996786175
Mean Absolute Error Accuracy:  0.07992502492513379
Root Mean Squared Error Accuracy:  0.116556282705668
```

As we can see, Gradient Boosting tested the highest with R-squared accuracy and having the lowest RMSE.

In addition, we also decided to create a feature importance chart to visualize how and what the most important features are that affect the sale price.



As we can see, the most important features are at the bottom, starting with Garage_quality. From the chart we can see that : GarageQual_TA,MSZoning_RH,Neighborhood,Roofstyle,Landcontour are the most important features with regards to the model.

Takeaways

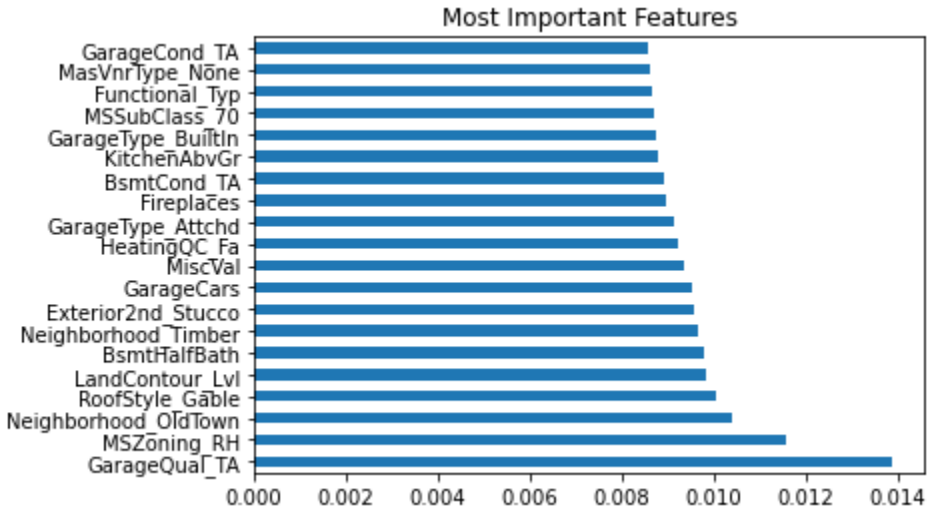
Our clients sales and marketing department can now use the communities by saleprice graphic as a key starting point with their clients to help setup expectations as to what they will be looking towards as far as saleprice is concerned relative to the neighborhood.

The size of the garage will become a factor. The most popular size garage is 1 car however, as far as price is concerned the best deal would be deciding a 1 car garage because it is lower than the mean saleprice. Or if there's a lot of cars/garage space that the client needs, then suggesting a 4 car garage home would be ideal because it is less than the price of 3 car garages.

Most people live within 6 or 7 communities in ames out of the 25 neighborhoods. That being said, using this data for the communities with the most people would be the most efficient way to utilize this data so that way it is more general.

The biggest takeaway our client has is now the ability to plug in a home and our predictive model will be able to predict the price of that home in the future with an RMSE <.11.

The best fit was using Gradient Boosting, which is great because of how fast and efficient it is. In addition, the most important features that affect a houses sale price, with regards to the model, are Garage(garage quality,size),Zoning(high density residential),Neighborhood,Roofstyle(Gable-which is best for cold weather, which is where Iowa is located),LandCountour(built on level ground).



Future Work

Would be interested in doing some deeper searching. Such as finding which neighborhoods have more successful homeowners vs others, but that would require getting Income and demographics data.

Another feature that would have been nice to add would have been time. Such as the change in price of these homes over time, the only information about the homes we have is what they last sold for but not every sale.

References

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

2. https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/
3. https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d
4. https://www.geeksforgeeks.org/lasso-vs-ridge-vs-elastic-net-ml/
5. https://www.geeksforgeeks.org/detecting-multicollinearity-with-vif-python/

https://github.com/astephens10/Capstone_2