Data Mining Final Project
House Prices: Advanced Regression Techniques

Authors: Sahar Hajiseyednasir, Haolan(Helen) Zhang

ABSTRACT

Predicting the house price is one of the most popular problems in the machine learning concept. Whenever the regression algorithm is the discussion topic the housing price problem shows up as a typical and common problem. There are many approaches to solve this problem, but here we used House Prices data from Kaggle.com website and we tried different algorithms to do parameter tuning and eventually we have the predicted results. Our main objectives for the project are 1) to gain facility in the end-to-end process of a data science project in a collaborative environment and 2) to better understand the implementation and evaluation of various supervised machine learning techniques.

KEYWORDS

Data prediction, Classification, Regression, Features Selection, Machine Learning

1 INTRODUCTION

Houses prices could vary with changing different parameters. For example, houses have different prices in different locations. In this project we used House Prices dataset from kaggle.com which has 79 explanatory variables describing every aspect of residential houses of the dataset. The kaggle.com provides us with two datasets which are train and test sets. In this project we followed different steps, Data processing, Mo n this project we followed different steps, Data processing, Modeling and Evaluation. Data processing which is the most important step is to get our data ready for modeling. In data analysis, how the data is generated to be read by the computer and kernel is very critical. Most of the time the raw data that we are provided with is not something that computers could understand. Because there are many missing values, irrelevant values, outliers, categorical values. What data scientists do in this significant step is to transfer the attributes the way that are interpretable by computers. Here in this project, we also did the same thing and we tried to generate the best dataset as our training set to make our model be a more accurate model.

House prices project is the project that the price of a large number of houses should be predicted using Linear Regression. In the Modeling step using just simple Linear Regression does not seem reasonable, why?! Because the model could be overfit or underfit to our training set and make it useless for applying to our test set. In this step we use different models for parameter tuning and using different training sets to avoid over and under fitting for our model. At the end, what we have done is that we compared these different models and use the average of these models to predict the values of prices for houses in our test set.

In conclusion, our workflow consisted of a full development cycle divided into five stages: exploratory data analysis and pre-processing, feature engineering, modeling, hyperparameter tuning, and ensembling.

2 DATA PREPROCESSING

Datasets are not always the way that we are looking for. Sometimes there are some columns or variables that are not useful for our prediction or they are not in the formatted that we are looking for. In this step we are making our data ready for prediction. ‘SalePrice’ is the variable we need to predict. Therefore, let's do some analysis on this variable first. There is no doubt that the year of house built will affect housing price. We try to visualize the relationship between them with boxplot. Below is what we get. It seems that the price of recent-built houses is higher.

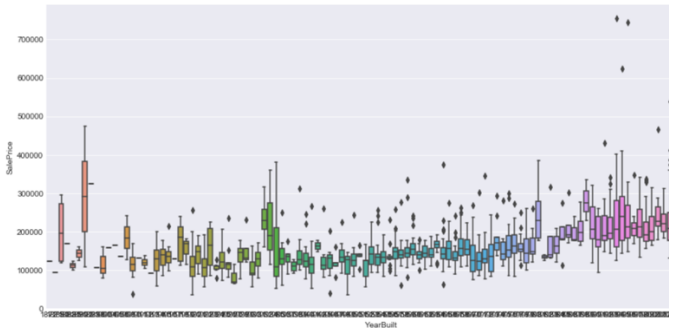

Figure 1: Relationship between 'YearBuilt' and 'SalePrice'

2.1 Outliers

Outliers removal is note always safe. We decided to delete these two as they are very huge and really bad (extremely large areas for very low prices). There are probably others outliers in the training data. However, removing all them may affect badly our models if ever there were also outliers in the test data. That's why, instead of removing them all, we will just manage to make some of our models robust on them. We go through all the columns to drop the outliers to avoid out model to be overfit.

To see the relationship between the variables in our training set, we plot the relationship below. It can be observed that we have two outliers in the GrLivArea.

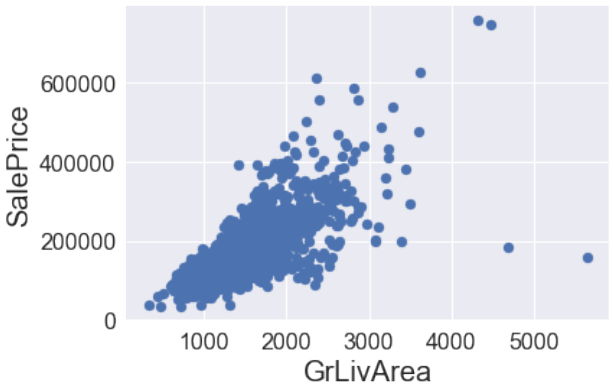

Figure 2: Scatter plot of relationship between GrLivArea and SalePrice

We drop these two points to avoid our prediction error to be large for the test set. After drop it the graph below is generated. We can see that 'SalePrice' there is a positive relationship between 'SalePrice' and 'GrLivArea'.

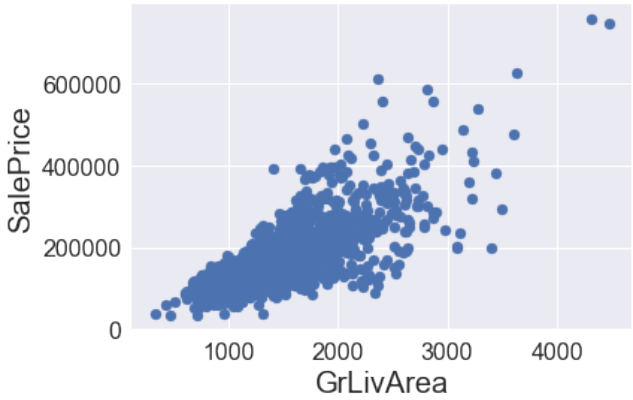

Figure 3: Scatter plot of relationship between GrLivArea and SalePrice after transformation

2.2Normalization

Linear Regression algorithm is a supervise learning algorithm. Therefore, we have labels in our training data that in this case it is the price of the houses in the training data. Linear Regression will use these labels to find a model between variables and the labels.

The EDA we perform here is to examine the distribution of the home sale prices. The histogram of home sale prices appeared to be right-skewed. We therefore performed a log transformation of the home sales prices to make the distribution more Gaussian. The following are histograms of home sales prices before and after the log-transformation:

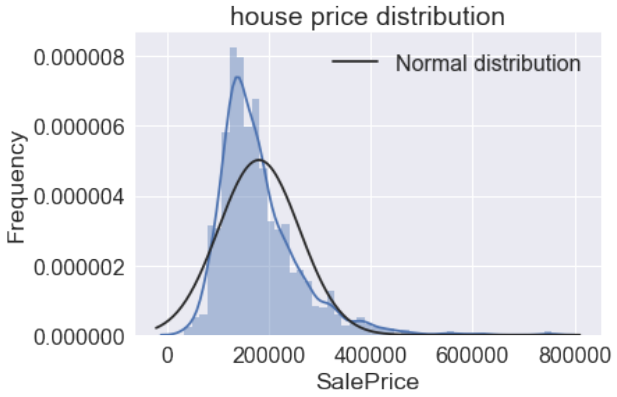

Figure 4: Histogram of housing price in dataset before transformation

As models love normally distributed data, we need to transform this variable and make it more normally distributed. We use Log-transformation. The skewness seems now corrected and the data appears more normally distributed. Statistically, the skewness and kurtosis are decreased from 1.88 and 6.52 to 0.12 and 0.80.

Figure 5: Histogram of housing price in dataset after transformation

we also explore the correlation between the various features using correlation matrices. The following correlation matrix shows the correlations between some of these features.
After the data transformation, we use the correlation matrix to find the most related. To this point, we have a basic knowledge of the dataset.

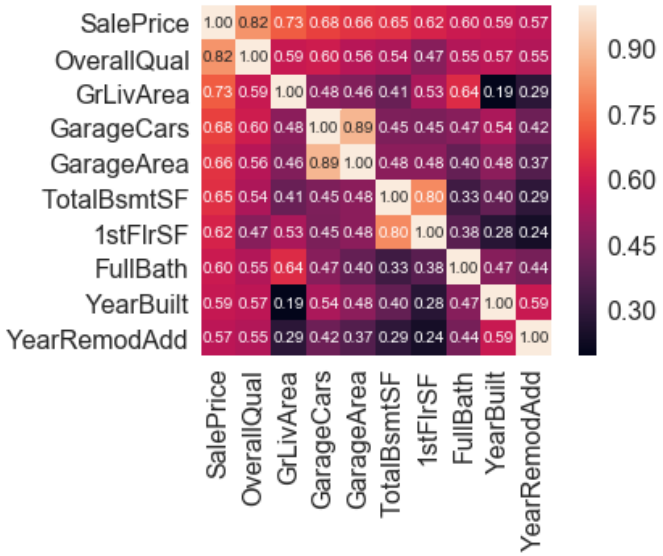

Figure 6: Correlation matrix between 'SalePrice' and other features

2.3 DATA CLEANING

In order for machine learning algorithms to provide meaningful insights, we needed to ensure that the data was relatively clean. For our dataset, we had to change some feature types and also handle missing values. The dataset cannot be processed when we have missing values in the rows. It usually throws an error says that it does not know how to handle the missing values. There are several ways that we take care of our missing values. For example, sometimes we drop the rows with missing values when we have large enough dataset after dropping the missing values included rows. However, most of the times it is not possible to get rid of them because we would lose a great number of our valuable data. What we can do here we can put median, mode or mean of a specific column's values instead of missing values. In this problem we have 79 columns and we used different approaches for each of the columns according to the properties of that column.

Type Conversion

First, we had to change the data types of the below features from numeric to string.

Features	MSSubClass, OverallCond, OverallQual, GarageCars, YrSold, MoSold

The features' value above represent different categories. Take MSSubClass as an example. The number encode different categories of houses, -Story 1946 and Newer (60) and 2-Story 1945 and Older (70). It is harder to discern in a feature like GarageCars, where each value seems to count something (cars) but in actuality represents the garage capacity, and therefore represents a category.
Usually there are some columns that they are categorical values like the colors. The values in a color column could be 'red', 'blue', 'green' and etc. The computer or our kernel is not able to understand and process these values. We need to change them to numeric values which are understandable for computers. To this end, we use get.dummies() function in Python Pandas library that transform the Categorical values to numeric Boolean values. Python pandas library has a function get_dummies() named that change the categorical variables to the numeric values.
This function makes a column for each value in each categorical column and puts 1 as that value exist for that row and puts 0 it that value does not exist for that row. For instance, we have three colors as values in the color column in our dataset. This function makes separated columns for each values in the color column (in this case three columns) and put zero or one for each color if the color of that observation is 'red', 'blue' or 'green'.

Missing Values
We used the below strategies for dealing with missing values.

Flagging as 'None'

Features	Alley, BsmtCond, BsmtQual, BsmtExposure, BsmtFinType1, BsmtFinType2, Fence, Functional, FireplaceQu, GarageCond, GarageFinish, GarageQual, GarageType, MasVnrType, MiscFeature, MSSubClass, PoolQC

The reason we need to do that is we want the model to treat observations with missing values as a separate category. Take 'PoolQC' as an example. From description, we know that the house does not have a swimming pool if 'PoolQC' shows missing value. However, it is important to let the algorithm know it too. Therefore, we flag the missing values as 'none'.

Impute Zero

Features	'BsmtFullBath', BsmtHalfBath', 'GarageYrBlt','MasVnrArea'

3. FEATURE SELECTION
Having said that, in this problem we are predicting the house price for houses in our test set. To do so, we are find our model using our training set. Once we are done with the data processing, we can now read and process our training set. overfitting and underfitting are the most concerns for machine learning experts. If the model that we generate is overfitted or underfitted the model will not be able to predict the prices properly.
To avoid these issues, what we did here is that we used Cross Validation function 5 folds to train the model better. Using cross validation enables our model to be tested with different set of datasets generated by cross validation function.
In addition to the cross validation, sometimes our variables coefficients are not tuned and this cause the problem that the model is overfitted and some variables are more important than what they should be. To automatically do feature selection and tune the coefficients of our variables in regression we used Lasso, GBM, XGB and Kernel ridge regression Methods.

3.1 LASSO
Lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.[1]
Lasso is quite similar conceptually to ridge regression. It also adds a penalty for non-zero

1 https://en.wikipedia.org/wiki/Lasso_(statistics)

coefficients, but unlike ridge regression which penalizes sum of squared coefficients (the so-called L2 penalty), lasso penalizes the sum of their absolute values (L1 penalty). As a result, for high values of λ, many coefficients are exactly zeroed under lasso, which is never the case in ridge regression.[2]
We use lassocv() function to do the parameter tuning for lasso model. Instead of simply conducting cross validation on all of the training data, it is advised to split it up into more traditional training set / validation set partitions.The Lasso is thus trained on the training set and then the hyperparameter alpha is tuned on the basis of results from cross validation of the validation set. Finally, the accepted model is used on the test set to give a realistic view oh how it will perform in reality. Seperating the concerns out here is a preventative measure against overfitting.
Thus, we get the RMSE score: 0.1131.

Here we have the importance of each features and residuals.

Figure 7: Plot of coefficients for lasso model

Also, we have the plot of residuals for LASSO model.

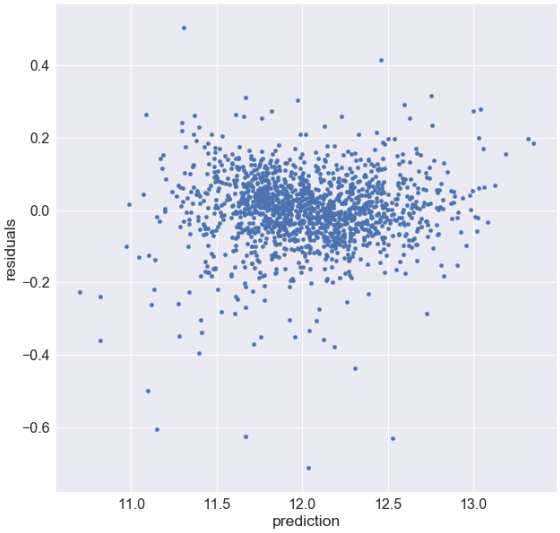

Figure 6: Plot of residuals for LASSO model

3.2 Gradient boosting and XGBoost

Tree-based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable to solving either classification or regression problems.
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 decisions tree. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.[3]
XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage.[4]

2 https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net

3 Breiman, L. (June 1997). "Arcing The Edge" (PDF). Technical Report 486. Statistics Department, University of California, Berkeley.

4 https://xgboost.readthedocs.io/en/latest/

It stands for Extreme Gradient Boosting; it is a specific implementation of the Gradient Boosting method which uses more accurate approximations to find the best tree model. It employs a number of nifty tricks that make it exceptionally successful, particularly with structured data. Both XGBoost and GBM follows the principle of gradient boosting. There are however, the difference in modeling details. Specifically, XGBoost used a more regularized model formalization to control over-fitting, which gives it better performance. From the XGBoost, we get 'GriLivArea', 'LotArea' and 'TotalBsmtSF' are the most relevant features to 'SalePrice'. Cross validation grid search is performed on each tree-based model with their respective hyperparameters. This resulted in the selection of the optimal hyperparameters based upon the average RMSLE score against the test set.
The RMSE score for XGBoost is 0.1273
And the RMSE score for GBM is 0.1157.

3.3 Kernel ridge regression

Kernel ridge regression is a non-parametric form of ridge regression. The aim is to learn a function in the space induced by the respective kernel k by minimizing a squared loss with a squared norm regularization term. [5] Kernel ridge regression combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space.[6]
We use KernelRidge() function to do the parameter tuning for lasso model.
The graphs below show the coefficients for attributes with the KRR model.

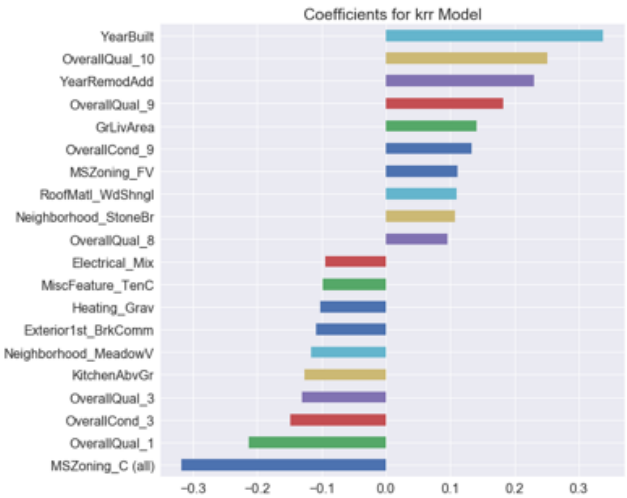

Figure 8: Plot of coefficients for KRR model

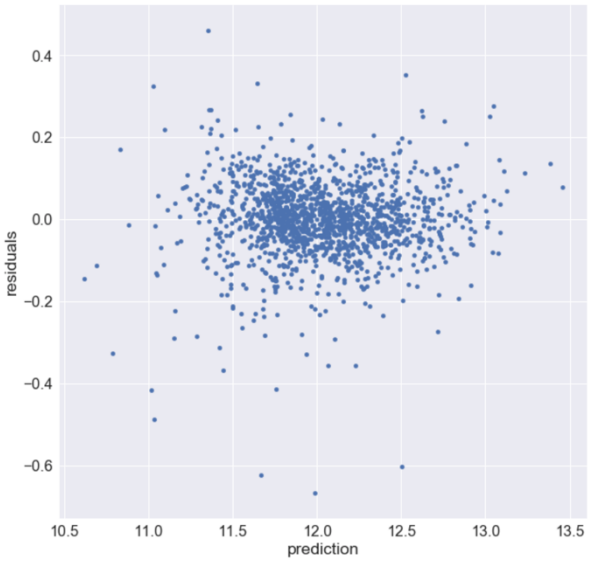

Figure 9: Plot of reiduals for KRR model

3.4 Stacking

After creating the linear and tree-based models above, we decided to combine them in an ensemble in order to increase the prediction accuracy and improve the overall confidence level of the predictions. The various models capture different aspects of the dataset, such as

5 Kernel ridge regression, Gaussian processes, and ensemble methods. CS281B/Stat241B (Spring 2008) Statistical Learning Theory Lecture: 10.

6 https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html

outliers, thereby making the ensemble more robust. A number of transformations and imputations were made to the dataset in this stage in addition to those made in the earlier stages before running the models in the ensemble.
Stacking (also called meta ensembling) is a model ensembling technique used to combine information from multiple predictive models to generate a new model. Often times the stacked model (also called 2nd-level model) will outperform each of the individual models due its smoothing nature and ability to highlight each base model where it performs best and discredit each base model where it performs poorly. For this reason, stacking is most effective when the base models are significantly different. Here we decide to use a simple stacking model to ensemble all the models together. From the four models mentioned before, we already get the rmse score we defined for each model. By serval experiments, we get even the simple stacking model improve out rmse score, same with the prediction. At last, we settle with ensembling LASSO, GBM AND KRR.

3.5 Result

The following is a summary of the RMSEs for the various models

LASSO	0.1131
KRR	0.1163
XGB	0.1273
GBM	0.1157
Ensemble (LASSO,GBM,KRR)	0.1103

Figure 10: Table for model score

4 CONCLUSIONS

In recent years, machine learning has been successfully deployed across many fields and for a wide range of purposes. One of its applications is in the prediction of house prices, which is the putative goal of this project. The dataset provides excellent learning material for us to perform data analysis, imputation, feature selection and machine learning models, such as linear-based model and decision-tree model. During this process, we gain a better understanding of the necessary steps to implement and evaluate various supervised machine learning techniques. To gain a sense of the relationship between housing price and other features, we use a lot of data visualization skills, including density plots, scatterplots, boxplots, and correlation plots. Also, for the purpose of using machine learning algorithms, we do data cleaning to gain a better insight. For our dataset, we had to change some feature types and also handle missing values. Last but not least, we ensemble machine learning models to improve our prediction performance. scatterplots, boxplots, and correlation plots. Also, for the purpose of using machine learning algorithms, we do data cleaning to gain a better insight. For our dataset, we had to change some feature types and also handle missing values. Last but not least, we ensemble machine learning models to improve our prediction performance. feature types and also handle missing values. Last but not least, we ensemble machine learning models to improve our prediction performance. scatterplots, boxplots, and correlation plots. Also, for the purpose of using machine learning algorithms, we do data cleaning to gain a better insight. For our dataset, we had to change some feature types and also handle missing values. Last but not least, we ensemble machine learning models to improve our prediction performance.

ACKNOWLEDGMENTS

I would like to thank Professor Ted Pawlicki for giving me this chance to do the data analytics work. It gives me a better understanding of data mining and also fosters me to have a deep interest to learn more about machine learning methods. This report was created as part of the final project for the course Data Ming at University of Rochester.

REFERENCE

[1] "Machine Learning: A Probabilistic Perspective" Murphy, K. P. - chapter 14.4.3, pp. 492-493, The MIT Press, 2012

[2] Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2

[3] https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard

[4] https://github.com/massquantity/Kaggle-HousePrices

[5] Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2. ISBN 978-0-07-042807-2.

[6] Alpaydin, Ethem (2010). Introduction to Machine Learning. London: The MIT Press. ISBN 978-0-262-01243-0. Retrieved 4 February 2017.