**House Price Prediction Project Report**

1. Business Understanding
   1. Objective

Predict the house price

* 1. Description

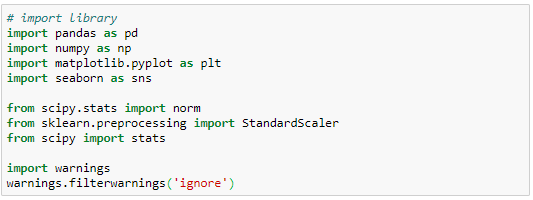
Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence.

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.

In this project, I will complete the analysis of which house feature will impact the house price. In particular, I will apply machine learning to predict the house price.

1. Data Understanding
   1. Import library

First of all, I need import python libraries containing the necessary functionality will need.



* 1. Load data

Next, load train.csv and test.csv dataset. Have a peek at the dataset size and variables.





* 1. Statistical summaries and visualization

To understand the data, I am going to consider some key facts about various variables including their relationship with the target variables, i.e. saleprice.

Here is the variable descriptions:

SalePrice - the property's sale price in dollars. This is the target variable that you're trying to predict.

MSSubClass: The building class

MSZoning: The general zoning classification

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access

Alley: Type of alley access

LotShape: General shape of property

LandContour: Flatness of the property

Utilities: Type of utilities available

LotConfig: Lot configuration

LandSlope: Slope of property

Neighborhood: Physical locations within Ames city limits

Condition1: Proximity to main road or railroad

Condition2: Proximity to main road or railroad (if a second is present)

BldgType: Type of dwelling

HouseStyle: Style of dwelling

OverallQual: Overall material and finish quality

OverallCond: Overall condition rating

YearBuilt: Original construction date

YearRemodAdd: Remodel date

RoofStyle: Type of roof

RoofMatl: Roof material

Exterior1st: Exterior covering on house

Exterior2nd: Exterior covering on house (if more than one material)

MasVnrType: Masonry veneer type

MasVnrArea: Masonry veneer area in square feet

ExterQual: Exterior material quality

ExterCond: Present condition of the material on the exterior

Foundation: Type of foundation

BsmtQual: Height of the basement

BsmtCond: General condition of the basement

BsmtExposure: Walkout or garden level basement walls

BsmtFinType1: Quality of basement finished area

BsmtFinSF1: Type 1 finished square feet

BsmtFinType2: Quality of second finished area (if present)

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

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: Number of bedrooms above basement level

Kitchen: Number of kitchens

KitchenQual: Kitchen quality

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

Functional: Home functionality rating

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

GarageType: Garage location

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

GarageCars: Size of garage in car capacity

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

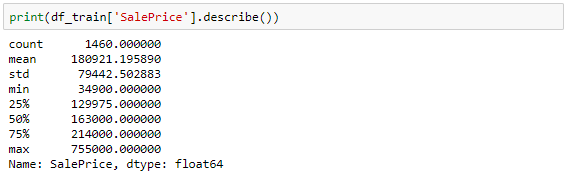
MoSold: Month Sold

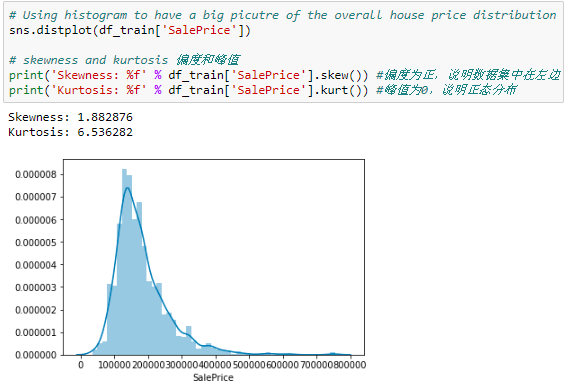
YrSold: Year Sold

SaleType: Type of sale

SaleCondition: Condition of sale

We can see from below, the saleprice is range from 34900 to 755000, while 75% of the price is lower than 214000. Also, the skewness is bigger than 0, which means most of the saleprice located in low price.



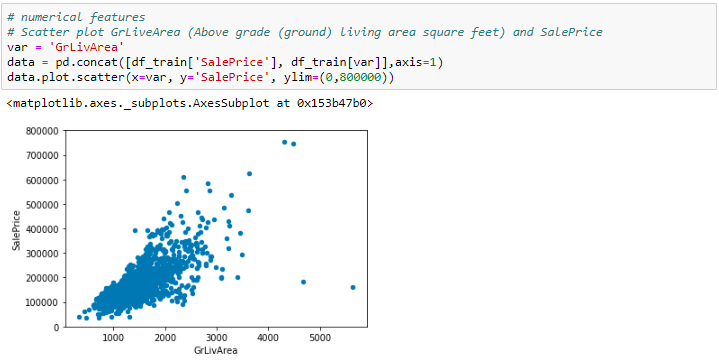


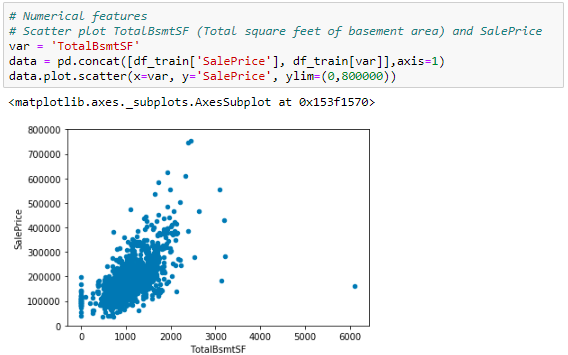
2.3.1 Investigating numerical variables

There are 2 kinds of variables, one is numerical variable and the other is categorical variable. Numerical variable is one with values of integers or real numbers, while a categorical variable is a variable that can take on one of a limited , and usually fixed number of possible values.

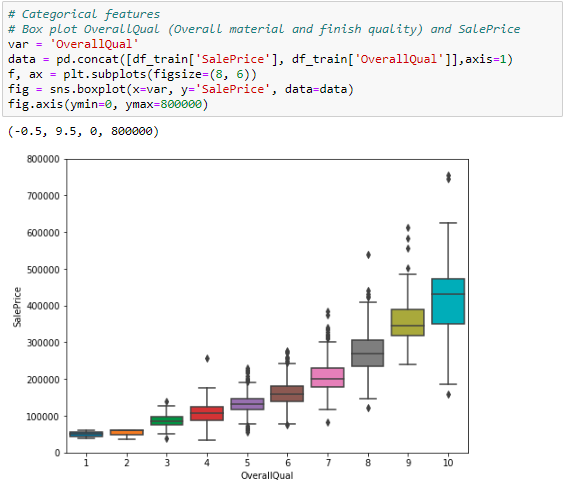
Since there are 80 variables in this case, I want to select several variables to see their relationship with saleprice.

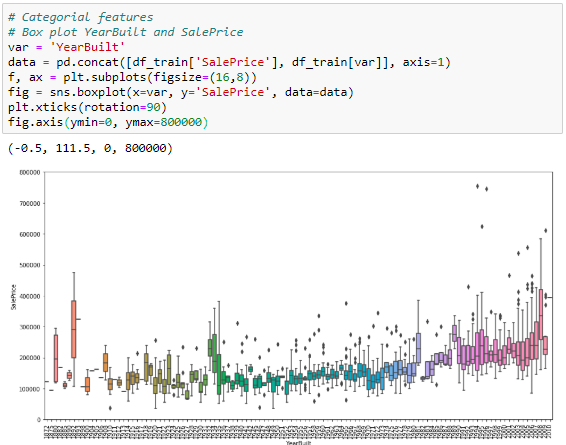
First, I start from numerical variables. Here I selected ‘GrLivArea’ and ‘TotalBsmtSF’, both of them seem to be linearly related with ‘SalePrice’, Both relationships are positive, which means that as one variable increases, the other also increases. In the case of ‘TotalBsmtSF’, we can see that the slop of the linear relationship is particularly high.





Next step, I select categorical variables. Here I selected ‘OverallQual’ and ‘YearBuilt’, both of them also seem to be related with ‘SalePrice’. The relationship seems to be strong in the case of ‘OverallQual’, where the box plot shows how sales prices increase with the overall quality.





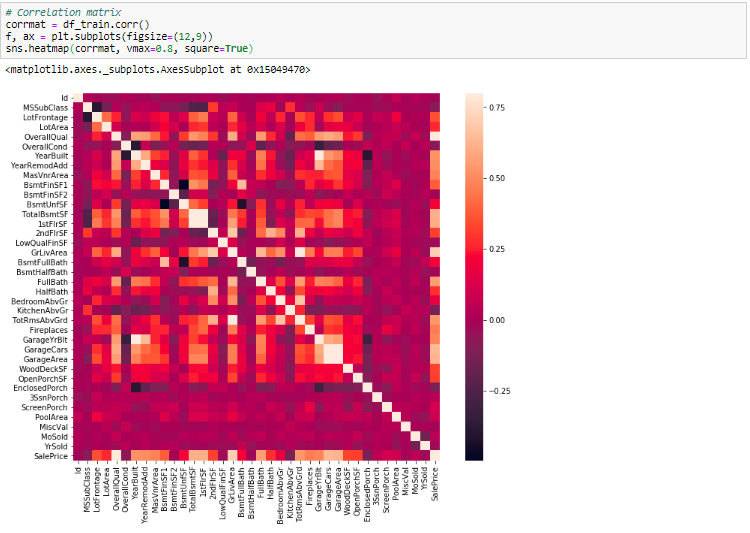
2.3.2 Variable correlation

Until now, I just follow my intuition and analyzed the variables I thought were important. But this method was subjective. As an engineer, I need a method able to withstand the winds of subjectivity. So, a heat map of correlation may give us an understanding of which variables are important, and the relationship between each variables.

To explore the universe, below is three things I plan to do:

* Correlation matrix (heatmap style)
* ‘SalePrice’ correlation matrix (zoomed heatmap style)
* Scatter plots between the most correlated variables (move like Jagger style)

2.3.2.1 Correlation matrix (heatmap style)

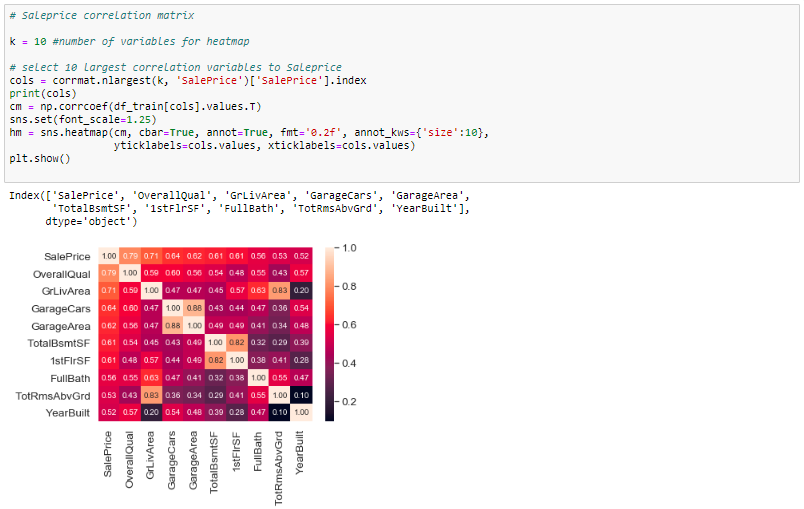


This heatmap is the best way to get a quick overview of the all variables and their relationship.

At first sight, there are 2 square that get my attention. The first one refers to the ‘TotalBsmtSF’ and ‘1stFlrSF’, and the second one refers to the ‘GarageCars’ and ‘GarageArea’ variables. Both cases show how significant the correlation is between these variables. Actually, this correlation is so strong that it can indicate a situation of multicollinearity. If we think about these variables, we can conclude that they give almost the same information so multicollinearity really occurs. Heatmap is great to detect this kind of situations and in problems dominated by feature selection.

2.3.2.2 ‘SalePrice’ correlation matrix (zoomed heatmap style)

Another thing that got my attention is the ‘SalePrice’ correlations. We can see ‘GrLivArea’, ‘TotalBsmtSF’ and ‘OverallQual’ and some other variables show strong correlations with ‘SalePrice’. That’s what I plan to do here: try to find the top 10 variables which has the strongest relationship with ‘SalePrice’.

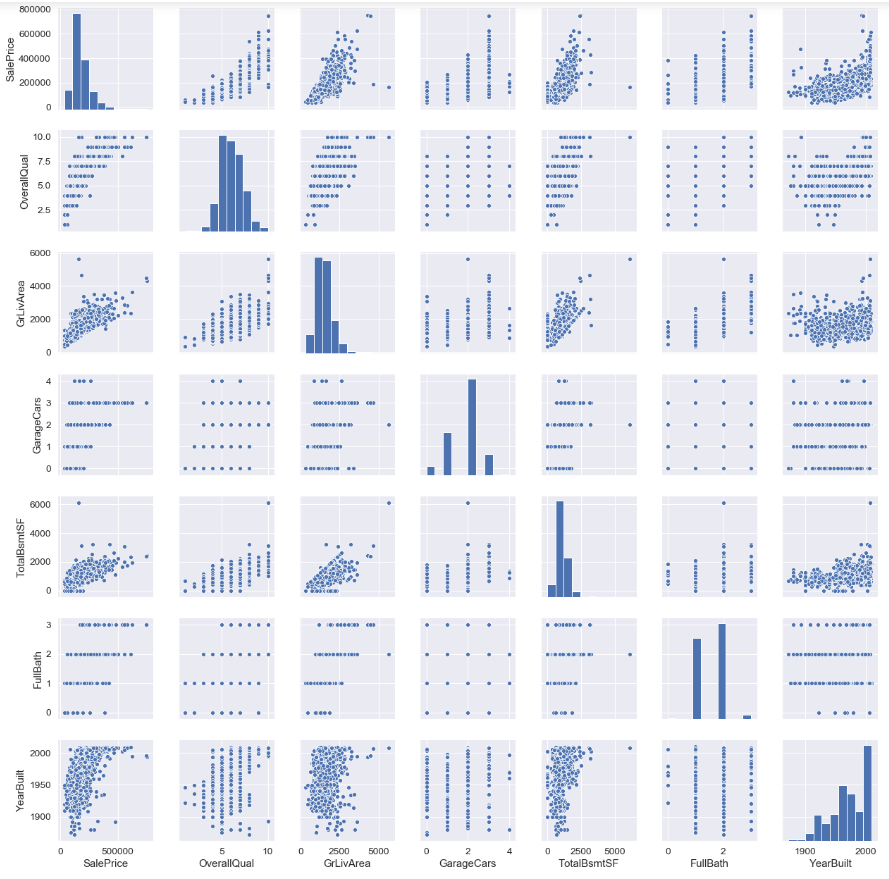


From above correlation heatmap, I find the top 10 variables most correlated with ‘SalePrice’.

* ‘OverallQual’, ‘GrLivArea’, ‘TotalBsmtSF’ are strongly correlated with ‘SalePrice’, this already be demonstrated in 2.3.1 with on the plot.
* ‘GarageCars’ and ‘GarageArea’ are also some of the most strongly correlated variables. However, the number of cars that fit into the garage is a consequence of the garage area. ‘GarageCars’ and ‘GarageArea’ are like twin brothers. You’ll never be able to distinguish them. Therefore, we just need one of these variables in our analysis. Here I keep ‘GarageCars’ since its correlation with ‘SalePrice’ is higher.
* ‘TotalBsmtSF’ and ‘1stFlrSF’ also seems twin brothers. Here I keep ‘TotalBsmtSF’ since it’s correlation with ‘SalePrice’ is higher.
* ‘TotRmsAbvGrd’ and ‘GrLivArea’ twin brothers again.

2.3.2.3 Scatter plots between the most correlated variables (move like Jagger style)





Although I already know some of the main figures, this mega scatter plot gives me a reasonable idea about variables relationships.

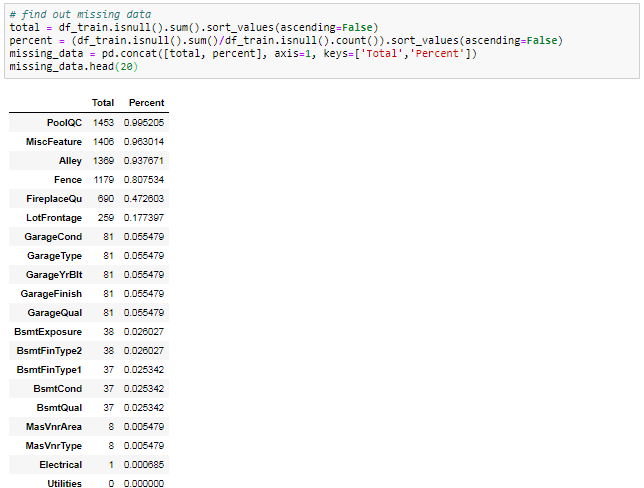
One of the figures we may find interesting is the one between ‘TotalBsmtSF’ and ‘GrLivArea’. In above figure we can see the dots drawing a linear line, which almost acts like a border. It totally makes sense that the majority of the dots stay below that line. Basement areas can be equal to the above ground living area, but it is not expected a basement area bigger than the above ground living area.

1. Missing data

Important questions when thinking about missing data:

* How prevalent is the missing data?
* Is missing data random or does it have a pattern?

The answer to these questions is important for practical reasons because missing data can imply a reduction of the sample size. This can prevent us from proceeding with the analysis. Moreover, from a substantive perspective, we need to ensure that the missing data process is not biased and hiding an inconvenient truth.



We’ll consider that when more than 15% of the data is missing, we should delete the corresponding variable and pretend it never existed. This means that we will not try any trick to fill the missing data in these cases. According to this, there is a set of variables (like ‘PoolQC’, ‘MiscFeature’, ‘Alley’ and etc.) that we should delete. The point is: will we miss these data? I don’t think so. None of these variables seem to be very important, since most of them are not aspects in which we think about when buying a house (maybe that’s the reason why data is missing?). Moreover, looking closer at the variables, we could say that variables like ‘PoolQC’, ’MiscFeature’ and ‘FireplaceQu’ are strong candidates for outliers, so we’ll be happy to delete them.

For ‘Gara

* 1. Categorical variables need to be transformed to numerical variables

The variables Embarked, Sex, Pclass are treated as categorical variables. Some of our model algorithms can only handle numeric values, so we need to create a new variable (dummy variable) for every unique value of the categorical variables.

For variable Embarked and Pclass, it will have a value 1 if the row has a particular value and a value 0 if not. Sex is a dichotomy, so it will be encoded as one binary variable (0 or 1).

* 1. Fill missing values in variables

Most machine learning algorithms require all variables to have values in order to use it for training the model.

From 2.3.1 I found that for numerical variables, only Age and Fare variable has missing value. The simplest method is to fill missing values with the average of the variable across all observations in the training set.

* 1. Feature engineering – creating new variables
     1. Extract titles from passenger names

If take a look at Name column, I can see there is a title in it. In that case, we might introduce an additional information about the social status by simply parsing the name and extracting the title and converting to a binary variable. Titles may have an influence on survival probability.

* + 1. Extract cabin category information from Cabin number

From 2.3.1, I can see out of 1309 cells, only 295 cells have cabin information, that means 77.4% data is missing for cabin variable. Due to high volume of missing data, one method is get rid of this column, another method is replacing the missing value as ‘U’ (means unknown) and extract cabin with cabin letter.

* + 1. Extract ticket class from ticket number

The ticket variable seems unsystematic, so let me extract the prefix of the ticket. If only digit in the ticket number, then replace it with ‘XXX’; otherwise extracting the prefix code.

* + 1. Create family size and category for family size

Here I want to use two variables Parch and SibSp to create the family size variable.

* 1. Assemble final datasets for modeling
     1. Variable selection

Select below features in the datasets:

* Imputed
* Embarked
* Pclass
* Sex
* Family
* Cabin
* Ticket
  + 1. Create datasets

Below I will separate the data into training and test datasets

* + 1. Feature importance

Selecting the optimal features in the model is important. Now, I am trying to evaluate what the most important variables are for the model to make the prediction.

1. Modeling

Now, I need select a model which I would like to try then use the training dataset to train this model and thereby check the performance of the model using the test set

* 1. Model selection

There are several options to choose from when it comes to models. So a good starting point is logistic regression.

* + 1. Logistic Regression
    2. Random forests
    3. Support vector machines
    4. Gradient boosting classifier
    5. K-nearest neighbors
    6. Gaussian Naïve Bayes

1. Evaluation

Now I am going to evaluate model performance and feature importance

* 1. Model performance

We can evaluate the accuracy of the model by using the validation set where we know the actual outcome. This dataset have not been used for training the model, so it’s completely new to the model.

We then compare this accuracy score with the accuracy when using the model on the training data. If the difference between these are significant this is an indication of over-fitting. We try to avoid this because it means the model will not generalize well to new data and is expected to perform poorly.

From 4.1, comparing the score for these models, I though gradient boosting classifier is the best model. It has the highest score for validation data, also the difference between training data and validation data is not big, that means the model does not have over-fitting problems.

* 1. Feature importance – selecting the optimal features in the model

有待补充

1. Deployment

Publishing the resulting prediction from the model to the Kaggle leaderboard.