Linear Regression models on Housing Data

Main Objective

- In this report, we will analyze the dataset for the houses that were sold in Ames, Iowa between 2006 and 2010.
- The objective of this report is to find out which aspects of a housing property are important in determining its selling price.
- The primary purpose of this regression-based report will be based on the interpretation of the various features in the dataset.
- In doing so we will use a variety of linear regression models and figure out the one that is best suited to our needs.

Brief Description of the Dataset

This dataset has 79 features that describe the various aspects of a home and one target column that is the Sale Price of the house. There are 1379 observations or data points which constitute the rows corresponding to each of the columns of our dataset.

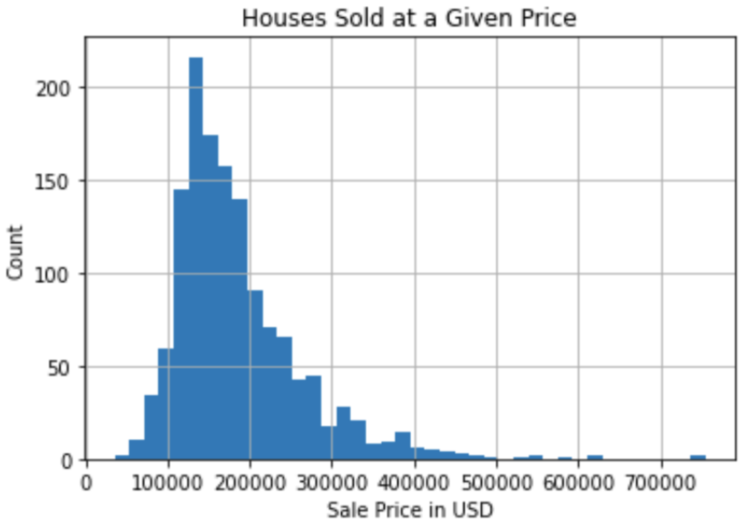
Since there are too many features to be described in detail, we will give an overview of some of the more important ones in our dataset below:

- Land area in square feet of the various floors in the house
- Land area of specific parts of the house like a basement, garage etc.
- Presence of a porch, fireplace, etc.
- Whether the unit has centralized cooling/heating
- The overall condition of the house and the condition of specific parts like bathrooms and kitchen.
- The style and shape of the house.
- The neighborhood in which the estate is located.
- The year in which the house was built.
- The year in which it was sold etc.

Overall, there are 43 categorical features and 36 numerical features in our dataset. Of the numerical ones, 20 are float type and 16 are int type. The target column, Sale Price, is of float type.

Data Cleaning

- There are no empty cells in our dataset, so we move straight to looking at the outliers.
- We start by plotting a histogram of the Sale Price column to find any outliers.



- There are 9 units sold for prices greater than $500,000 and are outliers in our dataset. To ensure our regression interpretation is accurate for most of our dataset, we shall drop the rows corresponding to these outlier values.

- Similarly, there are some outliers that were sold under $50,000. We remove them from our dataset as well.

- We also look at some other prominent features like the surface area of the first floor, the overall lot area, etc., and get rid of the outliers to improve the performance of our regression models.

Feature Engineering

First, we try to convert the various float features into normal distributions:

- We start by looking at the skew values of the various float-type columns in our dataset. We find that out of the 21 float-type columns in our data (20 features + 1 target), 19 of them have a skew value greater than 0.75 in magnitude.

- We log transform these features using the np.log1p() method. This reduces the skew value of 12 columns such that they are under 0.75.

- Looking at the remaining 9 columns, we find that their skewness arises from many cells being set equal to 0 and the remaining cells having a non-zero value. For e.g., the column for the area of the 3-season porch area is equal to zero for 1355 rows and is non-zero for only 24 rows. Such features cannot be converted into a normal distribution, and we have two choices on how to move forward with these features.

- First, we could convert such features into a Boolean type so that we only indicate the presence/absence of such features and forego the data about the size of such features.

- Alternatively, we continue using these features in their present form since there is some value to the added data that comes with the specific float value they carry, instead of just a Boolean indicating their presence/absence.

- We shall use the second approach and proceed by keeping their features in their present form since it is possible that the presence of a large porch can significantly affect the pricing of a house compared to a small one. Thus, we wish to preserve this information.

Now we move on to the categorical columns.

- There are 43 categorical columns and we one-hot encode them to convert them into numerical values which can be used by our regression models.

- In doing so, we remove the original column with string characters and replace it with the one-hot encoded columns.

- We also drop the first one-hot encoded column corresponding to each of the categorical to prevent highly correlated features from affecting our models.

After one-hot encoding, we have a total of 250 features. With these two steps of normalizing our float columns and one-hot encoding the categorical columns, we conclude feature engineering of this dataset and move on to building our regression models.

Linear Regression Models

In this section, we will use four models to first train our model on the dataset and then test it on another part of the dataset. In all the models, we will first use the standard scaler method that expresses all the x values in a column by transforming them as (x - μ)/σ, where μ is the mean of the column and σ is the standard deviation. Moreover, we use the KFold() method for each of the linear regression models, with the shuffle value set to 'True', the number of splits in our train and test data to be equal to 3 and having a constant random state value for each of the four models so that we can compare them.

1. For the baseline linear regression model, we get an R2 score of -1.23e24. This is a ridiculously large number showing that our model is not working well. This is happening because our model is overfitting the training data and the ability of our model to predict the Sale Price corresponding to the test data is very poor. This is seen in the fact that many coefficients have a large value with a total of 105 coefficients having a value greater than 0.01.

2. We then move on to adding the PolynomialFeatures method that increases the degree of our features. We set the degree of this method equal to 2. This method is called after the standard scaler in our pipeline. In this case, we get an R2 score of 0.68 which is a great improvement over the baseline linear regression method. In this case, we have a total of 31878 features. Out of this large number of features, only 4 coefficients have a value greater than 0.01. Thus, the PolynomialFeatures helps in alleviating the problem of overfitting.

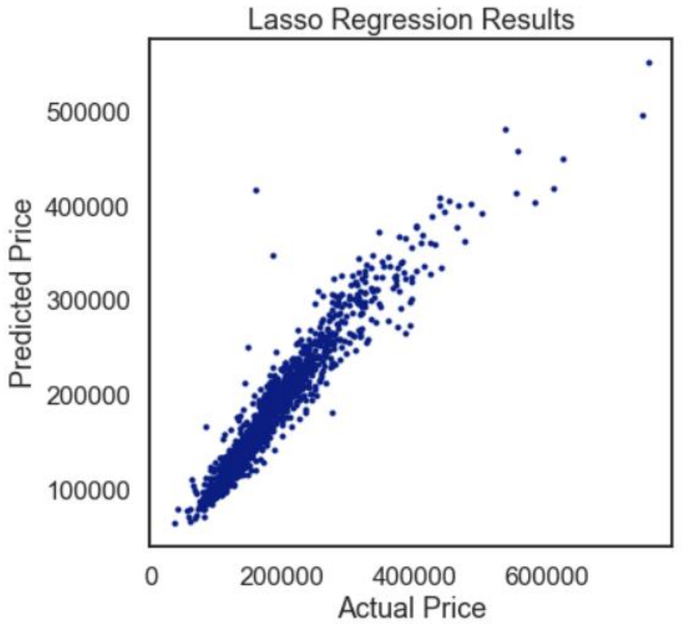
3. We now remove PolynomialFeatures from our pipeline and replace the ordinary linear regression with Lasso regression. Since Lasso regression depends on a hyperparameter called alpha, we use a few alpha values from 1e-4 to 1e-2 to find the best alpha corresponding to the best fit. We get our best fit for alpha equal to 0.0078. Corresponding to this alpha value, we get an R2 score equal to 0.9. Moreover, the problem of overfitting is significantly overcome because only 20 coefficients are greater than 0.01 in magnitude.

4. We then replace Lasso regression with Ridge regression. Once again we use a few alpha values from 1e-5 to 1e5 to find the best alpha fit. We get our best fit for alpha equal to 600. Corresponding to this alpha value, we get an R2 score equal to 0.91. Moreover, the problem of overfitting is significantly overcome because only 20 coefficients are greater than 0.01 in magnitude. Finally, a total of 38 coefficients are greater than 0.01 in magnitude.

The linear regression model that bets suits our needs is Lasso regression. Even though it has an R2 score that is slightly smaller than that of Ridge regression, 0.9 for Lasso compared to 0.91 for Ridge. This is because Lasso sets a total of 197 coefficients equal to zero while there are no coefficients set equal to zero for Ridge. Moreover, only 20 coefficients have a value greater than 0.01 compared to 38 for Ridge. Thus, Lasso will allow us to better interpret our results and to understand which features affect the Sale Price of our house the strongest.

Key Findings

Below we look at the predicted results from Lasso regression to see that it is a good predictor for the Sale Price of houses of Ames, Iowa.



We sort the coefficients obtained from Lasso regression in descending order to find the most important features. We find that some of the most important features that positively affect the Sale Price of an estate are:

- Ground Living Area
- Overall Quality
- Year Built
- Lot Area

Similarly, we look at which coefficients have a negative value to find which features negatively affect the Sale Price of a house. The three features that negatively affect the Sale Price of an estate in descending order of importance are:

- An above-ground kitchen
- Whether the estate is close to an off-site feature like a park
- Whether the house is in the Edwards neighborhood

Next Steps

As a possibility, we can use polynomial features along with Lasso regression and try to find any improvements in our model. While this wouldn't be too useful in improving the interpretability of our dataset, it will help in increasing our model's predictability.

A possible problem with our model might be that some features that have few non-zero values and this might offset our models. As discussed in the feature engineering section, this could be corrected by converting these columns to a Boolean 1 or 0 value informing us whether those features are present in the house or not.

Some additional data would also be helpful in improving our models. Data regarding crime in the neighborhood would one such feature.