CS 5512 - Machine Learning (Jan - Apr 2020)

COURSE PROJECT ON HOUSING PRICE PREDICTION

Muhammed Yaseen
Pre-final year UG in CSE
IIT Palakkad
111701032@smail.iitpkd.ac.in

Problem Statement

Project is to build a model for the sales price for each house. Along with the good generalization performance the model should have also explainability (i.e. which parameters are more important to decide a sale values). Your model also guide if a subset of features were enough for similar performance.

Use Train.csv from Kaggle[1] for showing all your results and analysis. To show generalization performance please do train test splits of 70% and 30%. Keep it in mind that training test data should be independent and should have the same distribution of all values . Do required validation steps to fixed hyper parameters of your model if exist.

Outline

The project mainly focuses on following parts:

1. Understanding the dataset
2. Cleaning the Dataset
3. Modelling the data

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

Methodology

Understanding the dataset

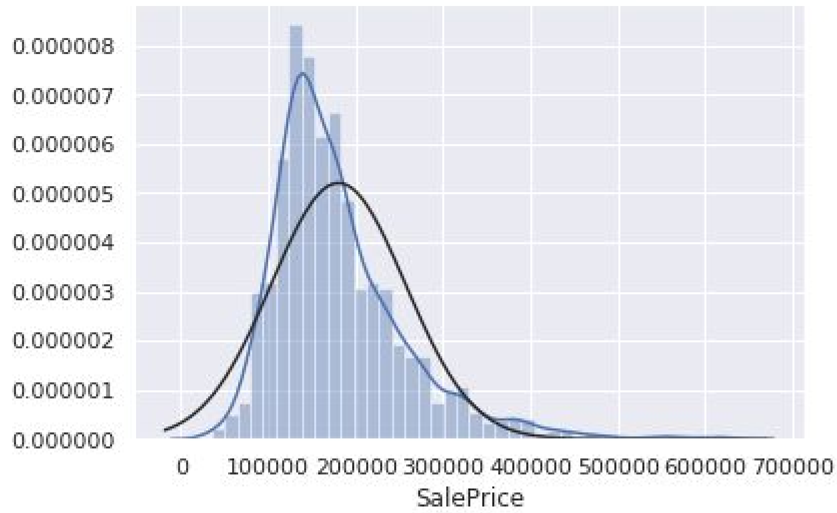
The first step is to understand the dataset in general. We had checked for statistical data of the dataset using pandas frameworks. It can be understood that the train data has 81 features and the test data has 80 features (all except for the target 'SalePrice'. There are 1460 data samples in the train data and 1459 data samples in the test data.

Cleaning the dataset

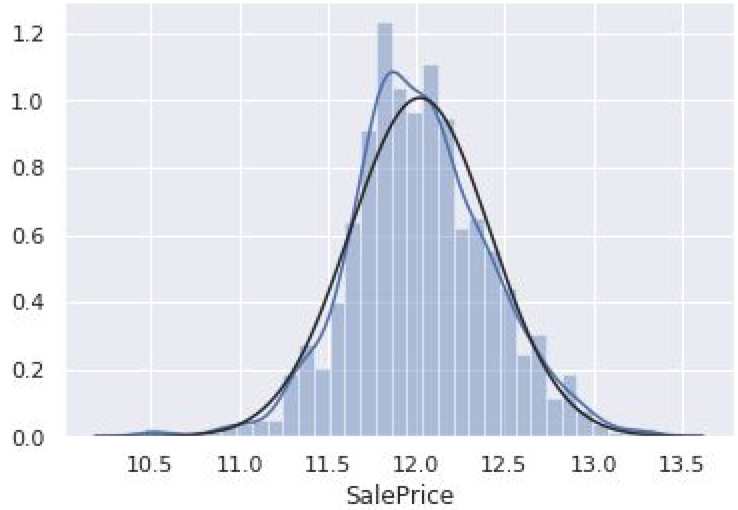
We will start by dropping the 'Id' feature which is irrelevant in the prediction of the data. The author of the Ames housing dataset recommends removing some outliers in the data. We restrict the dataset to the samples containing 'GrLivArea' less than 4000 based on this insight.

It is always advised to check for the correlation of features. We plot a correlation map between the values which have an absolute correlation value greater than 0.5 to identify the mostly correlated features. It can be noted that 'OverallQual', 'GrLivArea', 'TotalBsmtSF' displays maximum correlation with the target 'SalePrice'. It is also interesting to note that some features like 'GarageArea' and 'GarageCars' also display a high correlation which can be understood from the nature of the features. A plot between 'OverallQual' and 'SalePrice' can be found to be almost linear which can be understood from the high correlation value. In addition to this, we also plot some scatter plots between the top correlated values and the target 'SalePrice'.

Regression models will not work if it is not a normal distribution as it can affect the parameter calculations. We plot a graph to check the skewness of the 'SalePrice'.



It is clear that the above plot is skewed right from normal. Therefore, we use log normalization on the target ‘SalePrice’. The plot obtained after the normalisation is attached below.



If we log transform the response variable, it is required to also log transform the feature variables, which will be done at a later stage. This will be done at a later stage.

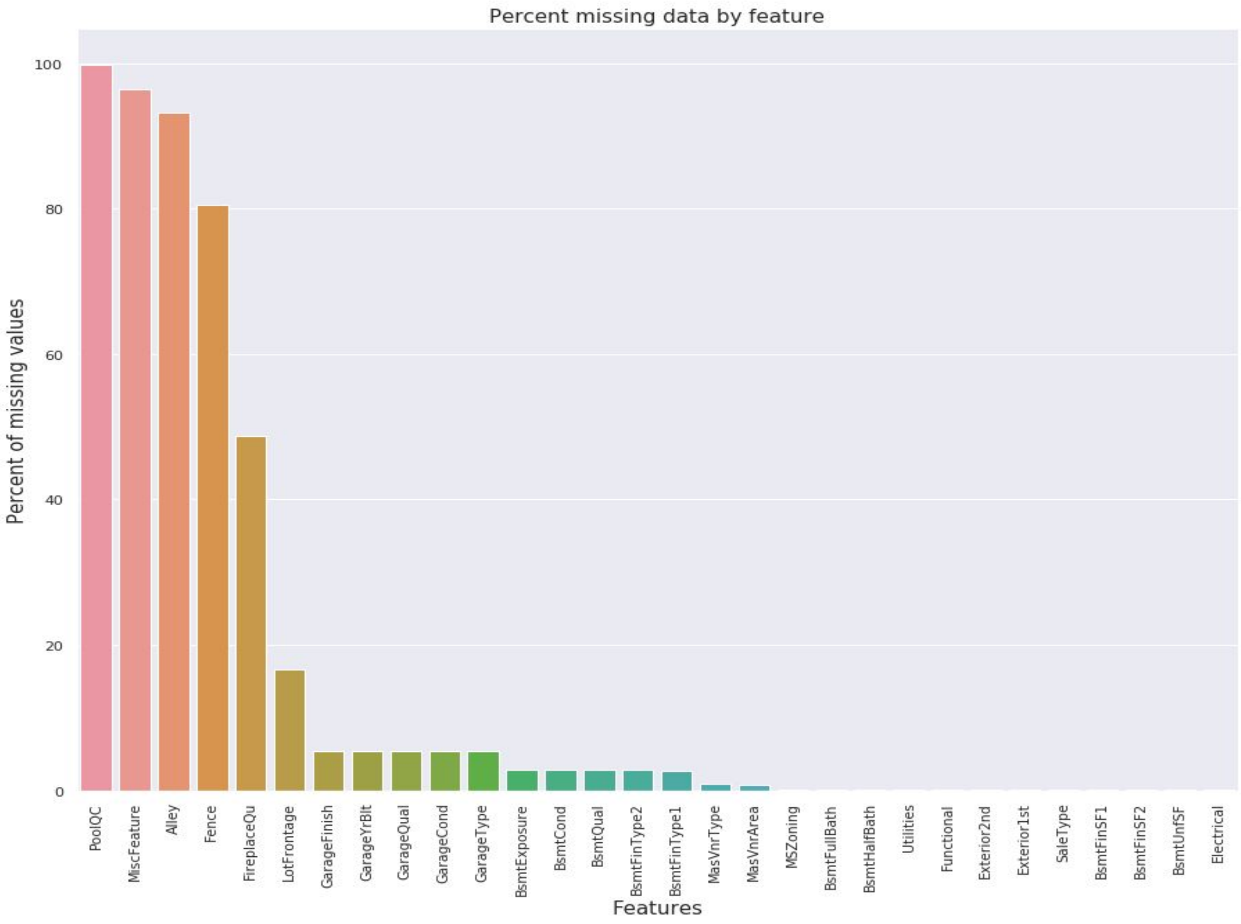
Feature Engineering

Feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features can be used to improve the performance of machine learning algorithms.[2]

We use some techniques to further polish the dataset for the modelling purpose. This includes handling of the missing data, deleting repetitive data, etc. It is important that all these techniques should be applied uniformly to the test data as well as the train data. Hence, we concatenate both of the data for the further processes. The count of train data and test data is noted for reverting it back once the transformations are done.

Handling the missing data

We plot the percentage of missing data for each of the features below.



[2] Definition taken from Wikipedia

Next step is to account for the missing data. We treat them individually to increase the quality of the dataset. We drop the feature ‘PoolQC’ as more than 95% of the ‘PoolQC’ values are missing. Also, we take a close look at the data description and fill the missing data with ‘None’ wherever it is suggested by the author. Similarly, numerical features are replaced by zero wherever suggested by the author of the dataset. We replace the ‘LotFrontageArea’ with the mean of the ‘LotFrontAgeArea’ in the corresponding ‘Neighbourhood’. Home functionality ‘Functional’ is replaced by the most used value ‘Typ’ wherever it was found to be missing. This technique is extended to other numerical variables, where we replace the missing value with the mode of the corresponding feature. We also observe that ‘Utilities’ and ‘MiscVal’ contains very redundant data (this insight is received from the skew plot of the features given later in this report). We drop both of these columns for refining the dataset.

Categorical Features

A categorical or discrete variable is one that has two or more categories (values). There are two types of categorical variable, **nominal** and **ordinal**. A nominal variable has no intrinsic ordering to its categories. For example, gender is a categorical variable having two categories (male and female) with no intrinsic ordering to the categories. An ordinal variable has a clear ordering. For example, temperature as a variable with three *orderly* categories (low, medium and high).

The categorical features have to be encoded into the number format for data modelling purposes. Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering. One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature. Label Encoding is used when the data is ordinal (a relative ranking exists) and One Hot Encoding is used when it is the opposite case. We also adopt the same strategy in our dataset.

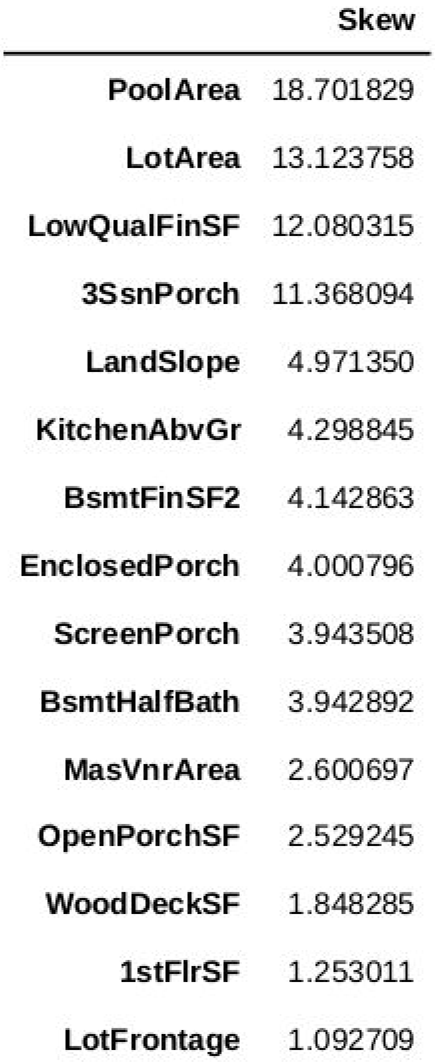
The TotalArea of a house is an important parameter which can decide the ‘SalePrice’. Based on this intuition, we add a new variable ‘TotalSF’ which is the sum of ‘TotalBsmtSF’, ‘1stFlrSF’ and ‘2ndFlrSF’.

Finally, we apply log transformation to other features other than ‘SalePrice’ to reduce the skew of the data. The following tables denote the skew values before and after the transformation.

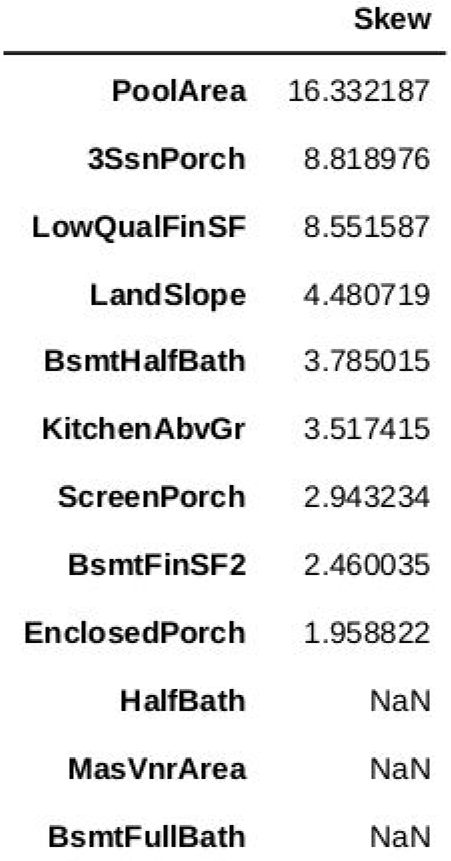
$$\text{Skewness} = \frac{3(Mean - Median)}{Standard\ Deviation}$$

This is done using the skew function from scipy.stats library. This helps us to give us an idea about how much skewed the data is to the left or right of the normal.

We can see that there is a noticeable improvement.



	Skew
PoolArea	18.701829
LotArea	13.123758
LowQualFinSF	12.080315
3SsnPorch	11.368094
LandSlope	4.971350
KitchenAbvGr	4.298845
BsmtFinSF2	4.142863
EnclosedPorch	4.000796
ScreenPorch	3.943508
BsmtHalfBath	3.942892
MasVnrArea	2.600697
OpenPorchSF	2.529245
WoodDeckSF	1.848285
1stFlrSF	1.253011
LotFrontage	1.092709



	Skew
PoolArea	16.332187
3SsnPorch	8.818976
LowQualFinSF	8.551587
LandSlope	4.480719
BsmtHalfBath	3.785015
KitchenAbvGr	3.517415
ScreenPorch	2.943234
BsmtFinSF2	2.460035
EnclosedPorch	1.958822
HalfBath	NaN
MasVnrArea	NaN
BsmtFullBath	NaN

The train and test data set is restored for the purpose of modelling the data.

Modelling the dataset

Linear regression models are used to show or predict the relationship between two variables or factors. The multivariate linear regression models discussed here are from the ones which were covered in the theory classes, namely Ordinary Least Square Regression, LASSO Regression and Ridge Regression.

Points to note in modelling

We use ‘GridSearchCV’ to choose the best hyperparameter from a set of values. This is done by a cross validation technique. The cross validation used here is 3-fold as the problem statement asks for a 70-30 train-test data split. The model is evaluated based on R2 Score. R Squared is the square of the correlation coefficient, *r* (hence the term r squared) as well as cross validation error using square loss. The coefficient of determination can be thought of as a percent. It gives you an idea of how many data points fall within the results of the line formed by the regression equation. The higher the coefficient, the higher percentage of points the line passes through when the data points and line are plotted.

If a vector of *n* predictions is generated from a sample of *n* data points on all variables, and Y is the vector of observed values of the variable being predicted, with $\overline{Yi}$ being the predicted values,

Mean squared error is given as,

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (Y - \overline{Yi})^2$$

The Sum of Squared Errors is given as,

$$SSE = \sum_{i=1}^{n} (Y - \overline{Yi})^2$$

Root Mean Squared Error is given as,

RMSE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y - \overline{Y}i)^{2}}$

Brief Note on Regression

The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.

Two popular examples of regularization procedures for linear regression are:

- Lasso Regression: where Ordinary Least Squares is modified to also minimize the absolute sum of the coefficients (called L1 regularization).
- Ridge Regression: where Ordinary Least Squares is modified to also minimize the squared absolute sum of the coefficients (called L2 regularization).

These methods are effective to use when there is collinearity in your input values and ordinary least squares would overfit the training data.[3]

Analysis

[3] Definitions from https://machinelearningmastery.com/linear-regression-for-machine-learning

The scores of various methods are as follows:

Model	R2 mean score	MSE mean score	Best parameter
OLS	-1.1941326838111813e+21	1.9222914495694723e+20	none
Lasso	0.9112493923128753	0.01393872254102986	0.001
Ridge	0.9118340189222528	0.013844886596199438	5

The performance of OLS is poor compared to other two models, which is evident from the negative scores. Lasso and Ridge models perform similarly, with a slightly better edge for Ridge Regression.

Summary

We can see that the LASSO and the Ridge models are performing similar. Upon multiple trials, it was found that the model predicted best when the final model was set to a weighted mean of both of these models, the factor for Ridge regression as 0.55 and the factor for Lasso Regression as 0.45, which can be understood from their respective performance scores. Kindly note that the 'SalePrice' is predicted in logarithmic form and hence, we explicitly convert it back by exponentiation at the end.

Result

The best model is the weighted mean of the LASSO and the Ridge models, with a slightly higher weightage for Ridge Regression. A common strategy will be to use PCA for the feature selection process. PCA combines similar (correlated) attributes and creates new ones, superior to original attributes. The data description of the data set was very clear to get more insights into the data, as to what all should be replaced for none, and what all are the meanings of each of the variables. Therefore, rather than involving PCA and getting a complicated model, we stick to a more understandable model by using our insights from the data description. Some features like 'Utilities' were dropped and some new features like 'TotalBsmtSF' were created in this fashion.

The model came out to be in the top 22% in the global Kaggle leaderboard with a rank 1084 (as on 15 May 2020).

Further improvements are possible by using advanced regression techniques.

References

1. Class slides by Dr. Sahely Bhadra
2. Introduction to Machine Learning by Dr. Andrew NG, Coursera
3. StatQuest with Josh Starmer, Youtube
4. Python, Pandas, Sklearn, Numpy documentations
5. www.towardsdatascience.com