

|  |
| --- |
| House Prices - Advanced Regression Techniques |
|  |
| December 1917  Machine Learning for Business Analytics  Authored by: Shivam Bhat Team Members: Aryan Sharma & Himanshu Malik |



# Business Background

# A house's value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. A house buyer will want to know if the house price matches the house value and if they are paying a fair price. A house seller will want to take advantage of the features that influence a house price the most, thus investing in making rooms at a small cost to get a large return.

# Problem Statement

# Our goal is to predict sale prices for homes in Ames, Iowa. We’re given a training and testing data set in CSV format. For each Id in the test set, we must predict the value of the Sale Price variable. We are going to take advantage of all of the feature variables available to us and use it to analyze and predict house prices.

# Our Approach

Machine Learning

# 

Feature Engineering

Data Cleaning

Data Ingestion & Exploration

# Data Ingestion & Exploration

In our housing dataset, the only difference in features between test and training data is “Sale Price”, the variable we are trying to predict.  The training set has 1460 rows and 81 features. The test set has 1459 rows and 80 features. There are two types of features in housing data, categorical and numerical.

By plotting the Sale Price histogram, we can see that the distribution of sale prices is right skewed. This is expected as it is fairly common to see a few houses in a neighborhood that are relatively expensive.

Chart, histogram

Description automatically generated Chart, bubble chart

Description automatically generated

Figure 1: Histogram plot for SalePrice Figure 2: Correlation Plot for Numeric Variables

To get an idea of relationships that exist in our housing data, we plot the correlation plot of the top 10 features that are most related to Sale Price. We see that OveralQual, GrVilArea, and GarageCars are at the top of the list.

# Data Cleaning

The first thing we want to check in our dataset is the count of missing values because missing data can imply a reduction in sample size.

We divided the features into two parts. The first part consists of features where the imputation of individual features cannot be predicted by using any other features in the dataset. The second part consists of features in which missing values in one feature can be predicted using the values present in other feature variables. For direct imputation, we have selected the variables Fence, Alley, MiscFeature, Utilities, Functional, Exterior1st, and Exterior2nd. We visualized the frequency distribution of these variables to get an idea of what value can be used to directly impute these variables.

Chart, bar chart

Description automatically generatedChart

Description automatically generatedChart, histogram

Description automatically generatedChart

Description automatically generated

Graphical user interface, table

Description automatically generatedGraphical user interface

Description automatically generated

Figure 3: Frequency distribution of features for direct imputation

From the frequency distribution graph, we have concluded that we will be replacing the NAs of Utilities with “AllPub”, Functional with "Typ", Exterior1st, and Exterior2nd with “VinylSd”. Here, replacement is done by using the maximum value that appeared for that variable. NAs for Fence, Alley, and MiscFeature have been replaced with None by looking at the description of these variables from data\_description.

After this imputation, we started with the second type of imputation where variables depend on other variables

1. Pool variables: We have assumed that the Pool quality can be determined by Pool area. We also concluded that where the Pool area is 0 means that there is no pool which can be attributed to pool quality as None. After applying this logic there are only 3 values that are left as NA for pool quality. We have imputed the missing PoolQC values with corresponding values of other houses having means of PoolAreas near to those of the missing PoolQC values.
2. Fireplace Variables: If a fireplace has a value equal to 0 means there is no fireplace. We can conclude that if there is no fireplace then the fireplace quality should be none corresponding to those values.
3. Lot Frontage Variables: Lot frontage variable is linked with the neighborhood. We have applied the logic that the house within the same neighborhood tends to have similar lot frontage. We have replaced the missing Lot Frontage value with the median of the lot frontage in the same neighborhood.
4. Garage Variables: We have changed the NA values of GarageType, GarageFinish, GarageQual, and GarageCond to None as here NA indicates that there is no garage. We also replaced the NA value of GarageYrBlt with the YearBuilt of the house. For numeric values GarageArea and GarageCars, we have replaced NA with 0.
5. Basement Variables: We have 11 variables for the basement. First, we observed that BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, and BsmtFinType2 have the common 79 NA values. The additional NA values for each column are replaced by using the mode for each of these variables. For the variables where NA means no basement (from data\_description) the NAs are replaced by None and for the integer variables the NA is replaced by 0.
6. Masonry Variables: We are changing all the NA values of MasVnrType to none where there is no area given. All the NA values of MasVnrArea have been changed to 0. We have only one value (house 2611 in the combined set) where the area is given but the type is missing. It has been replaced by the mode value of MasVnrType i.e. “BrkFace”.
7. MS Zoning and MS Subclass variables: The values of MS Zoning are linked with the MS Subclass. To visualize a distribution of house type vs zoning classification is created. From the histogram, we can see that house types 70 and 30 have zoning classifications of RL and RM. Type 20 is of zoning RL. The NA values are imputed accordingly.

Chart, bar chart

Description automatically generatedChart

Description automatically generated

Figure 4: Distribution of MSSubClass with Zoning Classification & SaleType with SaleCondition

1. Kitchen Variables: the NA values are replaced with the most common value i.e., “TA”.

9. Electrical System Variable: NA values are replaced with the mode value of the Electrical variable.

10. Sale Variables: Linked the variable sale type with the sale condition. The NA value in SaleType has normal sale conditions. The majority of normal sale condition is of type WD. So, the value in SaleType is replaced with “WD”.

# Feature Engineering

* We’ve applied a log transform to the sales price to compress outliers making the distribution normal. This is necessary because Outliers can have devastating effects on models that use loss functions minimizing squared error.
* We’ve also changed the character features into dummy variables.

# Model Selection, training, and fitting

We separated our combined data back into train and test data. We further split the train data into data\_train(80%) and data\_test(20%) so that we can predict the outcome (SalePrice) and calculate the accuracy by comparing the outcome of data\_test with the actual values of SalePrice in data\_test. Our performance metric will be the root mean square error (RMSE).

Our approach to selecting the models is based on the model's ability to deal with high-dimensionality datasets.

Linear Model: We started with the simplest linear model and applied the model on data\_train. The predictions were made for the values of data\_test using this trained model. To determine the accuracy of our model we calculated the rmse value for the data\_test set and it comes out to be 0.172515.

Next, we decided to use the Shrinkage Model such as Ridge and Lasso because we are dealing with a high dimensionality dataset.

Ridge Regression: We used the cv.glmnet function for cross-validation to choose the best lambda. The best lambda value with the smallest cross-validation error is 0.1913946. The RMSE value we obtained on the testing set using the best lambda is 0.9100479.

Lasso Regression: The matrices used were the same as used above in Ridge regression. We performed the cross-validation to get the best lambda. The value of the best lambda resulting in the smallest cross-validation error is 0.007205882. The RMSE we obtained on the testing set using the best lambda is 0.1024276.

Histogram

Description automatically generatedChart, histogram

Description automatically generated

Figure 6: Lambda plot for Ridge and Lasso Regression

GBM: We have used GBM as there is a high variance between the variables and it’s difficult for any single decision tree to fit all the training set variables. We have used 10-fold cross-validation which is repeated 5 times. The RMSE we obtained from this model on the test set comes out to be 0.1146263.

Ensemble Model: In this method we have used the weighted average method from all the models to get the best accuracy. Initially, we started by giving the accuracy based on RMSE obtained for all the models. We choose the Ridge to have 60% weightage, Lasso to have 20% weightage, and GBM to have 20% weightage. Using this method to obtain the RMSE on the test set resulted in a 0.13214 value. Using different combinations of weighted averages, the best RMSE on the test comes out from 20% weightage to Ridge, 20% weightage to Lasso, and 60% weightage to GBM. This combination gave us the best accuracy on the test set in the competition with an RMSE value of 0.12884.