DSF HW 2. Project Report

We started with our data analysis as the first step. We analysed various interesting aspects of this dataset. For example, the maximum number of house transactions happened in the month of June. Before any further data analysis, we needed to clean the data and handle the missing values intelligently.

How did I handle the missing data?

First of all, I checked the number fo missing values in each column. Having found that, I dropped off the columns which had more than 50% missing data. Now,out of the remaining dataset, I replaced the missing values with mean value of that column for each column. Before being able to replace with mean or even evaluate the mean, I had to make sure that all the columnd were of the type numeric so that the mean evaluation makes sense.

Merging the train and property dataset

After dealing with the missing values in the data, I merged the two given datasets- train dataset and property dataset over the column 'ParcelId'. Now, we are ready to dive deeper into further analysis for correlation and heatmap plotting.

We analysed different features, how they were distributed with logerror values. We plotted different plots to display the dependecies.

Finding the correlation values and draw the heatmap

To find the correlation , we first had to remove all the categorical columns as thy do not contribute in plotting correlations and gave erraneous results. After removing all the categorical values, we were left mainly with the numerical columns. Finally we plotted the bar chart of correlations and discovered that the over all correlations were really very low in all cases. Hence, we did not find very high correlation between any features.

Filtering out the features having very low correlation with logerror

From the heatmap, we realized that some features were rally redundant for linear regression as they had extremely low correlation values. Hence, we had to drop these columns as well prior to the linear regression of the final output dataset.

Linear Regression Model

Finally after all th above steps, we proceeded with linear regression model for our prediction model. I used sklearn library for the same.

I divided the fina output data (merged train and property data) into 80% training set and 20% test set. I built my linear regressor model on 80% train data and tested it's validation on the remaining 20% test data.

Results from Linear Regression:

Mean squared error: 0.028639

Variance score: 0.003542

Kaggle Score: 0.0650163

Random Forest Model

After linear regression model, I tried random forest model. But, to my surprise, I found that this performed even worse than the linear regression model. Following are the results obtained:

Kaggle Score:0.0829437

Kaggle rank: 2222

Gradient Boosting Model

This model also did not improve upon Linear regression model. Though it performed better than the random forest model.

Following are the results obtained:

Kaggle Score:0.0654311

Kaggle rank: 2222

LightGBM Regression Model

This model also did not improve upon Linear regression model. Though it performed better than the random forest model anmd Gradient Boosting Model.

Following are the results obtained:

Kaggle Score: 0.0651120

Kaggle rank: 2222

Combination of Linear Regression and LightGBM

Since all the above models did not improve upon the linear regression model, I finally tried, a combination of linear Regression and LightGBM. What I tried here was summing up half the value of predictions made by linear regression model and half the value of predictions made by LightGBM (the two best models till now).

This finally did improve the output a little bit as shown by the following results:

Kaggle Score: 0.0649504

Kaggle rank: 2153 (final rank on leaderboard now)