HOUSE PRICE PREDICTION USING ADVANCED REGRESSION TECHNIQUES

Professor Simon Shim,
CMPE 257 - Machine Learning

Group Members:
Sohan Shirodkar
Dhruvin Shah
Tanay Ganeriwal
Rohan Deshmukh
Tejas Mahajan

TABLE OF CONTENTS

INTRODUCTION

Ask a home buyer to describe his dream house, and they will probably not start with the height of the basement ceiling or the proximity to an east-west railway. But the Kaggle Competition's House Prices - Advanced Regression Techniques (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview) dataset proves that much more influences price negotiations than numbers of bedrooms or a white picket fence. With 79 explanatory variables describing (almost) each aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.

Real estate agents try not to disappoint their buyers. It is not easy for an agent to describe the sales price of a buyer's dream home. All buyers have different requirements, which makes it difficult for the agent to predict the house price. Here, Machine Learning enthusiasts / engineers attempt to fasten and ease the process. To increase the productivity of an agent, we applied several advanced regression techniques to predict the sale price of the house. Our model consists of various preprocessing techniques, visualizations, and detailed analysis. After different permutations and combinations, we came to a model with a low RMSE value. (Lower RMSE value == Extensive model).

Linear and logistic regressions are usually the first go to in data science. Due to their popularity, important analysts even conclude to assume that they are the only form of regression. The ones who are a little more involved believe they are the most important of all forms of regression analysis.

The truth is that there are countless forms of regression that can be carried out. Each form has its own importance and a certain condition in which it is best applied. Regression analysis is a form of predictive modeling that examines the relationship between a dependent (target) and an independent variable (s) (predictor). This technique is used to predict, model time series, and find the causal effect relationship between the variables.

The use of regression analyses has several advantages. They are as follows:

- It shows the significant correlations between dependent variable and independent variable.
- It indicates the strength of the impact of several independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, including the effects of price changes and the amount of advertising activities. These advantages help market researchers / data analysts / data scientists to eliminate and evaluate the best types of variables that can be used to build predictive models.

What type of regression techniques do we have?

There are different types of regression techniques to generate predictions. Three metrics mostly drive these techniques (number of independent variables, type of dependent variables, and shape of regression line).

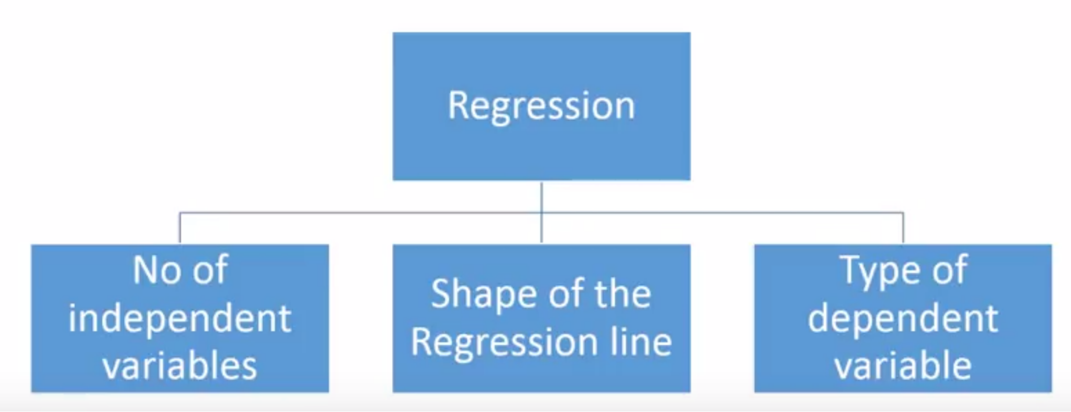

Regression
No of independent variables
Shape of the Regression line
Type of dependent variable

Figure 1: Types of Regression Techniques

Some of them advance regression techniques we have explored as below:

- Decision Tree Regressor
- Random Forest Regressor
- Gradient Boost Regressor
- XGBoost Regressor
- Bayesian Linear Regressor
- CatBoost Regressor
- Blending Model [XGB + LGBM]
- Stacking Regressor [Catboost + Bayesian]

LITERATURE REVIEW

Housing markets are unlike any other market (Smith, 2011b). The heterogeneity of the housing market, according to Meen (1996), makes it difficult to categorize: each house feature is different, and housing markets contain many varying factors, where no two houses are identical (considering the interior and exterior of the property). Cho (1996) provides evidence of the difficulty measuring a housing market, due to the heterogeneity of attributes that define houses. Housing markets are imperfect (Boelhouwer, 2011): there are many reasons for this, including recessions, which lead to delays in house prices and stagnation. According to Miles (2004), the UK housing market is widely considered a major transport mechanism for volatility in the UK economy.

If a household is applying for a housing loan, the bank and the household agree on a maximum fraction of the household's after-tax income available for mortgage repayment after other expenses are paid. The maximum is usually in the variety of 25 to 30%, depending on the country and economic situation at that time, but rarely exceeds 30% (Weicher, 1977; Hulchanski, 1995; Bourassa, 1996; Savage, 1999). From the household's income, fixed expenses, mortgage expenses and other information, the bank estimates the maximum obtainable loan and, therefore, the household's highest affordable price. As housing prices are fixed in the short run, the nominal affordability of the average house buyer and amounts of house buyers determine the average price of the house. The validity of this hypothesis is discussed in the next section.

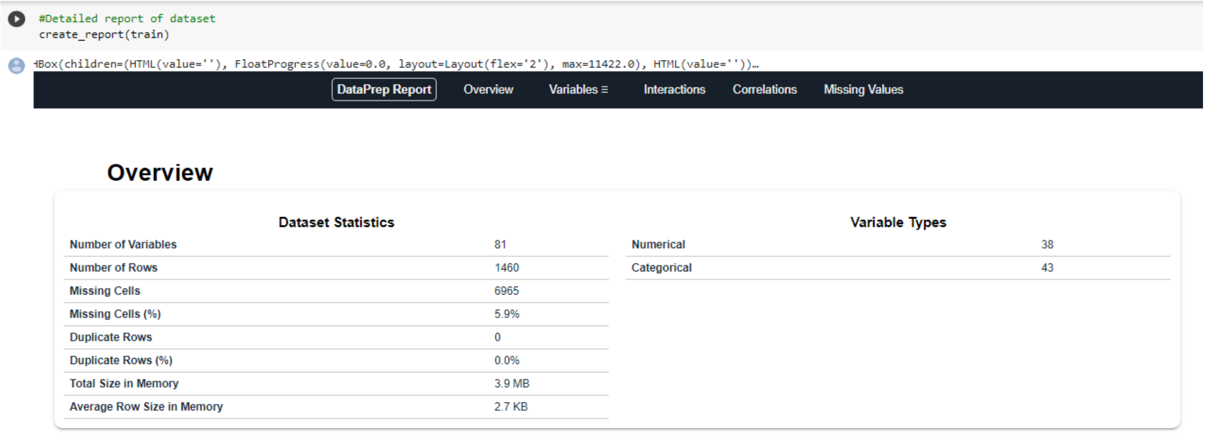
RELATED WORK

In the last two decades, forecasting the property value has become an important field. Significant research has been done on Artificial Neural Networks. This has helped many researchers focusing on real estate problem to solve using neural networks. In [4], the author has compared the hedonic price model and ANN model that predict the house prices. Some researchers like that in [5] have used classifiers to predict the property values. The author in the research article [5] has collected the data from Ames Housing Dataset. The author extracted approximately 2000 records from these sources, which included 91 variables. Subsequently, a test was used to select 47 variables as a preliminary screening. Some researchers have focused on feature selection and feature extraction procedure. The author in article [1] uses an open-source data set of the housing sales in King County, USA. There are about 20 explanatory variables. The author has compared various feature selection and feature extraction algorithms combined with Support Vector Regression. The author has collected approximately 210000 observations in a period of one year. The paper shows various data analysis performed on the data set.

Feature Selection is the process of selecting a subset of variables from a given set of parameters, either for their importance or their frequency. However, feature extraction is the process of reducing the dimensionality of the data. Initial set of data is transformed into derived values, which are equally informative and non-redundant. The three feature selection algorithms used are Recursive Feature Elimination (RFE), Lasso, Ridge, and Random Forest Selector, and the mean from each algorithm is calculated. Using feature selection, the author selects fifteen features out of twenty. The feature extraction algorithm used is Principal Component Analysis (PCA) and reduces the parameters from twenty to sixteen.

DATA EXPLORATION

The dataset is a subset of the Ames Housing dataset. The Ames Housing dataset consists of 2930 observations with 81 columns. Dataset contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010. The subset used in Kaggle competition consists of only 1460 observations. It consists of 38 numerical and 43 categorical features.



```
#Detailed report of dataset
create_report(train)
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, layout=Layout(flex='2'), max=11422.0), HTML(value='')))…
```

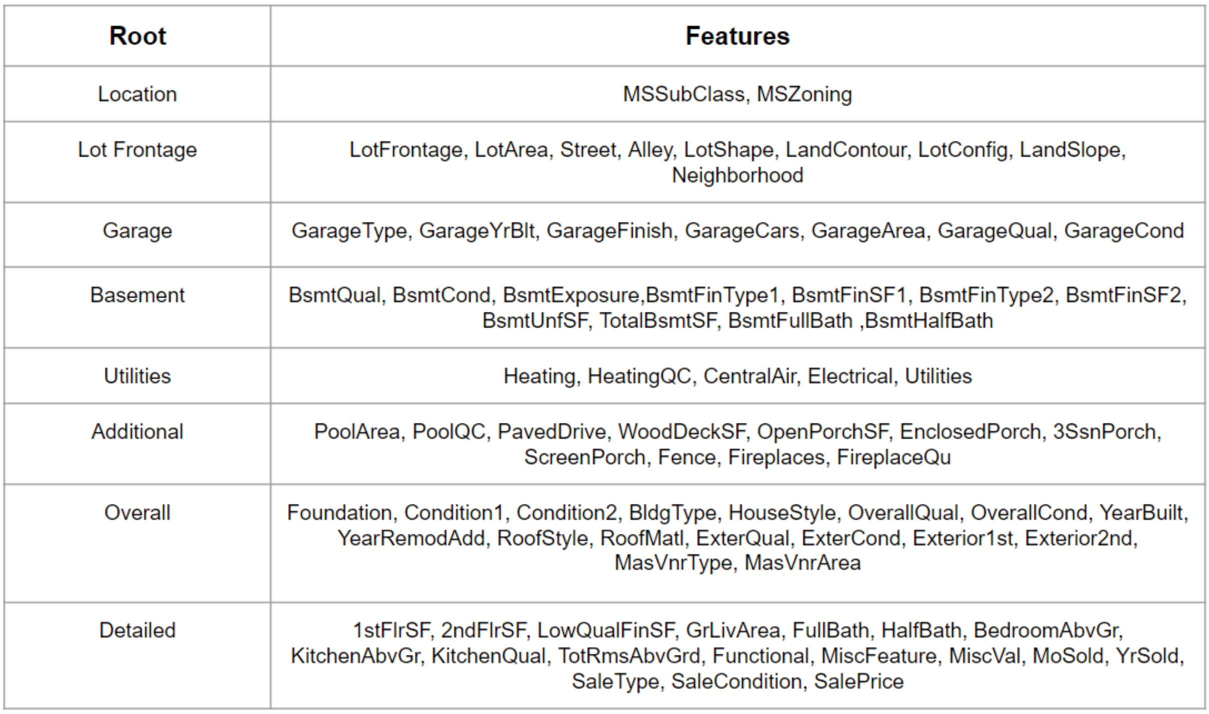
Overview

Dataset Statistics	
Number of Variables	81
Number of Rows	1460
Missing Cells	6965
Missing Cells (%)	5.9%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	3.9 MB
Average Row Size in Memory	2.7 KB

Variable Types	
Numerical	38
Categorical	43

Figure 2: Overview of Dataset

To understand the dataset better, clusters were created. Features were manually mapped to their root feature. This was beneficial in developing certain intuition. For instance, Overall Quality of the house should be very excellent (10, according to the dataset).



Root	Features
Location	MSSubClass, MSZoning
Lot Frontage	LotFrontage, LotArea, Street, Alley, LotShape, LandContour, LotConfig, LandSlope, Neighborhood
Garage	GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond
Basement	BsmtQual, BsmtCond, BsmtExposure,BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, BsmtFullBath ,BsmtHalfBath
Utilities	Heating, HeatingQC, CentralAir, Electrical, Utilities
Additional	PoolArea, PoolQC, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, Fence, Fireplaces, FireplaceQu
Overall	Foundation, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, ExterQual, ExterCond, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea
Detailed	1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, KitchenQual, TotRmsAbvGrd, Functional, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition, SalePrice

Figure 3: Data Description

DATA VISUALIZATION

Data visualization gives us a clear idea of what the information means by giving it visual context through maps or graphs.

We have implemented various types of visualization techniques:

- Pair Plot
- Word Cloud
- Heat Map
- Box Plot

1. Pair Plot

From the Pair Plot we can see that the bottom left triangle portion of the matrix is the same and the top right portion of the matrix, with the axes flipped. It displays the scatter plot for every feature against the corresponding feature. From the scatter plot we can infer the correlation of each column.
The diagonal row is nothing but simply a histogram of each variable and number of occurrences.

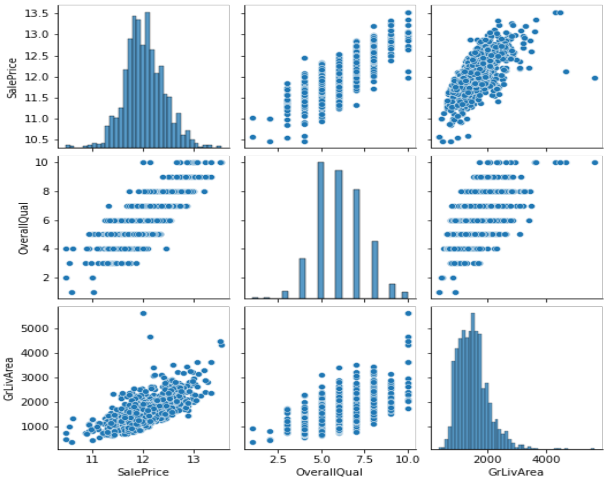
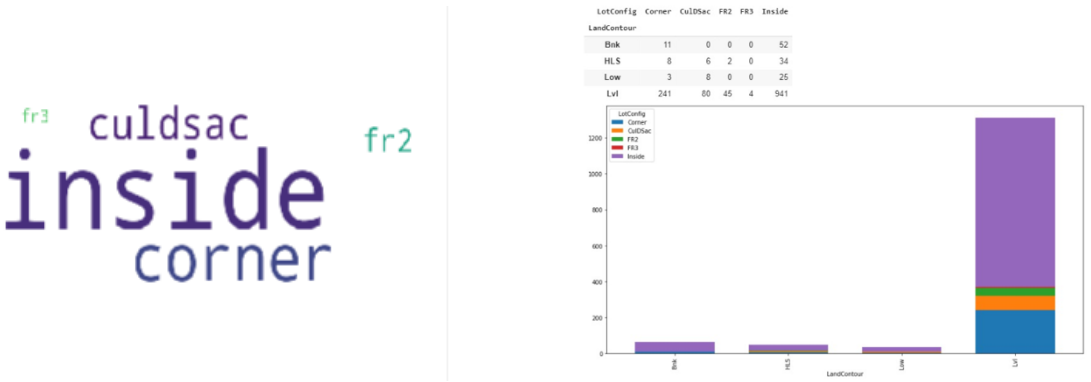


Figure 4: Pair Plot

2. Word Cloud

- Word Cloud is a simple way of data visualization.
- A word cloud is a collection, or cluster, of words depicted in different sizes.
- The bigger and bolder the word appears, the more often it's mentioned within a given text and the more important it is.
- From the below image we can see that the word "inside" has appeared 941 times for the column "LandContour".

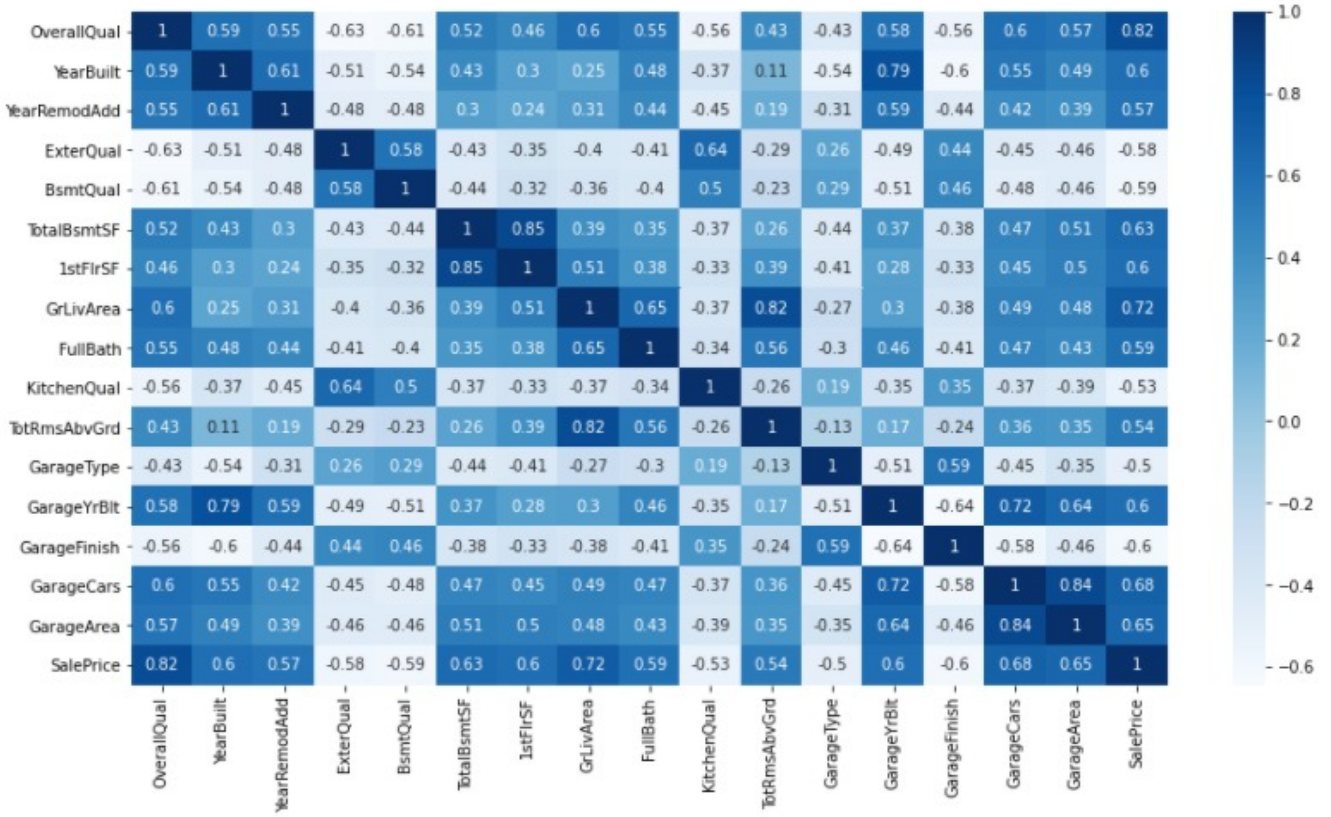


LotConfig LandContour	Corner	CulDSac	FR2	FR3	Inside
Bnk	11	0	0	0	52
HLS	8	6	2	0	34
Low	3	8	0	0	25
Lvl	241	80	45	4	941

Figure 5: Word Cloud

3. Heat Map

- The Heat Map uses a warm-to-cool color spectrum to show you which parts of a page receive the most attention.
- The features can be either positively correlated or negatively correlated from the target variable.
- Here the target variable is “SalePrice”.
- From the Heat Map below we can see that all the selected features has maximum correlation with the target variable.



	OverallQual	YearBuilt	YearRemodAdd	ExterQual	BsmtQual	TotalBsmtSF	1stFlrSF	GrLivArea	FullBath	KitchenQual	TotRmsAbvGrd	GarageType	GarageYrBlt	GarageFinish	GarageCars	GarageArea	SalePrice
OverallQual	1	0.59	0.55	-0.63	-0.61	0.52	0.46	0.6	0.55	-0.56	0.43	-0.43	0.58	-0.56	0.6	0.57	0.82
YearBuilt	0.59	1	0.61	-0.51	-0.54	0.43	0.3	0.25	0.48	-0.37	0.11	-0.54	0.79	-0.6	0.55	0.49	0.6
YearRemodAdd	0.55	0.61	1	-0.48	-0.48	0.3	0.24	0.31	0.44	-0.45	0.19	-0.31	0.59	-0.44	0.42	0.39	0.57
ExterQual	-0.63	-0.51	-0.48	1	0.58	-0.43	-0.35	-0.4	-0.41	0.64	-0.29	0.26	-0.49	0.44	-0.45	-0.46	-0.58
BsmtQual	-0.61	-0.54	-0.48	0.58	1	-0.44	-0.32	-0.36	-0.4	0.5	-0.23	0.29	-0.51	0.46	-0.48	-0.46	-0.59
TotalBsmtSF	0.52	0.43	0.3	-0.43	-0.44	1	0.85	0.39	0.35	-0.37	0.26	-0.44	0.37	-0.38	0.47	0.51	0.63
1stFlrSF	0.46	0.3	0.24	-0.35	-0.32	0.85	1	0.51	0.38	-0.33	0.39	-0.41	0.28	-0.33	0.45	0.5	0.6
GrLivArea	0.6	0.25	0.31	-0.4	-0.36	0.39	0.51	1	0.65	-0.37	0.82	-0.27	0.3	-0.38	0.49	0.48	0.72
FullBath	0.55	0.48	0.44	-0.41	-0.4	0.35	0.38	0.65	1	-0.34	0.56	-0.3	0.46	-0.41	0.47	0.43	0.59
KitchenQual	-0.56	-0.37	-0.45	0.64	0.5	-0.37	-0.33	-0.37	-0.34	1	-0.26	0.19	-0.35	0.35	-0.37	-0.39	-0.53
TotRmsAbvGrd	0.43	0.11	0.19	-0.29	-0.23	0.26	0.39	0.82	0.56	-0.26	1	-0.13	0.17	-0.24	0.36	0.35	0.54
GarageType	-0.43	-0.54	-0.31	0.26	0.29	-0.44	-0.41	-0.27	-0.3	0.19	-0.13	1	-0.51	0.59	-0.45	-0.35	-0.5
GarageYrBlt	0.58	0.79	0.59	-0.49	-0.51	0.37	0.28	0.3	0.46	-0.35	0.17	-0.51	1	-0.64	0.72	0.64	0.6
GarageFinish	-0.56	-0.6	-0.44	0.44	0.46	-0.38	-0.33	-0.38	-0.41	0.35	-0.24	0.59	-0.64	1	-0.58	-0.46	-0.6
GarageCars	0.6	0.55	0.42	-0.45	-0.48	0.47	0.45	0.49	0.47	-0.37	0.36	-0.45	0.72	-0.58	1	0.84	0.68
GarageArea	0.57	0.49	0.39	-0.46	-0.46	0.51	0.5	0.48	0.43	-0.39	0.35	-0.35	0.64	-0.46	0.84	1	0.65
SalePrice	0.82	0.6	0.57	-0.58	-0.59	0.63	0.6	0.72	0.59	-0.53	0.54	-0.5	0.6	-0.6	0.68	0.65	1

Figure 6: Heat Map

4. Box Plot

- A **boxplot** is a graph that gives you a good indication of how the values in the data are spread out.
- The boxplot is a standardized way of displaying the distribution of data based on a five-number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).
- It can tell you about your outliers and what their values are.
- From the below Box Plot, we see that “SalePrice” is plotted against “YearBuilt”. The figure also contains outliers that are clearly visible.

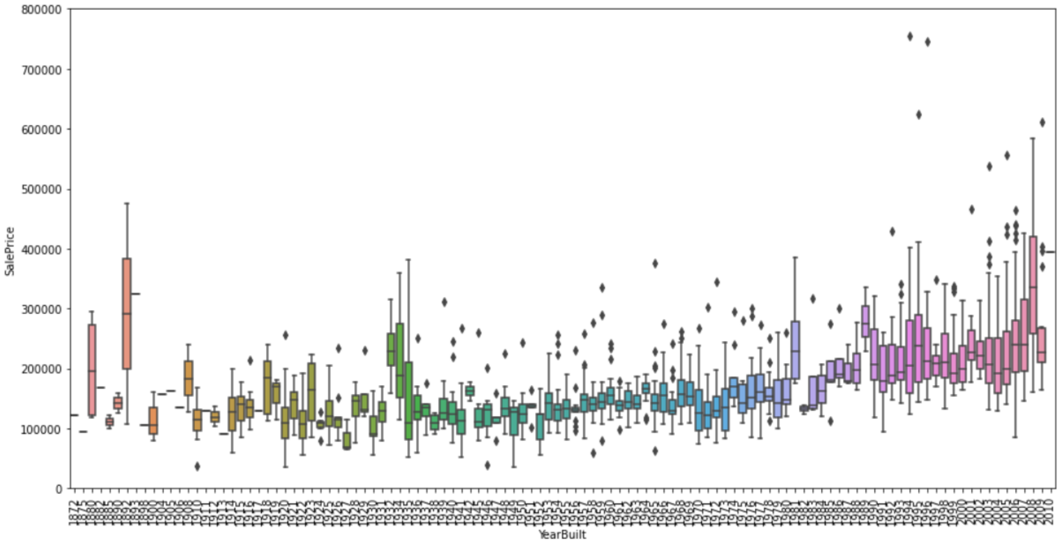
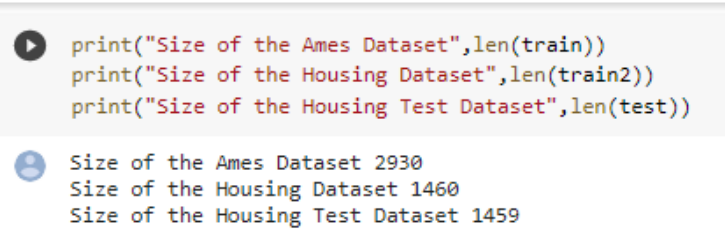


Figure 7: Box Plot

DATA PREPROCESSING

1) REDUNDANT VALUES:

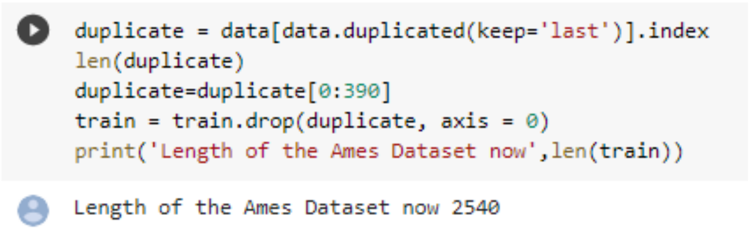
Dataset may incorporate information that are copies or nearly copies of each other. A significant issue when combining information from various, heterogeneous sources. Model: Same individual with various email addresses. Duplicate's ought to consistently be taken out, paying little mind to the model utilized. The cleaner the information, the better the outcomes. The screenshot includes the shape of the Ames Dataset before and after removal of duplicates rows is displayed below. It also gives an information regarding the duplicate values present in the Kaggle dataset (0 duplicate values).



```
print("Size of the Ames Dataset",len(train))
print("Size of the Housing Dataset",len(train2))
print("Size of the Housing Test Dataset",len(test))
```

```
Size of the Ames Dataset 2930
Size of the Housing Dataset 1460
Size of the Housing Test Dataset 1459
```

Figure 8: Before D-Duplication



```
duplicate = data[data.duplicated(keep='last')].index
len(duplicate)
duplicate=duplicate[0:390]
train = train.drop(duplicate, axis = 0)
print('Length of the Ames Dataset now',len(train))
```

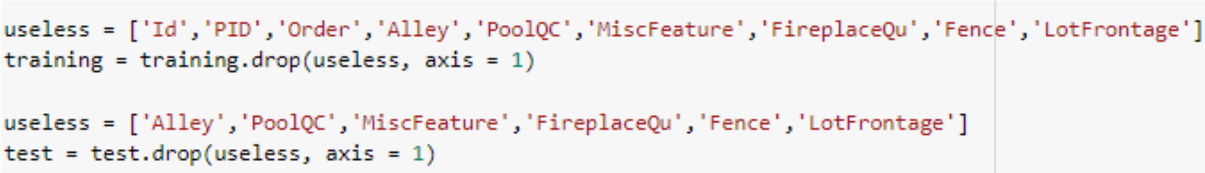
```
Length of the Ames Dataset now 2540
```

Figure 9: After D-Duplication

2) NULL VALUES:

Missing information presents different issues. To begin with, the shortfall of information lessens measurable force, which alludes to the likelihood that the test will dismiss the invalid theory when it is bogus. Second, the lost information can cause an inclination in the assessment of boundaries. Third, it can diminish the representativeness of the examples. Fourth, it might confound the investigation of the examination. Every one of these contortions may compromise the legitimacy of the preliminaries and can prompt invalid ends. Hence, it is compulsory to deal with missing/invalid/nan esteems. The techniques used are as follows:

A) Dropping columns which have more than 15% null values:

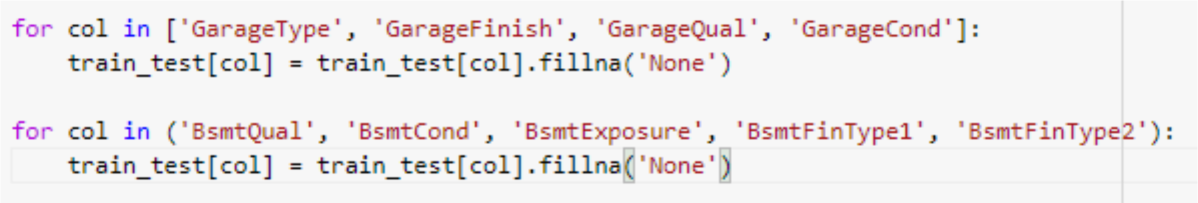


```
useless = ['Id','PID','Order','Alley','PoolQC','MiscFeature','FireplaceQu','Fence','LotFrontage']
training = training.drop(useless, axis = 1)

useless = ['Alley','PoolQC','MiscFeature','FireplaceQu','Fence','LotFrontage']
test = test.drop(useless, axis = 1)
```

Figure 10: Dropping Useless Columns

B) Filling with the string value, ‘None’:

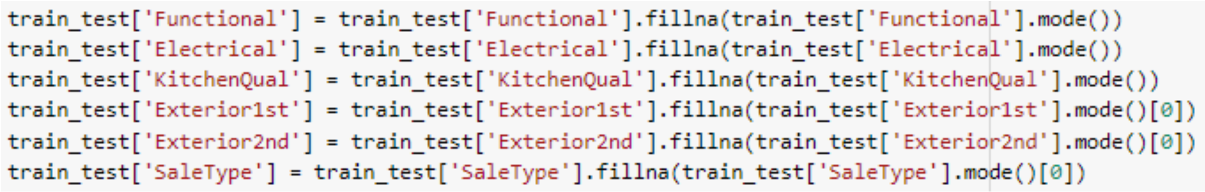


```
for col in ['GarageType', 'GarageFinish', 'GarageQual', 'GarageCond']:
    train_test[col] = train_test[col].fillna('None')

for col in ('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2'):
    train_test[col] = train_test[col].fillna('None')
```

Figure 11: Replacing with None

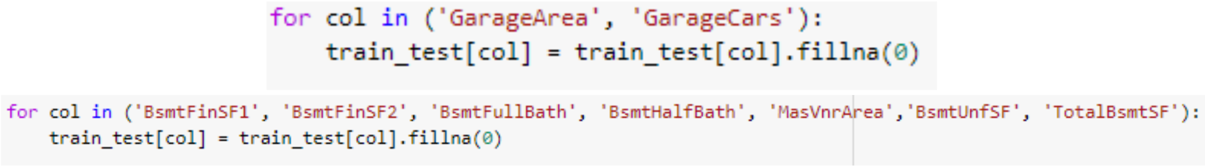
C) Filling with the mode of the column:



```
train_test['Functional'] = train_test['Functional'].fillna(train_test['Functional'].mode())
train_test['Electrical'] = train_test['Electrical'].fillna(train_test['Electrical'].mode())
train_test['KitchenQual'] = train_test['KitchenQual'].fillna(train_test['KitchenQual'].mode())
train_test['Exterior1st'] = train_test['Exterior1st'].fillna(train_test['Exterior1st'].mode()[0])
train_test['Exterior2nd'] = train_test['Exterior2nd'].fillna(train_test['Exterior2nd'].mode()[0])
train_test['SaleType'] = train_test['SaleType'].fillna(train_test['SaleType'].mode()[0])
```

Figure 12: Replacing with Mode

D) Filling with value 0:



```
for col in ('GarageArea', 'GarageCars'):
    train_test[col] = train_test[col].fillna(0)
```

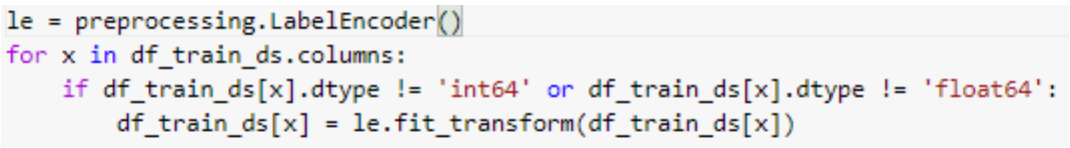
```
for col in ('BsmtFinSF1', 'BsmtFinSF2', 'BsmtFullBath', 'BsmtHalfBath', 'MasVnrArea','BsmtUnfSF', 'TotalBsmtSF'):
    train_test[col] = train_test[col].fillna(0)
```

Figure 13: Replacing with 0

3) ENCODING:

Machine learning models require all input and output variables to be numeric. This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model. Different techniques for encoding were used. They are as follows:

a) Label Encoding

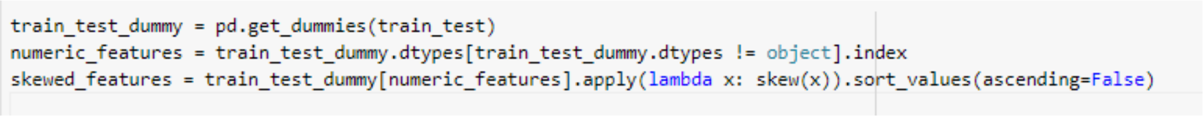


```
le = preprocessing.LabelEncoder()
for x in df_train_ds.columns:
    if df_train_ds[x].dtype != 'int64' or df_train_ds[x].dtype != 'float64':
        df_train_ds[x] = le.fit_transform(df_train_ds[x])
```

Figure 14: Label Encoding

b) Dummy Encoding

Dummy coding is used when there is a control or comparison group in mind. One is therefore analyzing the data of one group in relation to the comparison group: *a* represents the mean of the control group and *b* is the difference between the mean of the experimental group and the mean of the control group. It is suggested that three criteria be met for specifying a suitable control group: the group should be a well-established group (e.g., should not be another category), there should be a logical reason for selecting this group as a comparison (e.g., the group is anticipated to score highest on the dependent variable), and finally, the group's sample size should be substantive and not small compared to the other groups. In dummy coding, the reference group is assigned a value of 0 for each code variable, the group of interest for comparison to the reference group is assigned a value of 1 for its specified code variable, while all other groups are assigned 0 for that code variable.

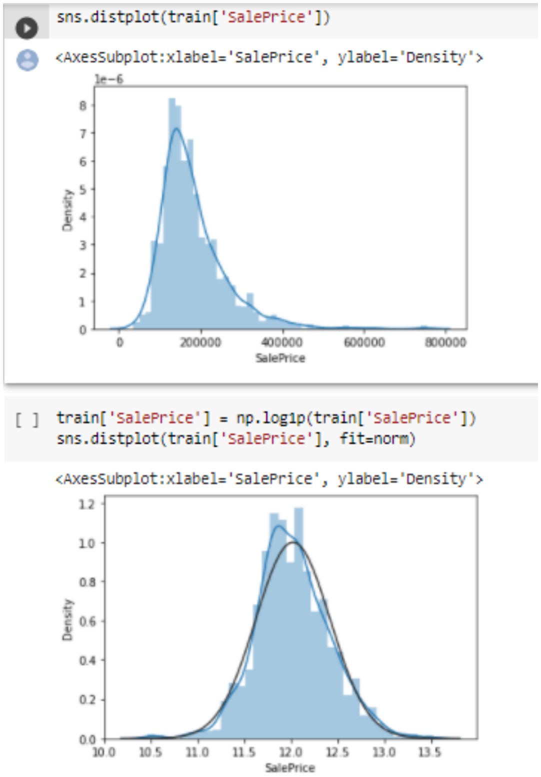


```
train_test_dummy = pd.get_dummies(train_test)
numeric_features = train_test_dummy.dtypes[train_test_dummy.dtypes != object].index
skewed_features = train_test_dummy[numeric_features].apply(lambda x: skew(x)).sort_values(ascending=False)
```

Figure 15: Dumming Encoding

4) DATA TRANSFORMATION (LOGARITHMIC TRANSFORMATION):

Among the numerical features, we chose to transform SalePrice and features, either log or Box-Cox transformations. We can clearly see that our target variable, Sale Price, is heavily skewed to the right, so we can apply a simple log transformation, so it follows a Gaussian distribution.



```
sns.distplot(train['SalePrice'])
```

```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```

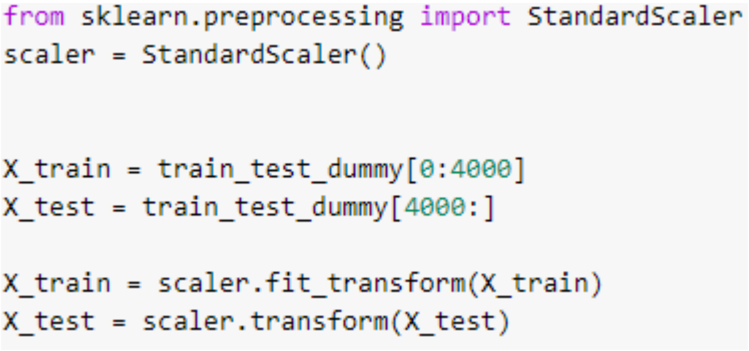
```
[ ]  train['SalePrice'] = np.log1p(train['SalePrice'])
     sns.distplot(train['SalePrice'], fit=norm)
```

```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```

Figure 16: Log Transformation

5) NORMALIZATION:

Standard Scaler- It removes the mean and scales each feature/variable to unit variance. This operation is performed **feature-wise** in an **independent** way. It can be influenced by **outliers** (if they exist in the dataset) since it involves the estimation of the empirical mean and standard deviation of each feature.



```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()


X_train = train_test_dummy[0:4000]
X_test = train_test_dummy[4000:]

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

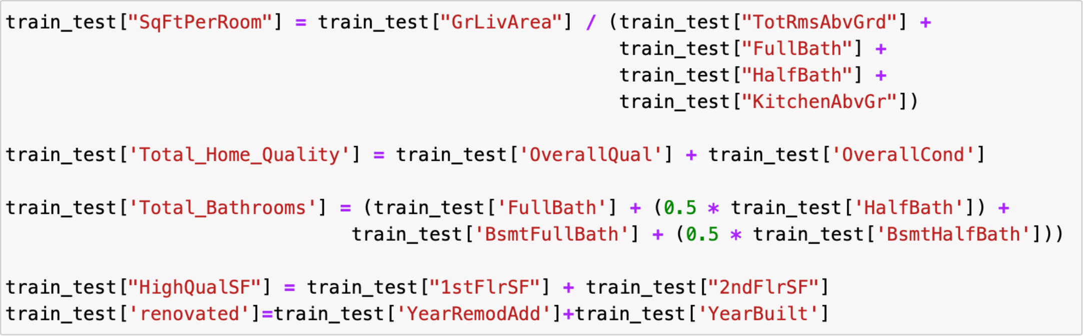
Figure 17: Normalization

FEATURE ENGINEERING

Feature engineering involves extraction of features from raw data along with the use of domain knowledge. Feature engineering is useful to improve the performance of machine learning algorithms and is often considered as applied machine learning. To avoid overfitting, sparse categorical variables are consolidated. We have established some new features using existing features.

For Example,

- SqFtPerRoom: - Established this feature by dividing Ground Living Area with the sum of Total Rooms Above Ground, Full Bath, Half Bath and Kitchen Above Ground.
- Total_Home_Quality: - Established this feature by adding Overall Quality and Overall Condition.
- Total_Bathrooms: - Established this feature by adding Full Bath, 50% of Half Bath, Basement Full Bath and 50% of Basement Half Bath.
- HighQualSF: - Established this feature by adding 1st Floor Square Feet and 2nd Floor Square Feet.
- renovated: - Established this feature by adding Year Remodeled and Year Built.



```
train_test["SqFtPerRoom"] = train_test["GrLivArea"] / (train_test["TotRmsAbvGrd"] +
                                                       train_test["FullBath"] +
                                                       train_test["HalfBath"] +
                                                       train_test["KitchenAbvGr"])

train_test['Total_Home_Quality'] = train_test['OverallQual'] + train_test['OverallCond']

train_test['Total_Bathrooms'] = (train_test['FullBath'] + (0.5 * train_test['HalfBath']) +
                               train_test['BsmtFullBath'] + (0.5 * train_test['BsmtHalfBath']))

train_test["HighQualSF"] = train_test["1stFlrSF"] + train_test["2ndFlrSF"]
train_test['renovated']=train_test['YearRemodAdd']+train_test['YearBuilt']
```

Figure 18: Feature Engineering

PROBLEM FORMULATION AND MODEL SELECTION

We used various machine learning models with a combination of different data pre-processing techniques discussed earlier. Along with these standalone models, we experimented with blending and stacking techniques, ultimately giving us the best results with stacking. With combination of various pre-processing techniques used in different models we got the best accuracy.

- Decision Tree Regressor:
  The Decision Tree Regressor was used along with GridSearchCV to find best parameters for the model, where a max depth of 8 proved to get better results with Decision Tree. The pre-processing for this model implementation consisted of filling the categorical features with mode and numerical features with 0. For converting the categorical variables to numeric we used get_dummies method. Thus, Decision Tree Classifier gave an RMSE of 0.20.

- Random Forest Regressor:
  In Random Forest Regressor implementation, features with more than 15% of null values were dropped, whereas a few remaining null values were filled with mode and 'None' for numeric and categorical features, respectively and used get_dummies method for encoding the categorical features. This model, with a max depth of 10, gave an RMSE of 0.15 on test set.

- Gradient Boosting Regressor:
  For Gradient Boosting Regressor, features with 95% or more null values were dropped and a combination of different techniques like median, forward fill, zero and 'None' were used to fill the remaining null values. The categorical variables were encoded using Label Encoder. This model achieved an RMSE of 0.15483.

- XGBoost Regressor:
  The pre-processing for the XGB Regressor included removing the features with more than 50% of null values, filling the remaining null values with a combination of different techniques. Further, to remove skewness, log transformation was used as well as PCA was used to reduce dimensionality. The XGB Regressor achieved an RMSE 0.1296.

- Blending (XGBoost Regressor & LightGBM Regressor):
  Blending technique was used to combine the predictions from XGB Regressor and LGBM Regressor. The pre-processing before using this technique consisted of dropping features with more than 95% null values and filling the remaining null values with median and mode. All the categorical variables were encoded using Label

Encoder. The blending technique used the output predictions from XGB and LGBM to make final predictions which achieved an RMSE of 0.13449.

- Stacking Regressor (Bayesian Ridge Regressor & CatBoost Regressor):
  The best RMSE score of 0.05 was achieved with the Stacking technique using Bayesian Ridge and Cat Boost Regressor. Stacking technique uses meta models where an output of one model is used to train another model and ultimately reach final prediction. With stacking we also used feature engineering to create new meaningful features. Thus, Bayesian Ridge and Cat Boost Regressor were used to fit the training data and the output from these was provided to a meta regressor for which Cat Boost regressor was used to make final predictions, ultimately getting the best score.

RESULT ANALYSIS

No.	Model	Result on Test Set	Result on Kaggle (RMSE)	Time Taken (seconds)
1.	Decision Tree Regressor	0.2056 (RMSE)	0.201	1.529
2.	Random Forest Regressor	89% (Score)	0.152	1.168
3.	Gradient Boost Regressor	89% (Score)	0.150	0.169
4.	XGBoost Regressor	0.12 (RMSE)	0.148	11.521
5.	Bayesian Linear Regressor	0.11 (RMSE)	0.141	8.319
6.	CatBoost Regressor	0.087 (RMSE)	0.138	12.689
7.	Blending Model [XGB + LGBM]	91% (Score)	0.134	13.957
8.	Stacking Regressor [Catboost + Bayesian]	0.0827 (RMSE)	0.053	25.638

- This section illustrates the results obtained with various settings from the most basic approach to the most advanced used in the project work. Thus, the section also corroborates the need for the experiments discussed and how they help in improving the model.

- Various models have been applied and tested against the test dataset to predict the price of residential homes using a plethora of features. The effectiveness of the models is judged by employing the metrics described below.

- Various preprocessing techniques were implemented on the dataset prior to applying the models. Finally, the best results were obtained by applying logarithmic transformations to remove skewness, performing feature engineering, filling null values, applying a OneHotEncoding technique and normalizing the dataset.

- The coefficient of determination or R2 score is calculated as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$.

- The mean squared error regression loss is calculated as:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_\text{samples}} \sum_{i=0}^{n_\text{samples}-1} (y_i - \hat{y}_i)^2.$$

- As expected, models like Gradient Boosting Regressor, Decision Tree Regressor and Random Forest Regressor provided mediocre results. Evidently, they were outperformed by CatBoost Regressor, LightGBM Regressor, Bayesian Linear Regressor and XGBoost Regressor. The model which resulted in providing the best RMSE was a Stacking Regressor which was a combination of CatBoost Regressor and Bayesian Ridge Regressor. The predictions generated by the Stacking Regressor were ranked **220 out of about 9,600 people in the Kaggle competition**. Thus, it was inferred that the performance of the Stacking Regressor excelled over the other regression models.

CONCLUSION AND FUTURE WORK

As the target variable ‘Sale Price’ was analyzed with most correlated variables and dealt with missing data and outliers, it took many models to reach the best one. Techniques like Blending and Stacking decrease the RMSE regression loss which makes the model better. After implementing numerous models, we finally concluded that the Stacking Regressor with a combination of Bayesian Ridge Regressor and CatBoost Regressor gave us the best results. As we achieved a **Kaggle rank of 220 out of around 9,600 people**, we chose this one as our best model.

With the help of the machine learning algorithms used in our project, we hope to aid the real estate agents using in order to help them in reducing the difficulty of predicting the prices of residential homes of the customers and display the dream homes according to the customers’ requirements.

This project is a fantastic learning experience that allows us to complete the entire process of building a machine learning model to solve a practical regression problem in the real world, from the beginning of data analysis, cleaning, preparation, etc., until the model stacking, evaluation, and delivery in the end. We touched on, thought about and finally either solved or acquired valuable knowledge / understanding of many major or minor practical problems, a very rewarding process. In the end, we all agreed and realized that feature engineering is often one of the most critical parts or differentiators on the final model performance.

Due to limited time, we have recognized, but have no time to examine many of the fascinating / promising directions / ideas, which are summarized as follows for an excellent reference in the future:

- As our dataset has relatively less datasets, the biggest thing we can do is scraping or imputing data for larger states, not for Iowa, but for the entire states with all different counties.

- Consider other advance models: XGBoost, SVR, Stacked model with Catboost and XGBoost with hyper-parameter tuning on both, and applying grid search CV etc., and tune with Bayes Optimizer.

- We don't need to convert categorical variables to dummy variables for tree-based models (H2O RF, etc.)

- We can try using different feature selection for different models: i.e., only dropping features for linear models, but not for tree-based non-linear models

- Examine more preprocessing choices, including BoxCox transformation, PCA, etc. For PCA, we may use the cross-correlation result among all the numerical variables, as shown earlier, to find highly correlated groups of variables, and only PCA on them and observe.

- Outlier check and removal

- Clustering analysis to generate new profitable categorical features.
- Feature selection: evaluate other advanced algorithms, e.g., Genetic algorithm, and simulated annealing, from the R-Caret package.

REFERENCES

[1] Bryant Homes, A Bryant Homes case study: Pricing the product http://businesscasestudies.co.uk/bryanthomes/pricing-the-product/conclusion.html. Accessed: May 18th, 2017.

[2] Bruce Archambeault, Ph.D., EMC Society Technical Advisory Committee Chair: The Importance of Technical Paper Writing, 2010

[3] John Yinger and Phuong Nguyen-Hoang : Hedonic Vices - Fixing Inferences about Willingness-To-Pay in Recent House-Value Studies

[4] Gaolu Zou Chengdu Economic and Development Institute of Conventions and Exhibitions, Chengdu University, Chengdu 610106, Sichuan Province, China : The Effect of Central Business District on House Prices in Chengdu Metropolitan Area: A Hedonic Approach

[5] Christos S. Savva, Department of Commerce, Finance and Shipping, Cyprus University of Technology : House Price Dynamics and the Reaction to Macroeconomic Changes: The Case of Cyprus

[6] Paul Emrath, Ph.D, Vice President for Survey and Housing Policy Research Economics and Housing Policy National Association of Home Builders : Breaking Down House Price and Construction Costs

[7] Stephan Abraham : Rental Properties: 4 Ways to Value a Real Estate Property

[8] House Price Factors, http://www.homeguru.com.au/house-prices

[9] RapidMiner, https://rapidminer.com/

[10] https://www.kaggle.com/c/house-pricesadvanced-regression-techniques (data source)