**CSCI461 : Introduction to big data**

**Project: Real state price prediction**

**Team members:**

Ahmed Mohamed Hashem 19105073

Abdelrahman Hassan Abdelhameed 19104406

Ammar Mahmoud Hamed 19105685

Mahmoud Ibrahim Hussien 19100242

**Presented to: Dr. Khaled Mohamed Fouad**

**Methodology:**

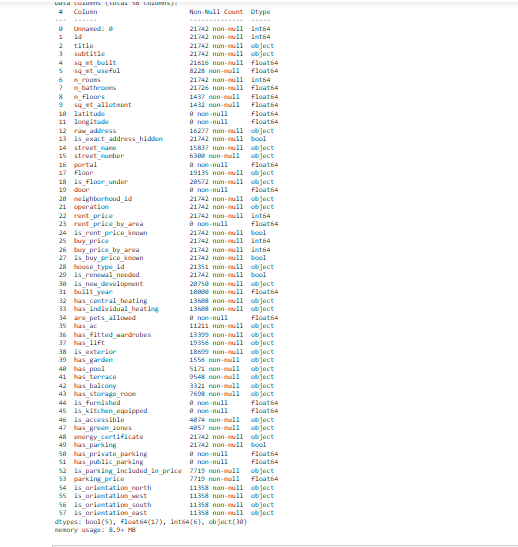
Our project aims to predict houses price in Madrid depending on some variables like; lift, central heating, number of bathrooms, number of rooms, area and parking.

We have cleaned our data using two ways by two libraries which are pandas and pyspark. We have visualized our data using “matplotlib” library. To predict the prices we made two models to predict it we have used linear regression model and Catboost model.

**Project Description:**

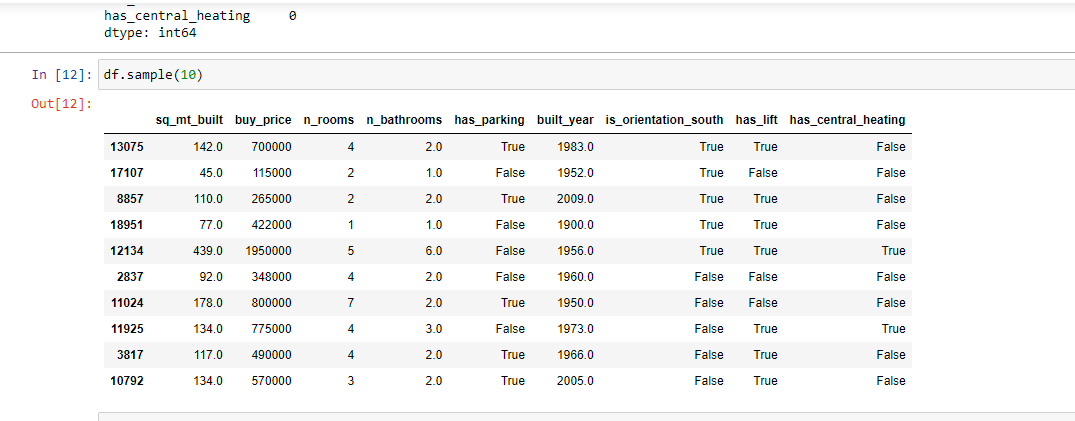
After some data cleaning and research, this project seeks to properly estimate real estate values in Madrid using a publicly available dataset. The project, in particular, is intended to be more of a learning experience for the author, since it evaluates four distinct ML strategies to produce predictions - linear regression using ordinary least squares (OLS).

**Dataset before cleaning:**



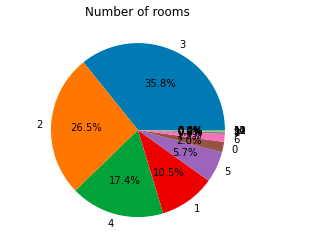
We used the function “info()” to view the info of our data frame. As we can see from the figure of “Dataset before cleaning”, there are many columns that contain null values or missing values and need to be removed, other than missing values there are columns that we will not need for our modeling. So we have removed a lot of columns.

**Here is an image after cleaning the dataset:**

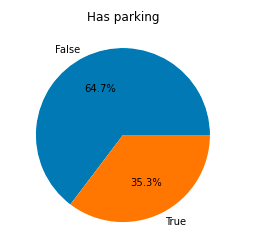


The figure above shows our data frame after we removed the columns that had null values or were not of much use to our modeling phase. We used the function “sample()” to get a sample of our data frame.

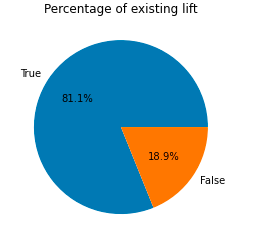
**Data visualization:**



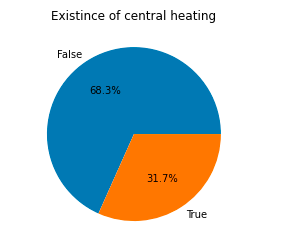
The image above shows percentages of number of rooms in real estates, we used “matplotlib.pie” to plot this pie chart using the column “n\_rooms” from our data frame. What we conclude from this image; majority of real estates for sale have from 2 to 4 rooms as you can see from the pie chart.



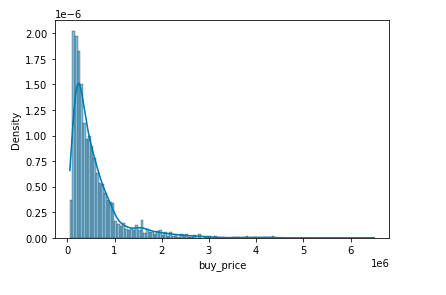
The image above shows percentages of real estates that has parking, we used “matplotlib.pie” to plot this pie chart using the column “has\_parking” from our data frame. What we conclude from this image; majority of real estates for sale have more than 60% has parking as you can see from the pie chart.



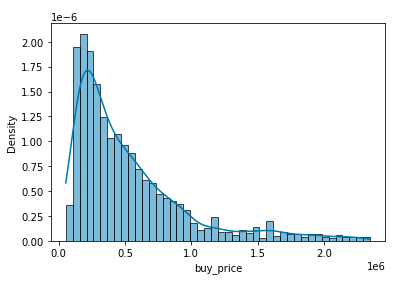
The image above shows percentages of real estates that have lifts in their property, we used “matplotlib.pie” to plot this pie chart using the column “has\_lift” from our data frame. What we conclude from this image; majority of real estates for sale have lifts as you can see from the pie chart.



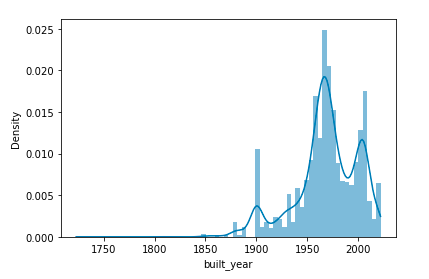
The image above shows percentages of real estates that has central heating. we used “matplotlib.pie” to plot this pie chart using the column “has\_central\_heating” from our data frame. What we conclude from this image; majority of real estates for sale does not have central heating as you can see from the pie chart.



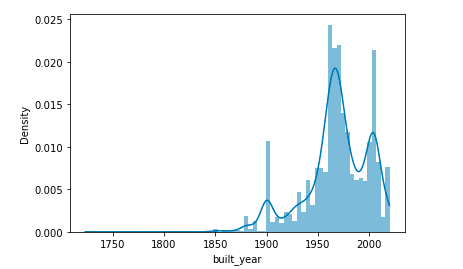
From the histogram above, we can conclude that there are some outliers present. Here, the outliers are situated around the higher prices (right side of the graph). So we have to remove the outliers. We used “seaborn.histplot” using the column “buy\_price” from our dataframe.

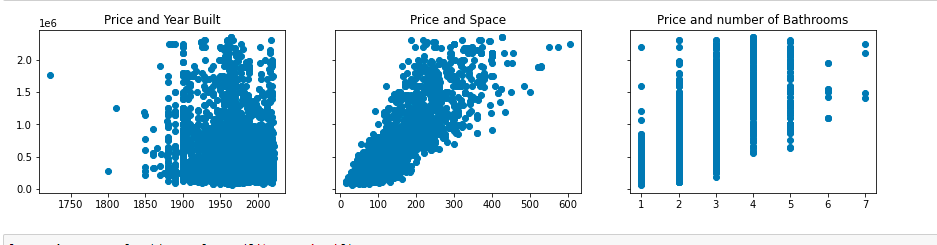


From the histogram above, There are some outliers present. Here, the outliers are situated around the higher prices (right side of the graph) and if the right side is excluded, the prices seem normally distributed. Outliers are a great issue for the model we will first use (ordinary least squares OLS). By removing the outliers, the graph is much more concentrated on the real data points now, it looks more 'normal' as in normally distributed.



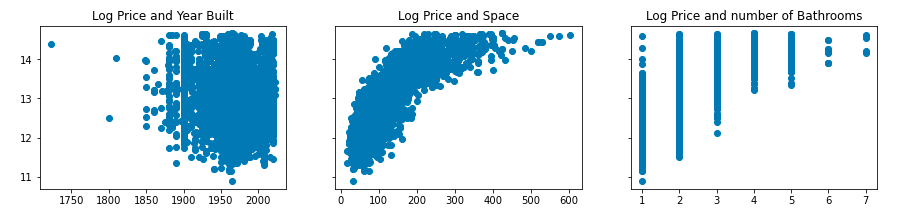
From the histogram above, we can conclude that there are some outliers present. Here, the outliers are situated around the higher prices (right side of the graph). So we have to remove the outliers. We used “seaborn.histplot” using the column “built\_year” from our dataframe.



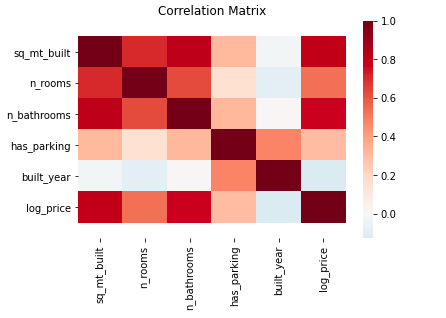


From the scatter above we conclude that it is slightly abnormal because the price is not in its normal form, so we had to change it.

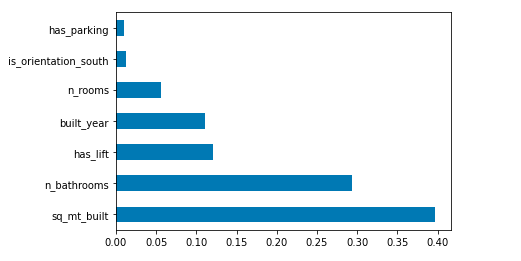
After transforming price to its normal form we can see that as long that the area increases the price gets higher which is logical same goes to the number of rooms.



A correlation matrix of the dataset:

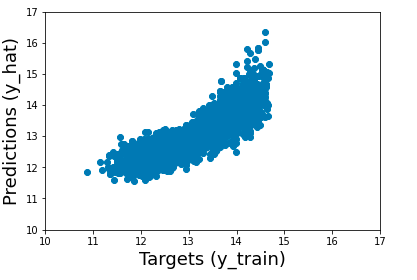


Now it's much more clear what the correlation is between all the datasets

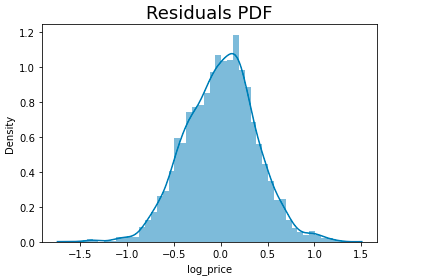


From the bar chart, we conclude that after collecting the variables that affect most on price are as follows; sq\_mt\_built, n\_bathrooms, has\_lift, built\_year, n\_rooms, is\_orientation\_south, and has\_parking.

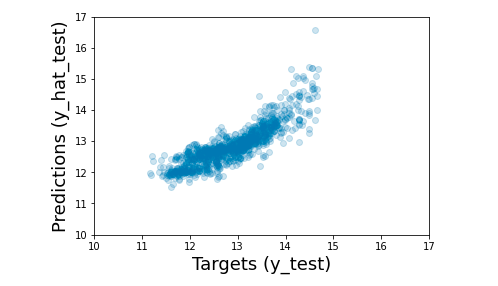
**After data cleaning and data visualization(exploration) we will start building our model.**



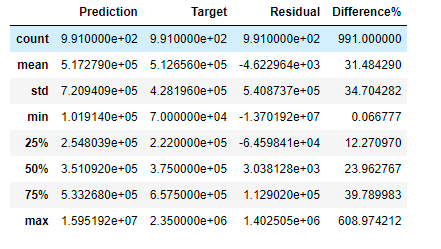
The simplest way to compare the targets (y\_train) and the prediction (y\_hat) is to plot them on a scatter plot. The closer the points to 45 degree line the better the prediction.



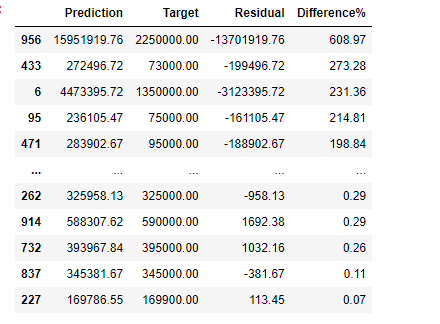
Plotting the residual PDF to check for anomalies.



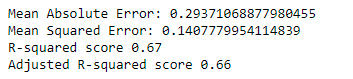
There are many predictions outside the 45 degree line, suggesting that the algorithm struggled to correctly predict prices especially in lower and upper price ranges. The algorithm performed much better for predicting the prices in the middle of the range. This is consistent with OLS's inherent limitation in dealing with outliers.



Exploring the descriptives here gives us additional insights.



To make the dataset clear, we can display the result with only 2 digits after the dot.



From this we conclude that the accuracy of the model is 0.66, which is not the best model, we identified that the algorithm suitable for modeling this dataset is Catboost with accuracy of 81.9%.

Catboost model

Chart, scatter chart

Description automatically generated

The simplest way to compare the targets (y\_train) and the prediction (y\_hat) is to plot them on a scatter plot. The closer the points to 45 degree line the better the prediction.

Chart, histogram

Description automatically generated

Plotting the residual PDF to check for anomalies. By histogram by diffrency between the y\_train and y\_hat

Chart, scatter chart

Description automatically generated

There are many predictions outside the 45 degree line, suggesting that the algorithm struggled to correctly predict prices especially in lower and upper price ranges. The algorithm performed much better for predicting the prices in the middle of the range. This is consistent with OLS's inherent limitation in dealing with outliers.

Text

Description automatically generated

**And here are the values of our catboost model**

**catboost**

**Text

Description automatically generated**

**Linear regression**

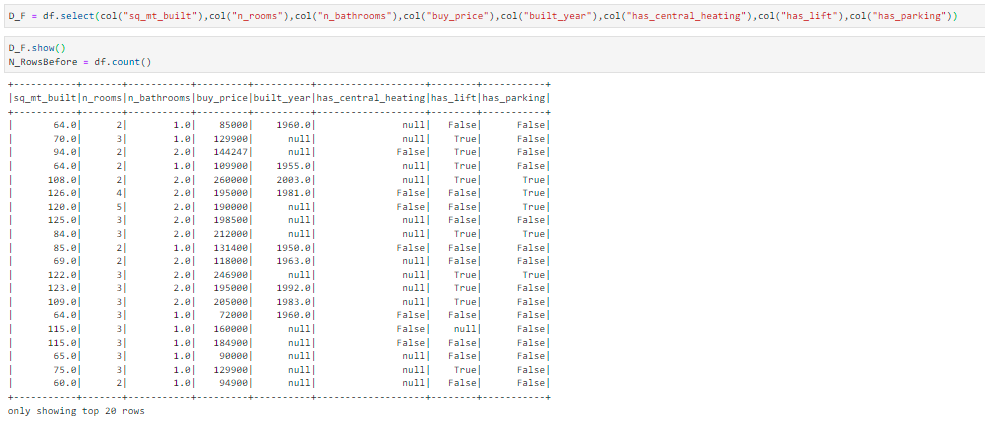
**Text, letter

Description automatically generated**

**As we see these are the two values for the two models and the catboost the way better than linear regression**

**Data preprocessing using “pyspark”:**



First we have to explore our data we have printed our schema using printschema() function, we have noticed that we have a lot of columns that will not be useful for our model so we had to filter out our data to be suitable for the model

We have used select method to filter our data to only select the columns that will be suitable for our model after cleaning, so we have 8 columns now. 7 will be variables and 1 which is buy price will be the y which is the target

Table

Description automatically generated

The figure above shows the values after cleaning. We have used spark to fill the null values by its mean values to avoid dropping a lot of rows. We have used the fillna method which is in Pyspark to fill the missing values.

