12/20/2022

Machine Learning

House Prices - Advanced Regression Techniques

Author: Himanshu Malik

Team Members: Aryan Sharma, Shivam Bhat

Logo, company name

Description automatically generated

**Problem Statement:**

Our goal is to predict sale prices for homes in Ames, Iowa on the given dataset. The dataset consists of 1460 observations for training set and 1459 observations for test set. The dataset consists of 79 variables (excluding Id and sale price). For each row in test dataset, we are going to predict the sale price of that home.

**Business Use Case:**

Buying a house depends on a lot of factors other than location, number of floors, bathroom, and area. Predicting the right price for a house makes life a lot easier for any Company that deals with Real Estate.

* Precise cost calculation can be very difficult and time consuming for any Real Estate Company. It also requires a lot of human labor which increase the cost of such work. In this case a Real Estate Company can use a machine learning model which can help them by predicting the sale price of a house as close as possible to its real value. This would save the company o lot of money they are using on on-field resources.
* The Model prediction will also give them a safety net of buying a house after which they can sell it at a **profit margin**. The Company will be able to accurately predict the price of a house and can compare with the price it is getting sold at to get an idea of whether it’s a profitable deal to buy a house on that price or not. The Company can also **estimate the marketing and additional cost** they can spend on a house after buying and still sell it on a profit margin. When a company is buying a house to sell. Money the company can spend on:

Marketing and additional Cost = (Selling Cost by owner - Predicted Cost by the model) - Profit they want to make.

**Executive Summary:**

* **Dataset:** Thedataset consists of 79 variables and 1460 observations for training dataset and 79 variables and 1459 observations for test dataset.
* **Approach:** Data has been cleaned using various imputation techniques such as converting the NA values to none or 0 where applicable. When variables are dependent on other variables in the dataset different methods like mean, median and mode are used to remove the NA values from them. After data cleaning log transformation and dummy variables are used for feature engineering.
* **Models used**: Models like Linear Model, Ridge regression and Lasso regression, GBM (Gradient boosting Machines), Ensemble model has been applied.
* **Result:** Ensemblemodel performs the best on the test data set with an RMSE value of 0.12884 (Model consists of 20% weightage to Ridge, 20% weightage to Lasso and 60% weightage to GBM.

**Approach:**

**Data Exploration:**

Initially, we started by focusing on understanding the dataset. The dataset consists of 79 variables (excluding Id and sale price) for the training dataset and 79 variables for the testing dataset which also exclude the “SalePrice” which needs to be predicted. The training dataset has 1460 observations and testing dataset has 1459 observations. By taking a glimpse into the data it was very clear that some features are integers while others are characters.

We started by combining the train and test data into one dataset called combined set to perform data cleaning and feature engineering. We started by plotting the SalePrice histogram to make some observations about SalePrice. We have noticed that the histogram plot obtained from SalePrice was skewed. This was expected as some houses are very costly compared to majority of houses in the dataset.

Chart, histogram

Description automatically generated Chart, bubble chart

Description automatically generated

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

Next, we plotted the correlation plot of all the numeric variables which has a correlation of at least 0.5 with SalePrice. We found that there are 10 such numeric variables out of which overall quality (OverallQual) has the highest correlation of 0.79 with SalePrice.

**Data Cleaning:**

Initially we started by checking the variables that contains NA value in the combined dataset. It turns out that PoolQC contains the highest number of NAs i.e., 2909 while there are many features of Garage and Basement contains only one NA value.

We divided the features into two parts. The first part consists of features where imputation of individual features can not be predicted using by using any other features in the dataset. The second part consists of features in which missing value 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**, **Exterior2nd**. We visualized the frequency distribution of these variables to get an idea of what value can be used to directly impute these variables.

From the frequency distribution graph below, we can conclude 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 appeared for that variable. NAs for Fence, Alley and MiscFeature has been replaced with None by looking at the description of these variables from data\_description.

Chart

Description automatically generatedChart

Description automatically generatedGraphical user interface, table

Description automatically generatedGraphical user interface

Description automatically generated

Figure 3: Frequency distribution of features for direct imputation.

After this imputation we started with 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 is only 3 values which 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 than the fireplace quality should be none corresponding those values.

**3). Lot Frontage Variables:** Lot frontage variable is linked with 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 lot frontage with the same neighborhood.

**4). Garage Variables:** We have changed the NA values of GarageType, GarageFinish, GarageQual, 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 basement. First, we observed that BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2 has the common 79 NA values. The additional NA values for each column is 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 has been changed to 0. We have only one value (house 2611 in 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 Sub class variables:** The values of MS Zoning is linked with MS Sub class. To visualize a distribution of house type vs zoning classification is created. From the histogram we can see that the hose type of 70 and 30 have zoning classification 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: i)Distribution of MSSubClass with Zoning Classification.

ii) Distribution of SaleType with SaleCondition.

**8). Sale Variables:** Linked the variable sale type with sale condition. The NA value in SaleType has normal sale condition. Majority of normal sale condition is of type WD. So, the value in SaleType is replaced with “WD”.

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

**10). Electrical System Variable:** NA values are replaced with the mode value of the Electrical variable.

**Feature Engineering:**

1. Log transformation is done to transform skewed numeric variables into their log terms. Because of this transformation the relative changes are more visible compared to the absolute changes.
2. Changed the character features into dummy variables.

**Model Selection, Training and Fitting:**

We separated our combined data back into train and test data based on SalePrice value. If SalePrice value is present the row belongs to the train data and if the value is NA or not present the row belongs to the test data. We further split the train data to data\_train (75%) and data\_test(25%) 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. We are using **root mean square error (RMSE)** to compare our model’s accuracy because that is the evaluation criteria set for our competition. Our approach to select a model is based on the model’s ability to deal with high dimensionality dataset.

To calculate RMSE we performed two methods to go with: 1). To use the outcome values same as given that gives a bigger RSME value because we are using the value of sale price which is a bigger value.

2). To use the log transformation on the outcome variable sale price to get a smaller RMSE vale.

For the report purpose second RMSE is used because it’s value make it easier to compare with the results of Kaggle RMSE value.

**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.** We have also applied the linear model with forward selectionbut there was not much change in the RMSE value that was obtained through linear model.

Next, we decided to use the **Shrinkage Model** such as Ridge and Lasso because we are dealing with a high dimensionality dataset. Ridge and Lasso methods penalize the model coefficients using lambda as a tuning parameter to reduce variance and result in a better fitting model.

**Ridge Regression:** We converted the test and train dataset into matrices because glmnet function requires matrixes. We implemented the gmlnet function over the long range of lambda to find out the best lambda for our model. We used cv.glmnet function for **cross validation** to choose the best lambda. The best **lambda** value with smallest cross validation error is **0.1913946 (lambda)**. The **RMSE** value we obtained on the testing set using the best lambda is **0.9100479 (RMSE)**.

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

Histogram

Description automatically generatedChart, histogram

Description automatically generated

Figure 5: Lambda plot for Ridge and Lasso Regression

**GBM (Gradient Boosting Machine):** We have used GBM as there is **high variance** between the variables and it’s difficult for any single decision tree to fit all the training set variables. So, we decided that we will use a **supervised learning algorithm that builds multiple trees for prediction**. The data split remains the same as in above mentioned models. We have used 10-fold cross validation which is repeated for 5 times. The **RMSE** we obtained from this model on testing set comes out to be 0.1146263 **(RMSE)**.

**Ensemble Model:** We picked such a model to incorporate the qualities of different models which we have applied. 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 test set resulted in 0.13214 value. Using different combinations of weighted average, the best RMSE on 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 **0.12882**

**Conclusion/Results:**

We have applied all the models that we thought would be the best for our dataset. From Ridge regression model we submitted our result and got an RMSE value of 0.13988 on the test set for which we need to do the final predictions. Lasso regression has resulted into an RMSE value of 0.13897 on our test set. GBM resulted into an RMSE value of 0.1323 on the test set for prediction. Even though the Ridge has the least RMSE value on the test data which was split from training data for testing purpose than other model except ensemble, it turns out that GBM performed the best on the test set for which we need to make predictions. The **Ensemble Model gives the best accuracy** with **RMS value if 0.12882** for the test set of competition. So, we can say that Ensemble model beats the Linear model, Linear forward model, Ridge Regression, Lasso Regression and GBM in terms of accuracy for this dataset.

**References:**

[**https://www.kaggle.com/code/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda/script**](https://www.kaggle.com/code/erikbruin/house-prices-lasso-xgboost-and-a-detailed-eda/script)