CSC 440: Data Mining Final Project (House Prices Prediction)

Sai Pranav Chittella, MS in Computer Science

ABSTRACT: There is a widespread belief that house prices are dependent on the generic factors like number of bedrooms and square area of house, and to prove this incorrect, The Ames Housing dataset proves that many other factors influence the final price of homes. This dataset contains 79 explanatory variables to describe almost every aspect of the house. The buyers/customers usually neglect this information, but it is greatly useful to other parties. As a result, their price estimation is very different from the actual prices. I have built a model to predict the prices of residential homes in Ames, Iowa, using advanced regression techniques. This will provide buyers will a rough estimate of what the houses are actually worth. This in turn will help them have better negotiation deals with sellers.

KEYWORDS

RMSLE, LASSO, XG BOOST, LG BOOST, Random Forest, Elastic Net

1 INTRODUCTION

One has to pay huge sums of money and invest many hours and even then there is a persisting concern whether it's a good deal or not. Buyers are generally not aware of features that influence the house prices. Almost all the houses are defined by the total area in square foot, the neighborhood and number of bedrooms. Sometimes houses are even priced at X dollars per square foot. This creates an illusion that house prices are dependent almost solely on the above stated factors. Alonside, there are various tools, like Zillow and Trulia, available online to assist a person with buying houses. These tools provide a price estimation of various houses and are generally free for use. These tools incorporate many factors to estimate the house prices by providing weights to each factor. For example, Zillow creates Zestimate of houses which is "calculated three times a week based on millions of public and user-submitted data points". The median error rate for these estimates is quite low.The primary goal of this project is to train a model based on the given data to predict the house price almost as accurate as possible evaluated by the Kaggle Leaderboard. The main components of the project include,

2 RELATED WORK AND METHODOLOGY

Housing market is important for economical activities (Khamis & Kamarudin, 2014). Traditional housing price prediction is based on cost and sale price comparison. So, there is a need for building a model to efficiently predict the house price. Khamis compares the performance of predict house price between Multiple Linear Regression model and Neural Network model in New York. Few Kagglers have come up with various regression techniques and neural networks approach [1], but the problem of choosing neural networks is the run time efficiency and computing requirements. Also, the goal of the project itself is to do a survey on a range of regression techniques and then do analysis on them. Alongside, previously people have tried to average several base models without stacking them[2] , and the people who did had different stacking methods and different scores.

Data analysis – Understanding the type of features in the dataset, how much missing values of different features, how many outliers are existed and whether specific features are skewed, are crucial in order to create a good model in the end.

Data preprocessing – Preprocess the data using the insights obtained from the exploratory data analysis, potentially including feature transformation, data type transformation, outlier detection and imputing missing values.

Modeling- – Creating a model using a standard technique for the problem in order to set up a benchmark for the future modeling improvement. Averaging the base models to improve the performance of individual model and stacking individual models to provide final results.

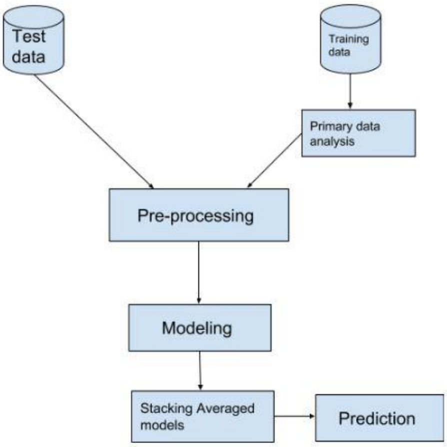


Figure 1: Project Methodology

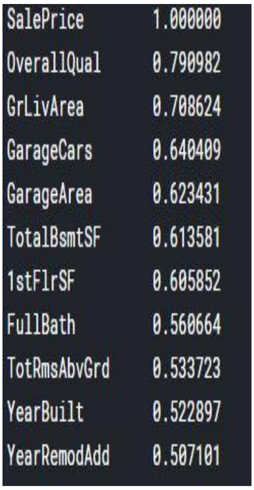
3 EXPERIMENTS

3.1 Data Analysis

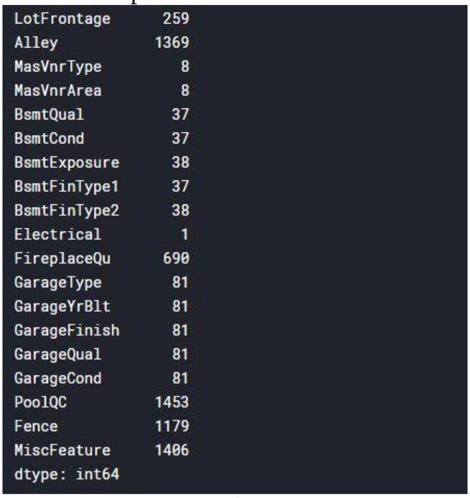
The training dataset is mixed with categorical and numerical features as well as some missing values. Specifically, there are 1460 observations, each with 81 features including the sale price of the house. The test dataset has 1461 observations, each with 80 features. For instance lot size, neighborhood, sq ft area, basement which a home buyer would want to know about a potential property. Our task is to fill the sale price of the test dataset using the model trained from the training dataset. 20 continuous variables relate to various area dimensions for each observation, from the available set of variables. Simple variables like typical lot size and total dwelling square footage can be found on most common home listings, other more specific variables are quantified in the data set. Area measurements on the basement, main living area, and even porches are broken down into individual categories based on quality and type. The large number of continuous variables in this data set lays platform to consider various methods of using and combining the variables.

The data analysis also said as the exploratory data analysis is the heart of the project although it is bit tedious and has consumed most of my time in the project. I have come across few projects which haven’t performed data analysis, due to which imputing the missing values in the data is done in a wrong way.

In order to proceed further, the correlation between the features and the target variable ‘SalePrice’ is studies closely and various forms of plots are obtained.



```
SalePrice       1.000000
OverallQual     0.790982
GrLivArea       0.708624
GarageCars      0.640409
GarageArea      0.623431
TotalBsmtSF     0.613581
1stFlrSF        0.605852
FullBath        0.560664
TotRmsAbvGrd    0.533723
YearBuilt       0.522897
YearRemodAdd    0.507101
```



```
LotFrontage      259
Alley           1369
MasVnrType         8
MasVnrArea         8
BsmtQual          37
BsmtCond          37
BsmtExposure      38
BsmtFinType1      37
BsmtFinType2      38
Electrical         1
FireplaceQu      690
GarageType        81
GarageYrBlt       81
GarageFinish      81
GarageQual        81
GarageCond        81
PoolQC          1453
Fence           1179
MiscFeature     1406
dtype: int64
```

Figure 2: Correlation between SalePrice and other dominant features and the missing values

I have also performed univariate, bivariate and multi variate data anlaysis on how the correlated features behave with the ‘SalePrice’ attribute and its performance can be seen in the graphs(only few of them are attached in the report due to the space issues).

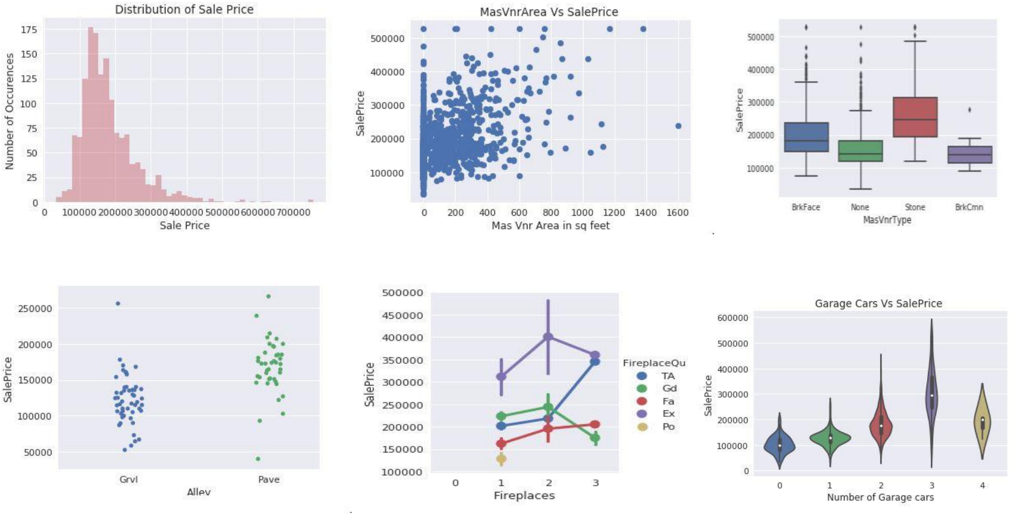


Figure 3: Exploratory Data Analysis with ‘SalePrice’ as one parameter

3.2 Data Pre-processing

The first step is to detect the outliers in the data and when a scatter plot is taken between the SalePrice and General Living Area, we understand that there are few outliers and for the sake of convenience we only delete two entries.

Second, the data seems to be right skewed and to reduce this tendency, transformation is done and the plot is as shown in figure 4.

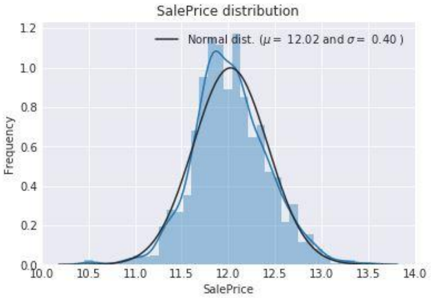


Figure. 4. Distplot when a log transformation is performed

In the next step , a correlation map is used as a whole to cross check with the data analysis results. The results is as below,

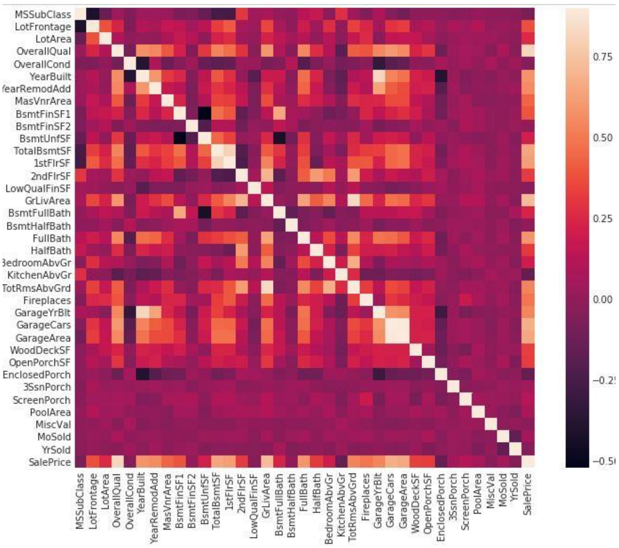


Figure 5: Correlation map

By getting results from the Data Analysis file, we can impute certain values into the table. The feature selection approach adopted here is a very simple and naïve one: delete features with high number of missing values. The categorical variables with the largest number of missing values are Alley, FirePlaceQu, PoolQC, Fence and MiscFeature. In the data exploration analysis, there are some features that are highly correlated with the sale price. We may treat some of these features as redundant variables and delete them from the training dataset. However, previous investigation shows that this will leads to a small decrease in both the local score and public score.
The categorical features are handled by the pandas get_dummies function that converts categorical variables into dummy or indicator variables. See pandas documentation for more details [3].

3.3 Modelling

3.3.1 RMSE

The Root Mean Squared Logarithmic Error (RMSLE) is used to as the evaluation metrics for this project.

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

Fig. 6. RMSLE.

where $\mathscr{E}$ is the RMSLE score, n is the total number of observations, p_i is the predicted number of the house price The value of RMSLE is higher when the differences between the predicted and actual house prices are larger. Compared to the most common Root Mean Squared Error (RMSE), RMSLE does not heavily penalize the huge difference between the predicted and actual values. Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally.

3.3.2 LASSO

“Linear Model trained with L1 prior as regularizer”[sklearn.linear model.Lasso. scikit learn. http://scikit-learn.org/stable/modules/generated/ sklearn.linear model.Lasso.html]. We use LassoCV() function from the sklearn.linear module. We do the same thing here. We take α equal to 0.005 and proceed with the calculations. The mean CV score for Regression obtained is 0.1128, which is better than few models previously.

3.3.3 Gradient Boost, XG Boost and LG Boost

“Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.”[Gradient boosting. https://en.wikipedia.org/wiki/Gradient boosting].
The mean CV score for Gradient Boost is 0.1169.
The implementation of the XG Boost algorithm is engineered for efficiency of compute time and memory resources. A design goal was to make the best use of available resources to train the model. The mean CV score obtained is 0.1161.
Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise algorithm can reduce more loss than a level-wise algorithm. The mean CV score is 0.1154.

3.3.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. The mean CV score is 0.1378.

3.3.5 Elastic Net

Elastic Net produces a regression model that is penalized with both the L1-norm and L2-norm. The consequence of this is to effectively shrink coefficients (like in ridge regression) and to set some coefficients to zero (as in LASSO).The mean CV score obtained is 0.1129.

3.4 Stacking and Averaging

The technique we implemented in this project is called stacking. Stacking, which stands for “stacked generalization”, was introduced by Wolpert (1992). The basic idea of “stacking” is to use another model or “stacker” to combine all previous model predictions in order to reduce the generalization error.

In each iteration, each base model will be trained using 3 folds and predict on the hold out fold. At the same time, each base model also need to provide a prediction on the entire test dataset. After the iteration over all folds, we will have the prediction of the entire training dataset for each model and 5 copies of the prediction of the entire test dataset for each model. Finally, we train second level model, or stacker, using the prediction in the training dataset as new features and use the average of the 5 copies of the test dataset predictions as the test input for the trained model to provide the final prediction.

4 CONCLUSIONS

The ensemble that is taken is ,
(stacked_train_prediction*0.90 + xgb_train_pred * 0.05 + lgb_train_pred * 0.05).
The RMSLE error is the lowest of obtained till and have made the comparisions with the other contestants whose approach is similar to the stacking. The RMSLE obtained is 0.07517 , which says that this model has the best prediction out of all the models I have presented above.[4]
We predicted the 'SalePrice' of the houses for the given Ames Housing dataset using two different methods. The first one included Ridge and Lasso model. The second one had Lasso and XGBoost. Since the second model performed better we the predictions provided by that model. This prediction data will help eventual buyers to have a better knowledge of the property. This will in turn help them to have better negotiation deals with the real estate agents.
This project opens several avenues for future work. One part is for data preprocessing, such as more advanced outlier detection techniques, more advanced feature selection techniques, fine-tuning of the log transform threshold of the skewed variable and so on. More creative feature engineering will be also extremely useful. Another part is for the regression techniques. My intuition for ensemble learning technique is that a better score may be achieved by increasing the size of the model library

ACKNOWLEDGMENTS

I would like to thank my instructor for the course Dr.Ted Pawlicki , for letting us choose the projects that we are interested in, as not only it would better our analytical skills, it helped me understand the process intuitively. Also I would like to thank the TA's for this course for responding patiently to my doubts and helping me in setting up the environment.

ENVIRONMENT SETUP

It is recommended to load the kernel to the Kaggle , as the data set I have used is on the cloud, and I haven't downloaded it. In case, if you are using the ipython notebook, as it will cover all the libraries included for this project and additionally download the XGB and LGB libraries using the steps mentioned on the readme file. The libraries used for this project are the following,

- Numpy
- Scikit-learn
- Seaborn
- matplotlib
- XGBoost
- LightGBM

REFERENCES

[1] https://www.kaggle.com/keerthirbollam/predicting-house-prices-neural-networks
[2] https://nycdatascience.com/blog/student-works/predicting-house-prices-using-machine-learning-algorithms.
[3] http://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html
[4] Ames data to Boston data set. http://www.amstat.org/publications/jse/v19n3/decock/ DataDocumentation.txt