**Raw House Data Report**

German Eduardo Baltzar Reyes

The original dataset contains 5,000 observations with 16 features describing the house sales of a particular region. The original dataset features are divided according to their data as follows:

* *Int Data* - MLS, zipcode, year\_built, bedrooms
* *Float Data* - sold\_price, longitude, latitude, lot\_acres, taxes, fireplaces
* *Categorical Data* - bathrooms, sqrt\_ft, garage, kitchen\_features, floor\_covering, HOA

However, observing the type of data shown in the header, we can notice that it is possible to make some adjustments to all of them to get a cleaner dataset.

The first step was to evaluate if there were some duplicated data. The next step determined how many NaN values they had. In this case, *'lot\_acres'* and *'fireplaces'* had 10 and 25. The rest was made by each feature individually.

For the *'bathrooms*' feature, it is inferred that it indicates the number of bathrooms a house has, hence being a numerical feature. The fact that it's a categorical one means there should be other representations than numeric data. This feature was transformed into a float one by determining its unique values, shown in Figure 1. It can be observed that the data is not only represented as a string value but that some variables have a *'None'* label.



Figure 1. Unique values for the *‘bathrooms’* column

For this project, the labels of *'None'* may refer to not having that feature on the house since there were rows with an actual NaN value. A 0 number then substituted the observations with this label, and the complete column was parsed into its floating form to eliminate the string representation.

The same procedure was followed for *'sqrt\_ft,' 'garage,'* and *'HOA.'* However, an additional method was made for the latter feature. Since most of the representations included a decimal point, every value that may have a comma was substituted by its corresponding point.

Since categorical data is more challenging to analyze with numerical values (even more when grouped with a large amount of text on the same cell), one of the best options is to encode the presence of each feature into a vector.

A standard encoder will be used instead of a one-hot vector since it's easier to have one single vector measuring the presence of every characteristic rather than having one single vector (of the same size) for every single observation. This encoding process was made for the *'kitchen\_features'* and *'floor\_covering'* columns.

For the *kitchen\_features* feature, it is necessary to separate the characteristics into different columns is essential. After that, we can observe which ones each house contains and then create a new column with the encoded representations. First, every single dataset observation was collected into a single list to determine which ones were the unique values. In this case, it was observed that multiple values shared the same initial keyword but included a brief description by using the ':' character. Hence, it was decided that only the keyword before those characters were representative, allowing us to reduce the number of unique values on the list. The final unique list resulted in a 55-D array that specified what kitchen features were included in each house. Finally, those encoders were included in the dataset as a new column, '*kitchen\_vectors*.'

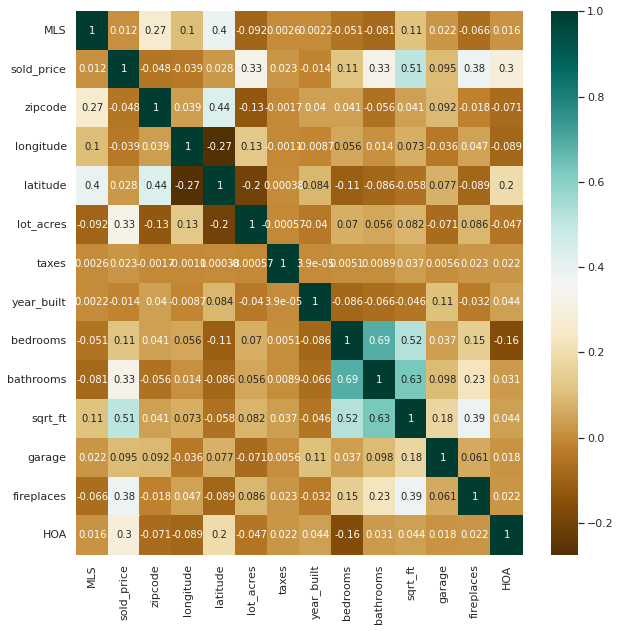
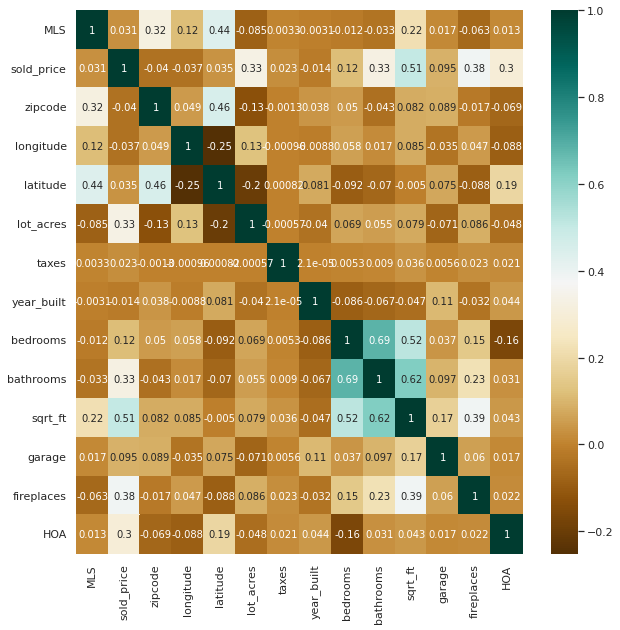
The *'floor\_covering'* column had the values after the ‘:’ token was kept since most observations had an *'other:'* behavior. The encoded version of this feature resulted in an 87-D array. Every observation in both columns was analyzed in its lower case form.

After modifying the representation of each column, the next instruction was to eliminate every NaN value and any possible outlier the dataset may have.

Since the NaN values of *'lot\_acres'* and *'fireplaces'* represented only 0.2 and 0.5% of its data, there was no problem eliminating those observations from the table. The proof of that is shown in Figure 2, where it can be observed how the correlation matrix of every single numerical column maintains a relatively similar correlation value with the rest of the observations after eliminating the null values.

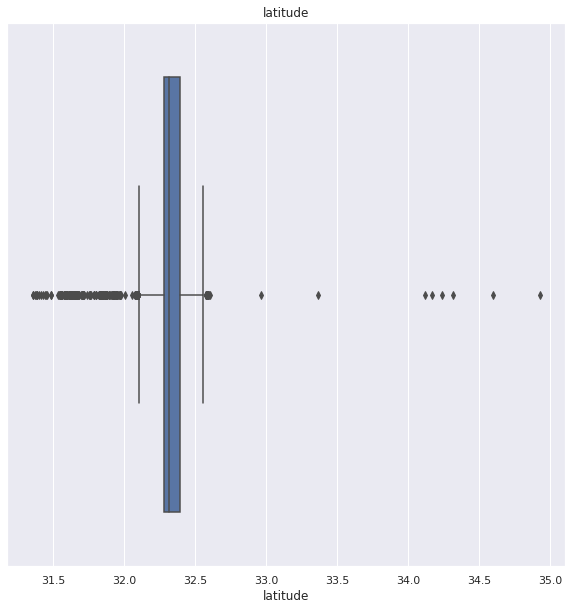
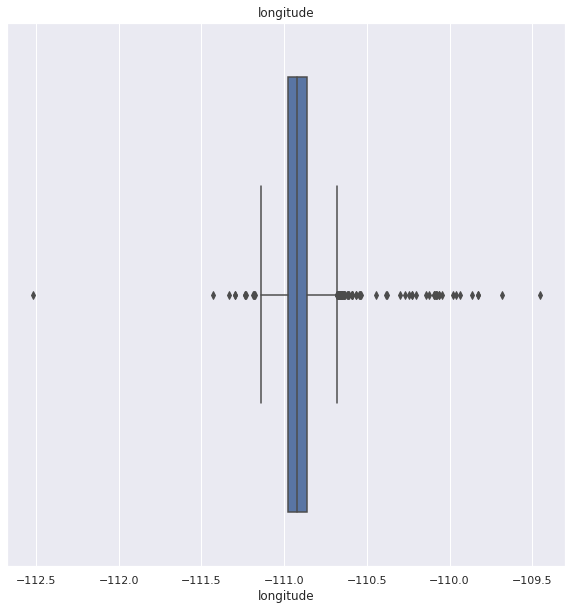
The outliers of every numerical feature were treated individually, according to the behavior of their corresponding boxplot.

In the case of the *'MLS'* column, it was observed that every single value was unique, so there was no significant information that the feature would provide to the dataset. For that reason, the column was deleted from the dataset.

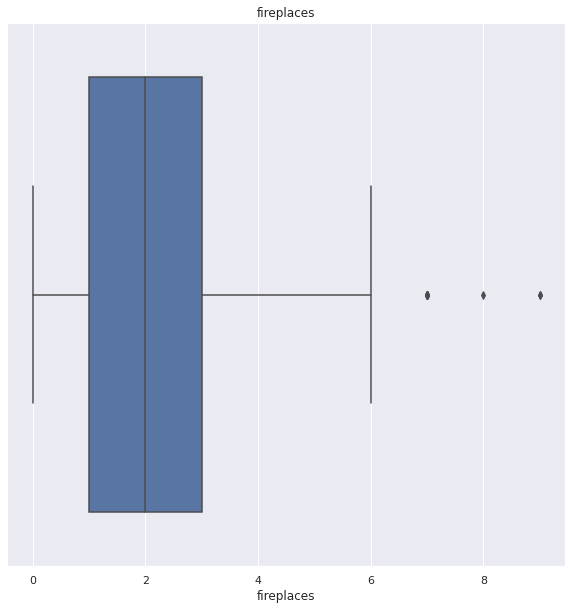


(a) (b)

Figure 2. Dataset correlation matrix (a) before and (b) after deleting null values



(a) (b)



(c)

Figure 3. Boxplot of (a) *longitude*, (b) *latitude*, and (c) *fireplaces*

The outliers of *'longitude,' 'latitude,'* and *'fireplaces'* were deleted by measuring their corresponding IQR scale. Their boxplots had a relatively big middle area between their first and third quartiles, as observed in Figure 3.

For the rest of the columns, the middle area of their boxplots was too small compared to the number of detected outliers. A particular rule was determined for every queue to reduce the number of observations deleted from the dataset.

In the case of *'taxes,'* the data with a zero value, or bigger than its Q3 value, was omitted. For *'sqrt\_ft’,* only the zero values were deleted, while the houses built before 1800 were deleted from *'year\_built'*. The *'sold\_price'* value bigger than 3 million was also deleted. The *'zipcode'* lower than 85400 and bigger than 86000 were ignored. Finally, the *'lot\_acres'* bigger than 500 were deleted from the dataset.

This series of rules eliminated 373 rows of the original dataset, a 7.46% of the original data. After these modifications, a final heatmap was obtained to observe how much the correlation between those features changed.

The graph shown in Figure 4 decreased the overall correlation between features, making it easier to determine which has a better relationship to determining the house cost based on given specifications. At the same time, by deleting those outliers and null values, only the categorical encoders of significant observations remained, indirectly cleaning the non-numerical data as well.

One procedure that was intended to do, but resulted unsuccessfully, was to look for a different way of dealing with the zero values of the column *'sqrt\_ft'*.

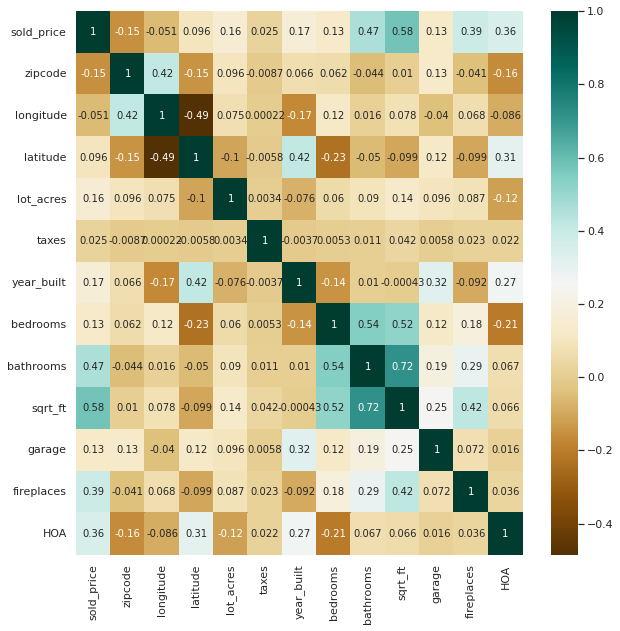


Figure 4. Final correlation matrix of the cleansed dataset

The idea was to determine which other feature had the best correlation with it and use it as a grouping reference to modify those zeros, based on the idea that there was no purpose in having a house if there was no terrain to construct it. The best correlation resulted from the *'bathrooms'* column. The following step was to create three groups based on the quartiles of *'bathrooms'*:

* The ones that had a value below Q1.
* The ones that were bigger than Q3.
* The ones that were between Q1 and Q3.

Once those groups were made, the following step would be to get the mean score of the 'sqrt\_ft' that belonged to the same groups and substitute any zero value with its corresponding mean value.

However, the data manipulation resulted in a series of errors and exceptions that could not be corrected, hence recurring the option of eliminating the zero values since the number of zero values was non-significant. Regardless of the decision made, the amount of data deleted from the original dataset was small enough to keep statistically significant observations.