Predicting Home Values in Los Angeles' South Bay

Springboard | Capstone #1 - Data Wrangling

By: Lauren Broussard

All code for this part of the project is saved here as a Jupyter notebook: "Capstone1_Data_Wrangling.ipynb".

Data Collection

The data was retrieved from Redfin.com. Redin allows you to get property information but limits downloads to 350 properties at a time. I downloaded files for each set of properties manually, by neighborhood. Downloads were broken up further by property type, number of bedrooms, and/or price as appropriate to get chunks of 350 properties or less. Finally, I gave each file a meaningful name (i.e. torrance_condos.csv) in case I needed to return to the original files later.

The dataset – after the initial download and merge – consisted of **19,527 rows and 29 columns from 84 separate files**, representing South Bay home sales for the last two years.

Data Cleaning

I mainly used the Pandas package in Python for data wrangling.

Merging Tables: I collected data in two main intervals from Redfin - about a week apart, I merged them into two separate files (Collection 1: 51 csv files, and Collection 2: 33 csv files). Prior to merging the two dataframes together, I added a "collection" column to be able to tell them apart if it became necessary later. I used the pandas.*concat()* method to merge each of the collections. I also added the filename to the end of each record. I kept home sale dates between February 06, 2018 and January 24, 2020, as these dates existed in both collections.

Dropping Columns: The following columns were dropped from the dataframe, because they contained information not useful to solving the problem, the columns were empty, or because the information included was identical down the entire dataframe: *'SALE TYPE', 'NEXT OPEN HOUSE START TIME', 'NEXT OPEN HOUSE END TIME', 'STATUS', 'FAVORITE', 'INTERESTED', 'URL', 'SOURCE', 'LOCATION', 'STATE'*

Change Variable Types: The SOLD DATE column was stored as an object. I used the pandas method *.to_datetime( )* to convert it to a mm-dd-yyyy datetime format. I did this during the import/merge steps. Additionally, I changed the type of 'YEAR BUILT' to int, and the the zip codes from floats to str.

Deleting Duplicates: After inspecting the duplicated data, I determined that most were either due to a difference in the neighborhood name (a column I created based on my downloads), or a difference in the number of days a house was on the market. I replaced the neighborhood columns in the duplicated rows with the city name, as they seemed to follow this pattern in most cases. I also took the max value for days on market when this was the cause of the duplicate.

Re-indexing: After concatenating files and cleaning the dataframe, I reset the index to be a range index.

Missing/Incorrect Values:

- **SOLD DATE:** I used the .dropna() method to remove properties with no sold date. These rows were also consistently missing data in other columns as well.
- **PROPERTY TYPE:** I dropped the rows from all property types except: *Single Family Home, Townhouse, Condo/Co-op, Mobile/Manufactured Home*. Many of the other property types were missing large amounts of data.
- **ADDRESS:** I looked up on Redfin any records that were missing addresses. I was able to determine that their addresses are listed as "Undisclosed" - I dropped these records.
- **CITY:** I looked up addresses that were missing a city, and updated them according to Redfin/Google. I also removed incorrect cities (i.e. a record where the city was San Bernardino - the data had been merged with another identical address in San Pedro).
- **ZIP/POSTAL CODES:** One property had a postal code outside of the range of CA zip codes (that begin with a 9) - I corrected the zip code.
- **PRICE:** I removed any rows in which the price was less than $10,000 (one was a property priced at $25). Many of the high priced homes seemed to have a very large number of bedrooms and bathrooms, so I kept these for now.
- **BEDS, BATHS:** I dropped rows with missing information on the number of bedrooms and/or bathrooms. I tried searching the information as well on Redfin and Zillow (another real estate site) to verify but could not find consistent information. I dropped 2 rows of single family homes with beds/baths higher than 10 that I could not verify between Redfin, Zillow, and the LA County Assessor's data.
- **LOCATION/NEIGHBORHOOD:** I dropped the original "Location" feature as the format did not seem to be standardized. Instead, I created a column called "Neighborhood" based on the neighborhood I looked up on Redfin and the name of the created csv file.
- **SQUARE FEET:** I dropped 5 Single Family Homes and 1 Townhouse with missing square feet information - I could not verify the information through another source. 52 Mobile homes with no square feet information were filled with the average square feet for other mobile homes in the area. Home with 0 square feet were fixed if I could confirm the square feet from two other home data sources, otherwise they were dropped.
- **LOT SIZE:** I filled the missing lot sizes with the median lot size by the property type.
- **YEAR BUILT:** The earliest standing home built in Los Angeles was built around 1818. I looked for any properties sold before this date or after 2020. One home was listed as being built in 2021. When I looked it up on Redfin, it is listed as sold as a vacant lot with the intent to turn it into a single family residence. I dropped this record.
- **$/SQUARE FEET:** I filled in the missing values with a calculation: Price / Square Feet
- **HOA/MONTH:** I chose to fill empty HOA/MONTH values with 0. I looked at other records where the HOA/MONTH was 0, and the data did not seem particularly different between the two.

My final dataframe, **'south_bay'** consists of **13631 rows and 18 columns.** It includes home sales data in South Bay neighborhoods between **February 06, 2018 and January 24, 2020**.