**House Prices-Advanced Regression Techniques**

**Executive Summary:**

I chose a dataset called ‘House Prices-Advanced Regression Techniques’ from Kaggle Competition as my project topic. I will predict the sales prices of houses by using approaches and models. The dataset contains 80 variables describing features of residential homes in Ames, Iowa including ID as indexes, SalePrice as target variables that I am going to predict and other features. The data package gives files ‘data description’ which describe types of categorical variables, ‘sample submission’, ‘test’ and ‘train’. Modeling with the train profile and predicting the sales’ prices. Comparing the predicted prices and the SalePrices in the test file, my goal is to reduce the difference between them by choosing approaches with the greatest score which means I need to minimize the Root Mean Squared Error between them.

**Three Solutions**:

I found a lot of solutions from other users which used different codes. Some of them give very specific visualization to see the variables correlations, outliers and missing data. I choose three of them. They are not the top ranked users but all of them have their own approaches.

I did not choose those who use all approaches like they combine all approaches and get the best one so that they get a great score. But I can not see any difference between them, so I choose three users who did not rank so high but each of them use different variables and models.

1. The first one I choose is ‘HousePricing’ made by Amrita Nigote with a 0.17551 public score.

Amiran uses RandomForestRegressor, she splitted 30 and 70 as train and test and random state as 42. She did not do a not of visualization with a pairplot only to see how each variable related with sales price.

1. Second one I chose is ‘house\_price\_reg’ made by Abdelmenam Tarek but he did not give his score.

Tarek uses the linear regression model to predict the sale price. She split the data to 33 and 67 as train and test and random state as 44. He did not give visualizations. But the model is simple and easy to understand.

1. The last one I chose is ‘#House-prices: LGBM tuned model (~0.12 - top 15%)’ made by Gershon Celniker and the score is 0.12 as known for smaller rmse, better model.

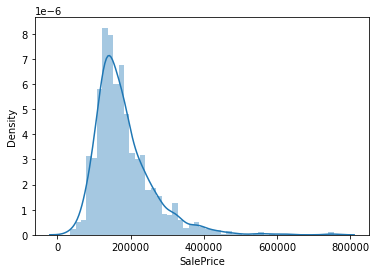
We can clearly know that Gershon uses an LGBM tuned model which I don't know much but glad to learn from his work. He use kfold to split data to 5 and then use LGBM model to predict and also he did very great visualization

|  |  |  |  |
| --- | --- | --- | --- |
|  | Features | Modeling | Performance |
| Amrita Nigote | All variables | RandomForest | 0.17551 |
| Abdelmenam Tarek | All variables | LinearRegression | none |
| Gershon Celniker | All variables,(choose top 32) | LGBM tuned model | 0.12 |

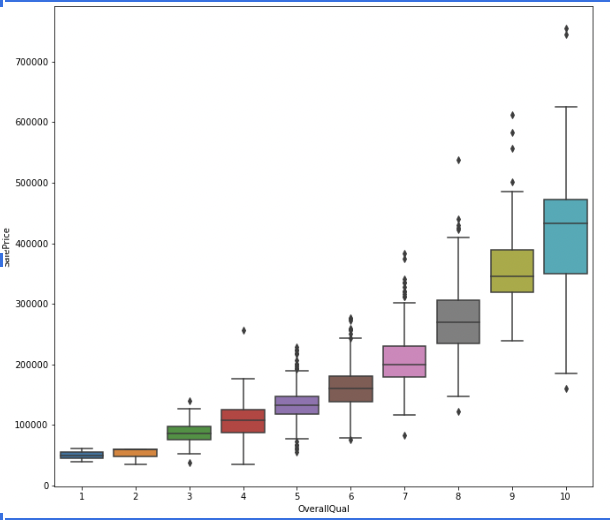
**Data description, EDA and Preprocessing:**

The data include a total 80 columns, and 4 of the dataset misses over 80%. There are 1460 samples and for the test file, it only has 1459 rows and 79 columns because it doesn’t have the target value.

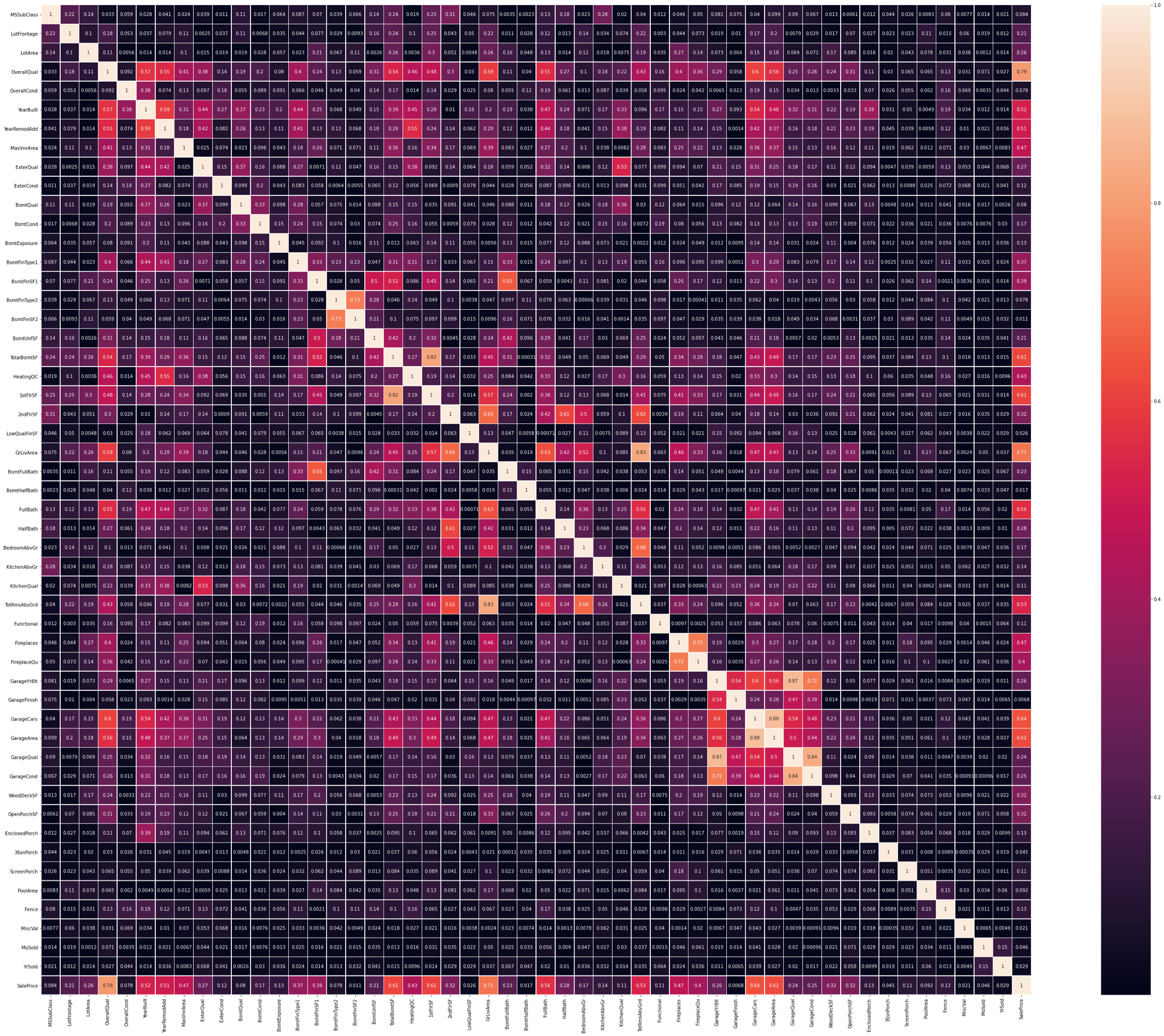
The first thing I want to see is how target values vary. Display the distribution of the target value SalePrice. The data is not normally distributed and right skewed which mean is greater than median.



And the box plot shows SalePrice is positively related with overall quality.



The heatmap of numerical variables to see which variables are most relative to target variables and the overall quality and the living area is most relative.



There are categorical variables and missing values in the dataset. First thing I do is to fill the missing value. I used SimpleImputer to replace missing categorical variables with their mean value and missing categorical value with most frequent values.

And then I used the ‘OneHotEncoder’ to transfer them to dummies and my features increased from 79 to 288.

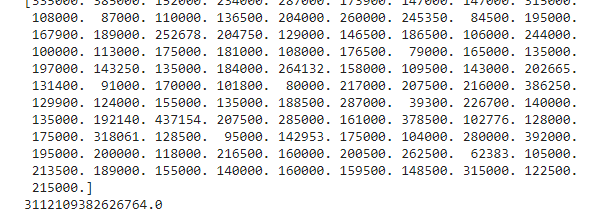
To better fit the following regression model and make it more accurate, I used the ‘MinMaxScaler’ to normalize features value and make it value range from 0 to 1.

Last thing I do is to split the train file to train data and test with a test size equals 0.3. Because the test file has no target value, so I have to build a model,test and sampling on my train file. After the split, the train data is 1022 and the test data is 438.

**Modeling**

**1.Linear Regression**

First model I use is the Linear Regression, I used to use it to figure out feature importance. I decide to see the first 100 prediction values and corresponding true values and compare them and calculate the root mean squared error.



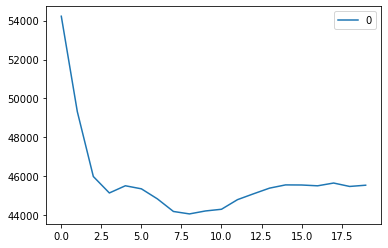
The RMSE value is extremely high because the test value range is 100 which is not the same as the train saleprice’s value range. There are some values too high or too low which are outliers. The reason should be some of the categorical values were transformed to dummies and most of their values may be mostly close to 0 and or mostly close to 1.

I find out the y\_train value’s min and max to replace those values in the prediction which are smaller than min with y\_train\_min and those are greater than max with max. After that, the RMSE value becomes 65256.

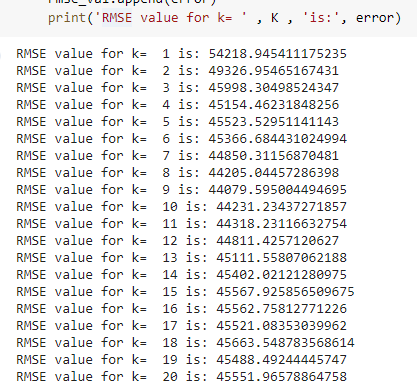
I still think it's way too high so I tried other models.

**2.K Nearest Neighbors**

The Second model I used is the KNN model. I planned to draw an elbow plot so I used a for loop to create an array of rmse values to collect RMSE values of different K-values from 1 to 20.



The elbow point is at round 2. So the best K value is at 2.



According to this result, the RMSE value at k=2 is 49326.95 which is much better than Linear Regression. However, I tried to run the code more times, each time the RMSE for k=1 to k=20 will change, so the RMSE value for knn at k=2 is not stable but will always be smaller than the one in the Linear Regression Model.

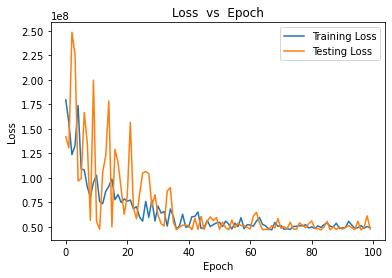
**3.Neural Network**

The library for my neural network is the torch library and I also used LogisticRegression.

At the very beginning, I created the data loader for the neural network using ‘tensor’ and ‘dataLoader’ batch size is set at 128.

The relu activation function I used with a 3-layer structure. Using ‘nn’ to create (288, 1024), (1024,256), (256,1).

Then I try to get the optimal loss value. First, I optimize the neural network with batches of data points and then test the neural network using a testing dataset. I train the neural network with 100 epochs. Using a for loop to find out all epochs can be divided by 10. Last, I plot a training loss and testing loss over time. The range is 100.



From the plot, we can see the model is set well. To calculate RMSE value, I used the last 20 test\_loss values. The RMSE value equals 7076.74 which is the best and much better than the other two models.

**Conclusion and Learning**

In my mind, the most difficult and greatest challenge for me in this project is data preprocessing and the Neural Network Model. There are too many features in the dataset which I never used and a lot of variables are categorical variables. In class,we usually get used to transforming to different numbers like transform [a,b,c,d] to [1,2,3,4]. But in this project I think there are too many categorical variables and the difference between the distance of 1to 4 and 2 to 3 may affect the accuracy of the models. So, I decided to transform them into dummies and then I got 288 features which made the dataset more complex and I need to be cautious when I was working on it. I was not familiar with the Neural Network because of lack of practise, So I spent a lot of time learning how it works.

To Conclude, the Neural Network is best, KNN model is better than Linear Regression in my model. The reason should be the data obtained by the KNN model is always close to each other not like Linear Regression which will get outliers. Replacing those outliers with min and max is not as stable as KNN.

The Neural Network is great for complex dataset.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear Regression | KNN | Neural Network |
| RMSE | 65256 | 49326.95(Not fixed) | 7076.74 |

**Links:**

Competition:

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

Interesting notebooks

<https://www.kaggle.com/amritanigote/housepricing/data>

<https://www.kaggle.com/abdelmenamtarek/house-price-reg>

<https://www.kaggle.com/celniker/house-prices-lgbm-tuned-model-0-12-top-15>

Kaggle notebooks and data:

<https://github.com/ZijianWang12/Intro_to_machine_learning_project_spring2021.git>