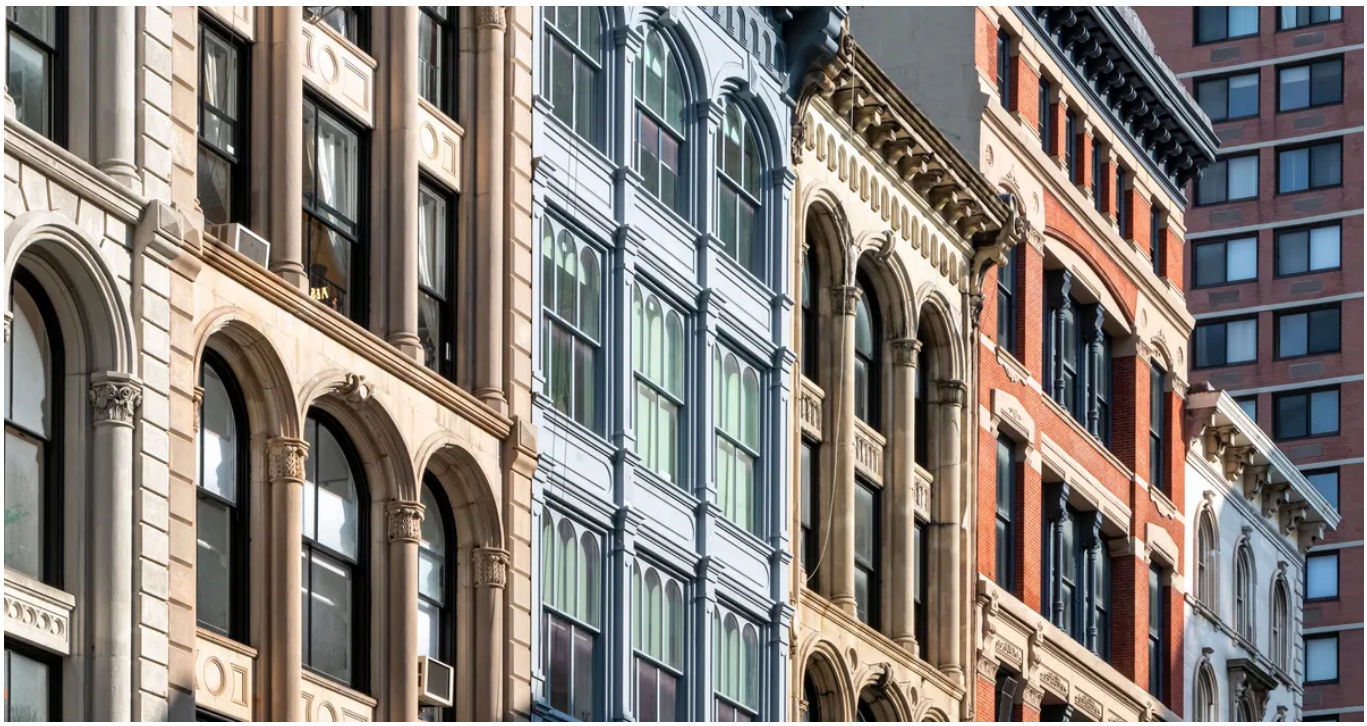
Second Capstone Project

NY Housing prices prediction model

By: Omar Alnimasi



Problem Identification:

We are working with a dataset of housing prices in New York in the year 2017. Using the available dataset, how can we build the best possible model to predict house prices based on the features provided such as the year it was built in, the year it was remodeled, the area, and much more, to provide the best possible model to predict the price. The dataset has 79 different features that we could use to build our model off of. Using this dataset will provide enough variables to use as the basis of our model and create the best possible model for this project. The prices of properties is a vital piece of information for any real estate agency, which is why the model is being built.

The model would have to be as close to the actual prices as possible with minimal variation from the actual prices. Creating a model that has around 90% r squared score is ideally what we are trying to accomplish. We are focused on building the best possible model to predict house prices for our dataset while we test it against some subsets of the dataset to check its validity. However, the dataset has a variety of features some of which are going to be essential and some of which are going to be useless. We need to find the useful features within them and also make sure not to disregard any that are essential. Also, some of the features have over 80% of missing values, and in some cases, it is about 99%, which will require us to figure out the ideal way to deal with that.

The dataset can be found in the following link: https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

Data Wrangling:

As discussed earlier, some of our features are missing most of its data as shown in figure 1.

	count	%
PoolQC	1453	99.520548
MiscFeature	1406	96.301370
Alley	1369	93.767123
Fence	1179	80.753425
FireplaceQu	690	47.260274

Figure 1: features with the most missing data.

The best approach here, which is the approach we chose, is to eliminate all of these features from the dataset as they lack correlation to the price, and that they are missing a big portion of their data. With the other columns, we imputed the mean or a value that made sense, for example, the year it was built is not an ideal column to use the mean or median, so we imputed 9999 to highlight that the data is missing here. We also changed the null values in the object columns with the value “None”, as they implied null means “None” with the documentation provided with the dataset. We kept addressing the missing values according to the documentations whenever we could, and in other cases, we just looked for the best approach to move forward. In this step, we dealt with every piece of null values in our dataset.

Exploratory Data Analysis:

After cleaning up the data, we started to explore it and see the correlation between features and the price. We started with the numerical data, and explored the correlation between each variable with the sale price, and that was done using a heatmap, to illustrate the relation better, as in figure 2.

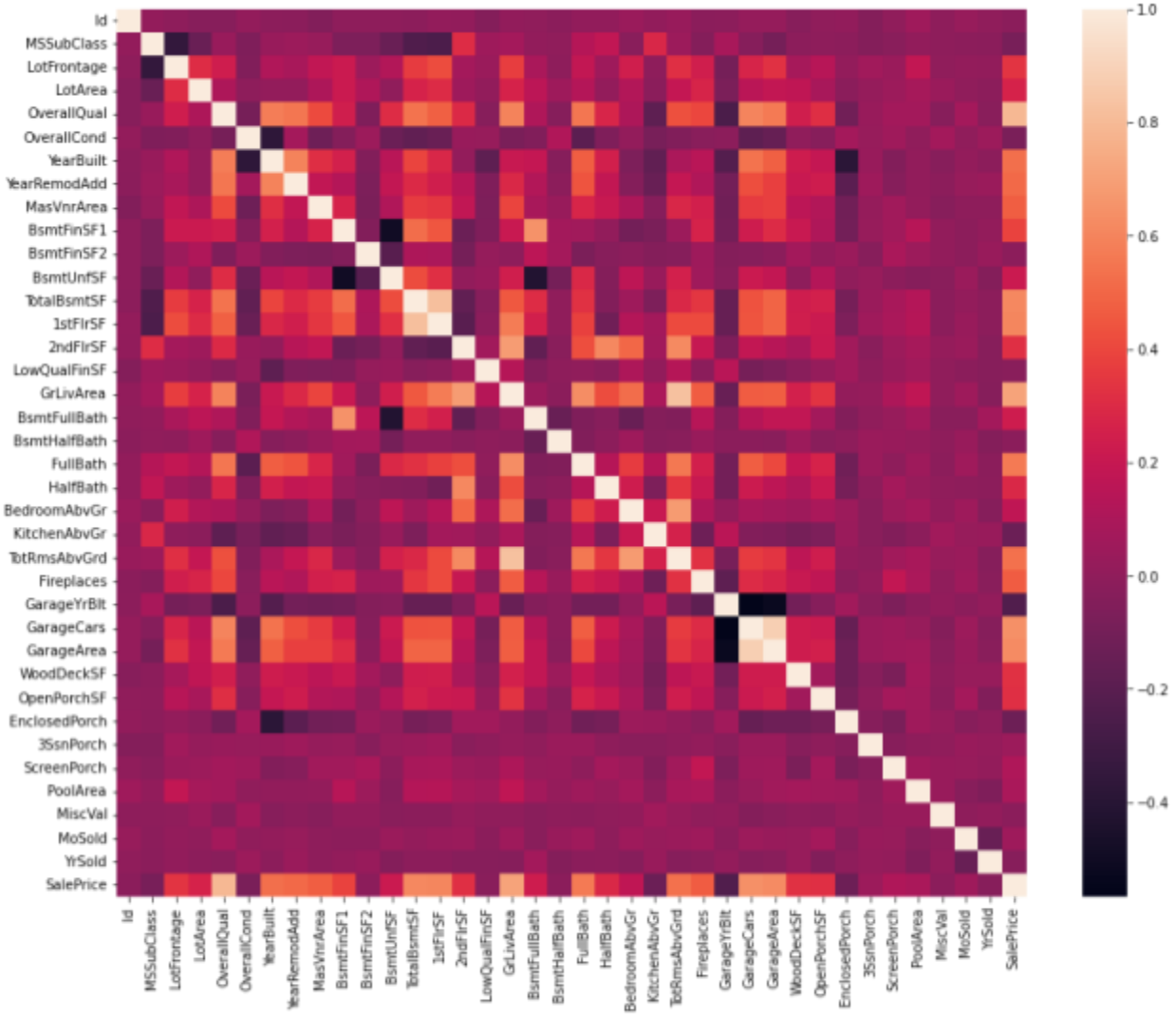


Figure 2: Heatmap of the numerical features.

Using this heatmap, we realized that the most correlated features are OverallCond, YearBuilt, MasVnrArea, 1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd, and GarageArea. We could also see some other features that are correlated to sale price, but they are also highly correlated with other features that we already included as well, so we would want to avoid using them to make sure the effect isn't doubled from these features.

Furthermore, we did some boxplots for the categorical features to see if they had any effect on the price, and the boxplots made it clear that the most correlated features are SaleCondition, SaleType, KitchenQual, BsmtQual, ExterQual, Neighborhood, and MSZoning. The one thing all these categorical features and numerical ones have in common is that they make sense. If you think about what would affect the price of a house, these are the first things that come to mind.

Pre-processing :

After cleaning up the data, and exploring the relation between features. We are now ready to pre-process the data before modeling. The first thing we need to do here is to break down categorical data using get dummy, which turns categorical data to tables of ones and zeros for each category.

For the numerical features, we used the standard scaler and we scaled all of our data using it to make sure that no column would dominate simply for having high values whenever we are building our model

Now that our data is completely numerical we can start by splitting the data into train and test. Here, we went for a 30% test set after shuffling to make sure we have a random 30% of the data in the test set.

Modeling :

Now that we have our data ready, we can start modeling. We started with a linear regression model, and we got a r squared value of 85.5%. We then tested a random forest tree model, which led to an 88.3% r squared value. One idea that was tested after this is to optimize the hyperparameters, this got our model to an 88.66% r squared score.

Finally, we tried gradient boosting with and without hyperparameter optimization. Surprisingly, we got a 88.65% r squared value with optimization, and a score of 88.73% r squared value without optimizing hyperparameters. This led us to the best model out of the models we tried and now we can work with gradient boosting for this project.

Finally, we checked out the best number of features using gradient boosting to see if any number of features is better than using all of them, and as it turns out, the best number is 63, and since we have 66 features, then this means the test ignoring some of the columns in the get dummy area. The effect of features is showcased in figure 3.

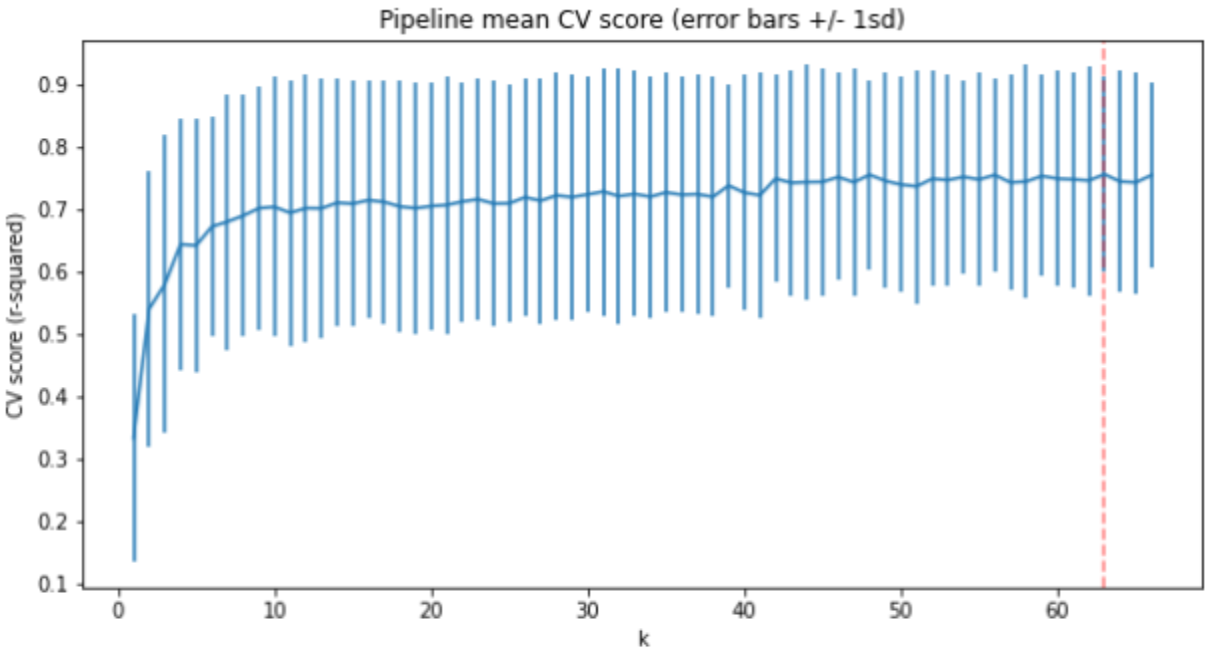


Figure 3: number of features vs cross validation scores.

Conclusion :

After cleaning, exploring, pre-processing, and modeling our data, we reached the conclusion that with a gradient boosting model, we can have an r squared score of about 88.7% to predict the housing prices.