Course: Data Mining and WareHousing

House Price: Advanced Regression Techniques
Kaggle Competition

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/

Team Name - Persistence
Team Member1 - Megha Agarwal (201506511)
Team Member2 - Kalpish Singhal (201505513)
Final Rank - 11
Final Score - 0.11449

Objective

Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

With 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

The potential for creative feature engineering provides a rich opportunity for fun and learning. This dataset lends itself to advanced regression techniques like random forests and gradient boosting with the popular XGBoost library. We encourage Kagglers to create benchmark code and tutorials on Kernels for community learning. Top kernels will be awarded swag prizes at the competition close.

Sold!! How do home features add up to its price tag??

Data Exploration

- The given Train Set has 1460 Rows and 79 Columns.
- Initially the train and the test data is concatenated.
- NaN values are replaced with significant values.
- The categorical variables which have some order(e.g. Poor, good, excellent) are replaced by numerical values [0, 1, 2, ...]
- Yes/No in certain features are replaced by 0, 1.
- The correlation of the features is calculated and the feature with highest correlation i.e Overall Quality and then Garage Live is considered to be of highest important.
- The features are trained accordingly using similar measures.

New Feature Creation:

- The feature extraction was quite minimal taking log transformations of numeric features and replacing missing values with the mean. It Didn't worked well - To improve it we added New features to it.

- Firstly the quality list is generated. Ordered categorical featured are divided into good and poor e.g. garage_poor_cond and garage_good_cond etc and are appended in a quality list. The idea is good quality should rise price, poor quality - reduce price.

- Exterior1st, Exterior2nd, RoofMatl, Condition1, Condition2, BldgType are converted to price brackets using SVM with kernel - rbf and gamma function - 0.001. Three price categories are created - pc1, pc2, pc3 owing to ranges (0, 150000), (150000, 220000) and (>220000).
- Months with numerical values had no significance and hence season features are also created.
- Taking into consideration the year sold, year built, YearRemoveAdd more features - recon_after_buy, build_eq_buy, etc are added.
- All the features are added to the original feature set, combining with different combinations of original features.
    - The final shape of the training data set was 1459 * 460 columns.

Scaling Features

- The features which are not of object type are fetched. First we have transformed the skewed numeric features(skew value is greater than 0.75) by taking log(feature + 1) - this will make the features more normal.
- Logarithm is applied to target value sale price as well.
- The categorical features are converted to numerical features for the models using get.dummies. The insignificant features hence produced are removed, which have many zeroes.
- Missing values are replaced by mean Value whereever left.
    - The outliers id's ([31, 463, 524, 633, 969, 971, 1299, 1325]) are calculated and the most significant ones - 523, 1298 are identified and dropped.
- More features are created using the chain and product of itertools. Different combinations of features are created and are appended to the final dataset. Finally we have 488 columns in the dataset so formed. The significance of each feature can be seen by plotting the graph.

Models

We have used 3 different types of models :-

Linear regression with Lasso Regularization - The linear regression is regularised using lasso regularization. The best parameters are computed using cross validation score. The idea is to try Lasso a few times on bootstrapped samples and see how stable the feature selection is. As we have mostly tuned the parameters and Lasso should perform better. So giving more weight to it .

XGBOOST - Xgboost model to our linear model to see if we can improve the score giving weight to its prediction

Elastic Net - Which is a regularized regression method that linearly combines the L1 and L2 penalties of the lasso and ridge methods.

Prediction

- No single model gave efficient outputs.
- We tried with combinations of different models using basic ensemble methods.
- Final prediction is done with giving different weights to the three models as w1*lasso_pred + w2*elastic_net + w3*xgboost where, w1+w2+w3 = 1
    - W1 = 0.50,W2 = 0.24,W3 = 0.26