A Report on Kaggle Competition - House Prices: Advanced Regression Techniques

1) Project Definition:

"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 playground competition's 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."

House Prices: Advanced Regression Techniques is the current ongoing competition on Kaggle where we have been given with the data that consists of multiple features which covers various aspects of house.

The task given is to predict the final sailing price for these houses using various machine learning algorithms. More than 5000 people have submitted their solution till now and to be precise, the number for total participants is 5190.

2) Dataset Description:

For this competition, **3** data files were given i.e., Train.csv, Test.csv and submission.csv.

Train.csv has in total **81** features and **1460** rows whereas the Test.csv has in total 80 features as it eliminates our dependent variable sale Price which we are going to predict and **1459** rows. So, we have one observation less in test data file compared to the train data file. The third file which is submission.csv has the actual sale price for the given test data and it is provided to us so that we can evaluate our model to see how well it did.

Talking about the type of data, it is distributed between Categorical and Numerical type. From total **80** features after eliminating the ID column, train file has in total **37** numerical and **43** categorical data. Kaggle has also given the description file which contains every detail for all the feature including all the categories that a feature possesses. I will discuss more about it in the Data Exploration section.

3) <u>Data Exploration:</u>

This is the initial and one of the important steps in data analysis where a large dataset is explored in an unstructured way which can further help us in finding out the important characteristics such as size, quantity, trends etc. in order to better understand the nature of the data.

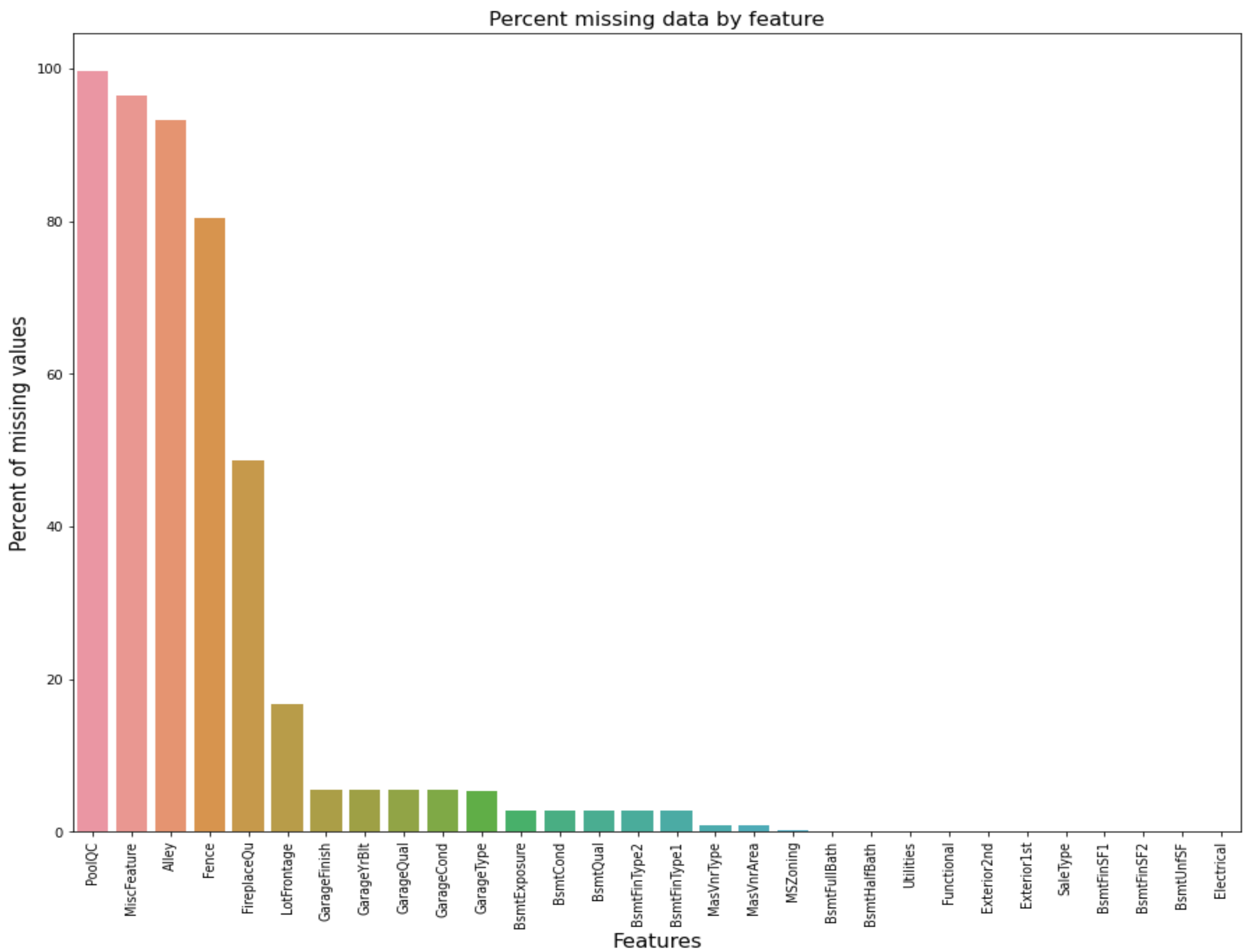
This first thing that I did in data exploration was to find out some information of data like the size of the data, type of the data and the amount of data that is missing. The .**info ()** method gave me all these information**.** Below is the image for percentage missing of all the features for train and test dataset.

	Missing Ratio
PoolQC	99.794380
MiscFeature	96.504455
Alley	92.666210
Fence	80.123372
FireplaceQu	50.034270
LotFrontage	15.558602
GarageYrBlt	5.346127
GarageCond	5.346127
GarageQual	5.346127
GarageFinish	5.346127
GarageType	5.209047
BsmtCond	3.084304
BsmtExposure	3.015764
BsmtQual	3.015764
BsmtFinType1	2.878684
BsmtFinType2	2.878684
MasVnrType	1.096642
MasVnrArea	1.028101
MSZoning	0.274160
BsmtFullBath	0.137080
BsmtHalfBath	0.137080
Utilities	0.137080
Functional	0.137080
Exterior2nd	0.068540
Exterior1st	0.068540
SaleType	0.068540
BsmtFinSF1	0.068540
BsmtFinSF2	0.068540
BsmtUnfSF	0.068540
KitchenQual	0.068540

Figure 1:Percentage missing data in Train File

	Missing Ratio
PoolQC	99.794380
MiscFeature	96.504455
Alley	92.666210
Fence	80.123372
FireplaceQu	50.034270
LotFrontage	15.558602
GarageYrBlt	5.346127
GarageCond	5.346127
GarageQual	5.346127
GarageFinish	5.346127
GarageType	5.209047
BsmtCond	3.084304
BsmtExposure	3.015764
BsmtQual	3.015764
BsmtFinType1	2.878684
BsmtFinType2	2.878684
MasVnrType	1.096642
MasVnrArea	1.028101
MSZoning	0.274160
BsmtFullBath	0.137080
BsmtHalfBath	0.137080
Utilities	0.137080
Functional	0.137080
Exterior2nd	0.068540
Exterior1st	0.068540
SaleType	0.068540
BsmtFinSF1	0.068540
BsmtFinSF2	0.068540
BsmtUnfSF	0.068540
KitchenQual	0.068540

Figure 2: Percentage missing data in Test Fil



In my initial approach, I decided to remove all those features which were having missing data more then 50% but then when I saw the description file, I found out that for most of the categorical features like PoolQC which defines the standard of pool of the house, MiscFeature, Fense, Alley and many more where missing percetage is higher then 90%, has a special meaning for NAN values. For example, NAN value in PoolQC states that the house does not have a pool or NAN in feature Fence tells us that the house does not have a Fence. Below are the images of few of them.

```
BsmtQual: Evaluates the height of the basement

       Ex       Excellent (100+ inches)
       Gd       Good (90-99 inches)
       TA       Typical (80-89 inches)
       Fa       Fair (70-79 inches)
       Po       Poor (<70 inches
       NA       No Basement
```

```
BsmtQual: Evaluates the height of the basement

       Ex       Excellent (100+ inches)
       Gd       Good (90-99 inches)
       TA       Typical (80-89 inches)
       Fa       Fair (70-79 inches)
       Po       Poor (<70 inches
       NA       No Basement
```

```
Fence: Fence quality

       GdPrv    Good Privacy
       MnPrv    Minimum Privacy
       GdWo     Good Wood
       MnWw     Minimum Wood/Wire
       NA       No Fence
```

```
PoolQC: Pool quality

       Ex       Excellent
       Gd       Good
       TA       Average/Typical
       Fa       Fair
       NA       No Pool
```

For this kind of features, who have a special meaing for NAN, I have filled them with the value given in the description file (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data?select=data_description.txt). For exmple NA values in PoolQC got filled with it's equivalent value "No Pool". For rest of the remaining data, I decided to fill out features with numerical type with their median values and categorical type with their mode. Instead of doing this process seperaltely for both test and train file, I merged them into one.

4) **Data Processing:**

The first thing that I did in data processing after filling out all the missing values was to find out the distribution of the dependent variable SalePrice. One of my model out of three is with LinearRegression. For Linear Regression, It is very much imporant that the data possess following qualities:

1) Data should be normaly distributed

2) All independent features should have linear relationship with the dependent feature but not with each other.

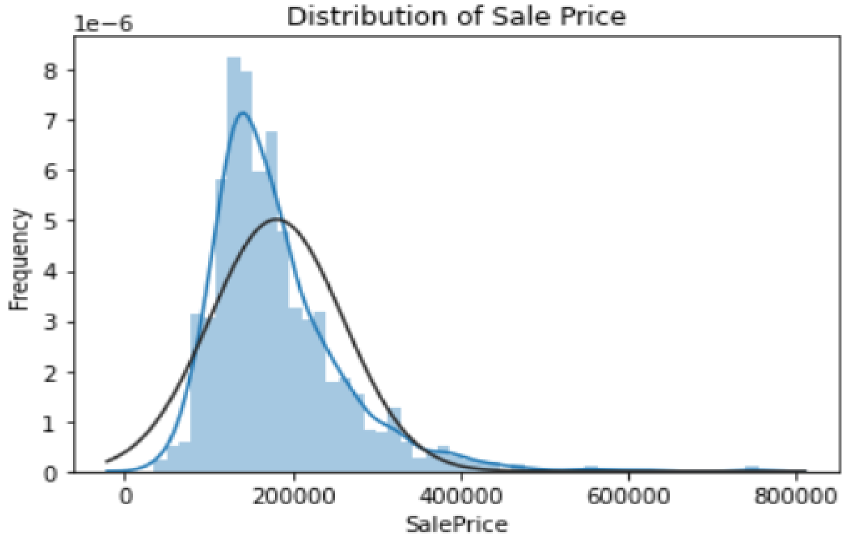


Figure 4: Distribution of Sale Price

By looking at the figure, it can be seen that the data is skwed r ight. To make this distribution normal, I performed **log transformation** on it. There are various other techniques like BoxCox transformation, Exponential transfromation, Reciprocal transfromation and square root transformation.

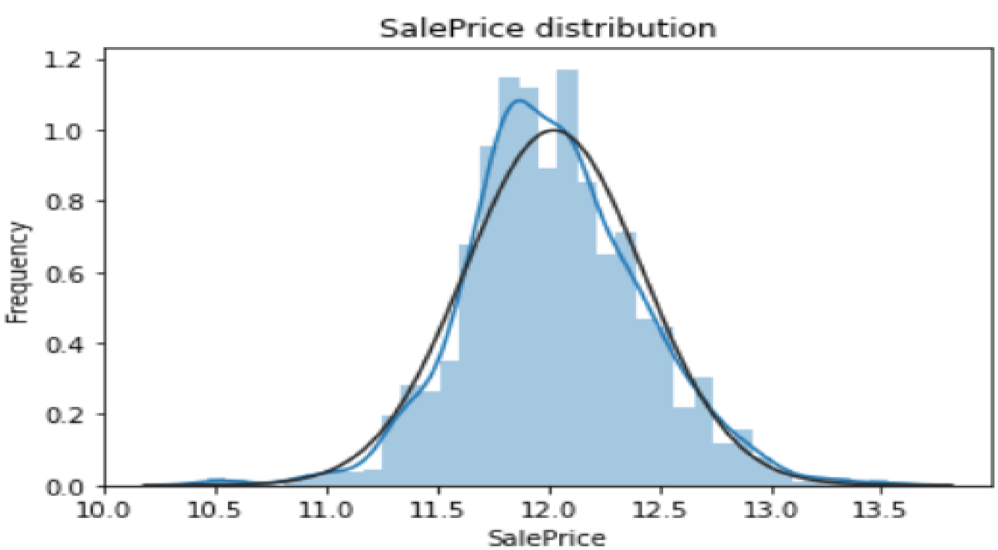


Figure 5: Normally distributed sale Price

After making the distribution of sale price normal, I checked the correlation of all the features with sale price and each other. I divided the features in to three section.

1) Highly correlated features: These are the features that have correlation value more then 0.5 with the dependent variable Sale Price.

```
highly_corr = dfTrain.corr()
highly_corr[(highly_corr['SalePrice']>0.5)].index
```

```
Index(['OverallQual', 'YearBuilt', 'YearRemodAdd', 'TotalBsmtSF', '1stFlrSF',
       'GrLivArea', 'FullBath', 'TotRmsAbvGrd', 'GarageYrBlt', 'GarageCars',
       'GarageArea', 'SalePrice'],
      dtype='object')
```

2) Features with Weak Correlation: These are the features that have some correlation with the dependent variable sale price but not strong.

```
Index(['LotFrontage', 'LotArea', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
       'BsmtUnfSF', '2ndFlrSF', 'BsmtFullBath', 'HalfBath', 'BedroomAbvGr',
       'Fireplaces', 'WoodDeckSF', 'OpenPorchSF', '3SsnPorch', 'ScreenPorch',
       'PoolArea', 'MoSold'],
      dtype='object')
```

3) Features with Zero Relation: These are the features that have zero or negative correlation with the dependent variable sale price.

```
Index(['MSSubClass', 'OverallCond', 'LowQualFinSF', 'BsmtHalfBath',
       'KitchenAbvGr', 'EnclosedPorch', 'MiscVal', 'YrSold'],
      dtype='object')
```

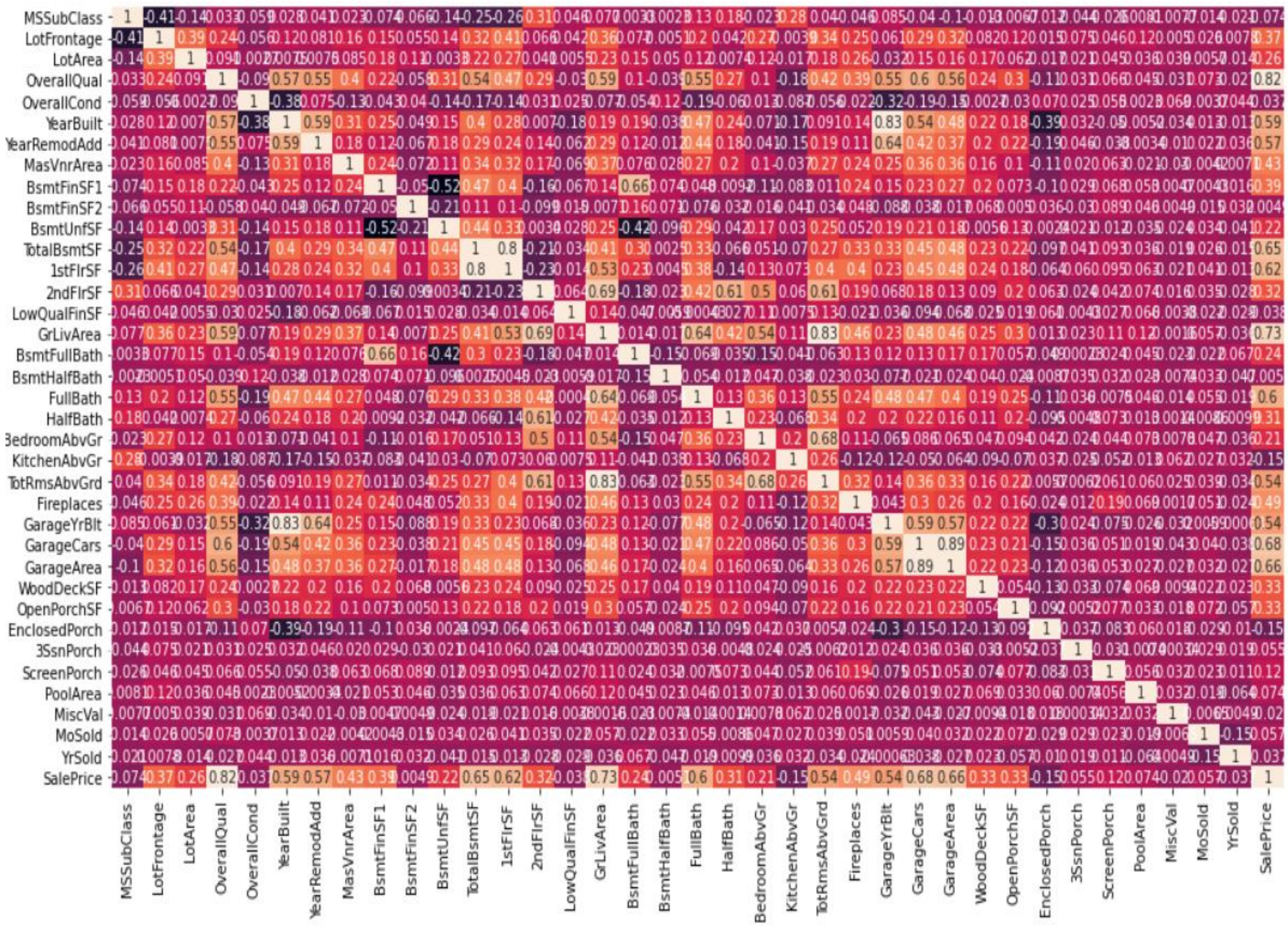


Figure 6: Correlation Matrix

From the correlation matrix, it can be seen that features such as OverallQual and GrLivArea shares highest correlation with sale price.

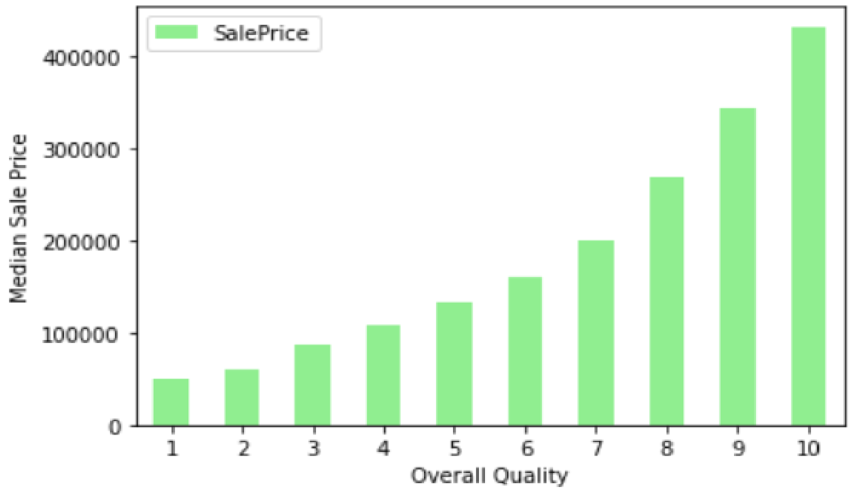


Figure 7: Relation of OverallQual with sale price

We can see that the sale price increases with increase in the overall quality.

Removing Outliers: Removing outliers is not always safe. I have only removed outliers from OverallQual and GrLivArea as they are the ones which are highly correlated.

Figure 8: GrLivArea before outlier removal

From the scatter plot, it can be seen that the two points down in the right corner are outliers as they have living area more then 4000 but their price is way less. Removing tose two points from the data.

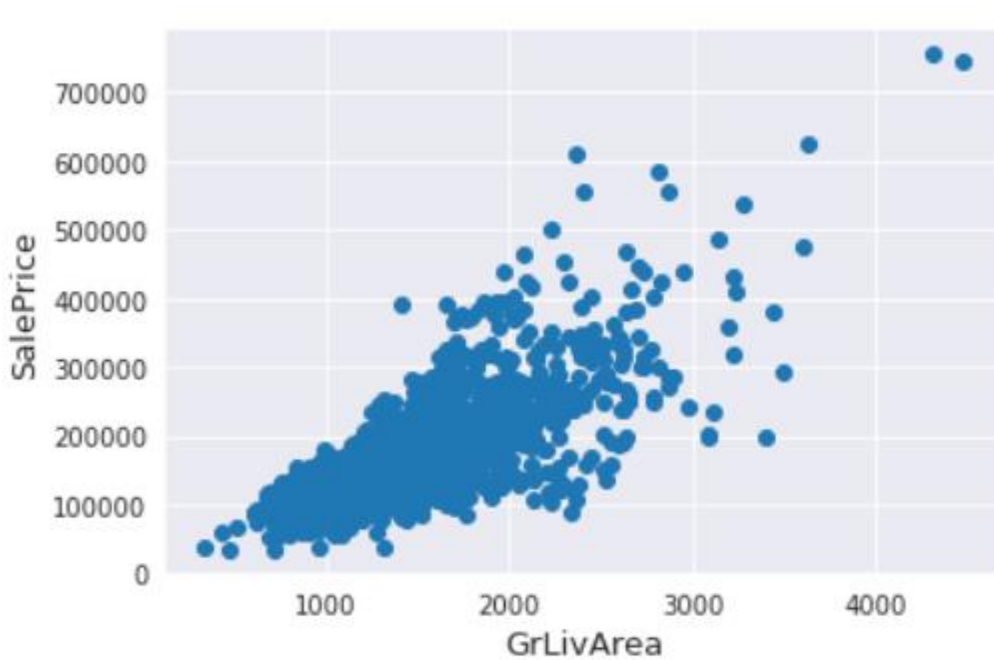


Figure 9: GrLivArea after outliers' removal

Checking Data Distribution:

I already made the distribution of sale price normal and now it is time to check the distribution of other remaining numerical data. For this, I divided my file into two different files, where in which on contains all the features that are of numerical type and the other which contains all categorical ones. During this, I found out that there were few features which were shown in the numerical file like overcond which defines the overall condition of the house but were of categorical type. I firstly converted them into categorical by changing their type from float to string.

Skew in numerical features:

	Skew
MiscVal	21.939672
PoolArea	17.688664
LotArea	13.109495
LowQualFinSF	12.084539
3SsnPorch	11.372080
LandSlope	4.973254
KitchenAbvGr	4.300550
BsmtFinSF2	4.144503
EnclosedPorch	4.002344
ScreenPorch	3.945101
BsmtHalfBath	3.929996
MasVnrArea	2.621719
OpenPorchSF	2.529358
WoodDeckSF	1.844792
1stFlrSF	1.257286
LotFrontage	1.103039
GrLivArea	1.068750
TotalSF	1.009157
BsmtFinSF1	0.980645
BsmtUnfSF	0.919688
2ndFlrSF	0.861556
TotRmsAbvGrd	0.749232
Fireplaces	0.725278
HalfBath	0.696666
TotalBsmtSF	0.671751
BsmtFullBath	0.622415
OverallCond	0.569314
HeatingQC	0.485534
FireplaceQu	0.332611
BedroomAbvGr	0.326568
GarageArea	0.216857
OverallQual	0.189591

Performed BoxCox transformation on these and made their distribution normal. As I mentioned earlier, I had to do this only because of model with Linear Regression.

5) <u>Model Creation & Analysis:</u>

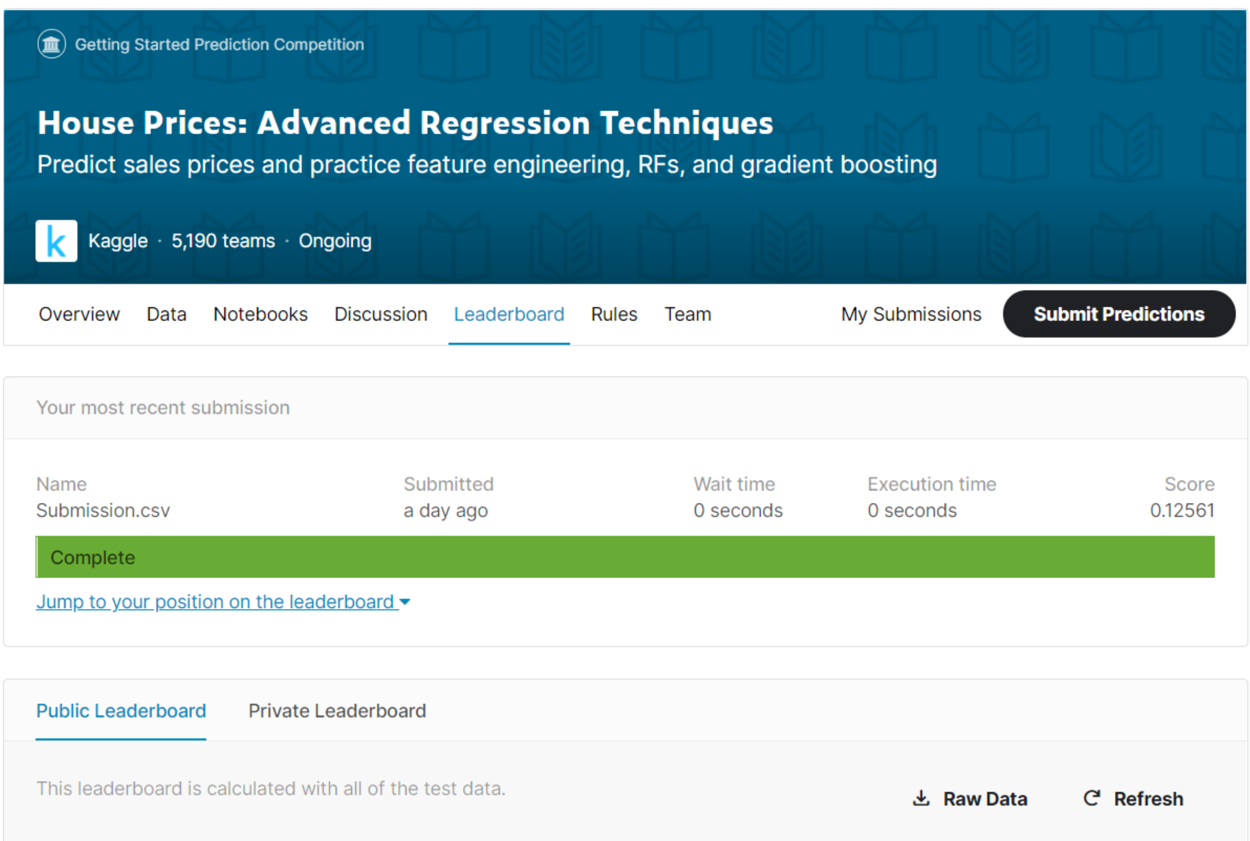
For this project, I created 3 models with different algorithms. The first model that I made is with Linear Regression. I used linear regression because I wanted to start with the simple model first before going forward with complex models. Then I made my second model with Random Forest and third with XGBoost. For my own purpose, I also tried to create on model with SVM but I haven't included that in this file.

I have used all the features except **Utlities** as it only had one category and so it would't have any better imapct on the performance of the model. Below is image of the score that I obtained after testing the model on the Train Data.

	Model	R2 SCORE	MAE	MSE	RMSE
0	Linear Regression	0.894274	0.085541	0.017325	0.131626
1	Random Forest Regressor	0.862293	0.096621	0.022566	0.150220
2	XGBoost	0.890856	0.085708	0.017885	0.133736

Figure 10: Model Evaluation on Train Data

The best result that I got is from model 3 which is using XGBoost algorithm and I have submitted that in Kaggle for this competition. Kaggle evaluates the score using Log based RMSE. My best score till now is 0.12 and my rank in the leaderboard is 1191 out of 5190



1190	JiaZhang000	0.12561	3	1mo
1191	PandyaHarsh	0.12561	11	1d

Your Best Entry ↑

Your submission scored 0.12561, which is not an improvement of your best score. Keep trying!

1192	Iris Theofilou	0.12563	2	6d

6) Comparing with Previous Score:

submission (2).csv 9 days ago by PandyaHarsh add submission details	9.45830
submission.csv 7 days ago by PandyaHarsh add submission details	0.14480
submission (1).csv 9 days ago by PandyaHarsh add submission details	0.19378

I have done multiple submissions in this competition, but I am only adding the 3 of them here which includes best and worst score. The first score I got which is 9.4 was when I trained my model with only those features that were having higher correlation.

The second score which is 0.19 was an improvement to my previous score. Here removed all the outliers from train and test file using IQR technique.

For the last attempt (Although I am still working on it) where I obtained my best score of 0.12, I kept outliers in all the features except the two highly correlated one's.