House Price Prediction
Capstone Project Report

Neeraj Walia

Abstract

House search for a first time home buyer is a daunting task. It involves research for house, talk to realtors, applying for bank loan, getting finances in order. There is tremendous amount of data available related to any house and its past sales history. In this project, I want to tackle "research for house" part of the home buying experience and make it easier for the home buyer to do their analysis and narrow down the choices to few neighborhoods or houses to pick from.

Introduction

As a home buyer, there are many tools (or websites) available where you can search for the houses that are already in market and check the asking price. But there is a need for a tool that will help home buyer research homes and predict the price of the house based on their wish list. It could be combination of requirements that user may have like location, number of bedrooms, bathrooms, single family or townhouse, square footage etc.
This project is to provide an easy to use tool to home buyers where they can enter their requirements and find out potential prices for the house. Using this tool they can compare their budget with the predicted price of the house and make adjustment to prioritize their requirements.

Context

I am building a website which is very visual and easy to understand. The website is divided into two parts "history of homes sold" and "Potential prices of houses not currently in market (predictive)".
Technology am using is python for data exploration, analysis, building model for prediction. For website building am using Tableau Public, it's a free service available for up to 10GB of data. It is a few restrictions for free version as which data sources we can connect to but for this project it will work. Tableau is picked for its impressive visualizations.
Data is available on kaggle website under the following link
https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

Question

The goal of this website is to predict the prices of the houses before they are put on sale by their owners. This can be based on various factors like neighborhoods, sq. footage, number bedrooms, bathrooms, house condition, year built, garage, type of roof, type of air conditioning etc. There are over 80 variables to pick from and make prediction.

Data

There are two sets of data available train and test set. Both have 1460 records in them. Each row represents a house and its information. Train set has 81 features including Sales price (target) which we

will use to train our model. Test data set have 80 features and we will make prediction the houses in this data set.
Data is available on kaggle website. And the link is provided above under context section. Data cover all details about the house. There are total 81 features and details everything related to the house. It has both categorical and numerical features. There are 37 numerical and 44 categorical features.
Lots of effort has been gone into translating categorical features into integer without changing its weightage. So features translate equally when they were as categorical as well as numerical

Below is the brief description about each feature. More detailed description is at the end of this report under data dictionary section.

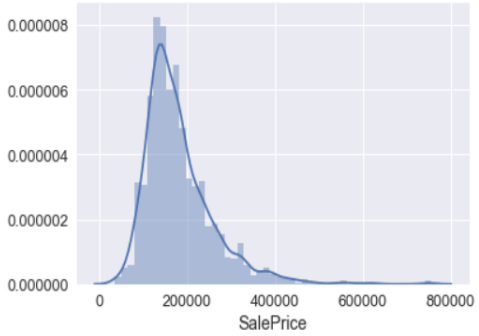
MSSubClass: type of dwelling like DUPLEX - ALL STYLES AND AGES, 2 FAMILY CONVERSION - ALL STYLES AND AGES, 2-STORY PUD - 1946 & NEWER etc.
MSZoning: zoning classification like agriculture land, commercial, Floating Village Residential etc.
LotFrontage: linear feet of street connected to property
LotArea: Lot size in square feet.
Street: type of road access to property like paved gravel.
Alley: type of alley access to property like no alley access, paved, gravel.
LotShape: shape of property like regular, slightly irregular etc.
LandContour: flatness of the property like near flat/level, banked, hillside, low etc
Utilities: type of utilities available like all public, noSewr, electric only etc.
LotConfig: Lot configuration like inside lot, corner, cul-de-sac
LandSlope: Slope on property like gentle, moderate, sever etc
Neighborhood: Physical location name of street or location
Condition1: proximity to various locations like artery, adjacent to feeder street, normal etc.
Condition2: proximity to various conditions like adjacent to arterial street, normal etc.
BldgType: type of dwelling like single family, duplex etc.
HouseStyle: 1 story or 2 story etc
OverallQual: Rates the overall material and finish of the house like excellent, good, average, poor etc.
OverallCond: Rates the overall condition of the house like excellent, good, average, poor etc.
YearBuilt: Original construction date
YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)
RoofStyle: Type of roof like gable, gabrel, hip etc
RoofMatl: Roof material
Exterior1st: exterior covering on house
Exterior2nd: Exterior covering on house if more than one
MasVnrType: Masonry veneer type like brick common, brick face etc
MasVnrArea: Masonry veneer area in square feet
ExterQual: Evaluates the quality of the material on the exterior
ExterCond: Evaluates the present condition of the material on the exterior
Foundation: type of foundation like Brick & Tile
BsmtQual: Evaluates the height of the basement like Excellent (100+ inches), Good (90-99 inches), Typical (80-89 inches), Fair (70-79 inches), Poor (<70 inches), No Basement
BsmtCond: Evaluates the general condition of the basement
BsmtCond: Evaluates the general condition of the basement
BsmtExposure: Refers to walkout or garden level walls
BsmtFinType1: Rating of basement finished area
BsmtFinSF1: Type 1 finished square feet
BsmtFinType2: Rating of basement finished area (if multiple types)
BsmtFinSF2: Type 2 finished square feet
BsmtUnfSF: Unfinished square feet of basement area
TotalBsmtSF: Total square feet of basement area
Heating: Type of heating like floor, GasA, GasW, Grav
HeatingQC: Heating quality and condition
CentralAir: Central air conditioning
Electrical: Electrical system
1stFlrSF: First Floor square feet
2ndFlrSF: Second floor square feet
LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet
BsmtFullBath: Basement full bathrooms
BsmtHalfBath: Basement half bathrooms
FullBath: Full bathrooms above grade
HalfBath: Half baths above grade
Bedroom: Bedrooms above grade (does NOT include basement bedrooms)
Kitchen: Kitchens above grade
KitchenQual: Kitchen quality
TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
Functional: Home functionality (Assume typical unless deductions are warranted)
Fireplaces: Number of fireplaces
FireplaceQu: Fireplace quality
GarageType: Garage location
GarageYrBlt: Year garage was built
GarageFinish: Interior finish of the garage
GarageFinish: Interior finish of the garage
GarageArea: Size of garage in square feet
GarageQual: Garage quality
GarageCond: Garage condition
PavedDrive: Paved driveway
WoodDeckSF: Wood deck area in square feet
OpenPorchSF: Open porch area in square feet
EnclosedPorch: Enclosed porch area in square feet
3SsnPorch: Three season porch area in square feet
ScreenPorch: Screen porch area in square feet
PoolArea: Pool area in square feet
PoolQC: Pool quality
Fence: Fence quality
MiscFeature: Miscellaneous feature not covered in other categories
MiscVal: $Value of miscellaneous feature
MoSold: Month Sold (MM)
YrSold: Year Sold (YYYY)
SaleType: Type of sale
SaleCondition: Condition of sale

Analysis

Sales price of the house is the target of our analysis and its distribution is as follows.

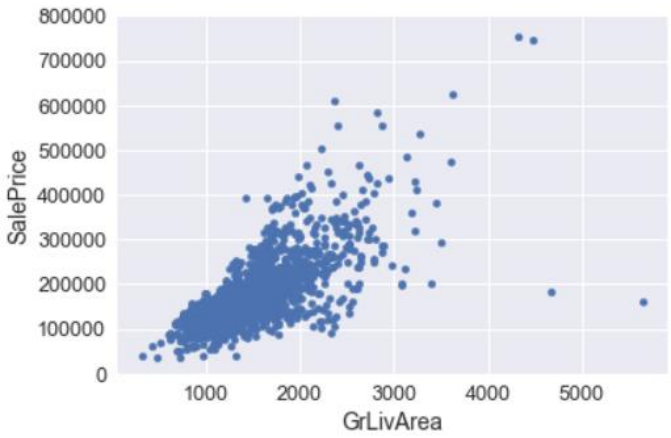
count	1460.00
mean	180921.19
std	79442.50
min	34900.00
25%	129975.00
50%	163000.00
75%	214000.00
max	755000.00



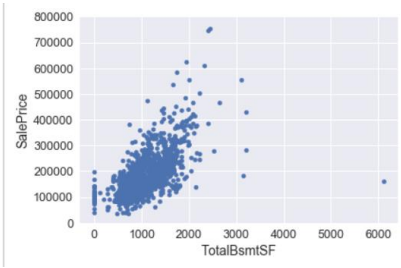
The skewness of the data is 1.88 meaning it is highly skewed. We could use log transformation to make it normally distributed.

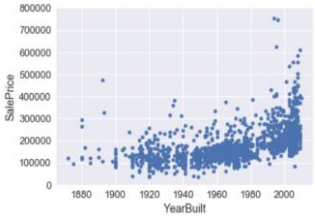
We will plot lots of scatter plot to see relation between sales and each individual feature to see their relationship.

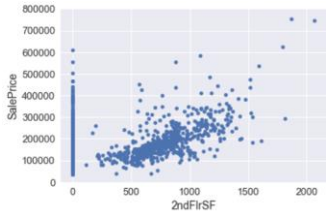
Above ground living area in square feet has a direct relationship with Sales Price. Higher the area higher is the sales price so it looks like a linear relationship.

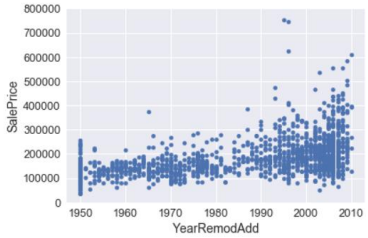


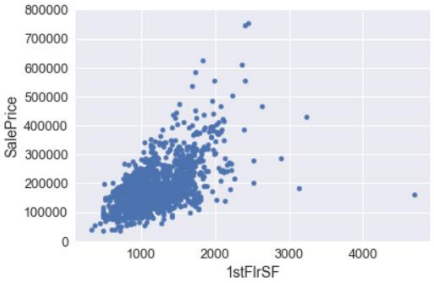
Similarly basement square feet has a direct relation to sales price

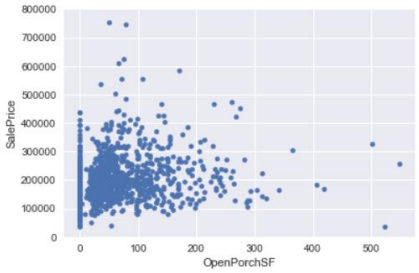




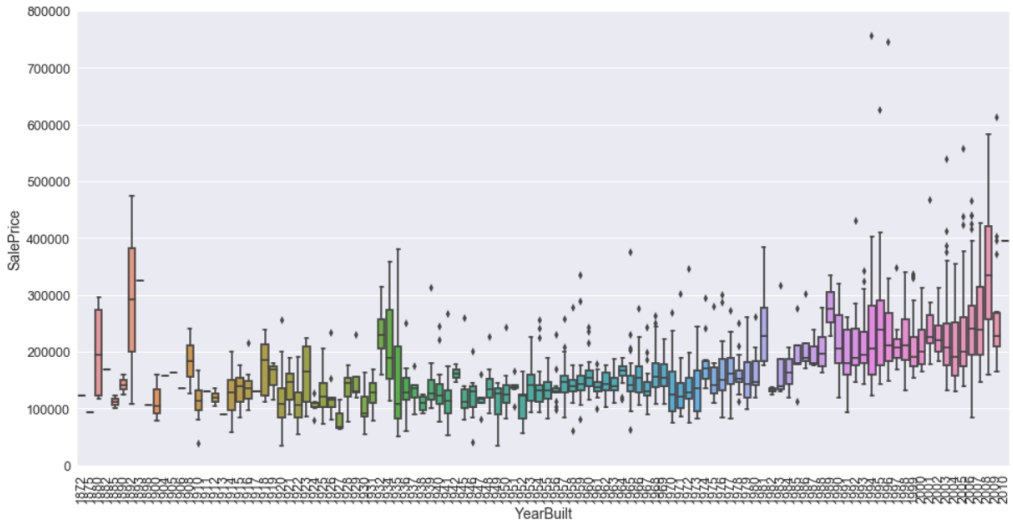






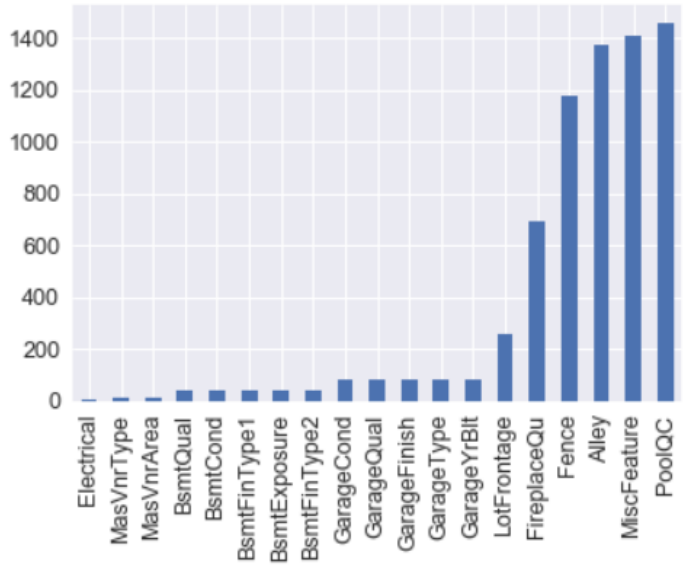


Following graph shows year when the house was build vs sale price. Price of older homes and newer homes has wide range of sales price. This also depends on other features like living square footage, condition of the house etc.



Missing Values: There are many NA values in our data set as shown by the graph below. But upon closer looks it shows that NA has specific meaning for that particular feature. For example, PoolQC there are over 1400 values with NA but that means no pool in the property.

NA values in Dataset



MiscFeature feature "miscellaneous feature not covered in other categories" defines NA as none. Similarly, Alley NA is defined as no alley access. Therefore, we need to keep these values as this provides information about the house.

Few features are removed from both Train and Test dataset either because of large NA values or their information was already available in other features. Features that are removed are LotFrontage, MasVnrType, MasVnrArea and GarageYrBlt.

I am going to convert categorical values into numerical by using one_hot_encoding method from sklearn. This transformer converts categorical feature into one hot numeric array. So, this transformer assigns binary integer to distinct categorical values in each feature and creates a new column for it. For example:

Old Alley **New columns** to identify three values of Alley

Alley
Grvl
NA
Pave

Alley_Grvl	Alley_Pave
1	0
0	0
0	1

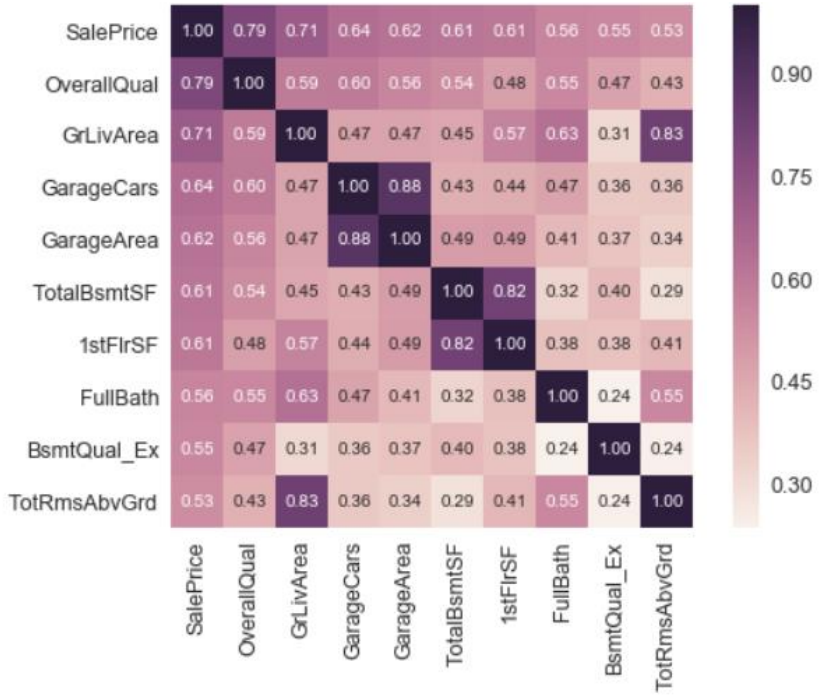
Final train data set after One_hot_encoding transformation was applied has 282 Columns. Now all categorical values are represented as numerical values. We also need Test dataset to have same features as train model for prediction model to work. Issue is test does not have all the data as of train dataset. For example as shown above "Alley" may not have any data for Paved alley in the test dataset. And due to this scenario Alley_Pave column will be missing from final test dataset. To avoid this situation am going to do left join with train data so all columns that are there in train will be in test as well with NULL values.

```
one_hot_encoded_training_predictors = pd.get_dummies(df_train)
one_hot_encoded_test_predictors = pd.get_dummies(df_test)
final_train, final_test = one_hot_encoded_training_predictors.align(one_hot_encoded_test_predictors, join='left',  axis=1)
```

And for the new columns that are added in test dataset, am going to replace NULLS with zeros as would imply that those feature is not present in the house.

```
#fill missing values in final_test with 0 since those features are not available
final_test.fillna(0, inplace = True)
```

Below is the heat map for top 10 variables in correlation matrix from final train data



Feature Selection

At the time of model building, training the model on a sample data and use prediction function to predict same data leads to mistakes called **overfitting**. To avoid this situation data should be data

should be split into test and train data, so model is trained on train data and tested on test dataset. But the percent of split between train and test data is still controlled manually and can be tweaked to receive favorable results. This problem can be solved by solution called cross-validation, this approach is based on k-folds cross validation. Data is split into k samples and model is trained on (k-1) samples. Resulting model is tested against remaining data. This process is repeated multiple times and performance is measured by the average of the values computed in the loop.
I will be using sklearn cross-validation cross_val_score function for feature selection in this project.

```
cross_validation.cross_val_score(model_name,X,y,cv=5, scoring='mean_squared_error')
X: Original Data (final train in our case)
Y: target data (Sale Price)
cv: Number of times cross validation will repeat
scoring = Mean squared error
```

Linear regression

I am going to build linear regression model because as discussed above there are linear relations between Sales price and independent variables.

```
lm = LinearRegression()
lm.fit(X_train, y_train)

score =cross_validation.cross_val_score(lm,X,y,cv=5, scoring='mean_squared_error')
mse_score = -score
rmse_score = np.sqrt(mse_score)
rmse_score.mean()
```

```
Root mean square of Linear regression is 35154.66
```

Random forest Regressor

Random forest is an estimator that fits classifying decision tree on various samples of data and uses average to improve the predictive accuracy and control overfitting.

```
rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)
rf.fit(X_train, y_train)
```

n_estimator is the number of trees in random forest.
Random_state is the see used by the random number generator.
Fit builds the random forest regressor model for the train data.

```
rf.predict(final_test)
```

Predicts the data for the final test data set.

rf = RandomForestRegressor(n_estimators = 1000, random_state = 42)
# Train the model on training data

```
rf.fit(X_train, y_train)
score1 =cross_validation.cross_val_score(rf,X,y,cv=5, scoring='mean_squared_error')
mse_score1 = -score1
rmse_score1 = np.sqrt(mse_score1)
rmse_score1.mean()
```

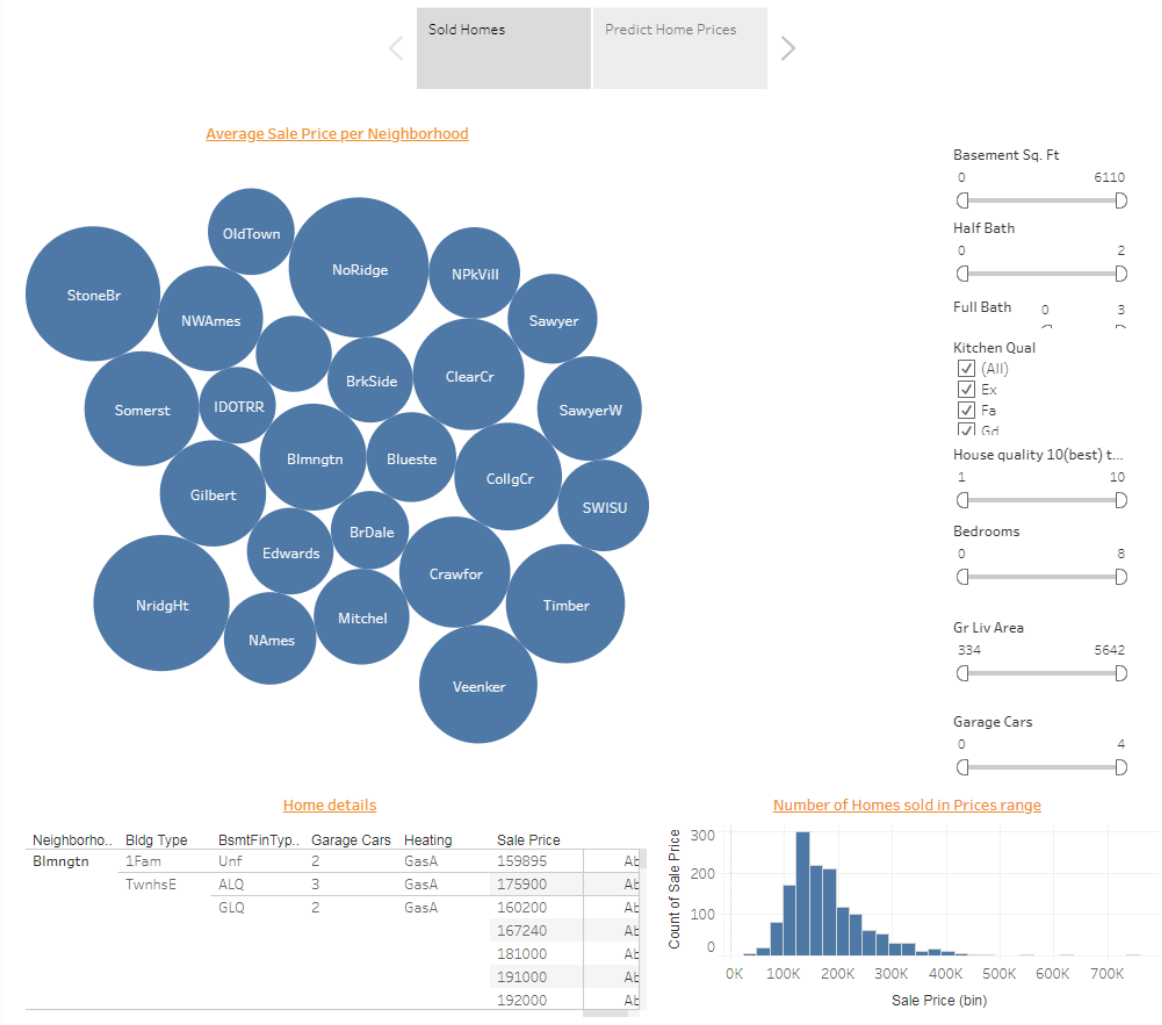
Here we will build the random forest regressor model with 100 estimators, fit the model with training data. Run cross validation cross_val_score function five times on the random forest regressor model on training data. Calculate the Root mean square error.

```
Root mean square error of Random Forest Regressor is 29921.69
```

The Random Forest Regressor model is the winner of this analysis with lower root mean square error. The prediction nfor the final_test data is going to be made based on this this model.

Website

Using the historical data (train dataset) and new predictive model (test dataset), a new website is developed and its link is given below. I have used Tableau for the design, development and hosting of the website.
https://public.tableau.com/views/Capstone_73/AmnesHousing?:embed=y&:display_count=yes&:toolbar=no



Data Dictionary:

Attached is the data dictionary file for detailed description of each feature.

data_description Detail.txt