Importing libraries is part of the process but what is more important is the feature engineering steps that you choose and apply on the dataset.

In the Bangalore house price prediction train dataset, there were quite a few missing values in features like society, balcony, bath, size and location. While society was an encrypted feature as explicitly mentioned in the summary of the hackathon, ID and availability (imbalanced with the category) was something I noticed that could be dropped off which does not have much implication on the price of the house.

Now, we could drop the rows containing missing values because the composition of these values could be ignored when compared with the total size of the dataset.

The size feature had formatting issues and also was categorical in nature. So, I split the values based on a common character and fed the integer value as required in another column and dropped the original column.

Created functions and applied them to convert the total\_sqft feature considering it had values in different units in the original dataset and then typecasted the feature to float datatype.

Then performed outlier treatment since the price did not match the size of the house of the bath that the house offered. So dropped rows where the size and bath of the house was greater than 15.

At this point, I had successfully treated my train dataset for all the inconsistencies.

Time to utilize the other datasets that were provided (avg\_2bhk\_rent and dist\_from\_city\_centre).

Merged these two new datasets based on the location feature available in our rain dataset and saw that both the newly added features had considerable number of missing values.

To treat the missing values in the avg\_2bhk\_rent, I applied a simple logic of mean of the avg\_2bh\_rent \* 1.5 times the value because while the original column had values for only 2BHKs, our main dataset had houses with more than 2BHKs. Hence the said logic was applied to make the avg\_rent values for 3BHK and 4BHK (comprising majority of the houses) as reasonable as possible.

To treat the missing values in dist\_from\_city\_centre, I applied the mode value of the feature.

Now I also scraped the website makaan.com using beautiful soup for additional feature price/sqft based on location and cleaned the newly created dataset which was categorical in nature and also converted the unit to lakhs to be on the same unit as our target variable and then merged this new data with the main train dataset. Treated the missing values in this scarped data using the mean of the feature.

Now comes the encoding of the area type and location columns which are categorical. Used Ordinal Encoding on area\_type considering the ordinal nature of the feature while label encoded the location column.

Now split the dataset into independent and target datasets to apply and fit the XGBoost Regression model.

(I tried multiple regression models and found that XGBoost Regressor performed the best based on my feature engineering steps).

Now comes the application of these methods on the test dataset. We need to perform the same feature engineering methods done on the train dataset because in real world case we will not have the scope of treating the new incoming data points. So, the best option is to test the model by applying the same feature engineering steps carried out on the train dataset.

However, when we come to encoding the location column, there are additional values in the test dataset that were not present in the train dataset. To tackle this situation, I identified these values and categorized them as a new class named <unknown> and then appended this new class to the label encoder classes that were identified from the train dataset.

We should note that, we are not allowed to drop any rows in the test dataset because in real world we will be applying the model on ever new datapoint and not ignore it. Ignoring could lead to business losses. Also, the hackathon submission needed us to submit the prediction for all the rows that were present in the test dataset.

Now comes the time to bring back the ID column as it is required to check the RMSE predicted for our test dataset.

Since I did not shuffle my data or applied train test split on my datasets, I was allowed to directly bring in the index of the test dataset as my ID column. I created a new dataframe with this ID column and applied the model’s predict method on my transformed test dataset to get the price of the houses.

This dataset was then finally written on a .csv file as per the requirement of the hackathon.