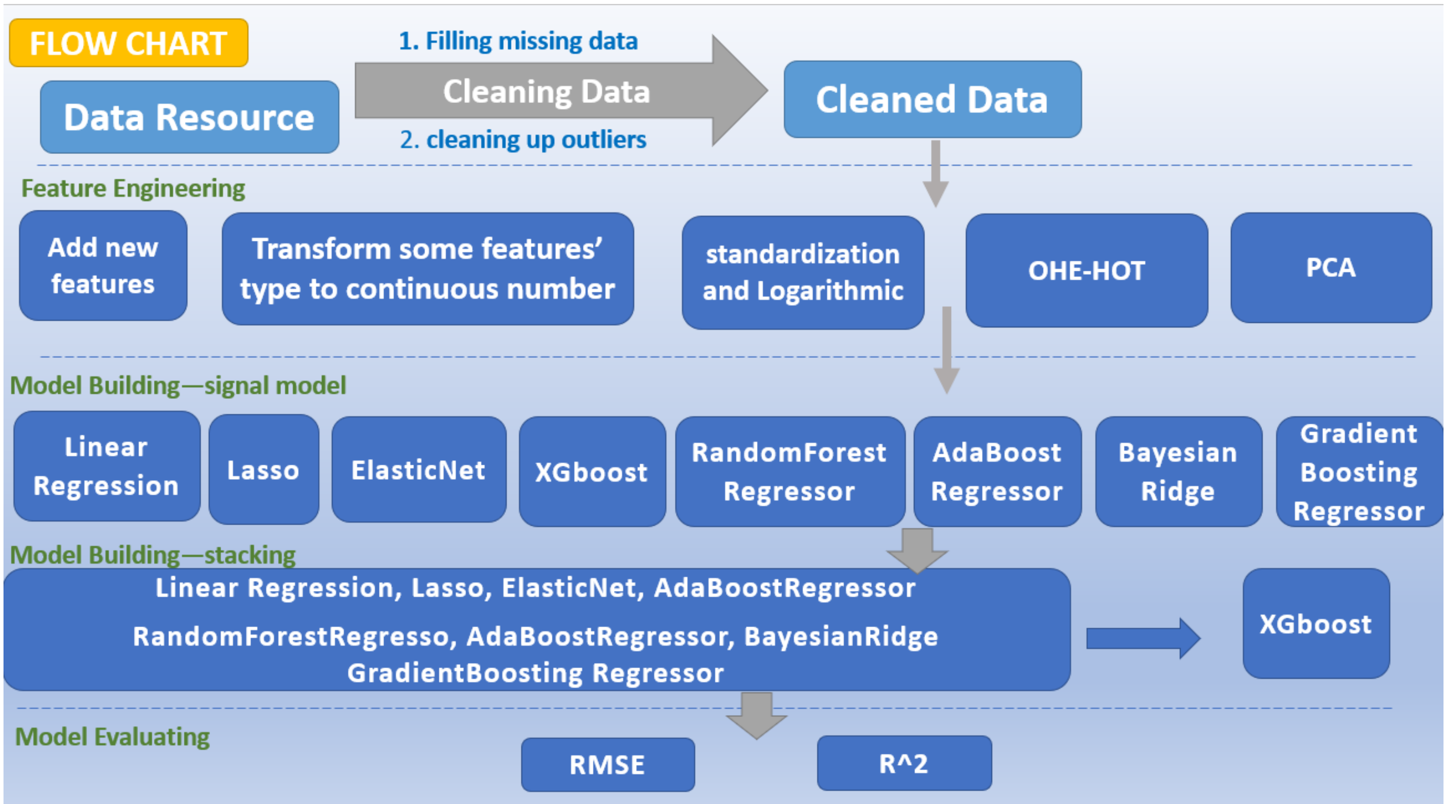
# Data Analytics Report of House Price

Siheng Huang siheng_huang@163.com




## Ⅰ Data source and dataset Introduction

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

This dataset contains 79 features, and the types of them are numeric are character-type discrete values. With analyzing the dataset, we can predict the price of house which only has some information.

For more, please refer to https://www.kaggle.com/c/house-prices-advanced-regression-techniques

## Ⅱ Exploration, Statistics, and Visualization

### a. about the numeric values

Considering to draw their heat map

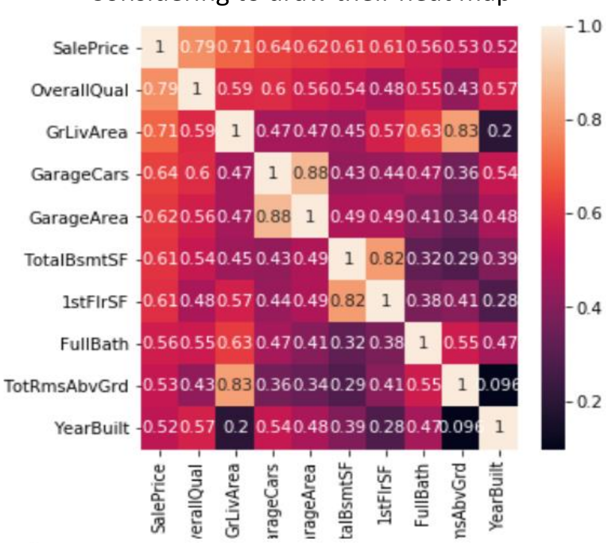



**Table 1 Top 10 variables with highest correlation with sales prices**

| Variable | Description | Correlation |
|---|---|---|
| OverallCond | Rates the overall condition of the house | 79% |
| GrLivArea | Above grade (ground) living area square feet | 71% |
| GarageCars | Size of garage in car capacity | 64% |
| GarageArea | Size of garage in square feet | 62% |
| TotalBsmtSF | Total square feet of basement area | 61% |
| 1stFlrSF | First Floor square feet | 61% |
| FullBath | Full bathrooms above grade | 56% |
| TotRmsAbvGrd | Total rooms above grade (does not include bathrooms) | 53% |
| YearBuilt | Original construction date | 52% |
| YearRemodAdd | Remodel date (same as construction date if no remodeling or additions) | 51% |

By analyzing the color depth of heat map, we can find out the relationship with the dependent variable (House Price) According to the heat map, we can find some variables that are related to each other, such as OverallQual and SalePrice have strong relation between them with the corr is 0.795. TotRmsAbvGrd and GlivArea is 0.83, which means that we can predict the SalePrice of some house with signal feature owing to their strong relationship with HousePrice, and we can also make combination of some certain features because the combination of them may have strong relationship with our dependent variable.

### b. about the character-type discrete values

***The charts are in appendix***

From the 1st chart, before 1990 the SalePrice do not change much over time by YearBuild and YearRemoveAdd, but after 1990 the influence increases tremendous. From 2rd chart to final charts show the SalePeice is affected by serious discrete variable, especially assessment type variable, and some charts show the effect is not linear.

## Ⅲ Data Cleaning

### a. Firstly, finding the features which have missing data

```
X=pd.concat([Train,Test],axis=0)
X1=pd.DataFrame(X.isnull().sum()[X.isnull().any()],columns=["values"])
X2=pd.DataFrame(X[X1.index].dtypes,columns=["types"])
lost_values2=pd.concat([X1,X2],axis=1).sort_values(by="values",ascending=False)
lost_values2
```

*the consequence is:*

| | values | types |
|---|---|---|
| PoolQC | 2909 | object |
| MiscFeature | 2814 | object |
| Alley | 2721 | object |
| Fence | 2348 | object |
| SalePrice | 1459 | float64 |
| FireplaceQu | 1420 | object |
| LotFrontage | 486 | float64 |
| GarageFinish | 159 | object |
| GarageQual | 159 | object |
| GarageYrBlt | 159 | float64 |
| GarageCond | 159 | object |
| GarageType | 157 | object |
| BsmtCond | 82 | object |
| BsmtExposure | 82 | object |
| BsmtQual | 81 | object |
| BsmtFinType2 | 80 | object |
| BsmtFinType1 | 79 | object |
| MasVnrType | 24 | object |
| MasVnrArea | 23 | float64 |
| MSZoning | 4 | object |
| Utilities | 2 | object |
| Functional | 2 | object |
| BsmtHalfBath | 2 | float64 |
| BsmtFullBath | 2 | float64 |
| GarageCars | 1 | float64 |
| Exterior2nd | 1 | object |
| KitchenQual | 1 | object |
| Exterior1st | 1 | object |
| Electrical | 1 | object |
| BsmtUnfSF | 1 | float64 |
| BsmtFinSF2 | 1 | float64 |
| BsmtFinSF1 | 1 | float64 |
| SaleType | 1 | object |
| TotalBsmtSF | 1 | float64 |
| GarageArea | 1 | float64 |

By analyzing all the data of the dataset, we can create some rules to fill the missing data:

(1) To some numeric data, we can fill them with 0, for the reason is that the meaning of missing data of some features is its value is empty, these features are as follows: LotFrontage, MasVnrArea, BsmtFullBath, BsmtHalfBath,GarageCars, BsmtFinSF1 and BsmtFinSF2;

(2) To some data, we can use their mode to fill them, because the meaning of missing data of some features is the empty of its value, and it may be existing probably, for example some data. And this kind of missing data is as follows: PoolQC, BsmtQual, BsmtCond, FireplaceQu, GarageFinish, GarageQual, BsmtExposure, Electrical, MSZoning, Exterior1st, Exterior2nd, KitchenQual and SaleType.

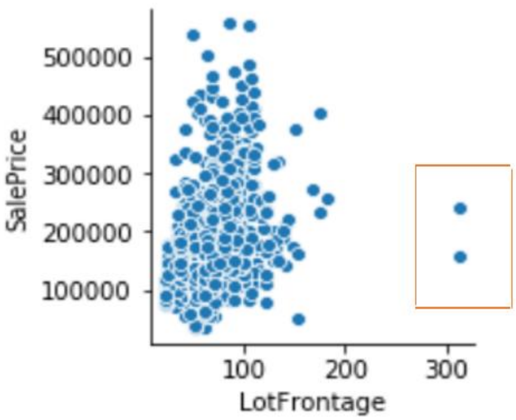
(3)And about other data, we cannot know about the reasons for them to have missing data. So, we use “missing” to fill them.

### b. Then, clean up outliers

**(1) Delete the SalePrice data which is beyond [data.mean-5*data.std ,data.mean+5*data.std]**

```
Data1=X[X.SalePrice <= X["SalePrice"].mean()+5*X["SalePrice"].std()]
```

**(2) Use rational graph**
Using the scatter diagram to Elimination of outliers




| Type | Delete number | Type | Delete number |
|---|---|---|---|
| 1stFlrSF | 4 | BsmtFinSF1 | 1 |
| 2ndFlrSF | 2 | BsmtFinSF2 | 1 |
| 3SsnPorch | 2 | LotArea | 4 |
| MasVnrArea | 1 | MiscVal | 3 |
| BedroomAbvGr | 1 | LotFrontage | 2 |
| TotalBsmtSF | 1 | | |

Taking "LotFrontage" for example, we can find that there are a spot diverge obviously, so we have to delete this data to reduce the variance .

```
value = 1
Data1=Data1.sort_values(ascending=False)[value:]
```

And finally, I select 11 features to do this process.

## IV Feature Engineering

### a. Add new features

After understanding every feature, I choose to add some new features to the dataset

```
Data["TotalFlrSF"] = Data["1stFlrSF"]+Data["2ndFlrSF"]
Data["TotalPorch"] = Data["3SsnPorch"]+Data["EnclosedPorch"]+Data["OpenPorchSF"]
                     +Data["ScreenPorch"]
Data["TotalBath"] = Data["HalfBath"]+Data["FullBath"]
Data['YearsSinceRemodel'] = Data['YrSold'].astype(int) - Data['YearRemodAdd'].astype(int)
```

### b. Transform some features' type to continuous number

| Feature | Replace By |
|---|---|
| Ex | 5 |
| Gd | 4 |
| Ta | 3 |
| Fa | 2 |
| Po | 1 |
| missing | 0 |

The type of some data is discrete object originally, they can be quantifiably represent. So I use the numeric value to represent its degree, for example, ExterQual, it has 6 kinds of values—"Ex", "Gd" , "TA" , "Fa" , "Po" and "missing", and each values means different degree, so I use 0 to 6 represent them.

### c. standardization and Logarithmic

To make the calculated amount reduce and make the model fit better later, we Standardize the data.

```
for column in Data_.columns:
    if Data_[column].dtypes != "object":
        Data_[column] = (Data_[column]-Data_[column].mean())/Data_[column].std()
```

And then, after standardization, the value will be between 0 to 1, then I use Logarithmic to deal with the data after standar dization. It can magnify the absolute value of the number between 0 and 1, and compress the number greater than 1.

```
def addlogs(res, ls):
    m = res.shape[1]
    for l in ls:
        res = res.assign(newcol=pd.Series(np.log(1.01+res[l])).values)
        res.columns.values[m] = l + '_log'
        m += 1
    return res
loglist=skewness[abs(skewness)>0.15].index.tolist()
Data_ = addlogs(Data_, loglist)
```

### d. One-Hot

For many machine learning model cannot deal with discrete variable, using the method "get_dummies" can make object variable transform to numeric variable.

```
Data_pre=pd.get_dummies(Data_,columns=Data_.select_dtypes(include=["object"]).columns)
```

### e. Reduce Dimensions--PCA

After the last step, the dataset's features increase to *313*, so we have to use PCA to reduce some features and at the meanwhile, the information of the remaining feature should contains most of the information

```
from sklearn.decomposition import PCA
x = PCA(n_components=0.975,svd_solver="full").fit_transform(Data)
```

And finally, the number of features is 85, far less than the formal number—313.

## V Model Building

According to the dataset, I choose 8 models to fit the data.
At the beginning, spilt the data.

```
Xtrain,Xtest,Ytrain,Ytest = train_test_split(X_1,y,test_size=0.1,random_state=666)
```

### I Signal Model

| Model | Parameters | rmse | r^2 |
|---|---|---|---|
| Linear Regression | / | 20972.2487 | 0.9044 |
| Lasso | alpha = 1.0 | 20969.3383 | 0.9044 |
| ElasticNet | alpha = 1e-10 | 20972.2487 | 0.9044 |
| XGboost | n_estimators = 400 | 21235.6509 | 0.8887 |
| RandomForestRegressor | n_estimators = 150 | 21151.6706 | 0.8896 |
| AdaBoostRegressor | n_estimators = 250<br>learning_rate = 1.0<br>loss = exponential | 24193.3710 | 0.8587 |
| BayesianRidge | / | 20845.4079 | 0.9039 |
| GradientBoosting Regressor | n_estimators = 350<br>loss = ls | 20349.2373 | 0.9012 |

## Ⅱ Stacking

By trying  the models above, I will use stacking to combine these models, and the first floor is LinearRegression, Lasso, ElasticNet,  RandomForestRegressor, AdaboostRegressor, BayesianRidge and GrandientBoostingRegressor; the second floor is XGboost.

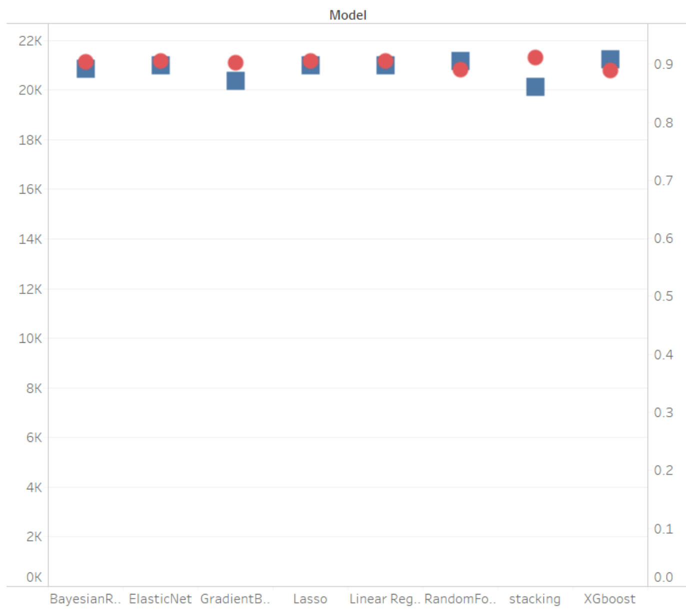
```
stacked = StackingCVRegressor(regressors=(rgc,la,ela,rfr,ada,bay,mlpr,gbr),meta_regressor=xg
                             ,use_features_in_secondary=True)
```

| | |
|---|---|
| 1st floor | LinearRegression, Lasso,ElasticNet,  RandomForestRegressor, AdaboostRegressor, BayesianRidge and GrandientBoostingRegressor |
| 2nd floor | XGboost |

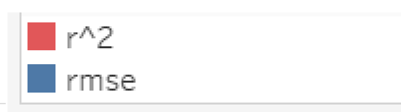
| | |
|---|---|
| RMSE | 20110.6972 |
| R^2 | 0.9108 |

## Ⅵ Evaluating

<Evaluation>




**We can read from the chart that the stacking has the least rmse and its r^2 is also good among these features.**




## Ⅶ Future Steps

I think the error mainly comes from the processes of filling missing value, feature engineering and PCA.

(1) While filling missing value, I use “0” and mode to fill them, however this way will take the inaccuracy to the model building and finally make the error great.

(2) And about the feature engineering, while using numeric value to represent the degrees of some discrete variables, it cannot easily replace the variable with the continuous number because we cannot do quantification accurate. For example, in this case, we use “5” to represent “Excellent”, use”4” to represent “Good”, use”3” to represent “Average”, but it may be incorrect because the gap between “Excellent” and “Good”, and between “Good” and “Average” may not be the same, but if we use “5”, “4” and “3” to represent them, it will mean that we default they are in the same.

(3) After the process PCA, the new features will just contain 97.5% information of the formal, this will also make the error great.

# APPENDIX

## Ⅰ exploration graph

Uploaded to Tableau (need VPN to read)

https://us-west-2b.online.tableau.com/#/site/sihenghuang/workbooks/191294?:origin=card_share_link

## Ⅱ code

Uploaded to GitHub

https://github.com/Alphonse-HUANG/HOUSE_PRICE-report/blob/master/house%20price%20code.ipynb