CS539 Final Project

Predicting House Prices in London

Irakli Grigolia

Fall 2022

Abstract

This is a final project for the CS539(Machine Learning) class offered in fall of 2022 at Worcester Polytechnic Institute. Dataset for house prices in London was chosen and different ML algorithms applied in order to find the best one for predicting the price of unknown house in London based on different variables(features). Mean Absolute Error(MAE) and Mean Absolute Percentage Error(MAPE) were used for evaluating the performance of the model. The technique that performed the best is XGBOOST with MEPA of 17% and MAE of $ $238860

Overview and Motivation

Goal of this project was to create a Machine Learning model that can predict housing prices in London,UK. My plan was to find out the degree of correspondence between Location/Postal Code and the price of the house. Obviously, size plays a big role in price estimation but I wanted to know how big of a factor was its location. It is interesting to figure out which feature combinations gave the best price estimate. I thought all of the variables in this dataset should have played a significant role except the "No. of Receptions" but the main ones were "Location/Postal Code","Area in sq ft", and the "House Type".

The main reason I chose this project is that, In Georgia, the country I am from, there is no automated way of estimating house prices. A person needs to go and check every house manually. So I wanted to simplify this process and maybe create an app(depends on performance I get) where a user would put in all the different variables and get the price. A friend of mine works in one of the banks in Georgia an he has been talking about this problem for quite some time already so I thought it would be a fun project to work on. Plus he can provide me a bigger dataset later.

Related Work

While trying to gather various materials that would help me, I stumbled upon an article on"towardsthedatascience.com". It is called "Predicting House Prices with Machine Learning”. Here is the link: https://towardsdatascience.com/predicting-house-prices-with-machine-learning-62d5bcd0d68f . The main goal of that project was similar to mine so it helped me to get started and showed me the outline and general structure that my own project should have.

In addition to that article, I found this paper to be an interesting read as well : https://www.hindawi.com/journals/sp/2021/7678931/ .

Initial Questions

The question I wanted to answer is "Can we predict with 80% accuracy the price of the house in London depending on House size, Location. Number of bedrooms ,Total Number of Rooms, House Type and/or Postal Code".

Machine learning question it answers is : How Much?(what is the price?).

After several iterations of this project, the question remained the same, but method answering the question changed a bit.

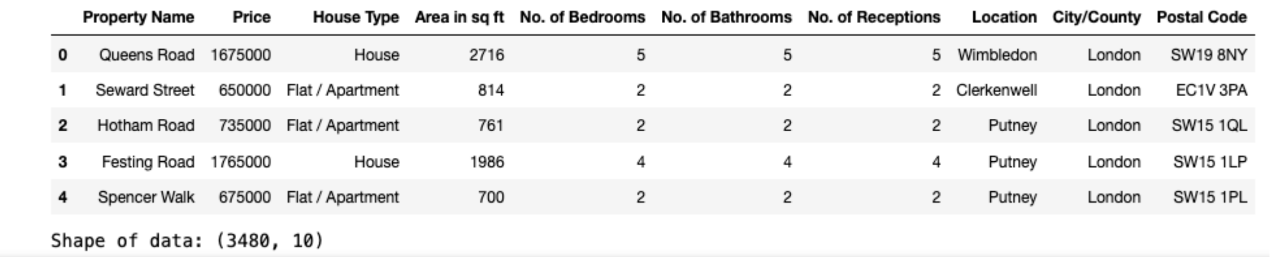
Data and EDA

Dataset

The dataset I used is called "Housing Prices in London" which I found on Kaggle. Here is the link: https://www.kaggle.com/datasets/arnavkulkarni/housing-prices-in-london

EDA

Dataset has 3480 data points and I planned to use all of it if there were no issues with samples. It has 10 features:*"PropertyName","Price","House Type", "Area in sq ft", "No. of Bedrooms", "No. of Bathrooms, "No. of Receptions", "Location","City/"County", "Postal Code"*. I encountered no issues with getting and importing the data.



	Property Name	Price	House Type	Area in sq ft	No. of Bedrooms	No. of Bathrooms	No. of Receptions	Location	City/County	Postal Code
0	Queens Road	1675000	House	2716	5	5	5	Wimbledon	London	SW19 8NY
1	Seward Street	650000	Flat / Apartment	814	2	2	2	Clerkenwell	London	EC1V 3PA
2	Hotham Road	735000	Flat / Apartment	761	2	2	2	Putney	London	SW15 1QL
3	Festing Road	1765000	House	1986	4	4	4	Putney	London	SW15 1LP
4	Spencer Walk	675000	Flat / Apartment	700	2	2	2	Putney	London	SW15 1PL

```
Shape of data: (3480, 10)
```

Data Cleaning

The first thing I noticed was that the # of Bedrooms, Bathrooms, and Receptions are equal for all the houses which I thought was odd. I changed their values randomly to either 1 or 2 to make dataset more diverse. A House can have more bathrooms especially if it is a big one, but randomly generating numbers would not work in this case because we could have gotten a house with 2 bedrooms and 3 or more bathrooms which would not make any sense so I chose to stick to maximum of two bathrooms per house. Same reasoning went behind changing "No.of Receptions values."

Dataset had 962 data points with missing value for Location column so I decided to drop those data samples and I was left with 2518 data points. Next, I searched for outliers and found a few in ‘Price” and “Area in sq ft” columns so I got rid of them to have data more uniformly distributed.

```
lt.show()
```

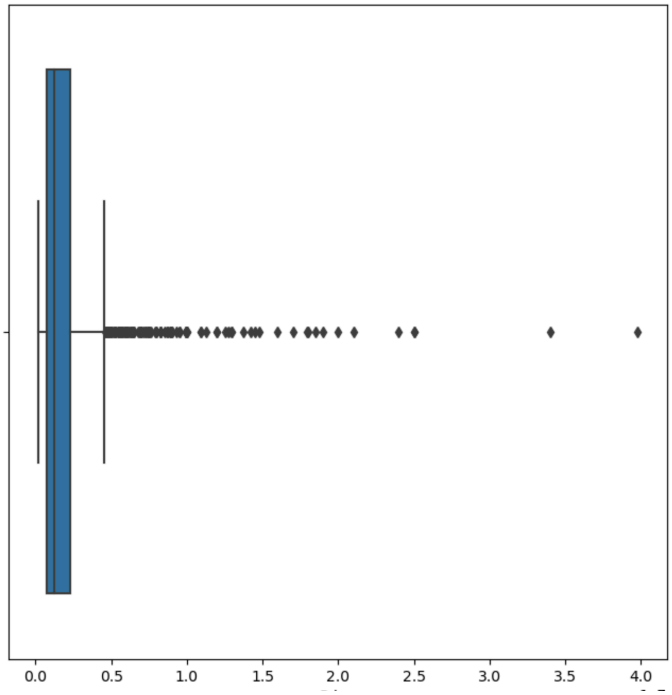


Figure 1: Boxplot of ‘Price” columns

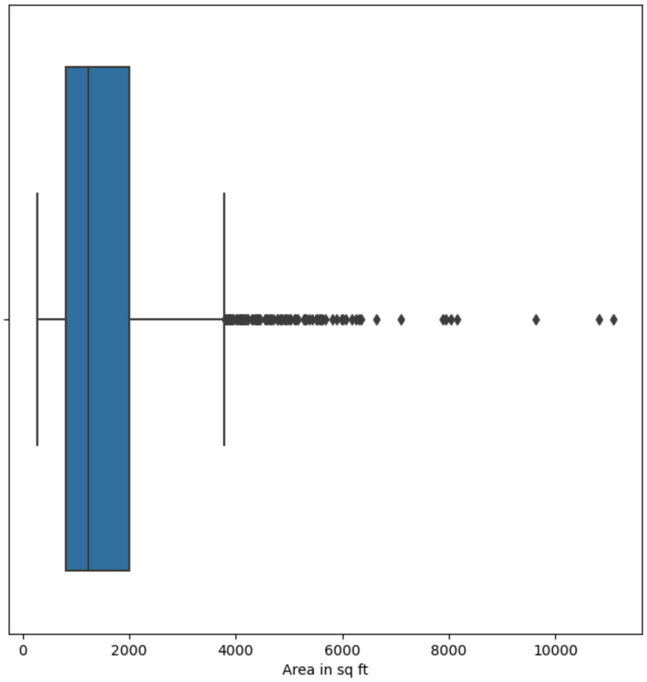
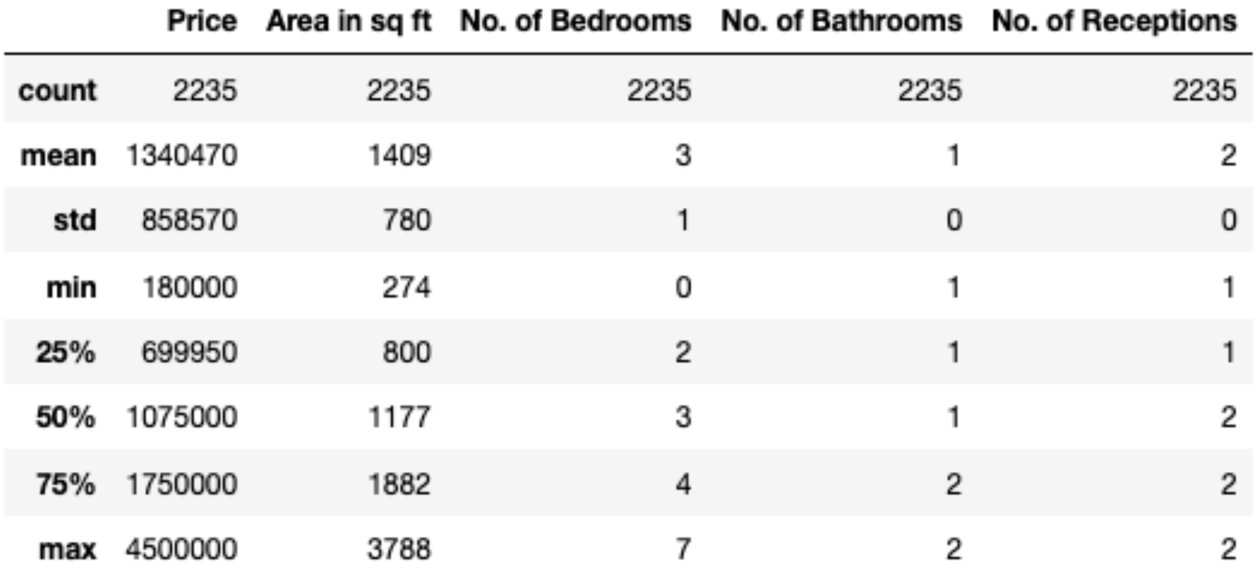


Figure 2: Boxplot of “Area in sq ft”

After Removing the outliers I was left with 2235 data points in my dataset and here is the summary of it.



	Price	Area in sq ft	No. of Bedrooms	No. of Bathrooms	No. of Receptions
count	2235	2235	2235	2235	2235
mean	1340470	1409	3	1	2
std	858570	780	1	0	0
min	180000	274	0	1	1
25%	699950	800	2	1	1
50%	1075000	1177	3	1	2
75%	1750000	1882	4	2	2
max	4500000	3788	7	2	2

Figure 3: Summary of Dataset

After exploring the data a bit more from the plots which can be found in .pynb file in my GitHub repository, I found out that

1) the highest correlation was between “Price” and “Area in sq ft” but that was before I converted categorical features to numerical so it made sense so far.
2) 30.7% of the houses/apartments have 2 bedrooms which is the largest number, 20.6% of the data samples have 3 bedrooms and having 8 or more bedrooms is almost an outlier
3) Top 3 most popular city/county is London, Surrey ,and Middlesex.

Model Revision

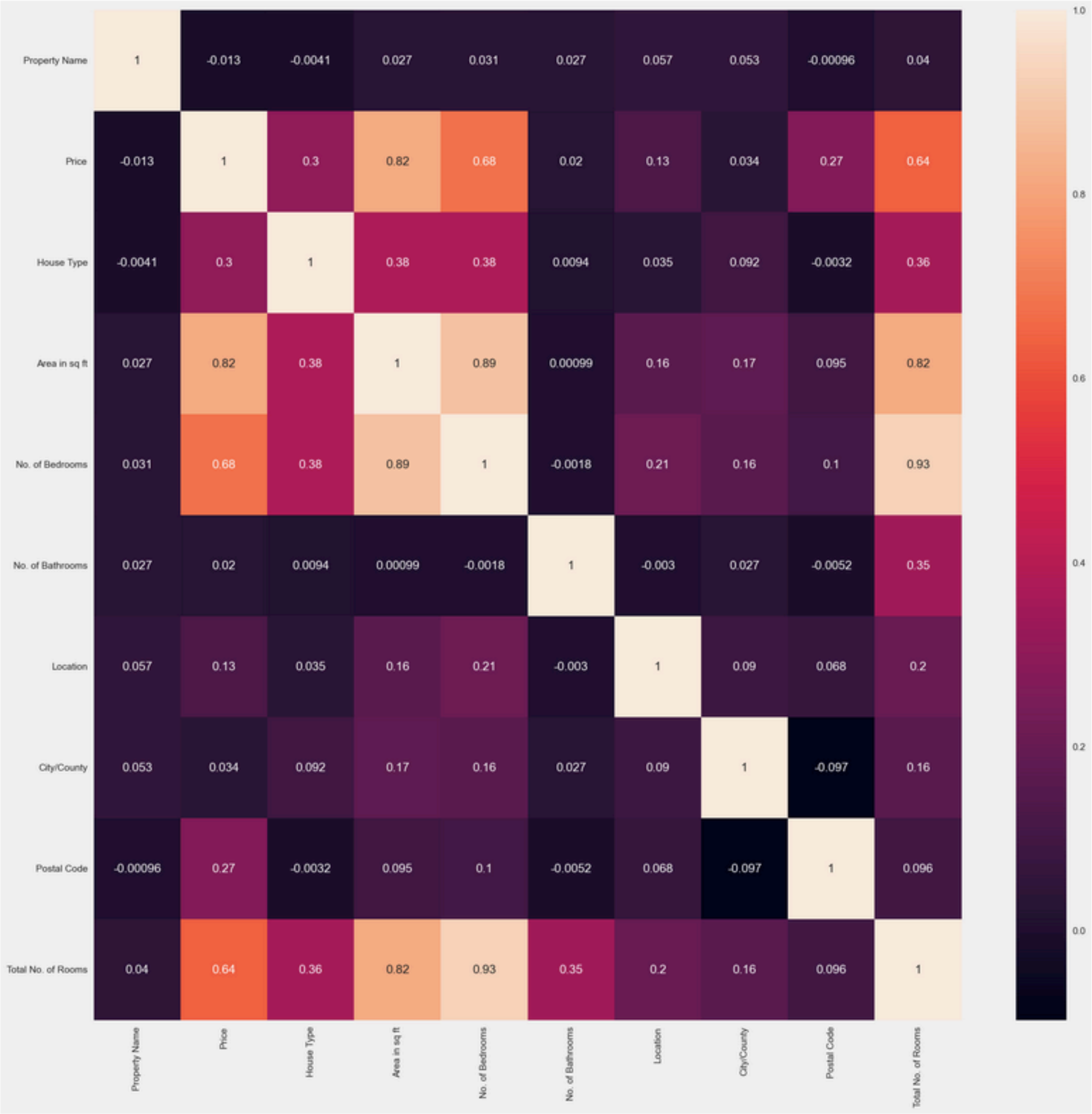
Next, I decided to transform categorical features such as ‘Location’, ‘House Type’, ‘Postal Code’, ‘Property Name’, ‘City/County’ to numerical ones so I could use them for training the model. I used Label Encoder for transformation, then got rid of the ‘No. Of Receptions’ feature because it did not seem to add any value since it was randomly generated and could potentially make model perform worse. Next, I created a new feature called ‘Total No. Of Rooms” which is number of bedrooms + number of bathrooms. After that I plotted the correlation matrix using Spearman’s correlation.

Spearman’s correlation

Spearman’s correlation is a non-parametric rank correlation measure.

It evaluates how effectively a monotonic function can capture the connection between two variables. While Pearson's correlation evaluates linear relationships, Spearman's correlation evaluates monotonic relationships; the Spearman's correlation between two variables is equivalent to the Pearson correlation between the rank values of those two variables (whether linear or not). A perfect Spearman correlation of -1 or 1 happens when one variable is a perfect monotone function of the other when there are no repeated data values.

It seems sense that the Spearman correlation between two variables will be higher when observations rate the two variables similarly and lower when observations rank the two variables differently.



	Property Name	Price	House Type	Area in sq ft	No. of Bedrooms	No. of Bathrooms	Location	City/County	Postal Code	Total No. of Rooms
Property Name	1	-0.013	-0.0041	0.027	0.031	0.027	0.057	0.053	-0.00096	0.04
Price	-0.013	1	0.3	0.82	0.68	0.02	0.13	0.034	0.27	0.64
House Type	-0.0041	0.3	1	0.38	0.38	0.0094	0.035	0.092	-0.0032	0.36
Area in sq ft	0.027	0.82	0.38	1	0.89	0.00099	0.16	0.17	0.095	0.82
No. of Bedrooms	0.031	0.68	0.38	0.89	1	-0.0018	0.21	0.16	0.1	0.93
No. of Bathrooms	0.027	0.02	0.0094	0.00099	-0.0018	1	-0.003	0.027	-0.0052	0.35
Location	0.057	0.13	0.035	0.16	0.21	-0.003	1	0.09	0.068	0.2
City/County	0.053	0.034	0.092	0.17	0.16	0.027	0.09	1	-0.097	0.16
Postal Code	-0.00096	0.27	-0.0032	0.095	0.1	-0.0052	0.068	-0.097	1	0.096
Total No. of Rooms	0.04	0.64	0.36	0.82	0.93	0.35	0.2	0.16	0.096	1

Figure 4: Correlation Matrix

As it can be seen from the figure 4, correlation greater than 0.5 that price has is only with ‘Area in sq ft’(0.82), ‘No of Bedrooms’(0.68), and with new feature I added ‘Total No of Rooms(0.64). Other features have either really small or even negative correlation. One possible explanation could be that we are just unable to capture the relationship between those feature which probably is not linear.

Performance Metrics

For the evaluation metrics I used Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE).

1. Mean Absolute Percentage Error is one of the most common metrics of model prediction accuracy and the percentage equivalent of MAE. MAPE measures the average magnitude of error produced by a model, or how far off predictions are on average.

2. Mean Absolute Error is the magnitude of the difference between the individual measurement and the true value of the quantity is called the absolute error of the measurement. The arithmetic mean of all the absolute error is taken as the mean absolute error of the value of the physical quantity

Using MAPE we can evaluate accuracy of our model. So for example if MAPE value is ~0.5, that means that our model is correct only approximately 50% of the times which would not be good. On contrary, if MAPE score is 0.2 or less that means that model accuracy is 80% or more and that means that we achieved the score we initially aimed for which would be a success.

In addition to MAPE using MAE we can tell by how much our model misses the target. For example if MAE score is 200000 that would mean that on average our model is off by $20000.

Goal is to get both MAPE and MAE scores as small as possible.

Other measures I looked into were Median Absolute Error, R2_score and Mean Poisson Deviance, but in the end I chose MAPE and MAE because I found them easier to interpret.

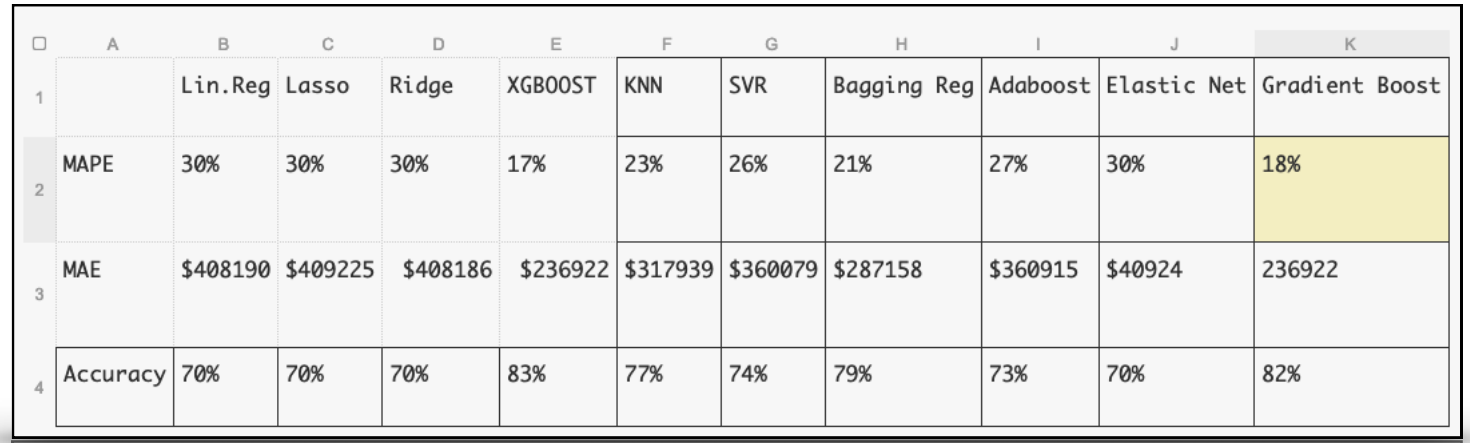
Methodology, Results and Final Analysis

I wanted to compare performances of different algorithms and choose the best one. I tried 10 different ones including : Linear Regression, Lasso and Ridge Regression, AdaBoosting, Regression Boosting, XGBR, KNN. I compared their accuracy based on MAPE and MAE.

The whole process was the following: Data Cleaning + Log Transformation + Standard Scalar to normalize the dataset and use different algorithm to predict the price of the house.

Different plots of the results can be found in .pynb file in my GitHub repository.

Below is the results of all the algorithms.



A
B
C
D
E
F
G
H
I
J
K
1
2
3
4

	Lin.Reg	Lasso	Ridge	XGBOOST	KNN	SVR	Bagging Reg	Adaboost	Elastic Net	Gradient Boost
MAPE	30%	30%	30%	17%	23%	26%	21%	27%	30%	18%
MAE	$408190	$409225	$408186	$236922	$317939	$360079	$287158	$360915	$40924	236922
Accuracy	70%	70%	70%	83%	77%	74%	79%	73%	70%	82%

Figure 5: Results

Project summary In steps:

1) Found the dataset on Kaggle.

2) Performed data cleaning. Dataset was not ideal to begin with because number of Bathrooms and Receptions were always equal to the number of Bedrooms for every house, which is not the case in real life therefore indicating that there was something wrong with the data. To fix it I got rid of "No. of Receptions" feature and randomly generated bathroom number for every house in a range from 1 to 2 that made it more realistic.

3) Date had number of missing values so I got rid of them as well (962 data points in total).

4) Analyzed different statistics of the data to get better understanding of the dataset.(e.g highest and lowest priced house, most popular location, etc.)

5) As it can be seen from the plots in .pynb file, dataset had a few outliers which I got rid of.

6) Visualized the dataset with various graphs and plots(e.g histograms, boxplots, etc.)

7) Used Label Encoder to convert Categorical values to numerical.

8) Looked for correlations between different features.(Price has highest correlation with Area and with Number of Rooms.

9) Added new feature called "Total No. of Rooms" which is bedrooms + bathroom number.

10) Chose Mean Absolute Percentage Error and Mean Absolute Error for performance metrics.

11) Tried different ML models and compared their performance.

12) Out of 10 different algorithm, the one that performed the best is XGBOOST with MAPE of 17% and MAE of $238860.

Conclusion

In the end having 83% accuracy is good but I think to improve it and have lower MAE more data is needed(2518 data points is not enough).In addition I think the number of features were not enough as well, first of all, there was an issue with 2 features(No.of Bathrooms and No. of Receptions) which were exactly the same for every data point and I had to randomly generate them which did not help and even maybe made it worse. Secondly, much of the categorical features such as Location, Postal Code, City/County are basically different name for the same feature. Each of them identify the location of the house so I don't think they added much value at all.With more features and more data I believe my model would perform much better.

In Conclusion, it is definitely possible to predict a house price with relatively good accuracy and low margin of error; the best algorithms to use from what I have observed are ensemble models that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The one thing I regret is that I did not use Neural Nets properly. I tried a simple architecture but it was performing really bad and I decided not to included it at all. I did not have enough time to come up with more complex one and try fine-tuning and testing it.I want to do it on my own, outside of this project scope. I will upload the results if I see any performance improvements.

In a real-world, there are many other factors/features that play a big role in estimating the house price, for example in what condition is the house, how old is it, etc. If I want to continue working on this project, I would need to get a better dataset with more features, ideally with some images to evaluate in what condition the house is in. But that is for the future. The main goal of this project for me was to get the idea of what is possible and have some hands-on experience with working on Machine Learning project.

Overall, I believe that the project can be considered successful, because I got the accuracy which I intended from the beginning, even though, for sure there is a room for improvement.