Machine Learning: Predicting House Prices

Group 3: Anjali, Yalda, Ning, Vincent

Thinking Process

1. Data Collection
   a. What was used?
   b. What was gathered?

2. Tableau
   a. Graphical data

3. Machine Learning
   a. Models

NEW LISTING | House | 3 bds , 4 bths

$3,999,900

3390 Shuswap Road E
Kamloops, BC

Save + View >

NEW LISTING | House | 7 bds , 7 bths

$3,999,000

13443 Mclaughlin Rd
Caledon, ON

Save + View >

NEW LISTING | House | 4 bds , 4 bths

$3,800,000

231 Arts Lane
Port Severn, ON

Save + View >

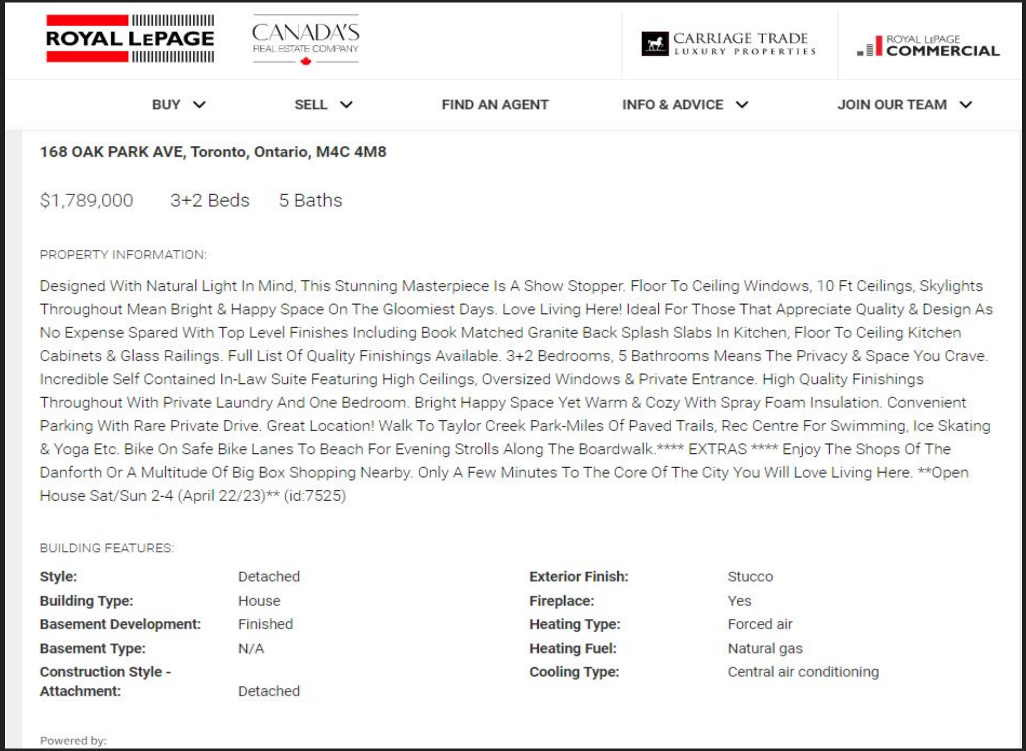
Data Collection - What was gathered?

House Information

- Price
- # of bed's

Building Features

- Fireplace
- Style (Detached, Semi-detached)



ROYAL LEPAGE | CANADA'S REAL ESTATE COMPANY | CARRIAGE TRADE LUXURY PROPERTIES | ROYAL LEPAGE COMMERCIAL

BUY | SELL | FIND AN AGENT | INFO & ADVICE | JOIN OUR TEAM

168 OAK PARK AVE, Toronto, Ontario, M4C 4M8

$1,789,000 3+2 Beds 5 Baths

PROPERTY INFORMATION:

Designed With Natural Light In Mind, This Stunning Masterpiece Is A Show Stopper. Floor To Ceiling Windows, 10 Ft Ceilings, Skylights Throughout Mean Bright & Happy Space On The Gloomiest Days. Love Living Here! Ideal For Those That Appreciate Quality & Design As No Expense Spared With Top Level Finishes Including Book Matched Granite Back Splash Slabs In Kitchen, Floor To Ceiling Kitchen Cabinets & Glass Railings. Full List Of Quality Finishings Available. 3+2 Bedrooms, 5 Bathrooms Means The Privacy & Space You Crave. Incredible Self Contained In-Law Suite Featuring High Ceilings, Oversized Windows & Private Entrance. High Quality Finishings Throughout With Private Laundry And One Bedroom. Bright Happy Space Yet Warm & Cozy With Spray Foam Insulation. Convenient Parking With Rare Private Drive. Great Location! Walk To Taylor Creek Park-Miles Of Paved Trails, Rec Centre For Swimming, Ice Skating & Yoga Etc. Bike On Safe Bike Lanes To Beach For Evening Strolls Along The Boardwalk.**** EXTRAS **** Enjoy The Shops Of The Danforth Or A Multitude Of Big Box Shopping Nearby. Only A Few Minutes To The Core Of The City You Will Love Living Here. **Open House Sat/Sun 2-4 (April 22/23)** (id:7525)

BUILDING FEATURES:

Style:	Detached	**Exterior Finish:**	Stucco
Building Type:	House	**Fireplace:**	Yes
Basement Development:	Finished	**Heating Type:**	Forced air
Basement Type:	N/A	**Heating Fuel:**	Natural gas
Construction Style - Attachment:	Detached	**Cooling Type:**	Central air conditioning

Powered by:

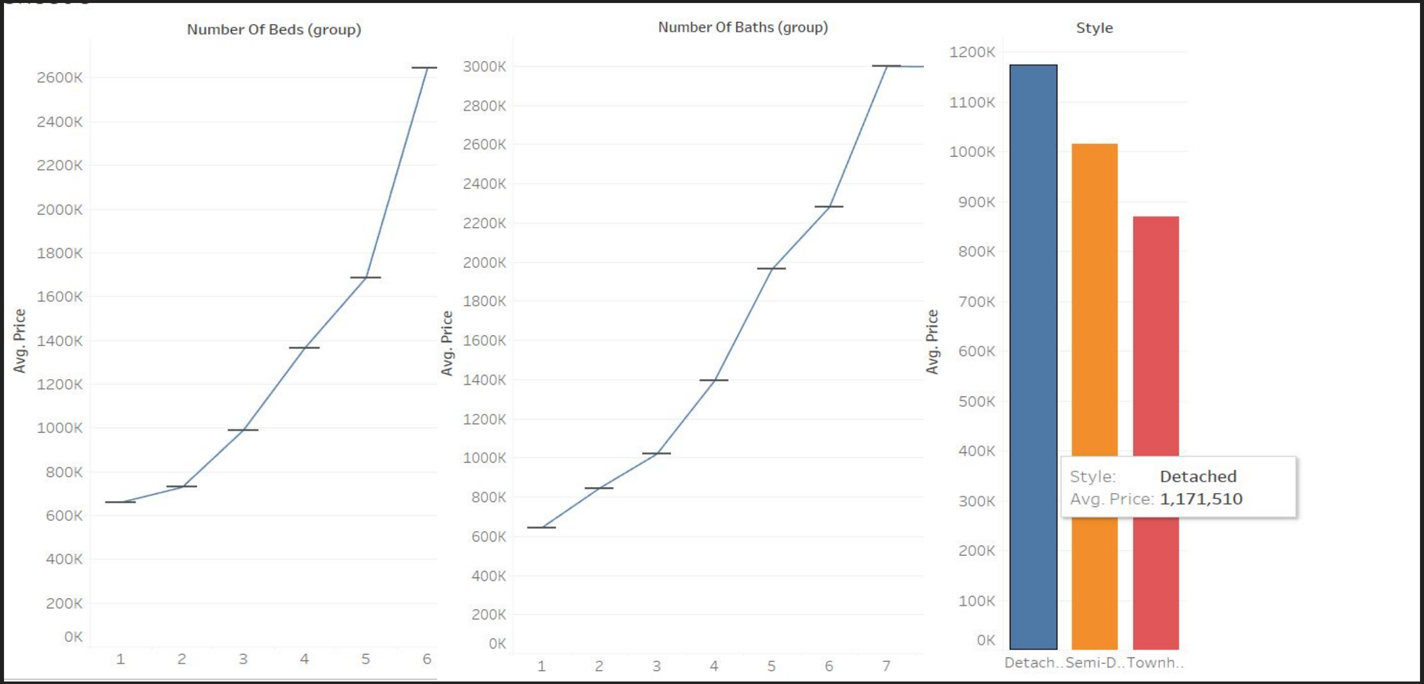
Gathered Data

	lat	lon	city	style	building_type	cooling_type	heating_type	price	number_of_beds	number_of_baths	extra_space	fire_place
0	43.33341	-79.8829	Waterdow	Townhouse	Row / Townhouse	Central air conditioni	Forced air	749900	2	3	0	0
1	43.85637	-79.3377	Markham	Detached	House	Central air conditioni	Forced air	2998000	5	8	1	1
2	43.67089	-79.3169	Toronto	Detached	House	Central air conditioni	Forced air	2599900	3	4	0	1
3	43.58962	-79.6444	Mississau	Semi-Detached	House	Central air conditioni	Forced air	2499850	4	4	1	0
4	43.44744	-79.6667	Oakville	Detached	House	Central air conditioni	Forced air	1999000	5	4	0	1
5	43.65001	-79.4828	Toronto	Detached	House	Wall unit	Hot water radiator heat	1499000	3	2	0	0
6	43.15614	-79.9162	Mount Ho	Detached	House	Central air conditioni	Baseboard heaters, Forced	1425000	3	3	0	0
7	43.88012	-79.4393	Richmond	Townhouse	Row / Townhouse	Central air conditioni	Forced air	1199000	3	3	0	0
8	43.88012	-79.4393	Richmond	Detached	House	Central air conditioni	Forced air	1098000	3	2	0	0
9	43.68581	-79.7599	Brampton	Detached	House	Central air conditioni	Forced air	999900	4	2	0	1
10	43.89756	-78.8635	Oshawa	Detached	House	Central air conditioni	Forced air	949900	4	3	1	0
11	43.58962	-79.6444	Mississau	Townhouse	Row / Townhouse	Central air conditioni	Forced air	875000	3	3	0	1
12	45.42088	-75.6901	Ottawa	Detached	House	Central air conditioni	Forced air	866000	3	2	0	0
13	43.64134	-79.3998	Toronto	Townhouse	Row / Townhouse	Wall unit	Baseboard heaters	838000	2	2	1	1
14	43.68581	-79.7599	Brampton	Townhouse	Row / Townhouse	Central air conditioni	Forced air	749888	3	3	0	1
15	45.42088	-75.6901	Ottawa	Townhouse	Row / Townhouse	Central air conditioni	Forced air	645900	3	2	0	1
16	45.01542	-75.6446	Kemptvill	Detached	House	Wall unit	Baseboard heaters, Other	599999	2	1	0	1
17	43.14082	-80.2632	Brantford	Townhouse	Row / Townhouse	Central air conditioni	Forced air	599000	3	3	0	0
18	45.04141	-74.734	Cornwall	Detached	House	Central air conditioni	Forced air	579000	2	2	1	0
19	45.82133	-77.1105	Pembroke	Detached	House	Central air conditioni	Forced air	550000	3	2	0	1
20	42.10133	-83.1086	Amherstb	Detached	House	Central air conditioni	Forced air	549900	3	2	0	0
21	43.25608	-79.8729	Hamilton	Townhouse	Row / Townhouse	Central air conditioni	Forced air	400000	2	3	1	0
22	43.25608	-79.8729	Hamilton	Townhouse	Row / Townhouse	Central air conditioni	Forced air	400000	2	3	1	0
23	42.96929	-81.2053	London	Townhouse	Row / Townhouse	Central air conditioni	Forced air	399900	4	2	0	0
24	43.62186	-79.9439	Halton Hil	Detached	House	Central air conditioni	Forced air	2999000	4	4	0	1
25	44.35968	-78.7422	Kawartha	Detached	House	Central air conditioni	Forced air	2900000	6	6	0	1

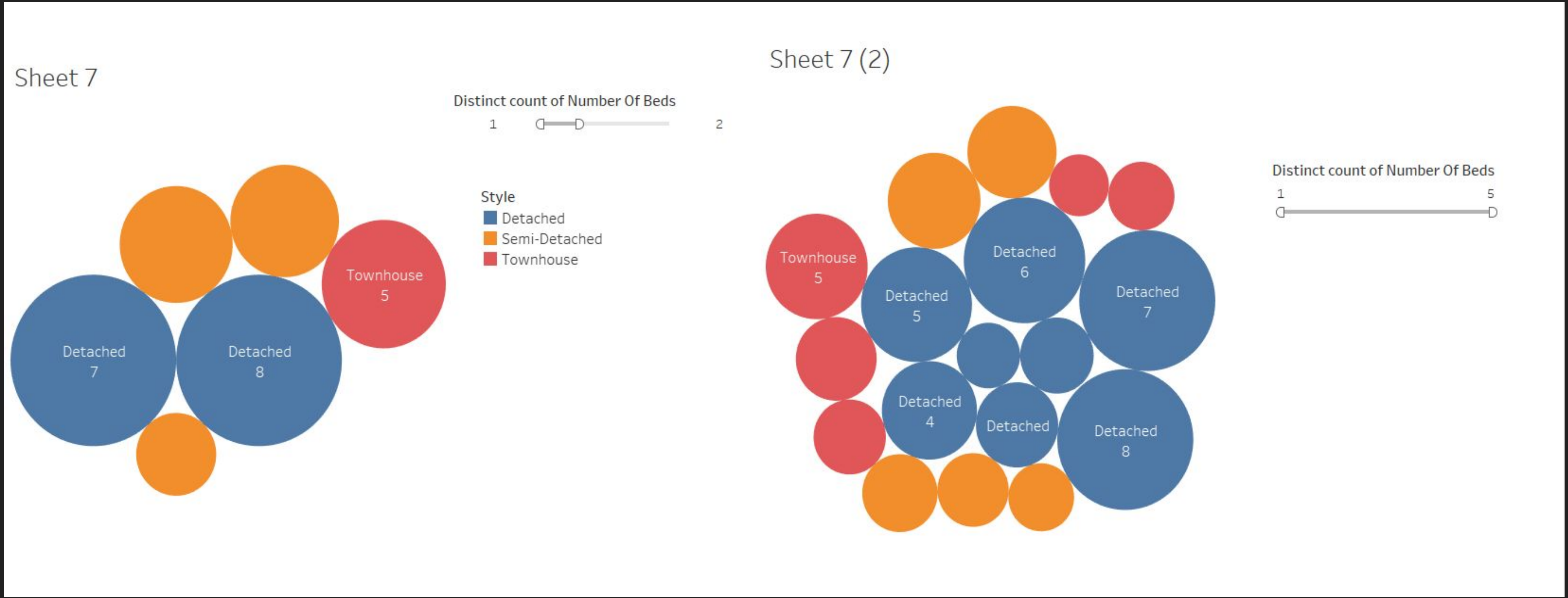
predicting_house_price

Data Visualization with Tableau

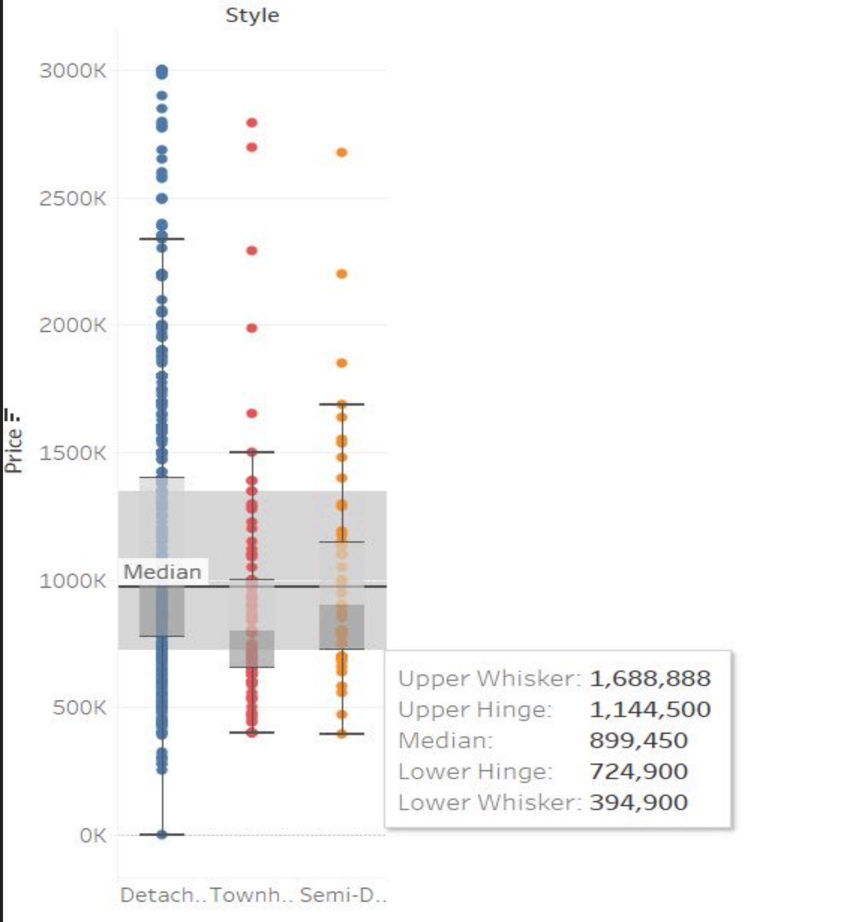
From these graphs we can see price for the house depends on the beds , baths and style of the house.



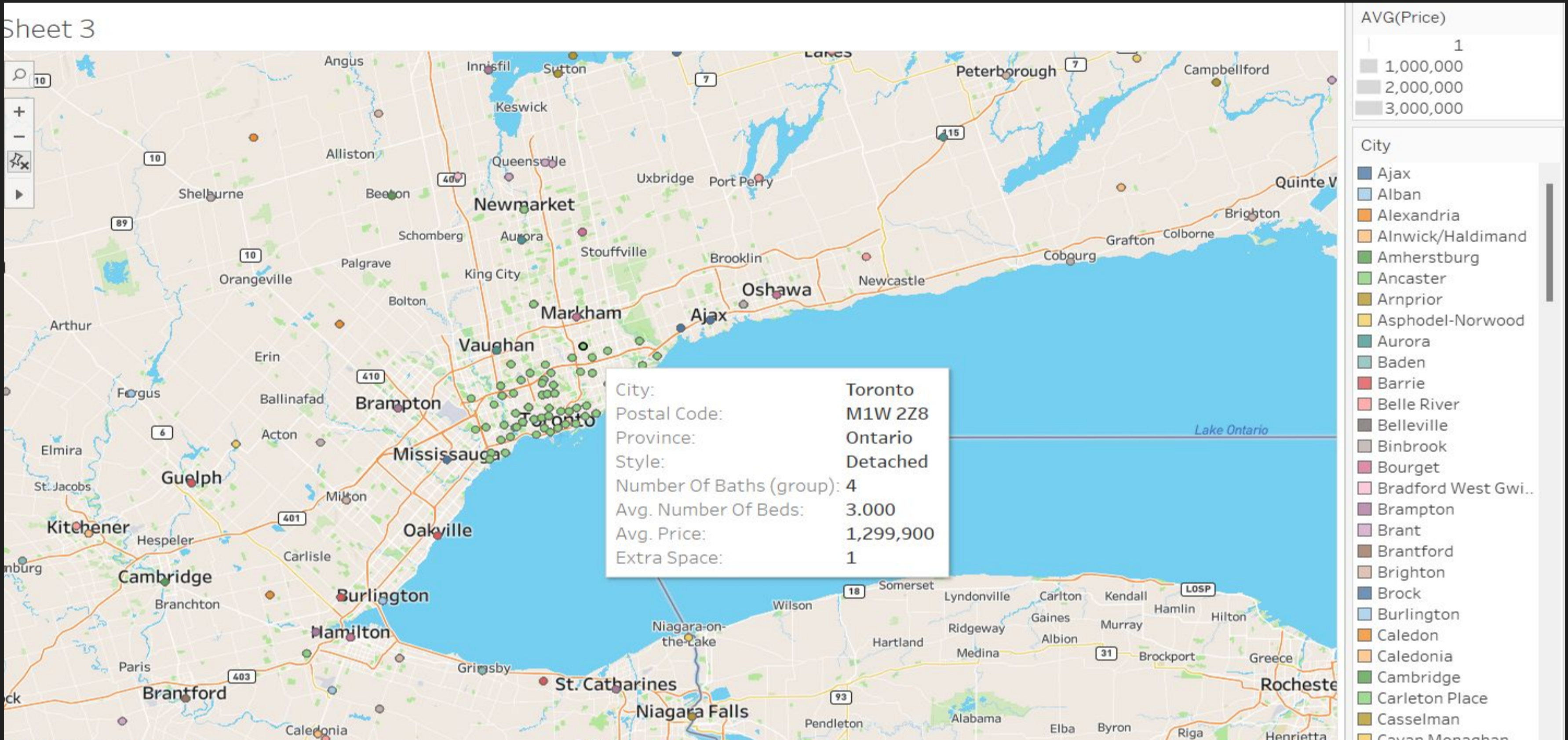
Here in this visualization we can see the style whether it is detached , semi-detached or townhouse , it depends on the number of beds and baths. As in here we have much less houses if we filter number of beds to 1-2.



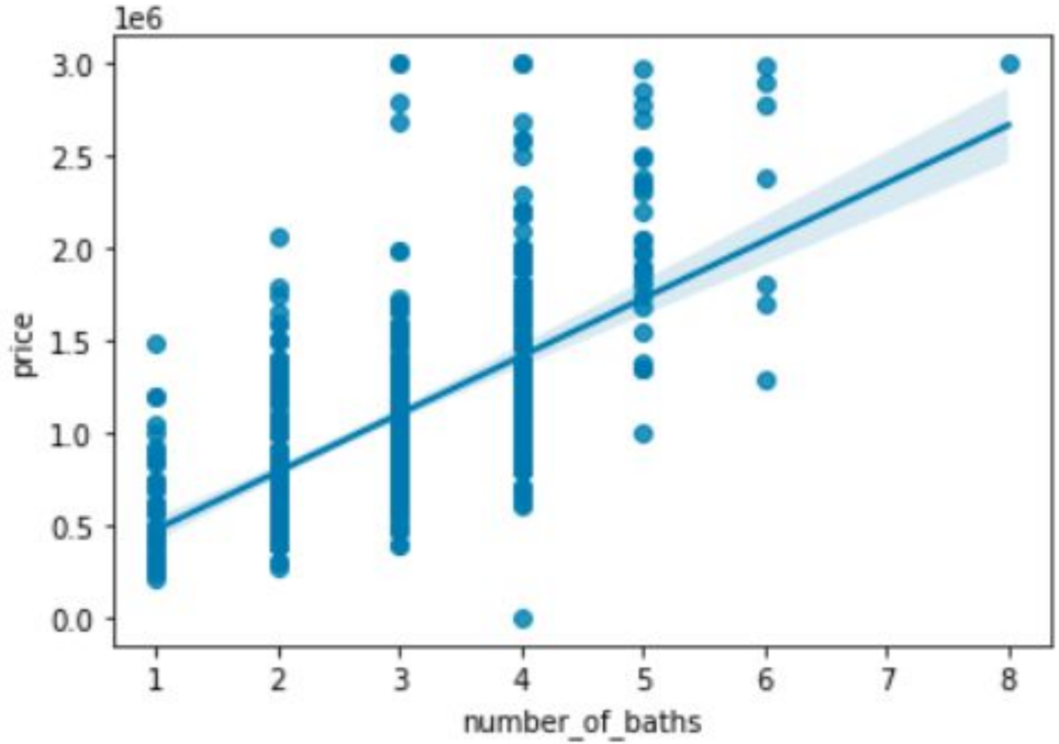
In this box plot we can see how the data is distributed and also the outliers in the graph.



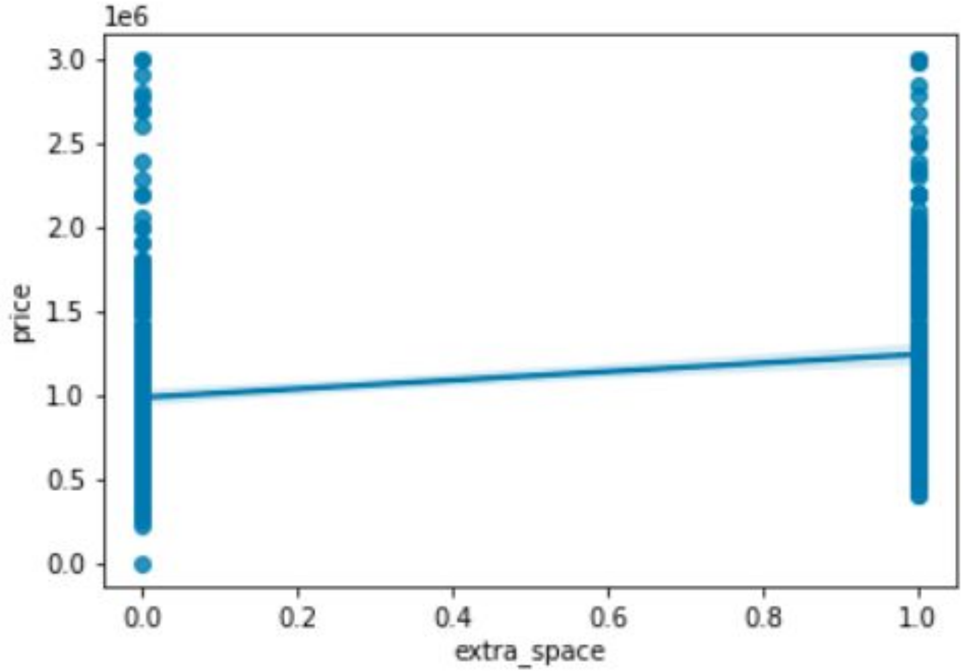
For the map we use lat and long from predicting_house_price_csv which gives us detailed information about the houses.



Out[90]: <AxesSubplot:xlabel='number_of_baths', ylabel='price'>



Out[91]: <AxesSubplot:xlabel='extra_space', ylabel='price'>



```
Out[128]: array([ 1.01585000e+06,  1.56945771e+14,  1.55946000e+06,  7.31953000e+05,
                  1.24823900e+06,  7.47800000e+05,  1.02449100e+06,  1.84256900e+06,
                  8.60310017e+13,  2.92055000e+05,  1.45721800e+06,  8.68272000e+05,
                  5.49327000e+05,  8.60310014e+13,  1.22818400e+06,  1.57012100e+06,
                  9.60157000e+05,  1.01606000e+06,  1.08099600e+06,  8.75368000e+05,
                  1.16884600e+06,  1.14375200e+06,  1.35823300e+06,  1.06866300e+06,
                  1.07616200e+06,  5.56999000e+05,  2.02168200e+06,  1.29882600e+06,
                  2.23158800e+06,  8.60310018e+13,  8.61417000e+05,  8.77717000e+05,
                  9.20583000e+05,  1.01358800e+06,  1.01101700e+06,  1.30957000e+06,
                  1.56945770e+14,  1.12527100e+06,  9.71176000e+05,  9.72932000e+05,
                  1.32770700e+06,  4.51835000e+05, -4.09490264e+14,  1.53429000e+06,
                  4.39205514e+14,  1.07075400e+06,  6.16012000e+05,  1.71434300e+06,
                  1.65613300e+06,  7.19220000e+05,  3.39068344e+14,  1.21616100e+06,
                  1.00097600e+06,  7.98285000e+05,  1.29615200e+06,  7.80019000e+05,
                  1.45792000e+06,  8.60310017e+13,  1.12165200e+06,  2.06074800e+06,
                  1.65035900e+06,  1.25313200e+06,  8.33179596e+13,  1.23697300e+06,
                  6.30908000e+05,  1.81828700e+06,  1.07091400e+06,  4.97357000e+05,
                  1.56142700e+06,  1.15737700e+06,  5.94689000e+05,  1.01517600e+06,
                  6.07801000e+05,  1.70729600e+06,  1.83378100e+06,  1.01822300e+06,
```

Try with different set amount of futures for Linear Regression

[30]:

```
y=df['median_house_value']
r2_result=[]
for i in trim_option:
    X_test_frame=df_data_scaled_transformed.iloc[:,i::]
    model_loop = LinearRegression()
    X_train, X_test, y_train, y_test = train_test_split(X_test_frame, y,random_state=1)
    model_loop.fit(X_train,y_train)
    LinearRegression()
    y_predict=model_loop.predict(X_test)
    r2 = r2_score(y_test, y_predict)
    r2_result.append(r2)
```

[32]:

```
r2_result
```

t[32]:

```
[0.62154995142617,
 0.6092621922301564,
 0.6091581153212526,
 0.6091442221149217,
 0.5779790129316478]
```

20]:

```
y_predict
```

20]:

```
array([1348056.29551521, 1005081.55982772, 1005081.55982772,
       1005081.55982772, 1348056.29551521, 1005081.55982772,
       1005081.55982772, 1348056.29551521,  319132.08845274,
       1005081.55982772, 1005081.55982772,  662106.82414023,
        662106.82414023, 1348056.29551521, 1005081.55982772,
       1348056.29551521, 1005081.55982772, 1348056.29551521,
       1691031.0312027 , 1348056.29551521, 1005081.55982772,
       1348056.29551521, 1005081.55982772, 1348056.29551521,
        662106.82414023, 1005081.55982772, 1348056.29551521,
       1005081.55982772, 1005081.55982772, 1348056.29551521,
       1005081.55982772, 1348056.29551521, 1005081.55982772,
       1005081.55982772, 1005081.55982772, 1691031.0312027 ,
       1005081.55982772, 1005081.55982772, 1005081.55982772,
       1005081.55982772, 1005081.55982772,  662106.82414023,
        662106.82414023, 1348056.29551521,  662106.82414023,
       1005081.55982772, 1348056.29551521, 1005081.55982772,
       1348056.29551521,  662106.82414023, 1005081.55982772,
       1005081.55982772, 1005081.55982772, 1005081.55982772,
       1005081.55982772, 1005081.55982772, 1348056.29551521,
```

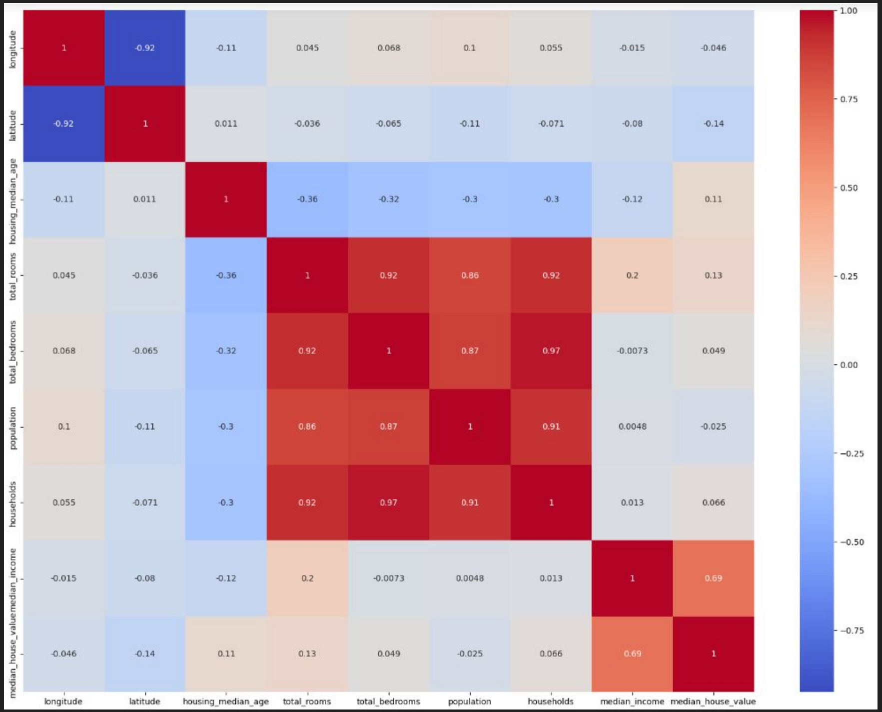
Methods used to increase the accuracy of the model

After not achieving the desired accuracy, another dataset with more number of columns was used for a random forest regressor model which works better for more complex models and different methods were implemented to improve the accuracy of the model such as:

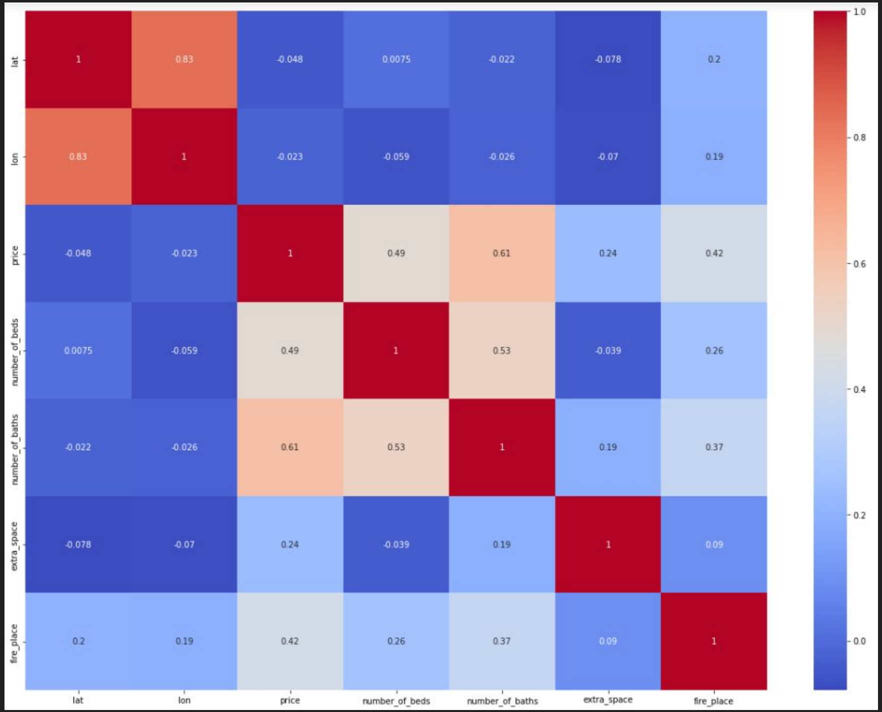
1. Cleaning and preprocessing the data
2. Finding the optimal value for the other hyper-parameters
3. Removing the least relevant columns from the features
4. Trying bagging to reduce variance in the model by creating random subsets of the training dataset and training multiple decision trees based on them
5. Using boosting to reduce the bias in the model by iteratively adjusting the weight of the training data
6. Increasing the size of the training dataset

Using the mentioned methods, helped to increase the accuracy from 70% on the first iteration to 76%.

Correlation Matrix of the new dataset



Correlation Matrix of the old dataset



Analyzing the features' importance

```
[(0.5675130203384309, 'median_income'),
 (0.16272857179933684, 'ocean_proximity_INLAND'),
 (0.07388171641296075, 'population'),
 (0.0669345253793256, 'housing_median_age'),
 (0.05431434066732454, 'total_bedrooms'),
 (0.03441077479484053, 'households'),
 (0.030205587219674666, 'total_rooms'),
 (0.004990653452852285, 'ocean_proximity_NEAR OCEAN'),
 (0.0034462953614574653, 'ocean_proximity_<1H OCEAN'),
 (0.001574514573796469, 'ocean_proximity_NEAR BAY')]
```

Thank you for listening!!