**CAPSTONE PROJECT**

**Problem statement:**

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, you want to sell a house and you don’t know the price which you can take — it can’t be too low or too high. To find house price you usually try to find similar properties in your neighbourhood and based on gathered data you will try to assess your house price.

1. **Summary of problem statement, data and findings**

**Abstract:** The proposed project is about determining the ‘price’ of a house based on all the features of the available dataset and not only on the location and square footage. The problem statement also clearly states that the value of the house is not being predicted only from the buyer perspective but also from the seller perspective with the given dataset to decide on the correct ‘price’ to be tagged for the house. To understand the given dataset we started our work by analysing it in excel and we were able to identify outliers and wrong or missing values in few attributes of the dataset. We then performed univariate analysis which showed us that almost all the attributes are right skewed with uneven distribution and bivariate analysis showed 100% correlation between total\_area and lot\_measure, so one attribute can be removed during the model development phase. The bivariate analysis also showed that few attributes had good correlation with the target variable ‘price’. With this basic information we started developing a model using different methods, as the dataset was continuous we found that only few methods supported it. Out of the different model developing methods Gradient Boost(GB) and Random Forest(RF) are the two methods which gave a better model score for the existing dataset. So we utilized these two methods for getting the score of the tuned model. We used 5 iterations for tuning the model. With the proper tuning, the dataset was able to provide 89.3% as the maximum model score for predicting the ‘price’ with only 17 attributes out of 22 attributes. So our tuned model can perform with almost 90% accuracy for predicting ‘price’ with the given dataset.

1. **Overview of the final process**

* The objective of the proposed project was to determine the price of a house using the given data set. Before the data set was tested using python notebook, we analysed the given data set in excel workbook using filter which gave some interesting insights on the given dataset,

1. The room\_bed attribute had a value 33, which when compared with the total area, indicates this value could be an outlier.
2. The room\_bath(no of bathrooms/no of bedrooms) attribute has some integer non zero value when the room\_bed has ‘0’ value, which indicates wrong values in the data set.
3. The attribute lot\_measure15 is only a part of the total area, but in few cases the value is greater than total\_area which is not possible.
4. The attribute ceil (Total floors (levels) in the house) has few values in decimals. How can the total floors be in decimals
5. The attribute year\_renovated has few values ‘0’ but the lot\_measure15 has greater value than lot\_measure in few cases. This indicates that few values of ‘0’ might be missing values in year\_innovated.
6. The correlation between attributes is not clear from excel workbook, which can be seen using python notebook.

* With this inference it made us easy to start our analysis for the given dataset.
* The univariate analysis showed that almost all the attributes are right skewed with uneven peaks. The boxplot was used for all the attributes to find the outliers. Except 3 attributes(ceil, yr\_build, zipcode) all the attributes were having outliers, which can affect the performance of the model for the given dataset. We also found the count of the outliers which can help us for further analysis.
* The bivariate analysis was done using ‘Pairplot and Correlation heat map’. This clearly showed, a. total\_area and lot\_measure has a correlation of '1' so any one column can be neglected.

b. living\_measure, quality, ceil\_measure, furnished & roon\_bath have better correlation with the target variable ('price'). So for predicting the ‘price’ of a property these attributes play a major role in this dataset.

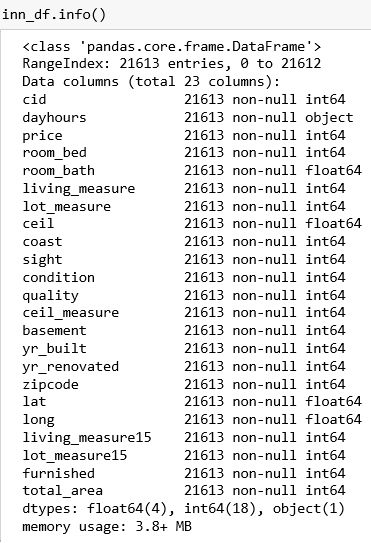
* With this information we started developing a model using different methods. While testing the given dataset using a model we found out that the given dataset is **continuous** and so the dataset works fine only for few model developing methods and for others it gives a score of only less than 1%. Among all the model developing methods **Gradient Boosting (GB) and Random Forest (RF)** are the two methods which provides better model score compared to other methods. So with these two methods we started fine tuning the model using different iterations. The different iterations were,

1. Finding the model score for the given dataset without doing any changes on the given dataset.
2. Removing the outliers using (mean+3\*SD) on the existing dataset and developing a model using GB and RF method.
3. Replacing ‘Zeros’ from the attributes which are providing wrong information with their respective column medians and developing a model using GB and RF
4. Using PCA to analyse the minimum number of attributes needed for developing a better model using GB or RF which can provide atleast 95% variance.
5. Analysing the model performance using GB and RF by removing attributes.

Thus by performing all these iterations on the given dataset, a tuned model was developed for predicting the target column ‘price’ with a maximum score of 89.3%. The model used around 17 attributes out of 22 available attributes for providing the maximum score.

1. **Step-by-step walk through the solution**

* As a first step we imported the dataset to the python notebook.
* Then we used the info() function to see the type of each attribute which helped us to sort between int, float and object.



From the info() function we can clearly see that the attribute ‘dayhours’ is of the type ‘object’, so using this attribute data analysis is cannot done in python notebook. Either this attribute has to be converted to integer or float type if it is going to provide more value to the dataset or it can be removed if the attribute is not that important. In this dataset ‘dayhours’ attribute is not that important so it can be removed for further analysis.

* Then we used the describe function to find the statistical values for each attribute. The statistical values helped to understand the distribution of the dataset.
* Before starting the univariate analysis we removed the cid and dayhours attributes as they were not support the model in predicting the target column, (i;e predicting the price of the house).

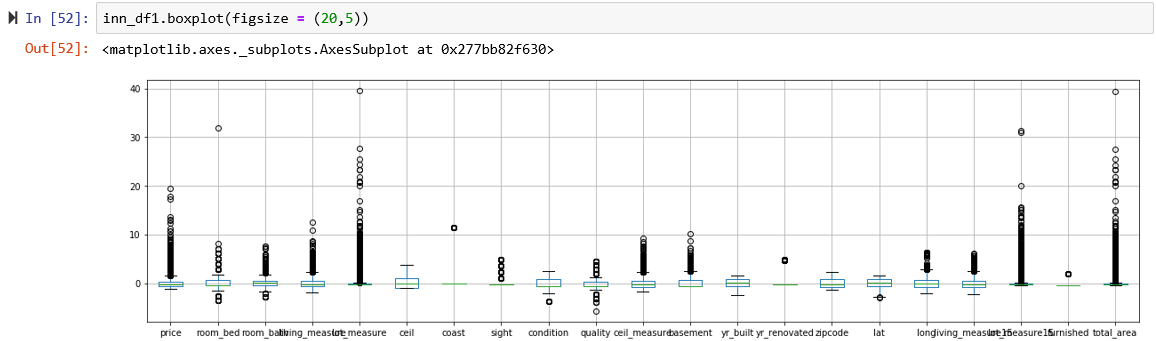
**Univariate Analysis**

* The univariate analysis was performed with the distance plots to find the distribution of each attribute.
* We see that almost all the attributes are right skewed, this might be because of the outliers in the attributes
* To find whether my attribute is normally distributed or not we used skewness and kurtosis to check whether it is within the range of [-2,2]
* To find the range of outliers we used boxplot.
* We also measured the count of outliers, so that it helps us see how much amount of outliers can be removed.

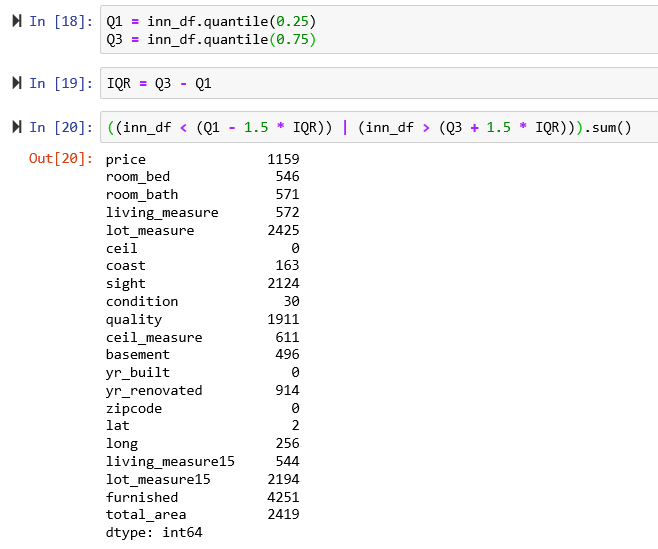
**Univariate Analysis – Distance Plots**



* From the distance plot we can see that more than 10 attributes are categorical. Among those attributes many are having non-gaussian type distribution. The other attributes which are not categorical are mostly right skewed. So as most of the attributes are having non-gaussian type distribution they possess outliers. The outliers can be visually seen using boxplot.



* From the boxplot we can clearly see that except few attributes like ceil, yr\_built & zipcode all the other attributes having outliers.
* For better analysis of dataset the outliers have to be removed, this can done only when the no of outliers for each attribute is found.

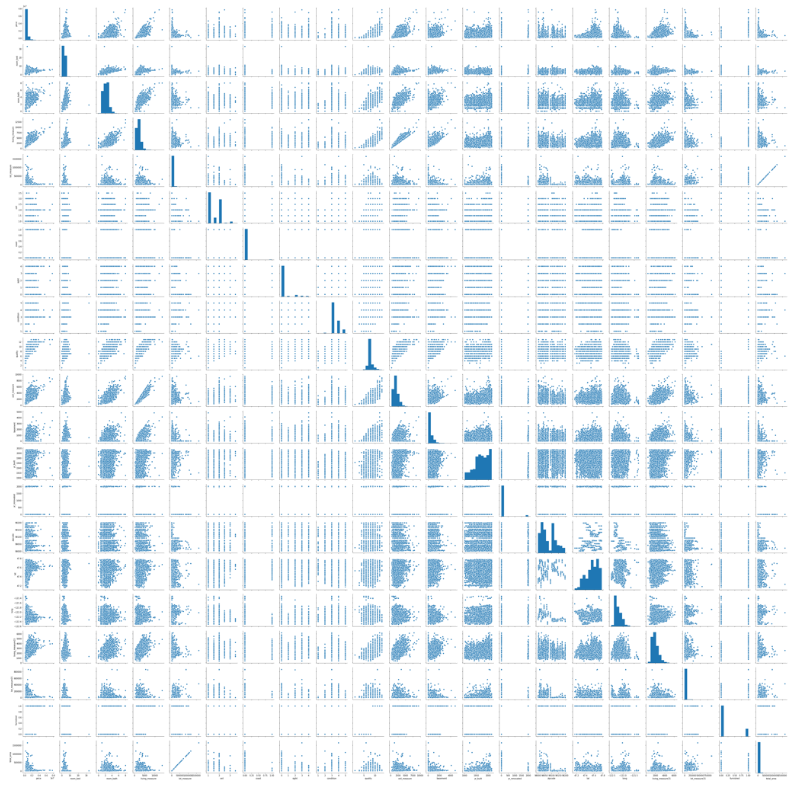


* This clearly gives us information on which attributes have more outliers. With this we started our analysis for reducing the outliers and better utilization of the dataset.

**Bivariate Analysis**

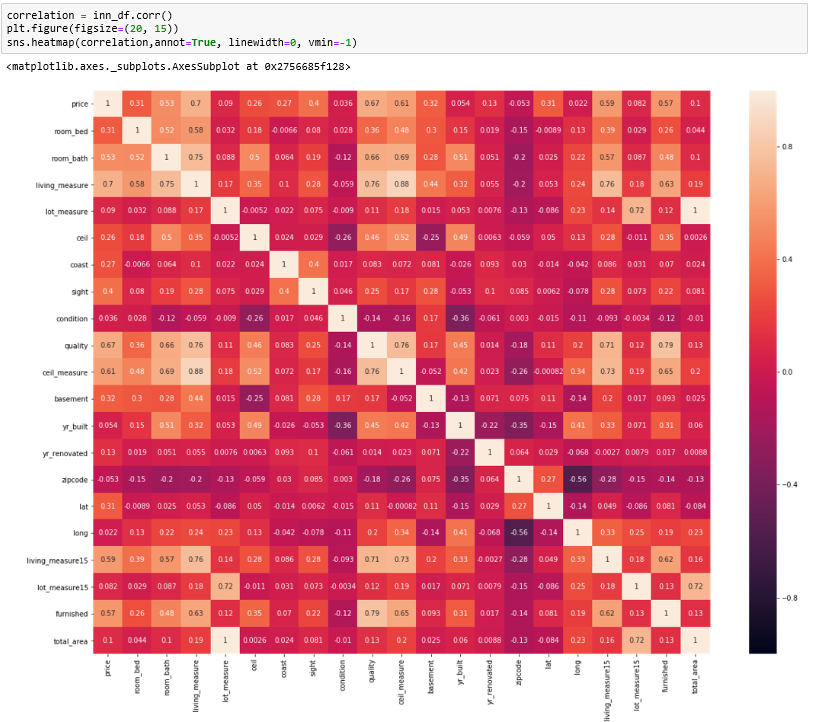
* We then started our bivariate analysis to understand the relation between attributes.
* There are two methods which can clearly show the relationship between attributes, one is pairplot and other is correlation heat map.

**Pairplot**



* From the pair plot we can see linear correlation between few attributes and few has clustered correlation with no proper relation with each other.
* Let us do the correlation analysis between each attribute using heat map to exactly find the relation between two attributes.

**Correlation Heatmap**



* From the heat map it is very clear that total\_area and lot\_measure has a correlation of '1' so any one column can be neglected.
* living\_measure & ceil\_measure has a correlation of '.88' which means all the houses are almost built with a ceiling.
* living\_measure has also got a good relationship with quality & room\_bath, which means with greater living capacity the quality is good and the no of bathrooms/bedrooms increases.
* living\_measure15 is just an extension of living\_measure, so has good correlation
* All the furnished houses have better quality.
* living\_measure, quality, ceil\_measure, furnished & room\_bath have better correlation with the target variable ('price'). So for predicting the price of a property these attributes play a major role in this dataset.

**Deciding Models and Model Building**

* Before we started to modify the dataset, we developed a simple linear model to find how it performs on the existing dataset.



* We splitted the dataset into train and test data, then we trained the dataset using simple Linear Regression.
* We then tried to find the score for test data set which gave us around 70%
* The target column of the dataset is said to be continuous. So we cannot use any classification model for testing the dataset.
* We then tried to find the score using Logistic Regression and SVM, which showed us an interesting error that the dataset is said to be continuous.
* Then we understood that even few model developing methods in regression also cannot be used for testing the continuous data directly. So we used a label encoder to convert continuous dataset to numerical form which made the dataset to be suitable for testing using Logistic Regression and SVM. But even after that the score was less than 1%. So these models are not suitable for the given dataset.
* Ridge & Lasso methods gave a model score of 70%
* Decision Tree Regressor gave a model score of 76%
* Gradient Boost Regressor(GB) gave a model score of 87%
* Random Forest Regressor(RF) gave a model score of 87.2%
* Comparing all the model scores Gradient Boost Regressor(GB) & Random Forest Regressor(RF) seem to perform well on the given dataset.

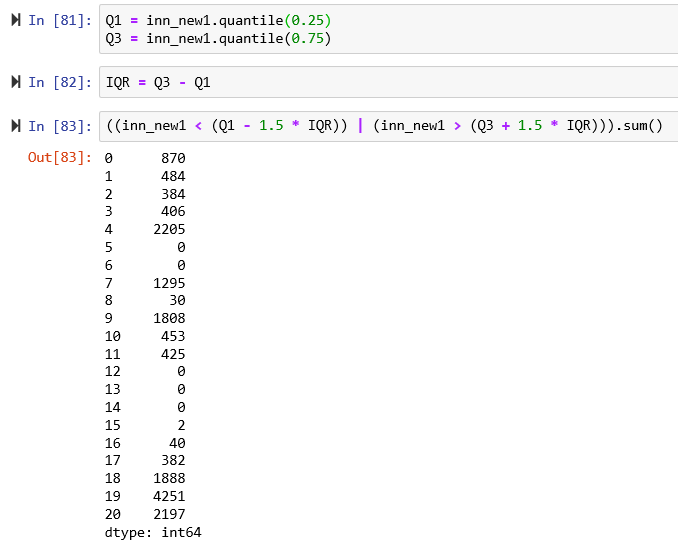
**4. Model Evaluation**

Now we have decided on which model to be used, so our next objective is to fine tune the selected model. This can be done through different iterations which are mentioned below.

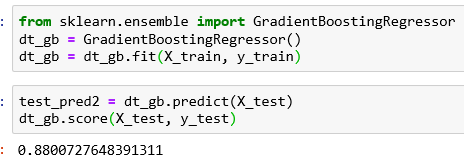
1. Removing the outliers using (mean+3\*SD) as the limit on the existing dataset and developing a model using GB and RF method.
2. Replacing ‘Zeros’ of the attribute ‘room\_bath’ with its respective column medians and developing a model using GB and RF
3. Tuning the learning rate and the estimators in GB & RF
4. Using PCA to analyse the minimum number of attributes needed for developing a better model using GB or RF which can provide atleast 95% variance.
5. Analysing the model performance using GB and RF by removing attributes.

**Removing the outliers using (mean+3\*SD) as the limit on the existing dataset and developing a model using GB and RF method.**

* We removed the outliers using the limit for each attribute as (mean+3\*SD).



* The no of outliers has certainly reduced compared to the original dataset outliers
* We then tested using GB & RF which gave a maximum model score of 88%.



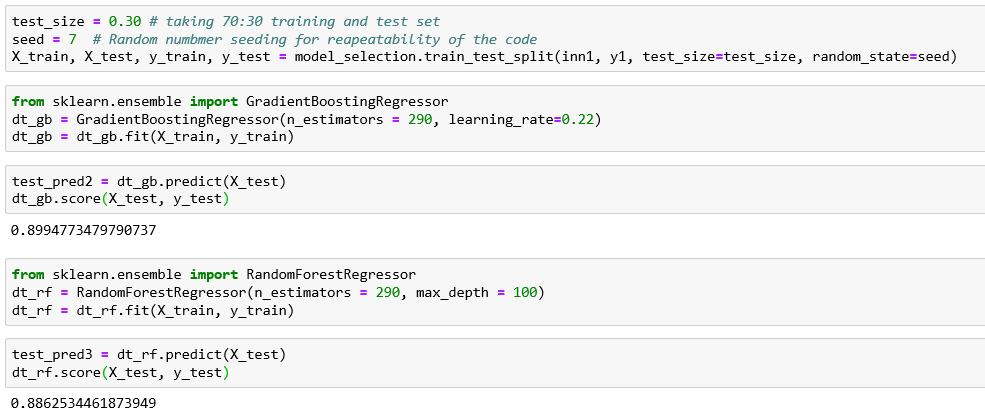
* Compared to the normal dataset, the model score has been increased by 1%, so we proceeded with the removed outlier dataset for further tuning.

**Replacing ‘Zeros’ of the attribute ‘room\_bath’ with its respective column medians and developing a model using GB and RF**



* Result clearly shows that there is no change in the model output compared with the previous dataset. So we can continue with the previous dataset for further tuning by leaving this iteration.

**Tuning the learning rate and the estimators in GB & RF**

****

* From the results it is clear that GB performs better than RF. After more number of iterations we found that the best no of estimators was 290 and the best learning rate was 0.22. With this the model was able to give a score of 89.94 in Gradient Boost Regression method.
* With the best estimators and learning rate we can use GB method for further analysis whether the model can be tuned to get better score.

**Using PCA to analyse the minimum number of attributes needed for developing a better model using GB or RF which can provide atleast 95% variance.**

* PCA can provide elbow, which is normally used for finding the clusters in the dataset, which is normally used for classification problem. But the target column here is continuous, so there is no need to analyse elbow curve.
* From the imaginary PCA curve we can see that around 17 to 18 variables are needed to get 95% of variance, which when tested with the model, can give a score of only 82%, which is very less compared to the previous tuned results.
* But if we want only few attributes to be utilized to develop a model then PCA gives us a better solution with minimum attributes providing model score of 80%

**Analysing the model performance using GB and RF by removing attributes.**

* By removing single attribute along with cid and dayhours attributes the model score obtained are shown below,
* room\_bed - The model score was 89.896%
* room\_bath - The model score was 89.73%
* living\_measure - The model score was 89.744%
* lot\_measure - The model score was 89.869%
* ceil - The model score was 89.755%
* coast - The model score was 89.23%
* sight - The model score was 89.47%
* condition - The model score was 89.45%
* quality - The model score was 89.298%
* ceil\_measure - The model score was 89.665%
* basement - The model score was 89.938%
* yr\_built - The model score was 89.576%
* yr\_renovated - The model score was 89.635%
* zipcode - The model score was 89.6%
* lat - The model score was 88.588%
* long - The model score was 89.30%
* living\_measure15 - The model score was 89.546%
* lot\_measure15 - The model score was 89.839%
* furnished - The model score was 89.918%
* total\_area - The model score was 89.672%
* **From the results we can clearly see that all the model score decreases if we remove anyone attribute. Let us try and check if any two or more attributes are removed simultaneously apart from cid & dayhours, whether the the model score increases or not???**
* lot\_measure & basement - The model score was 89.983%
* lot\_measure, lot\_measure15 & basement - The model score was 89.936%
* lot\_measure, furnished & basement - The model score was 89.989%
* **After all the analysis of removing attributes, removal of lot\_measure, furnished & basement can yield the best score of 89.989%. Thus the proposed model can perform with an accuracy of 90% using only 17 attributes out of the 22 attributes from the given dataset, which is what we found from PCA as the minimal no attributes for providing 95% variance.**
* **Thus model evaluation is done using 5 iterations and the model is tuned to provide best score of 89.989%.**

**5. Comparison to benchmark**

How does your final solution compare to the benchmark you laid out at the outset? Did you improve on the benchmark? Why or why not?

* The initial approach we followed for this problem is Linear regression,
* We see that we have achieved accuracy of 70 -75 % without and with removal of outliers on the dataset. From the problem statement, the benchmark was set as 85%.
* Finally we used Gradient Boost Regressor(GB) as our model because it was giving the best model performance with a score of 87% for the given dataset.
* By our tuning methods we were able to improve the model score to approximately 90% from 87%.

**So we have improved our base model using different iterations, which helped us in obtaining more details about the underlying dataset. The randomness added has made the model to predict accurately on the incoming unseen data and hence the model is more prepared for real world datasets.**

**6. Visualizations**

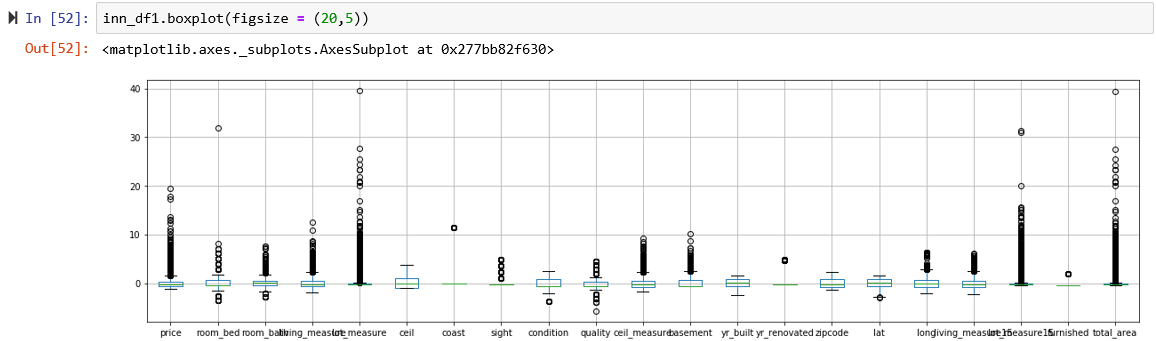
In addition to quantifying your model and the solution, please include all relevant visualizations that support the ideas/insights that you gleaned from the data.

* The visualizations are the univariate analysis, bivariate analysis and PCA.

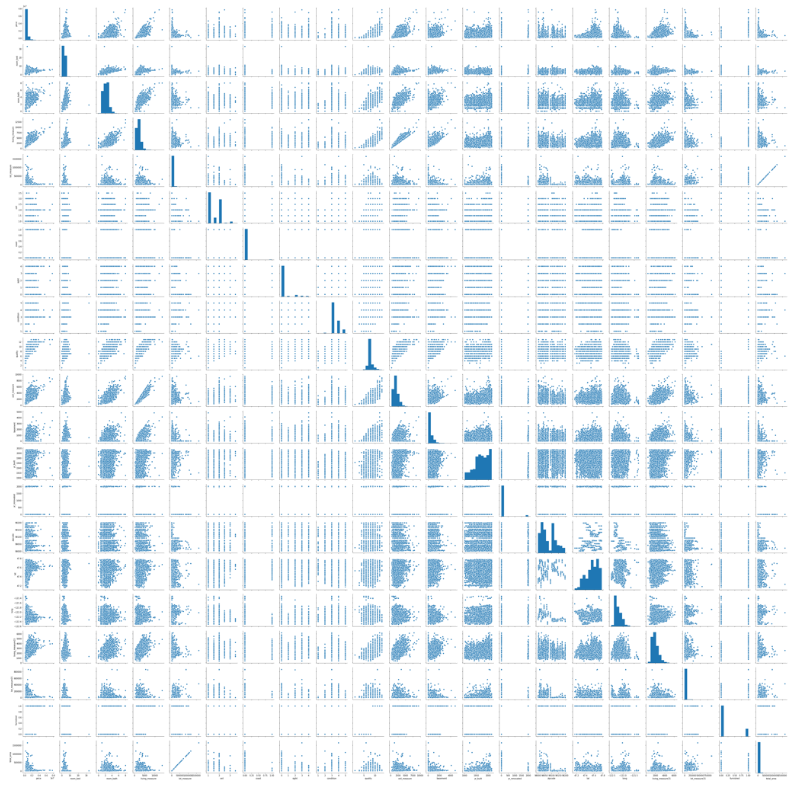
**UNIVARTIATE ANALYSIS – DIATANCE PLOT**



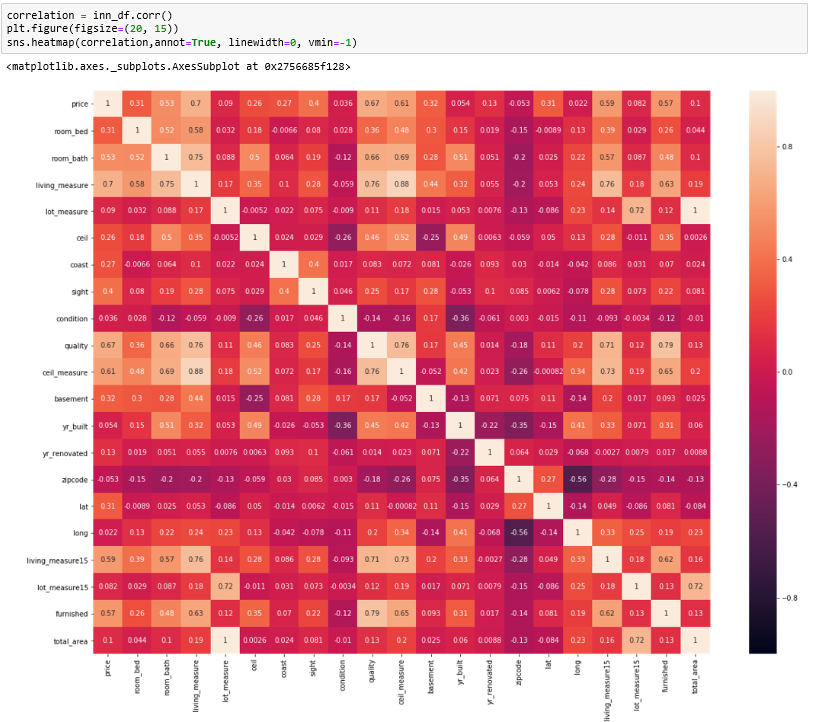
**BOXPLOT**



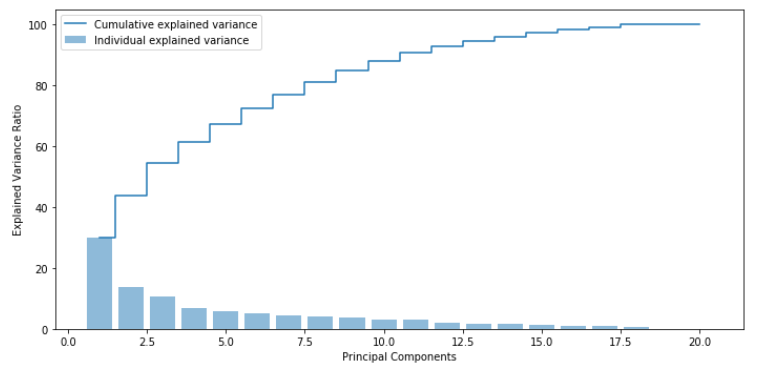
**BIVARIATE ANALYSIS - PAIR PLOT**



**CORRELATION HEAT MAP**



**PRINCIPAL COMPONENT ANALYSIS**



**7. Implications**

How does your solution affect the problem in the domain or business? What recommendations would you make, and with what level of confidence?

* Considering the Latitude and Longitude provided in the dataset, we can see that it is confined to a small part geographically and the sold date is also confined to a period of May 2014 to May 2015, so this prediction applies to a very minimal scope.
* There were cases, where the same house is being sold twice within the 1 year period and there is slight increase in the price in the latter, which clearly explains the value increase as time passes. So this should be considered if this model is being used for the data sets with data of longer period. With the given information, we were able predict the best score using Gradient Boost Regressor with a model score of 90%.

**8. Limitations**

What are the limitations of your solution? Where does your model fall short in the real world? What can you do to enhance the solution?

* Since our target variable is continuous we have to limit our modelling with regression and we cannot use any classifiers.
* The given dataset had some missing values and wrong information, when we try to replace the respective column medians, the model was getting overfitted.
* The model was not responding to polynomial features during tuning phase.

**9. Closing Reflections**

What have you learned from the process? What you do differently next time?

* Data set should be split as train, validation and test data to avoid over-fitting in production.
* Even while predicting same continuous variable each Regressor can perform differently.
* Depending on the domain even 85-90 is a good score, to avoid over-fitting.
* Importance of learning rate in GradientBoostingRegressor.
* Importance of PCA
* Next time while predicting data sets with wide variety of size in variables, they can be grouped and each of group can be dealt with different modelling.
* Like in this case, we could have grouped them on the basis of number of bedrooms.