Introduction:

Housing in India involves

Methodology:

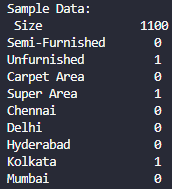
The data was divided into 10 categories which determines the features of each real estate property on various cities in India. Categories such as the Size of the area, Number of Bedrooms, Hall, and Kitchens, etc. For the Purposes of Experimentation, four out of eleven features of each of the houses were initially used in the training of the Regression Model. These Features are Size, Area Type, Furnishing Status, and City. The rationale behind using these features is their ease of re-encoding. In the case of Size, having a larger area for available for rent, can be beneficial for the tenant. Mainly due to business opportunities that could arise from having a large size. At the same time, it is also beneficial for the landlord, as they can charge more for their property. Meanwhile, for the other chosen features, it mainly boils down to the lack of outlier values they possess.

Experiments:

The first modification as seen above was change the features that are retained during the pre-processing step. Initially, for the first model, BHK, Bathroom count, Furnishing Status, and Rent were used as the features of the first model. Furnishing Status was re-encoded using One-Hot encoding in order to obtain the single bit representation of each furnishing status category. 20% of the data was set aside for testing and afterwards, it was standardized to normalize the distribution of the data. Using these features, we have achieved the following Mean Squared Error and Coefficient of determination r^2:



Afterwards, Forecasting was done with the following Data Point:



The results we’re unspectacular to say the least as seen below:



<insert comments here>

We will use these values as a baseline for future experiments and tweaks to our model.

The first tweak done to the model was to change the features that will be used to build the model. As stated above: Size, Area Type, Furnishing Status, and City we’re used as the new features of the model. The only other change done was to also re-encode Area Type and City to One-Hot encoding. Besides that, no other tweaks we’re made in splitting the data or anything else. Despite that, we have achieved a somewhat significant improvement as seen below:

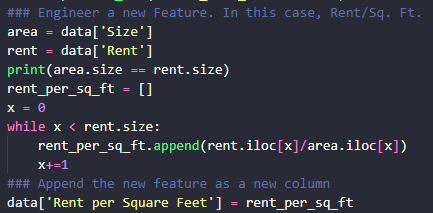


The mean squared error is still quite big, meaning our model still is not very accurate the data, but we we’re able to shave off a modest 793,573,577.45 over our loss, while improving the coefficient of determination by .20 points.



Afterwards, engineering a new feature was done. In this case, computing the value of the rent per square feet. As one of the things I have noticed was that the current rent values were not being considered by the model creation process. The rent per square feet is simply done by the ff. formula:

And was implemented with the ff. code snippet below:



Where the Size and Rent columns were isolated from the dataset, and iterated over when doing the computations.

Doing this reduced the mean squared error by a further 478,097,999.26 and increased the fit from 0.52 to 0.64. However, using the new model with revised weights barely improved the forecasted rent of the example:

