Advanced Regression Analysis To Predict Sales Price of House

Summary

Regression is a statistical tool used to understand and quantify the relation between two or more variables. Regression ranges from simple models to highly advanced equations. The two primary uses for regression in business are forecasting and optimization. In addition to helping managers predict such things as future demand for their products, regression analysis helps fine-tune manufacturing and delivery processes.

The most common use of regression in business is to predict events that have yet to occur. The model built in this project predicts the sale price of each house with 79 explanatory variables using skills as feature engineering and some advanced regression techniques.

Background

If we ask a person who wants to buy a house, they might think of several aspects, even some platforms like Zillow, Realtor.com, Trulia provide real estate properties for buying, selling, renting, etc. As a user who wants to visit platforms like these might not think about the aspects to get his/her dream house. Even people don't do deep dive analysis for a particular price on a particular house. These platforms do a lot of deep dive analysis based on major as well as minor aspects too and regression analysis for predicting the sale price of houses is one of them. All the information about the datasets used in this model can be found kaggle competition (link here)

Methods

The dataset used in this model proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. There is also a file called data_description.txt in the data directory of this repository where every variable is explained. The dataset is already splitted into train and test sets so we do analysis on them separately. Following are the methods used in the model:

1. **Data exploration, Visualisation and Analysis**

Firstly I explored some of the basic information of the train set and test set like shape, first 10 rows.

Univariate analysis on target variable

Our target variable or dependent variable is the Sales Price. Because of univariate there is only variable here i.e. Sales Price so we use a discrete probability distribution of one variable. There are many discrete probability distributions available. One such is the is binomial distribution:

Binomial Distribution Formula

$$P(x) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{(n-x)!\,x!} p^x q^{n-x}$$

where
n = the number of trials (or the number being sampled)
x = the number of successes desired
p = probability of getting a success in one trial
$q = 1 - p$ = the probability of getting a failure in one trial

Skewness	$\frac{q-p}{\sqrt{npq}}$

Below plot (fig-1) defines the univariate distribution of Sales price and skewness. As we can see there's positive skewness so we can improve the skewness by applying logarithmic function on the Sales price. After that we can see the improved skewness in the distribution (fig-2).

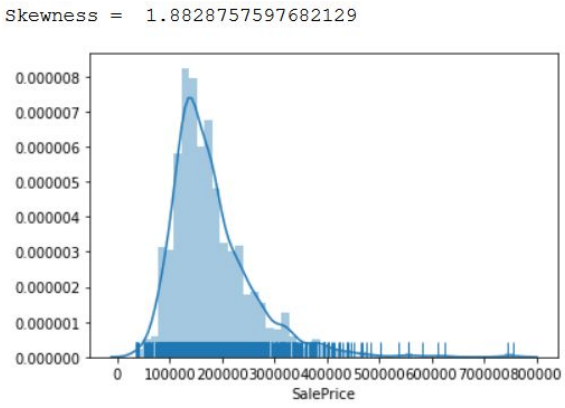

Fig - 1

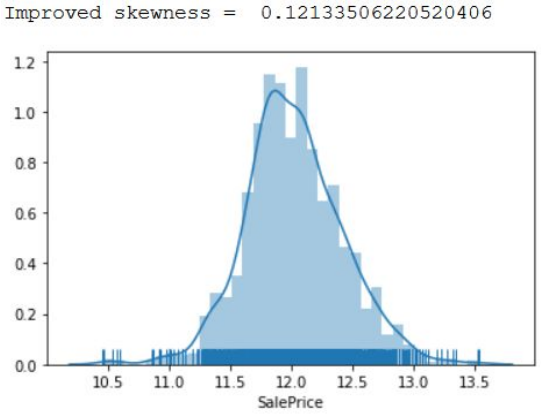

Fig - 2

Bivariate analysis

Using bivariate analysis we can see how the features are correlated with the Sales price. We find the pairwise correlation of all columns in the dataframe. Any missing/NaN values are automatically excluded. For any non-numeric data type columns in the dataframe it is ignored. The

DataFrame.corr() method of the pandas library uses the following methods to find the correlation coefficient:

- **pearson** : standard correlation coefficient
- **kendall** : Kendall Tau correlation coefficient
- **spearman** : Spearman rank correlation

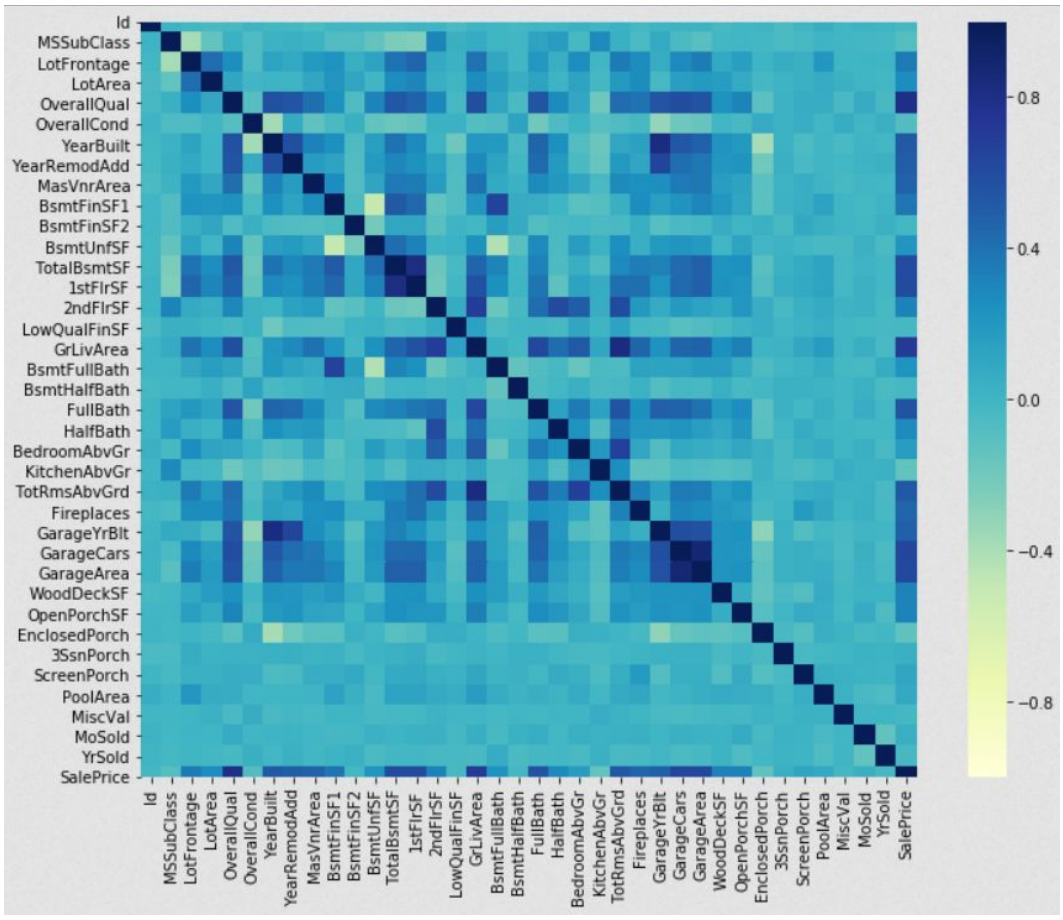

Fig-3

In the above heatmap we can get an idea of how a feature is correlated to the Sales Price (Last column from left or last row from top - Fig-3).

It's true that we can get an idea but that isn't enough for our analysis, so we use annotations that are the value of the correlation in the range -1 to +1. See fig-4.

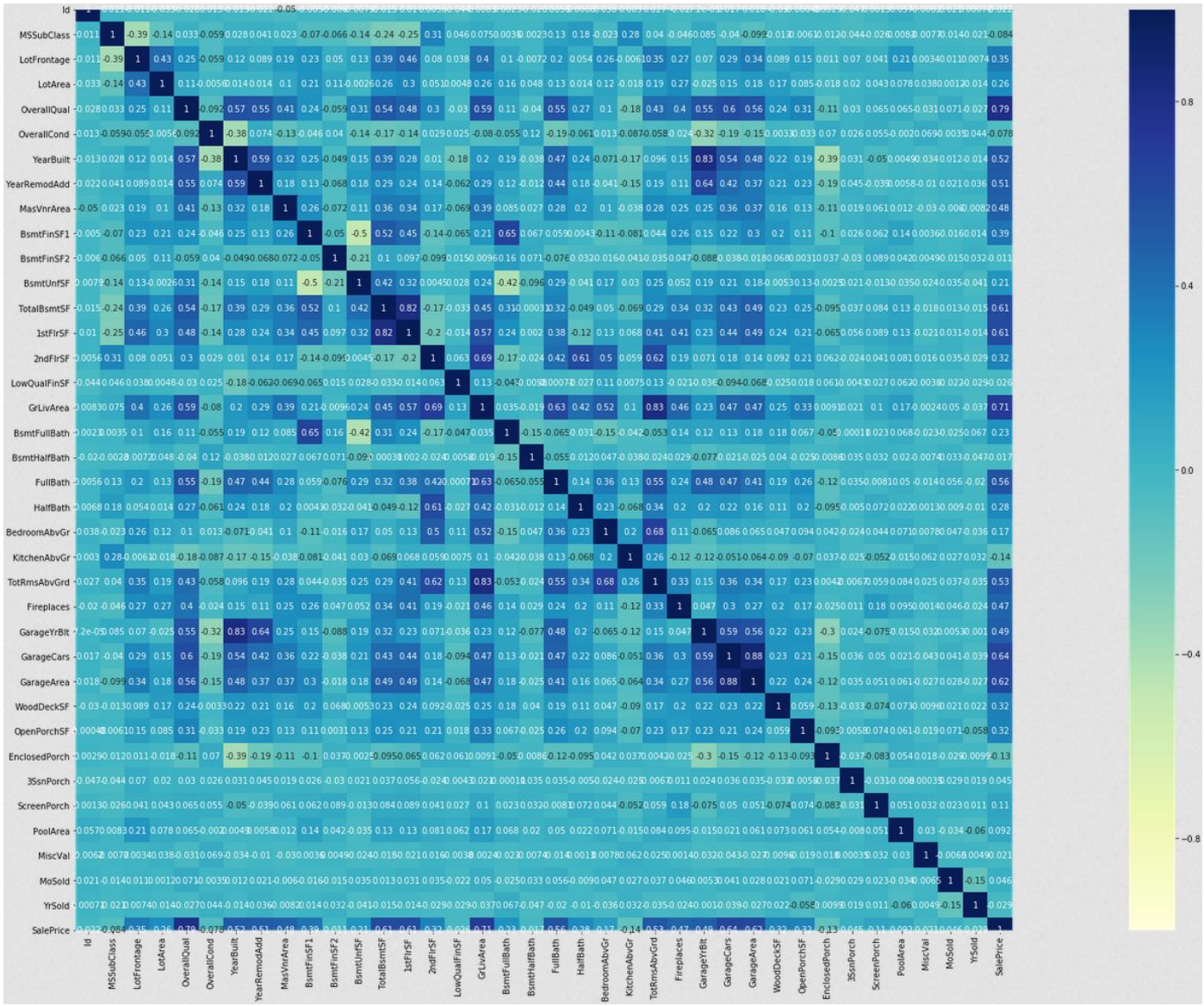

Fig-4

Features correlation with Sales price (Sorted)	
Feature	**Correlation Value**
SalePrice	1.000000
OverallQual	0.790982
GrLivArea	0.708624
GarageCars	0.640409

GarageArea	0.623431
TotalBsmtSF	0.613581
1stFlrSF	0.605852
FullBath	0.560664
TotRmsAbvGrd	0.533723
YearBuilt	0.522897

(**Note: Please refer data_description.txt in the data directory of this repository where every variable/feature is explained**)

From fig-4 and the above table we can see that some are highly correlated with values more than 0.5 and some are less than zero or near to -1. We can conclude that those features which are less correlated with Sales price can be dropped from the train set.

2. Data Preprocessing

Dealing with missing data

No of missing data in Train and Test set			
Train Data		**Test Data**	
Feature	**Total**	**Feature**	**Total**
PoolQC	1453	PoolQC	1456
MiscFeature	1406	MiscFeature	1408
Alley	1369	Alley	1352
Fence	1179	Fence	1169
FireplaceQu	690	FireplaceQu	730
LotFrontage	259	LotFrontage	227
GarageCond	81	GarageCond	78
GarageType	81	GarageQual	78
GarageYrBlt	81	GarageYrBlt	78

GarageFinish	81	GarageFinish	78
GarageQual	81	GarageType	76
BsmtExposure	38	BsmtCond	45
BsmtFinType2	38	BsmtQual	44

It can be seen that there are a lot of features like Pool Quality and Misc feature have a lot of missing values and if we visualize the missing data in the train set below in fig-5.

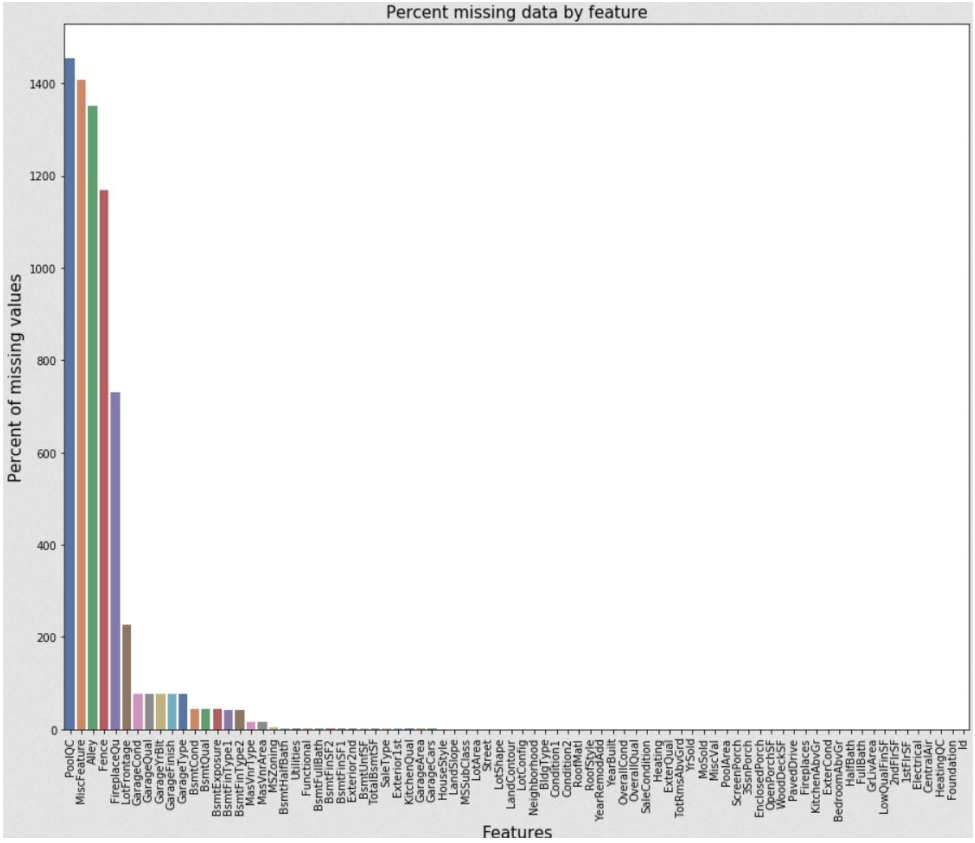

Fig-5

From the above missing data visual and the correlation analysis we can drop some features from the training and test datasets.

List of features (columns) to drop from the training and test set

['PoolQC','MiscFeature','Alley','Fence','FireplaceQu','GarageType','GarageFinish','GarageQual','GarageCond','MasVnrType']

Now the features which needed to be filled are as following:

- For Continuous Features/Variable we fill the missing data with mean
- For Categorical Features/Variable we fill the missing data with mode

Outliers

Formula for Outliers

A value is an outlier if it lies more than 1.5 times the IQR from the nearest quartile.

Thus, a value is an outlier if it is

$$< Q_1 - 1.5(IQR)$$

or

$$> Q_3 + 1.5(IQR)$$

Using the above outlier formula we find the outlier and visualize with scatter and box plots(Fig-6). Analyzing fig-6, it represents the Sales Price vs Ground Living Area trend, we can remove some outliers which have Ground Living Area greater than 4000 square feet. After dealing with the outlier we can see the change in fig-7

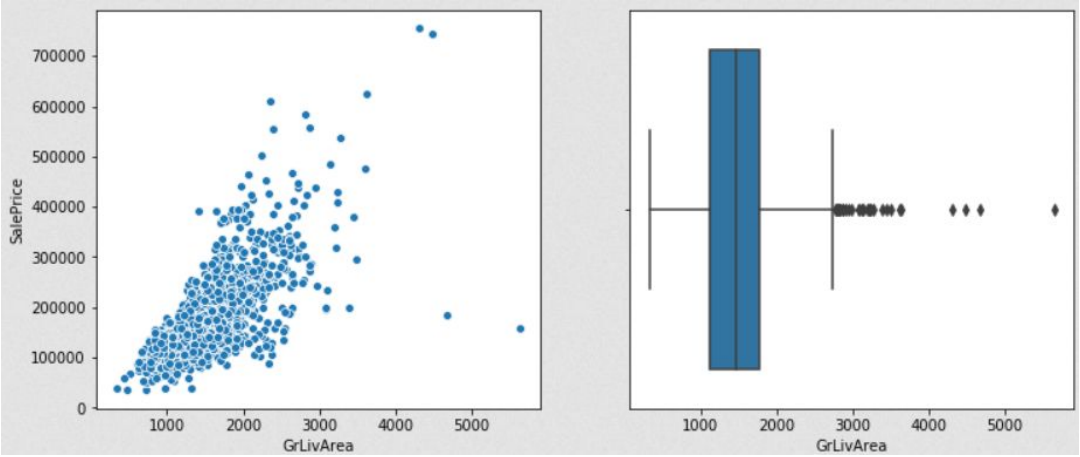

Fig-6

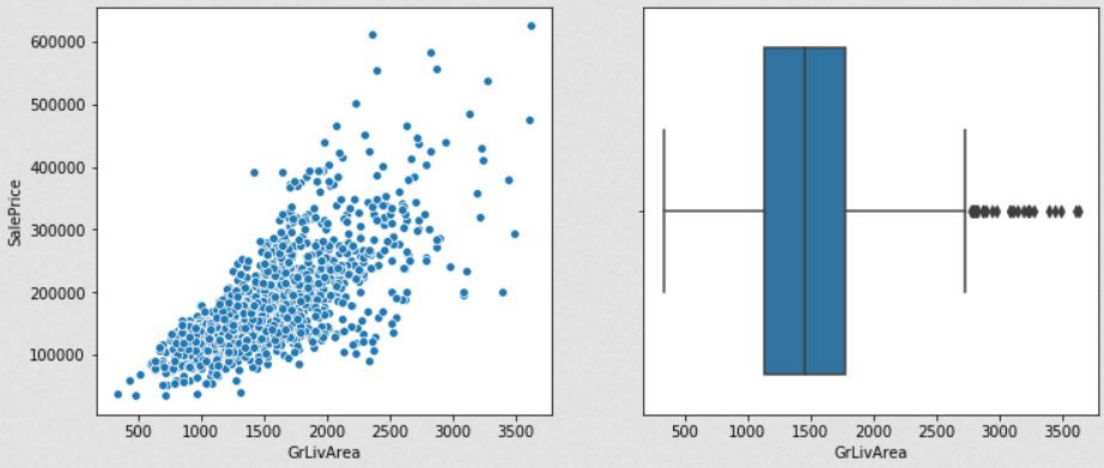

Fig - 7

Transforming and Encoding

Next dealing with the categorical data which are numeric so that we can apply LabelEncoding to them. First the numeric data is converted to string data then Encoding the data. Next step would be differentiating the dependent and independent variables.

Modelling and Tuning

The processed data to feed into a LinearRegression model which gives us a score of 0.8998795031894828.

Tuning :

- Gradient Boost:Improved Score = 0.9906745236620275
- XGBoost: Improved Score = 0.9980329342046588

Results

The final result of our model would be a file named as ‘predict_csv.csv’ file which consists of the final price of houses. The Sales price column values are the average of the predictions from the three regression models

Improvements

We can also improve the model based on users who visit a platform which implements the model we built for predicting the Sales price of the houses. If we can use A/B testing with the help of cookies we can generate more significant features(data collected from user interaction through the A/B testing) for the prediction.