House Prices - Advanced Regression Techniques

Mert KÖKLÜ
Student
Eti, Yükseliş Sk. No. 5
06570 Çankaya/Ankara
(+90) 5417144365
kklumert@gmail.com

Şeyda AYBAR
Student
Eti, Yükseliş Sk. No. 5
06570 Çankaya/Ankara
(+90) 05435024094
seydaybar@gmail.com

Emre BAŞAR
Student
Eti, Yükseliş Sk. No. 5
06570 Çankaya/Ankara
(+90) 5389538826
emrebasar345@gmail.com

ABSTRACT
We are drowning in data, yet starving for knowledge. Data is so important for our life that it affects us in every possible way such as product recommendations. Altough this is not limited by recommendations, data has been used in many ways and has been formed basis to the machine learning algorithms. In this paper, we present our data science and machine learning skills applied on 'Ames Housing' dataset that has been compiled by Dean De Cock[1]. We are going to start demonstrating our data handling process, as well as data visualizations and comprehensive exploratory data analysis approach. These processes will give a lot of intuition about data and modeling section, we will feed our organized and refined data to our machine learning models such as decision trees and ridge regresssion.

Our depth of concepts are central importance task for many data processing and analyzing projects. With this analysis and model training phase, we can predict for house price that has not seen by model based on features of houses.

1. INTRODUCTION
Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continues output variable based on the one or more predictor variables [2]. We present our regression estimators that has been predicted the very well score on the test set of the Ames Housing dataset. Most of the machine learning models requires refined and polished dataset for the input, so as well as we need models to predict outcome for the house prices, we need to handle and process data for best the best score.

We had the dataset but it was not what we are going to feed to our models. Unprocessed data would cause so much trouble that it requires from us to handle it. This is why handling data is so important to create state of the art models. Moreover besides handling the data, analyzing the data is necessary to create such well performed models.

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques[3]. Altough it is so important, it is overlooked by data scientist. By doing data visualizations and comprehensive exploratory data analysis, we can create new features based on knowledge that we have learnt from the dataset. Morover, exploratory data analysis will present correlation between price and other features of the house so we can get insight about which feature will play an important role of the prediction of price and which is not.

2. OUR APPROACH
2.1 Data Handling and Exploratory Data Analysis
We have gathered the 'Ames Housing' dataset that was compiled by Dean De Cook from the 'House Prices - Advanced Regression Techniques' Kaggle competition[4].

We start as importing some necessary data science libraries such as pandas, numpy, matplotlib, as well as seaborn for beautiful plotting. The dataset has been divided into two sets: train and test set. The sets has been comprised from house features and target value 'SalePrice'. As every machine learning problem has a task to solve, our task was predicting the 'SalePrice' of the houses in test set.

By reading csv file and describing the dataset using pandas functions, we can see that there are 80 features if we do not include target feature. These features vary from the type of roof (RoofStyle) to overall material and finish quality (OverallQual). At firt glance, some tasks such as deleding 'Id' column needs to be done. Morover after describing the dataset, some columns are object dtype (data type) that needs to be converted to the numeric types to be fed into the models. Also there are a lot of empty values in the features so because some models can't work with missing values, we needed to handle them too.

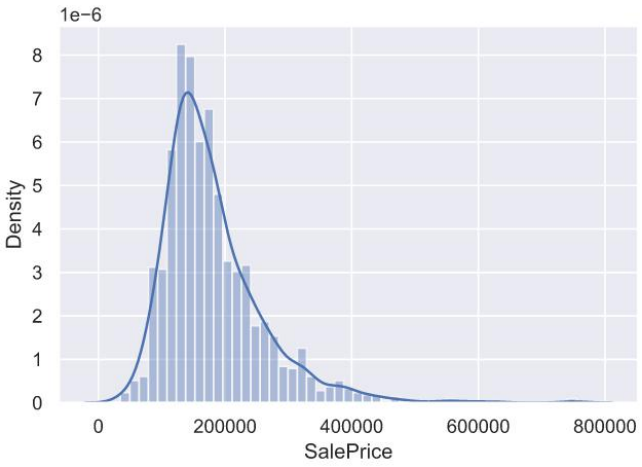

Figure 1. SalePrice distribution plot

Our target value 'SalePrice' is also a problem because of its skewness so we needed to fix that as well.

After calculating correlations between variables and plotting heatmap of it, we see that, the most important features that effects 'SalePrice' were table below.

Feature	Correlation	Feature Explanation
OverallQual	0.790982	Overall material and finish quality
GrLivArea	0.708624	Above grade (ground) living area square feet
GarageCars	0.640409	Size of garage in car capacity
GarageArea	0.623431	Size of garage in square feet
TotalBsmtSF	0.613581	Total square feet of basement area
YearBuilt	0.522897	Original construction date

According to this correlations table, we needed to keep eyes on some features. We had thought that YearBuilt will be important factor on house prices but it was not as we have expected. 'GarageCars', and 'GarageArea' has surprised us, yet these are strongly correlated among themselves because more cars means more garage area. We will understand how the dependent variable (SalePrice) and independent variables relate by doing bivariate regression analysis in the plots below .

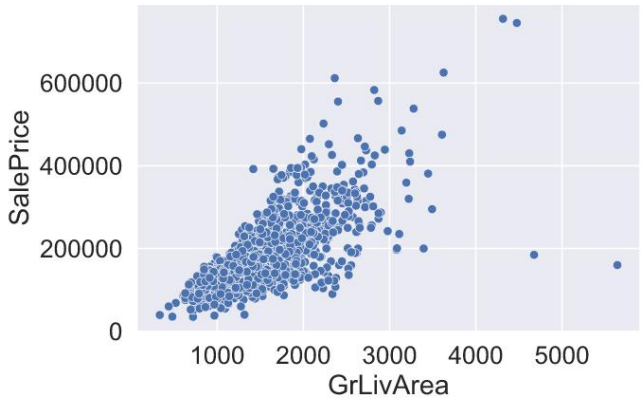

Figure 2. SalePrice with GrLivArea scatter plot

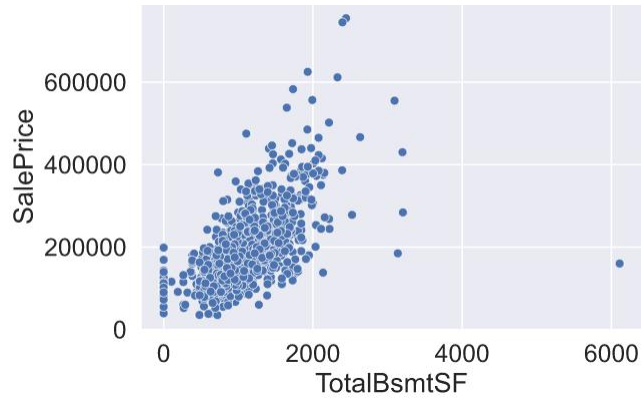

Figure 3. SalePrice with TotalBsmtSF scatter plot

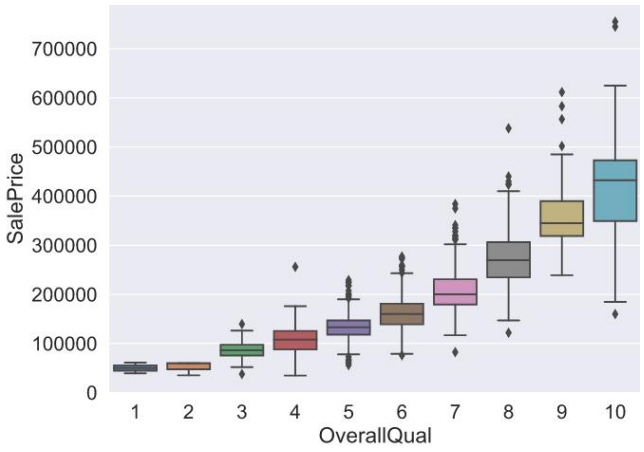

Figure 4. SalePrice with OverallQual box plot

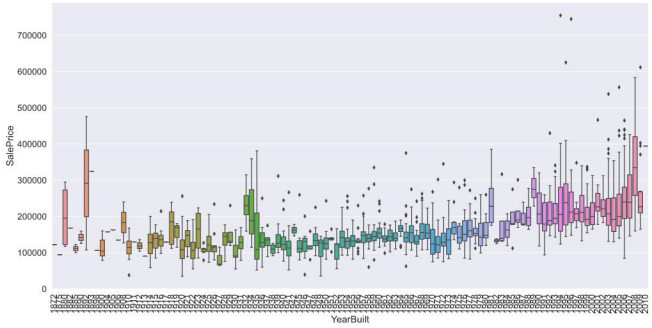

Figure 5. SalePrice with YearBuilt box plot

We can see positive correlation in the scatter plots. This means, there is a linear relationship that more 'GrLivAre' or 'TotalBsmtSF' means more 'SalePrice'. Morover we can see that in box plots too. It is not unexpected observing a linear relationship between 'OverallQual' with 'SalePrice'. Also sale prices has been slightly increased over the years as the correlation table has stated.

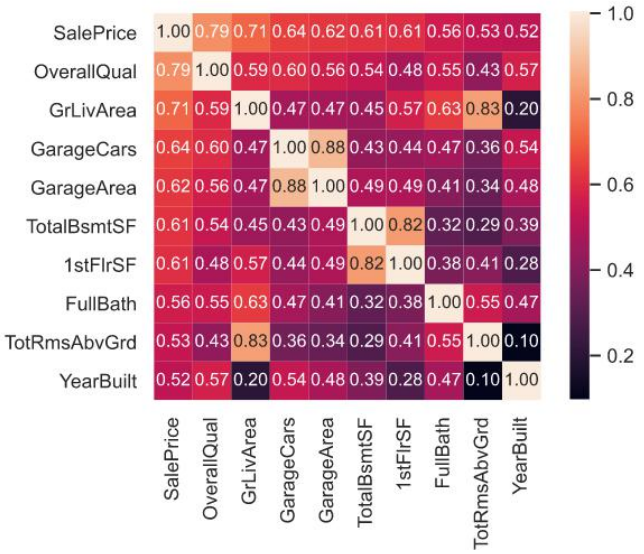

Figure 6. Correlation heatmap between attributes

The heatmap with annotations also gives a good insight about relationships between features and the target variable 'SalePrice'.

We have observed from 'SalePrice' with 'GrLivArea' scatter plot that 'GrLivArea' could had outliers. Outliers was a touble that we should be aware of. Because outliers can be markedly affect our models and can be a valuable source of information, providing us insights about spesific behaviours. The two values with bigger 'GrLivArea' seem strange and they are not following the crowd. We can speculate why this is happening. Maybe they refer to agricultural area and that could explain the low price. I'm not sure about this but I'm quite confident that these two points are not representative of the typical case. Therefore, we had defined them as outliers and deleted them. After that our 'SalePrice' with 'TotalBsmtSF' scatter plot becomes below.

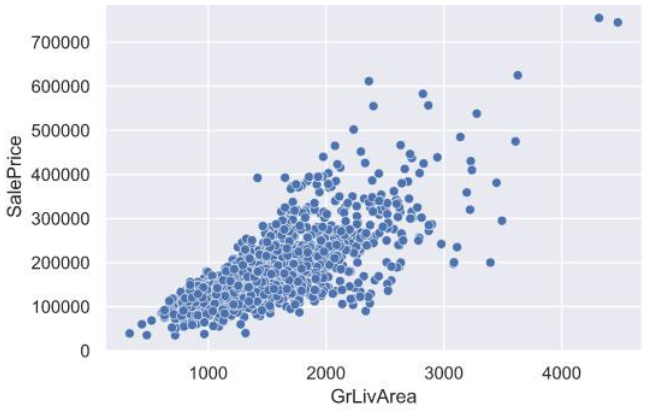

Figure 7. SalePrice with TotalBsmtSF scatter plot after removing outliers

Another thing that we though we need to fix was skewness. The 'SalePrice' was skewed to the right. This is a problem because most ML models don't do well with non-normally distributed data. We applied a log(1+x) tranform to fix the skew.

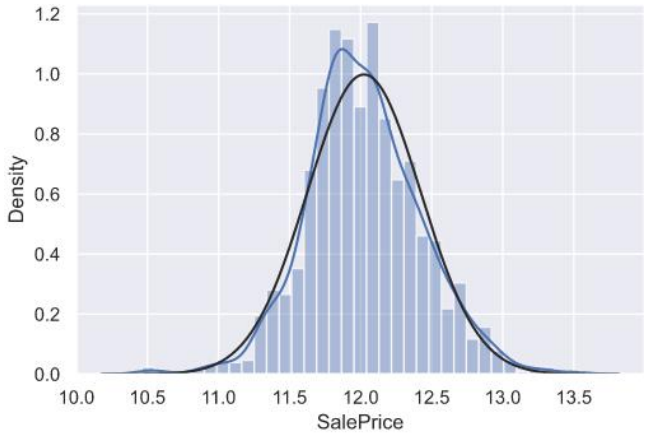

Figure 8. Distribution of SalePrice after fixing skewness

After applying the transform, we can see that the SalePrice distribution is like normal distribution (black line).

Analyzing correlations between features, fixing skewness on the target feature and deleting some outliers, it was turn for the handling missing values. As we have said earlier, the dataset has a lot of missing values as we observe from the table below.

Features	Total	Percentage
PoolQC	2907	99.69
MiscFeature	2811	96.39
Alley	2718	93.20
Fence	2345	80.41

Table 1. Total and percentage of the missing values per attribute

The dataset description page was saying that some of the features' missing values indicate that there is no such a thing. For example, If there is a house that has no pool, the dataset will indicate 'PoolQC' of the house as empty/missing value. Altough it is best to delete features if it has empty values more than %85, we avoid to delete in this problem. 'PoolQC' would be an important feature so if we delete it, we could lose some insightful information. Instead we had inserted 'None' value as indicating emptyness to the missing values. Morover this method was not applicable to the some features. Those are just missing values that does not indicate empytiness of that thing so we have filled missing values with mean or median value of the that particular feature.

Most of the machine learning models requires numerical attributes, ours included so we needed to convert attributes with pandas object dtype to the numeric dtype. First, there were some numeric types that were not numeric such as attribute 'YrSold'. We have converted them to the string types and after that, encoded all categorical features with the 'LabelEncoder' function from sklearn. After transforming, all of our attributes were numerical.

We have stated that feature engineering is important so we created one attribute 'TotalSF' that is sum of the 'TotalBsmtSF', '1stFlrSF', and '2ndFlrSF'. Altough, we could create more new attributes from the features, we did not prefer that to avoid from creating bias on some features.

Altough we have fixed skewness of the target feature, skewness of some numerical features also needs to be handled so we have fixed them and converted all categorical variables into dummy variables. After that, our data was ready to be fed by the models.

2.2 Regression Models

In this part, we have used sklearn library because of its rich machine learning algorithms. The regression models that we have applied with sklearn were,

- Linear regression
- Ridge regression
- Lasso regression
- Elastic Net regression
- Decision tree regression
- Random forest regression
- K neighbors regression

Some models requires appropriate parameters and it is exhausting to try all of them manually. The appropriate parameters for the models were very important that they could change the score drastically so we have defined some possible parameters for the model and used GridSearchCV[5] function from sklearn to do exhaustive search over defined parameter values for the

model/estimator. After fitting all the models, GridSearchCV has already found the best parameters for the estimators.

2.3 Prediction Phase: Score Evaluation

We had do prediction on the test set with the estimators that we have fitted. The score evaluation was based on RMSE (Root Mean Square Error)[6]. Kaggle competition that we gathered from the data was evaluating with RMSE so we thought it would be appropriate for us to evaluate in this way too.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

Figure 9. Formule of the RMSD (Root Mean Square Deviation)

Estimator	RMSE Score
Linear regression	0.1902
Ridge regression	0.1187
Lasso regression	0.1158
Elastic Net regression	0.1440
Decision tree regression	0.2141
Random forest regression	0.1387
K neighbors regression	0.2337

Table 2. RMSE score of all estimators

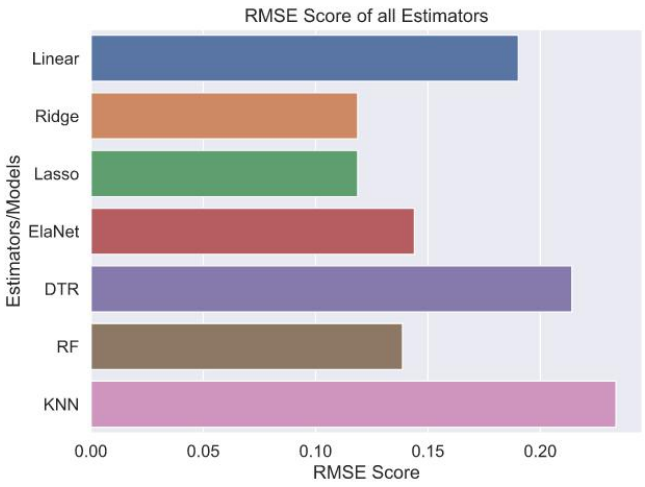

Figure 10. RMSE score comparison bar chart for estimators

We see that the best estimators that has the smallest root mean square error was Ridge and Lasso Regression estimators. Moreover, Random Forest Regressor and Decision Tree Regressor follows them with a very good RMSE score. Plotting correlations between estimators would give us the closeness between them.

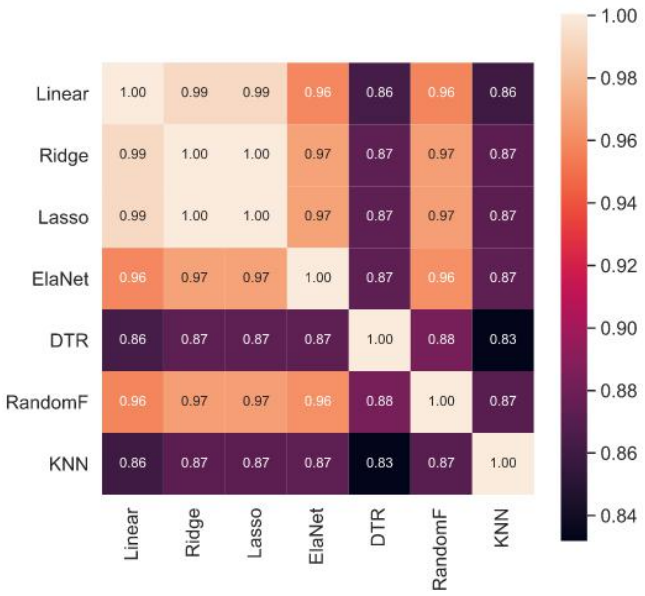

Figure 11. Correlation heatmap for RMSE scores of estimators

As we can see Lasso and Ridge regressors very high correlation but K neighbors regression estimator is less corralated with the other regressors.

2.4 Future Improvements

We could try more estimators such as LightGBM and XGBoost with appropriate parameters and use advanced approaches including stacking or voting regression scores on this task. We could also add more attribute with feature engineering and see if estimators perform well.

3. REFERENCES

[1] De Cock, Dean. "Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project." Journal of Statistics Education 19.3 (2011).

[2] Regression Analysis in Machine learning Web=https://www.javatpoint.com/regression-analysis-in-machine-learning

[3] Feature engineering, Wikipedia Web=https://en.wikipedia.org/wiki/Feature_engineering?oldformat=true

[4] House Prices - Advanced Regression Techniques Web=https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

[5] sklearn.model_selection.GridSearchCV Web=https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[6] Root-Mean-Square Deviation Web=https://en.wikipedia.org/wiki/Root-mean-square_deviation?oldformat=true