**BUILDING REGRESSION MODELS TO PREDICT LAGOS RENT**

**OBJECTIVE**: To explore different state-of-the-art regression models that can intelligently help predict rent of a house based on certain features and specific location in Lagos, Nigeria. For this task 16,800 properties in Lagos was web-scraped.

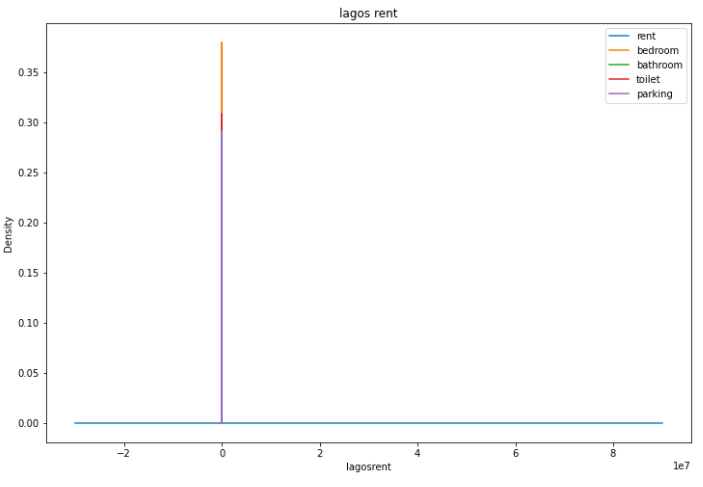
**EXPLORATORY DATA ANALYSIS**

The important libraries for the project were imported. That include numpy and pandas for my arrays and table, seaborn and matplotlib for my data visualizations and plotting, scikit learn for importing model for standardization.

The dataset was read in through pandas and the first five rows was displayed making clear a short outlook of my data. The **description** of the data was checked giving some information like the *count, minimum value, maximum value, 25, 50 and 75 percentile, the mean and the standard deviation*.

From the count, it was observed that only the ‘rent’ had a complete feature, ‘bedroom’ and ‘bathroom’ had same number of features, while ‘parking’ had the least features after ‘toilet’. In order to understand better the information the information of the lagosrent dataset was called. It was realized that null values was responsible for the variation in the *count*.

It could also be observed from the description that for the ‘bathroom’ column, while it has a mean of 4.639733, it has a high standard deviation of 9, this rose a suspicion about the nature of the data and suggested the presence of outliers.

The null values of the data was checked and only rent and location was without null values. Below are some visualizations of the data:

The figure above shows the kde distribution of lagosrent with ‘parking’, ’toilet’, and ‘bedroom’ being visible. Other visualizations for individual features are given below: Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated

**DATA CLEANING PROCESS**

Rather than making the whole null values same figure, we decided to treat the null value of each feature differently based on the nature of the feature.

The first thing noticed about the data set was that there is a certain row of repeated amount of null values having both the ‘location’ and ‘rent’ the same. The view is shown below:

A picture containing chart

Description automatically generated

Since the value above has none of the determining features, it was believed that it is best to drop them. A code was written to drop rows with null value of more than three which reduced the rows of the total data from 16800 to 15111 showing that those two rows of four null values amounted to about 1,689 rows.

On the ‘bathroom’ features, there is one hundred value count of 120 for a ‘bedroom’ of 4 which is not reasonable, it was decided that the 120 will be substituted for 4 from the assumption that the house has as much ‘bathroom’ as the ‘bedroom’. This increased the count of 4 in the ‘bathroom’ to 5160 from 5060. This obviously reduced the standard deviation of my ‘bathroom’ features from the previous 9 to approximately 3 and the mean from approximately 4.6 to 3.8.

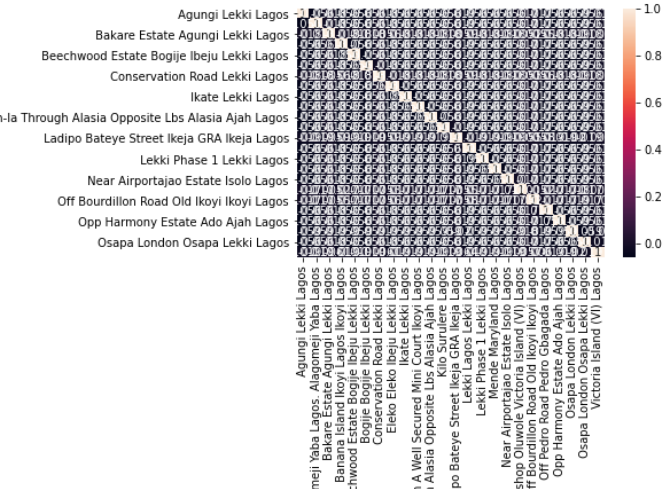
With the ‘toilet’ feature, there is a null value of about 100, we decided to make this null value one beyond the bathroom value because a house should have an extra general toilet, therefore filling the null values with 5 in all the repeated instances. Theis increased the value of our dataset from 15011 to 15111, with no more null values.

To clean the ‘parking’ feature, a column called ‘rowmin’ was created. This column takes the minimum of each row as its value. We then use the value of this ‘rowmin’ to fill the null value of the parking ‘feature’. This is done with the assumption that the parking space of an apartment may not necessarily be more than either the ‘number of toilets’, or ‘bathroom’, or ‘bedroom’.

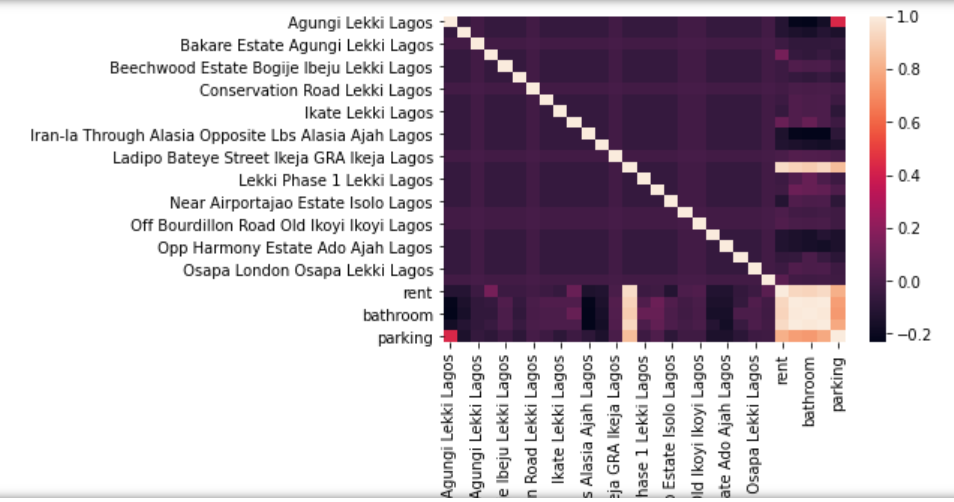
**FEATURE SELECTION AND ENGINEERING**

Since there are about 24 distinct location in our data, it was believed that the best engineering to be done to the data is One Hot Encoding where each of the distinct location takes on a distinct numerical value in order to be readable by the algorithm. When the encoding was done, we had extra 25 columns in which one was dropped to avoid multi-collinearity. The resulting new 24 columns was then added to the initial dataset with the source column for our encoding which is ‘location’ also dropped. This brought about a final 29 columns in the dataset called *finallagos\_clean*.

Using heatmap to visualize the correlation of my data, there was no correlation. A need to standardize our data arose.



Rather than using standard scaler before min-max scaler, min-max scaler was directly used and then our dataset features have correlation.



**TRAINING MY DATA**

My target y was defined as the ‘rent’ while the rest of the features aside rent was assign as x. Train test split was imported from Scikit learn model selection. 75% of my total data was used to train the model while 25% was used for test.

Linear regression was imported since we are to make the prediction of rent from other features.

**TESTING MY MODEL PERFORMANCE**

Checking my predicted value to my test value, we saw the similarity showing that the model was doing well.

Mean squared error was used in testing the model and the accuracy was closer to one.

**BY:**

**KULOGUN SAMUEL AYOOLA**