CSCI 550 - Project 2

Eliot Liucci, Eric Folsom, Nick Clausen, Christal O'Connell

2024-10-25

1 Executive Summary

2 Data Preprocessing and Exploration

The data received had a few issues that needed to be dealt with. First of all, the `Description` variable contained pieces of key information that were extracted (number of bathrooms, number of bedrooms, total number of rooms, and sell date).

```
# Extracting important information from descriptions
sell_date = as.numeric(0)
rooms = as.numeric(0)
bedrooms = as.numeric(0)
baths = as.numeric(0)

# Takes a minute to run, not too bad though
for(i in 1:nrow(data)){
    sell_date[i] = str_split(
        str_split(data$Description[i], "sold on ")[[1]][2],
         ", is a")[[1]][1]
    rooms[i] = str_extract_all(
        str_split(data$Description[i], "total of ")[[1]][2],
         "\\d")[[1]][1]
    bedrooms[i] = str_extract_all(
        str_split(data$Description[i], "total of ")[[1]][2],
         "\\d")[[1]][2]
    baths[i] = str_split(
        str_split(data$Description[i], "bedrooms, and ")[[1]][2],
         " of which are bathrooms")[[1]][1]
}
```

Once these variables were extracted, the `Description` variable was dropped from the data set. We noticed some houses with strange recording like "42 bathrooms and 7 rooms", so we dropped any listings where the number of bedrooms and number of bathrooms was greater than the total recorded number of rooms.

```
# Adding features of descriptions, removing descriptions
data = data %>%
    mutate(
      Sell_Date = mdy(sell_date),
      Rooms = as.numeric(rooms),
      Bedrooms = as.numeric(bedrooms),
      Baths = as.numeric(baths)
    ) %>%
    select(-Description)

# Removing rows where num bathrooms/bedrooms exceeds number of rooms
data = data %>%
    filter(Bedrooms < Rooms | Baths < Rooms)
```

Next, we dropped any listings with a missing value in at least one of the variables.

```
# Removing rows with at least 1 missing value
data = data %>%
    drop_na()
```

We also wanted to deal with outliers, so a function was written that would identify a listing as an outlier if it was greater than 3 standard errors away from the mean value and used this to remove outliers for variables where the maximum value was significantly higher than the 3rd quartile.

```
# Filtering extreme observations
is_outlier = function(x){
    result = abs(x - mean(x)) > 3*sd(x)
    return(result)
}

# Removing outliers for variables where max() is greater than q3()
data = data %>%
    filter(!is_outlier(`Sale Price`),
           !is_outlier(`Land Square Feet`),
           !is_outlier(Baths),
           !is_outlier(`Lot Size`),
           !is_outlier(`Town and Neighborhood`),
           !is_outlier(`Age Decade`),
           !is_outlier(`Age`),
           !is_outlier(`Estimate (Land)`),
           !is_outlier(`Estimate (Building)`),
           !is_outlier(`Building Square Feet`),
           !is_outlier(`Other Improvements`))
```

Finally, we removed all spaces from variable names and replaced them with underscores.

```
# Removing Spaces
data = data %>%
  rename(Property_Class = `Property Class`,
         Neighborhood_Code = `Neighborhood Code`,
         Land_Square_Feet = `Land Square Feet`,
         ...
         Age_Decade = `Age Decade`,
         Neighborhood_Code_Mapping = `Neigborhood Code (mapping)`,
         Town_and_Neighborhood = `Town and Neighborhood`,
         )
```

This cleaned data set was written as `data_cleaned.csv` so it could easily be reloaded for the remainder of the analysis.

```r
# Save cleaned data
write_csv(data, "data_cleaned.csv")
```

3 Model Development and Performance Evaluation