I have attached a python notebook in the email. Please do take a look at it. These are the operations that I have performed.

- The dataset consists of 1042 rows and 20 columns. This is a regression problem where we

can the target variable is 'price' which I have predicted using Machine Learning Modeling.

- Dropped the columns 'id', 'time_created','time_updated','external_id','url','latitude' and 'longitude' from the dataset, as these variables do not provide information significant in modeling.

- Here I have observed that the variable 'status' has only one value throughout the dataset i.e. 'active', hence I have can drop this variable as it is not providing us significant information.

- I observed that the variables 'bedrooms' ,'bathrooms', 'garages' ,'parkings' ,'offering' ,'erf_size' ,' floor_size' have missing values and the target variable 'price' also has missing values. Hence I took care of this by filling the missing values of the independent features and the target variable.

- After making the above observation I filled the two rows which have value '[None]' in the property_type column with 'house' as the value for the'agency' variable for these rows is 'rawson' and the mode for the variable 'property_type' for the agency 'rawson' is 'house' and also mode for the 'property_type' variable for the area 'Constantia' is also 'house'

- Predicted the missing Values Using Imputers From sklearn.preprocessing

- Here I used the KNNImputer to fill the missing values in the variables 'price', "garages","parkings","erf_size","floor_size" by predicting the values using the KNNImputer library.

- We go through a range of values from 1 to 20, for the parameter 'n_neighbors' in the KNNImputer, as we want to find which value of 'n_neighbors' gives the maximum value of correlation between the target variable 'price' and the feature 'floor_size'. The reason I have selected the variable 'floor_size' to calculate the correlation with the target variable 'price' is that, before imputing the missing values the target variable 'price' had the highest corrleation with the independent variable 'floor_size' which was 0.5319914806523912. Now I am finding the maximum correaltion value between the target variable 'price' and the variable 'floor_size' after the missing values are imputed using the KNNImputer, for different values of the parameter 'n_neighbors' and then compare it with 0.5319914806523912, whcih is the correlation for the original dataset whcih consists of missing values.

- Here we observe that the maximum correlation between the target variable

'price' and the independent variable 'floor_size' is 0.4233518730063556, when the value for 'n_neighbors' is 6. This value is less than the value of correlation for the orignal dataset, hence

we move on to another Imputer to fill the missing values as after the missing values were filled using the KNNImputer the correlation decreased whcih is not desirable.

- Here we observe that the correlation between the target variable 'price' and the independent variable 'floor_size' is 0.6703992976511615 after the imputation of missing values using IterativeImpueter. This value is more than the correlation value for the original dataset. Hence we allow the imputation of the missing values using IterativeImputer into the orignal dataset.

- Now while filling the variable 'bathrooms' and 'bedrooms'; there are 4 and 14 NaN values respectively. Hence I have decided to fill the values on a case by case basis. I have decided to fill the 'NaN' values based on their 'property_type'. So for filling the 'bathrooms' variable which has 'property_type' as 'house', I have filled these values with the mode for the 'bathrooms' and 'bedrooms' variable. Similarly I have done the same for the other 'property_type' 'apartment'.

- Performed Data Visualizations for the features to draw more insights.

- Here, you can see outliers in the target variable 'price' from the above figure. While price outliers would not be a concern because it is the target feature,the presence of outliers in predictors, in this case there aren't any, would affect the model's performance. Detecting outliers and choosing the appropriate scaling method to minimize their effect would ultimately improve performance.

- From the correlation matrix, we can see that there is varying extent to which the independent variables are correlated with the target. Lower correlation means weak linear relationship but there may be a strong non-linear relationship so, we can't pass any judgement at this level, let the algorithm work for us.

- Build the regression models Linear Regression, XGBoost, AdaBoost, Decision Tree, Random Forest, KNN and SVM.

- Performed Hyperparameter tuning for all the above algorithms.

- Predicted the prices using the above models and used the metrics RMSE, R -square and Adjusted R-square.

- As expected, the Adjusted R^2 score is slightly lower than the R^2 score for each model and if we evaluate based on this metric, the best fit model would be XGBoost with the highest Adjusted R^2 score and the worst would be SVM Regressor with the least R^2 score.

- However, this metric is only a relative measure of fitness so, we must look at the RMSE values.

- In this case, XGBoost and SVM have the lowest and highest RMSE values respectively and the rest models are in the exact same order as their Adjusted R^2 scores.

- This further confirms that the best fit model for this dataset is XGBoost and the worst fit model is SVM.

- Plotted the actual v/s predicted prices.

- Looking at the plots of actual vs predicted prices, you can also see that the data points in XGBoost are closer to each other.