

Data Wrangling Report

Introduction:

This document explains what kind of data wrangling and cleaning steps were performed, and how the missing values or the outliers handled on “House Sales in King County, USA” dataset. I loaded the dataset from Kaggle website in csv format and read it in the jupyter notebook after importing necessary libraries.

Data Source: <https://www.kaggle.com/harlfoxem/housesalesprediction>

Data Specifications:

The dataset has 21 house features columns, along with 21613 observations. Rows are specifications of houses sold in King County between 05/02/2014 and 05/27/2015.

List of attributes and explanations of features below.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
id                21613 non-null int64
date              21613 non-null datetime64
price             21613 non-null float64
bedrooms          21613 non-null int64
bathrooms         21613 non-null float64
sqft_living       21613 non-null int64
sqft_lot          21613 non-null int64
floors            21613 non-null float64
waterfront        21613 non-null int64
view              21613 non-null int64
condition         21613 non-null int64
grade             21613 non-null int64
sqft_above        21613 non-null int64
sqft_basement     21613 non-null int64
yr_built          21613 non-null int64
yr_renovated      21613 non-null int64
zipcode           21613 non-null int64
lat               21613 non-null float64
long              21613 non-null float64
sqft_living15     21613 non-null int64
sqft_lot15        21613 non-null int64
dtypes:
datetime64[ns] (1), float64 (5), int64 (15)
memory usage: 3.5 MB
```

id: notation for a house
date: Date house was sold
price: Price is prediction target
bedrooms: Number of Bedrooms/House
bathrooms: Number of bathrooms/House
sqft_living: Square footage of the home
sqft_lot: Square footage of the lot
floorsTotal: Floors (levels) in house
waterfront: House which has a view to a waterfront
view: Has been viewed
condition: How good the condition is (Overall)
grade: Overall grade given to the housing unit, based on King County grading system
sqft_above: Square footage of house apart from basement
sqft_basement: Square footage of the basement
yr_built: Built Year
yr_renovated: Year when house was renovated
zipcode: Zip Code
lat: Latitude coordinate
long: Longitude coordinate
sqft_living15: Living room area in 2015(implies-- some renovations) This might or might not have affected the lotsize area
sqft_lot15: LotSize area in 2015(implies-- some renovations)

Data Preprocessing:

1. I loaded the dataset in csv format and read it in the jupyter notebook after importing necessary libraries. I applied **df.head()** and **df.info()** to see some basic information about dataset. There are 21613 entries, and all features look like have no non-null entries. I will check it one more.

2. I wrote the explanation of column names for better understanding. “**View feature**” is explained as “**view: Has been viewed**” in the Kaggle website. I made a further investigation about this feature from King County official website and I reached out these results below:

TOTAL VIEW QUALITY: This is the sum of all view's quality. The view's quality can vary from 0 to 4, in 5 different categories; Puget Sound, City/Territorial, Lake Washington/Sammamish, Mountain, and Small Lake/River.

<https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=k>

VIEW: For each classification will display blank for no view or "Fair", "Average", "Good" or "Excellent" to reflect the quality of view for that unit.

<https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=k>

Later they changed the quality of view grades as: 0 = Unknown, 1 = Fair, 2 = Average, 3 = Good, 4 = Excellent

https://www5.kingcounty.gov/sdc/Metadata.aspx?Layer=parcel_ext

As a result, “view” feature indicates the quality of view for the house.

3. I checked both duplicated values and missing values in the dataset. “Id” column is a unique number for each houses. When I checked it (**house_data.id.unique()**), I saw that There are 177 duplicated rows. In these duplicated rows, everything is the same but price. I kept the last entries, dropped the first ones. After dropping duplicated rows , the dataset has 21436 entries. Also I checked any missing value with (**.isnull().any()**). There is no missing values.

4. I reduce the dataset by dropping columns that won't be used during the analysis.

I inspected the useless features. “id” column has only one unique value for each observations and that did not impact or change anything in the data. 'date', 'lat', 'long' columns are also has no meaning for analysis. For that reason, I dropped those four columns. (**h_data.drop(['id', 'date', 'lat', 'long'], axis = 1, inplace=True)**).

5. I checked the unique values for 'bedrooms', 'bathrooms', 'waterfront', 'view', 'condition', 'grade' columns. I saw that one house has 33 bedrooms and its 'sqft_living' (square footage of the house) is 1620. Most likely, there is a mistake and I dropped this line.

6. 10 bathroom and 13 bedrooms values are zeros. Most likely, there are mistakes about these observations. I did not want them to have negative effect on calculations, and I dropped them.

7. I changed datatypes of columns **'floors'**, **'waterfront'**, **'view'**, **'condition'**, **'grade'** into category, and column **'zipcode'** into str. As a result, total 21419 rows and 17 columns left in the dataset. Datatypes are : category(5), float64(2), int64(9), object(1). At first memory usage was 3.5 MB, and after changing the data types, it became 2.1 MB.