# Battle of Neighborhoods

## Finding Relationships Between Similar Clusters of Neighborhoods and Housing Prices

# Contents

# Introduction

Having one's own home is probably the desire of every individual on this planet. However, buying a house is not an easy feat and one needs to put a lot of considerations before the deciding to buy one, the price of a house being the most important.

Housing prices can vary on a lot of aspects, such as the size of the house, the number of bedrooms or bathrooms, etc. Also, the neighborhood where the house is located can count heavily in determining the price of a house as well.

In this Capstone Project, we would try to understand what impact the locality has in determining the housing prices in King County, USA. We would use Foursquare APIs to fetch the local venues for neighborhoods and then cluster the neighborhoods based on few top localities for each neighborhood and then try to understand if there is any correlation between these clusters and housing prices.

## Data

- **kc_house_data**: The dataset has been taken from Kaggle and is widely used for a hands-on experience on training regression models. We will leverage this dataset to add another feature to it and understand if we can find correlations between housing prices and the clusters of neighborhoods that we will be creating further in this project.
  In this dataset, the most important column for our project is the column 'zipcode'. We will find the unique number of zipcodes that we have and later group our entire dataset according to zipcode.

  https://www.kaggle.com/harlfoxem/housesalesprediction

  Below is the list of columns that the dataset has:

  id: a 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
  floors: Total 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
  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)

- **Coordinates using geolocator library**: Once we have the zipcodes, we will try to find the coordinates for each of them using geolocator library and put it together into a dataframe. We will further be using these coordinates to explore nearby venues using Foursquare APIs.
- **Foursquare data**: Once we have the coordinates, we will analyze the neighborhoods for each of the zipcodes using Foursquare data. The data that we will receive will be in a JSON format. We will be extracting the necessary components from the data (categories) and then convert the extracted data into a pandas dataframe using **json_normalize.** We will then cluster the areas defined by zipcodes and later try to find if neighborhood has any impact on prices or not.

# Methodology

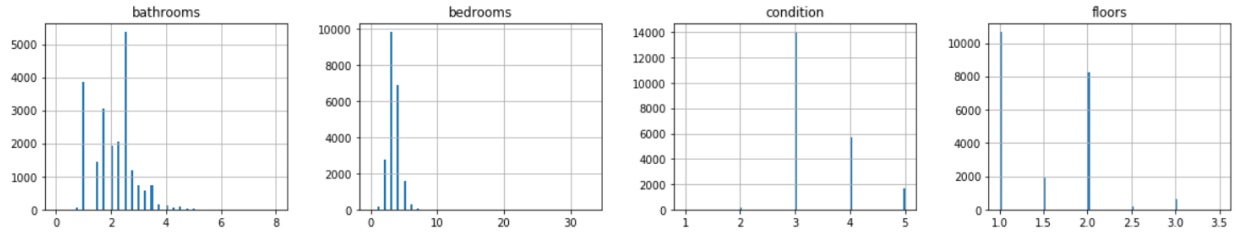
## Exploratory Data Analysis

We can quickly go through the kc_house_data dataset to understand the dataset. Below is a list of the columns and the dtypes for each of them.

```
kc_house_data.dtypes
id                int64
date             object
price           float64
bedrooms          int64
bathrooms       float64
sqft_living       int64
sqft_lot          int64
floors          float64
waterfront        int64
view              int64
condition         int64
grade             int64
sqft_above        int64
sqft_basement     int64
yr_built          int64
yr_renovated      int64
zipcode           int64
lat             float64
long            float64
sqft_living15     int64
sqft_lot15        int64
```

Here, price is the target value and all the following columns can be used as features. The columns id and date do not have much relevance; hence, we can drop those columns.
(The column date describes the date when the housing price value was recorded. On analyzing the dataset, we found that most of the dates belong to the year 2014-15. For learning purpose, we are assuming that the prices would still be the same and the venues in the neighborhoods (districts here) have grown proportionately).

Now, let's take a quick glance at the histograms for the numeric columns and try to understand the distributions for each of the features.
However, the two major columns that we are interested in are 'price' and zipcode. But we will just take a short look at each of the features for educational purpose. Also, it is interesting to look at the histograms as they reveal the distribution - whether it is normally distributed and if it is heavy on the either sides or if it follows a different pattern altogether.




From the above figures, we can say that all the four features: bathrooms, bedrooms, condition and floors are categorical in nature. Surprisingly, the number of bathrooms appear in decimal sizes. Since, it is quite intriguing, let us try to list out the unique values of bathrooms:

```
kc_house_data['bathrooms'].unique()
```

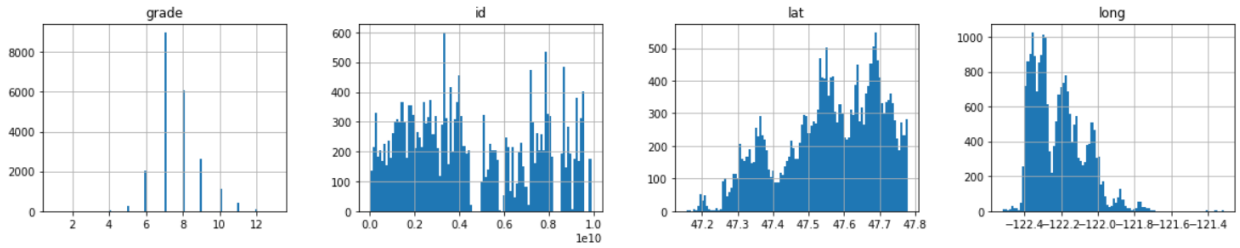
```
array([1.  , 2.25, 3.  , 2.  , 4.5 , 1.5 , 2.5 , 1.75, 2.75, 3.25, 4.  ,
       3.5 , 0.75, 4.75, 5.  , 4.25, 3.75, 0.  , 1.25, 5.25, 6.  , 0.5 ,
       5.5 , 6.75, 5.75, 8.  , 7.5 , 7.75, 6.25, 6.5 ])
```

Well, that's probably a truth and need to agree to it.

Moving further towards bedrooms, even though the data is categorical, it is tail heavy. Looks like we have houses with ~30-35 bedrooms!

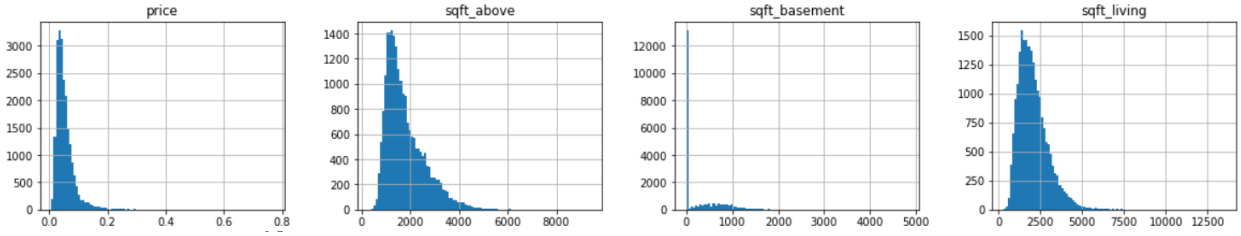
The condition of most of the houses seem to be in the range 3-5, 3 being the highest. However, we have very low number of houses that belong to categories 1 and 2.

Most of the houses have 1 or 2 floors with few exceptions

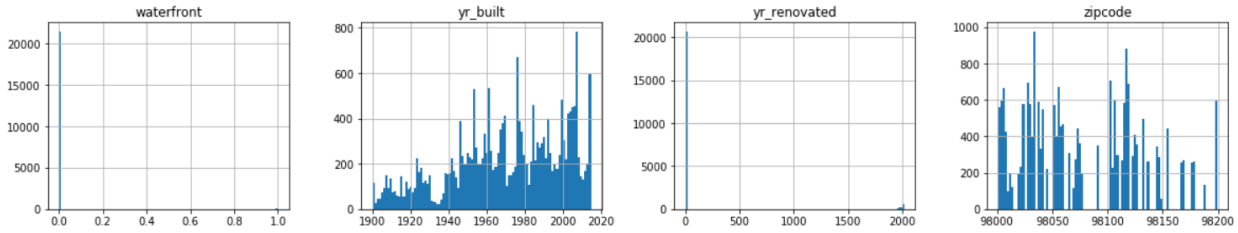



Let us consider the above graphs now. The histogram for grade is categorical, however, can be said to have a fairly normal distribution. The second graph for id can be ignored.

Looking at the coordinates, it seems like most of the houses are located at higher latitudes and lower longitiudes.



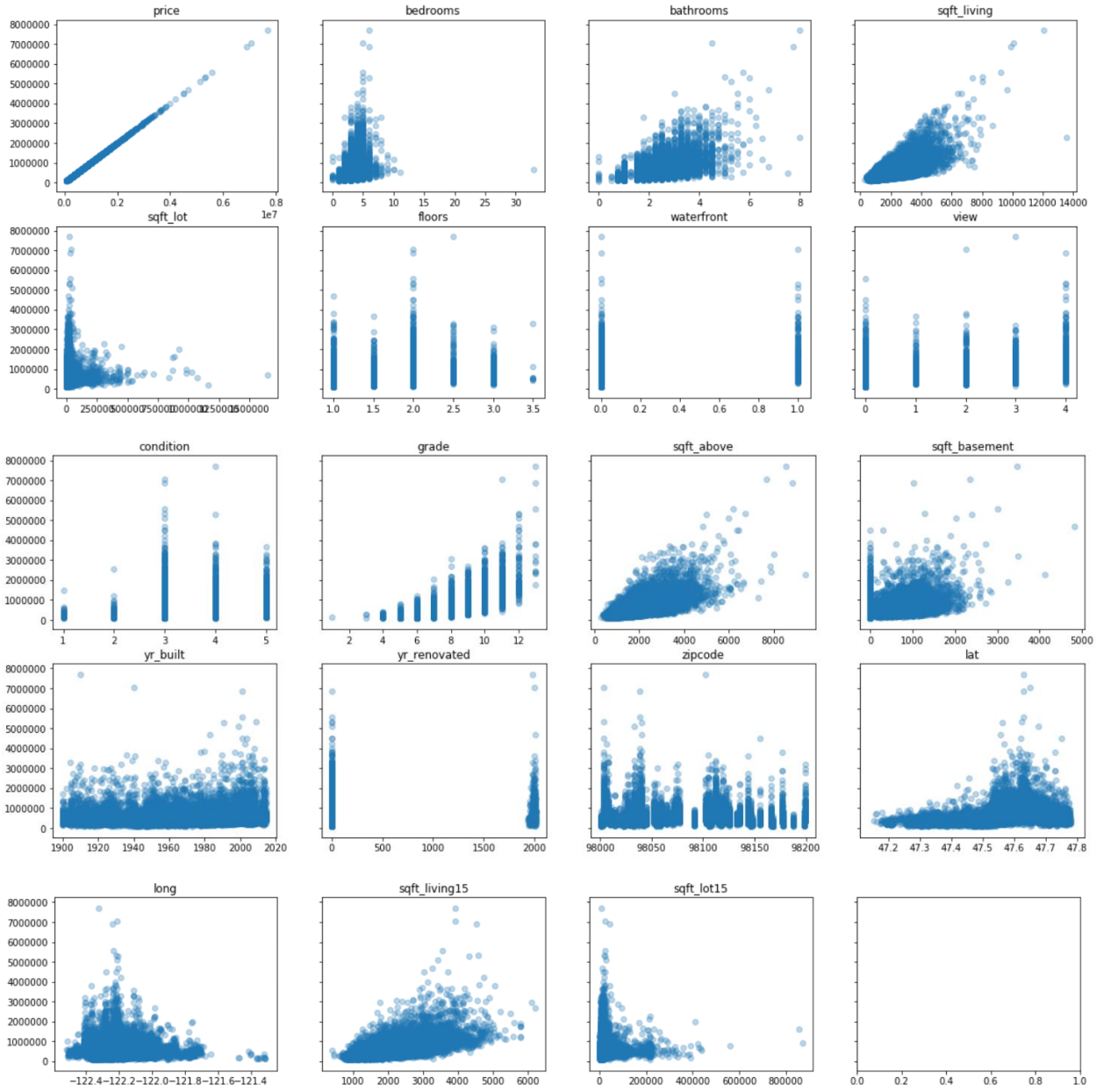

These histograms are quite interesting. All the four are fairly normal and tail heavy. Its easy to interpret each of the graphs except for the third one. We see a huge spike at point 0 which is indicative of the fact that most of the houses just don't have a basement.




It seems like most of the houses do have a waterfront. Also, with time newer houses were built. The yr_renovated column has a huge spike at 0 which means we don't have this data for most of the records. The distribution for zipcode is quite non-uniform and not much can be infered from this graph.

## Correlations

Now, let us try to understand the effect of these features on our target – price. We will draw scatterplots between the important parameters and find the correlation between price and the attributes based on the scatterplot we generate:




Following are the inferences we can easily draw from the above scatterplots:

1. There is a strong correlation between price and the features – bedrooms, bathrooms, sqft_living, sqft_above, and sqft_living_15
2. For the feature grade, it can be said to have a positive correlation. The data is categorical and there is an increase of prices with the grades.
3. It looks like for the feature sqft_lot, there is slightly a negative correlation.
4. For rest of the features, the correlation does not seem to be very dominant and we cannot say something concretely about them.

## Analyze Zipcodes

The most important column for our project is the column 'zipcode'. We try to find the unique number of zipcodes that we have and later group our entire dataset according to zipcode.

```
len(kc_house_data['zipcode'].unique())
```

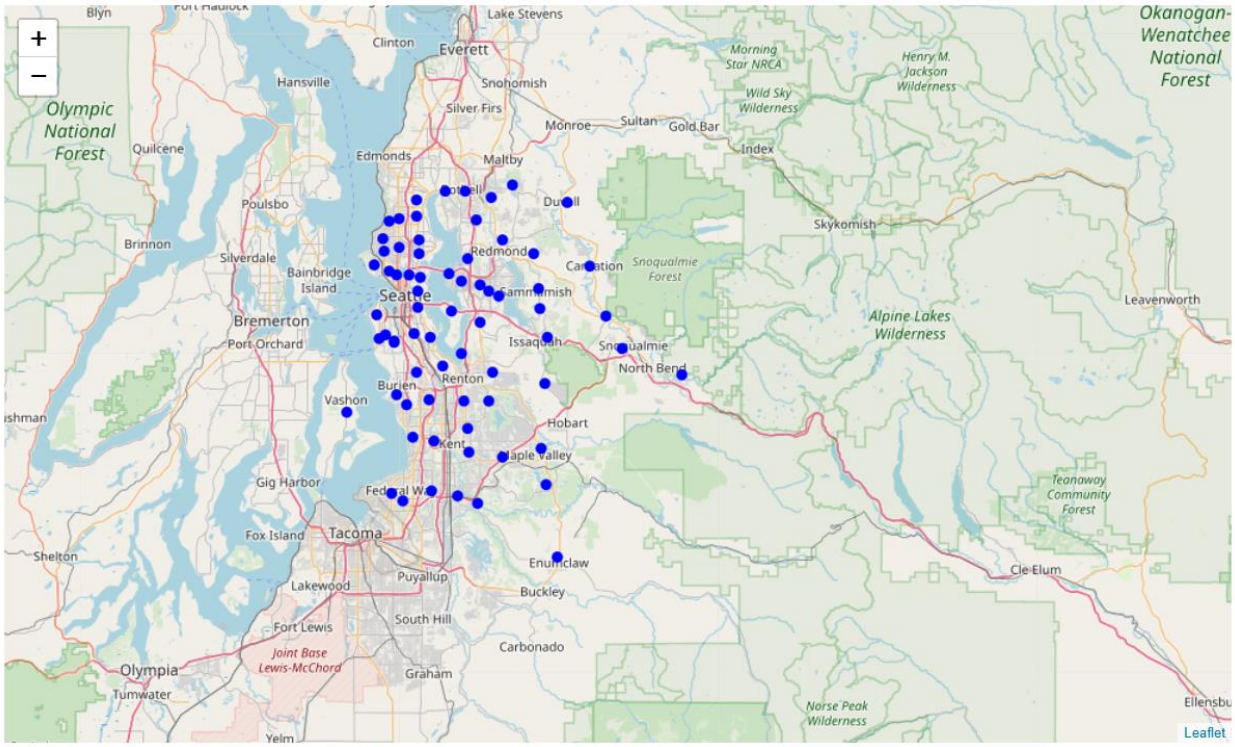
```
70
```

**Find coordinates for the zip_codes:** Now, when we have the zipcodes, we try to find the coordinates for each of them using **geolocator** library and put it together into a dataframe. We will further be using these coordinates to explore nearby venues using Foursquare APIs.

```
zipcode_coo.head()
```

| | zipcode | lat | lon |
|---|---|---|---|
| 0 | 98178 | 47.4986 | -122.248 |
| 1 | 98125 | 47.7215 | -122.305 |
| 2 | 98028 | 47.7569 | -122.242 |
| 3 | 98136 | 47.539 | -122.389 |
| 4 | 98074 | 47.6143 | -122.036 |

Now, we have the coordinates for our zipcodes and hence we will plot a map to see how the distribution looks like:




## Connecting with Foursquare

Now, since we have the coordinates, we will analyze the neighborhoods for each of the zipcodes using Foursquare data. We will then cluster the areas defined by zipcodes and later try to find if neighborhood has any impact on prices or not.

```
kc_venues = getNearbyVenues(zipcode_coo['zipcode'], zipcode_coo['lat'], zipcode_coo['lon'])
```

```
100.0
```

```
len(kc_venues['Neighborhood'].unique())
```

```
65
```

We observe that out of 70 zipcodes, we only have neighborhood data for 65 of them. Hence, while analyzing data for all neighborhoods, we will omit data for the five zipcodes for which we dont have the neighborhood data.

```
print('There are {} uniques categories.'.format(len(kc_venues['Venue Category'].unique())))
```

```
There are 198 uniques categories.
```

Now, we can see that there are 195 categories for the venues. Our objective is to find clusters of similar neighborhoods amongst these categories. To accomplish that, we will be using **KMeans clustering algorithm** for our project.

For the algorithm, the input feed needs to be in a format, where we have each of the 195 venues as features and a value for each parameter which indicates how strong the feature is. Hence, to achieve this, we can simply use one hot encoding to convert our data set into the required format and then for each neighborhood, calculate the mean for each of the features.

Below are some details as to how the data looks like after it is one-hot encoded.

```
kc_onehot.head()
```

| | Neighborhood | ATM | Accessories Store | Alternative Healer | American Restaurant | Antique Shop | Arts & Crafts Store | Asian Restaurant | Athletics & Sports | Auto Workshop | Automotive Shop | BBQ Joint | Bagel Shop | Bakery | Bank | Bar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 98178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 98178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 98178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 98178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 98178 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```
kc_grouped = kc_onehot.groupby('Neighborhood').mean().reset_index()
kc_grouped.shape
```

```
(65, 199)
```

Let's take a quick look as how many total venues do we have for each of the neighborhood.

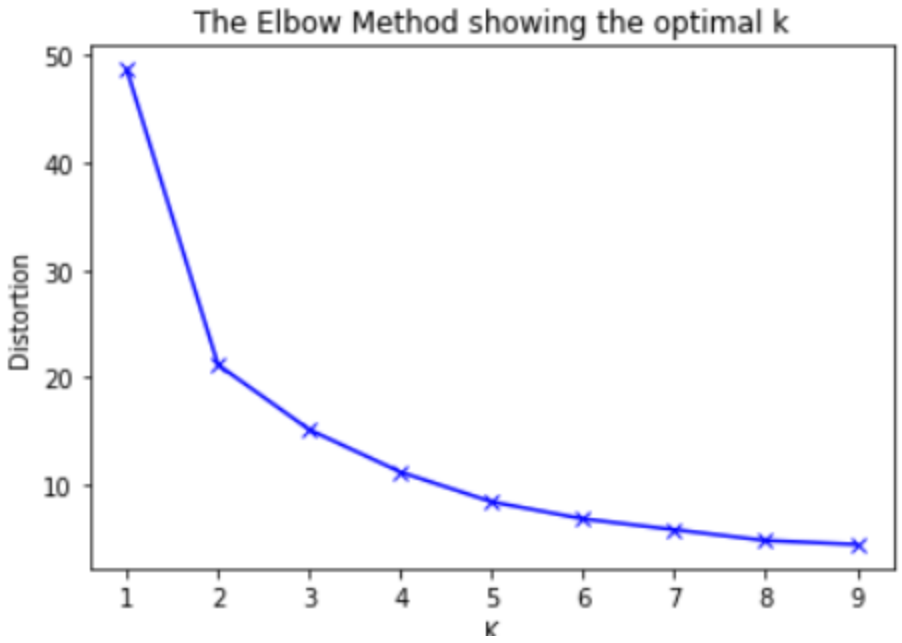
```
kc_grouped_by_count = kc_onehot.groupby('Neighborhood').count().reset_index()
kc_grouped_by_count['ATM'].value_counts()
```

```
4      9
5      9
3      8
1      6
2      5
7      4
9      4
10     3
23     2
6      2
19     2
11     1
100    1
43     1
17     1
18     1
21     1
27     1
29     1
30     1
36     1
12     1
Name: ATM, dtype: int64
```

## Applying the Machine Learning Algorithm

Now, we have our data ready to be fed into the KMeans machine learning algorithm. This is an unsupervised learning approach where the data is segmented into **k** clusters in which each data point belongs to the nearest mean. Here, our clustering will be based on the types of venues in a neighborhood, i.e., how similar are neighborhoods based on the most common venues they have.

Even when the algorithm is unsupervised, we do need to provide the number of clusters that the dataset will be divided into, which can be a hard problem to solve. Hence, we will take a step further and run our clustering algorithm for clusters within the range 1-9 and try to find out the optimum value of clusters that we need to provide using the elbow method.



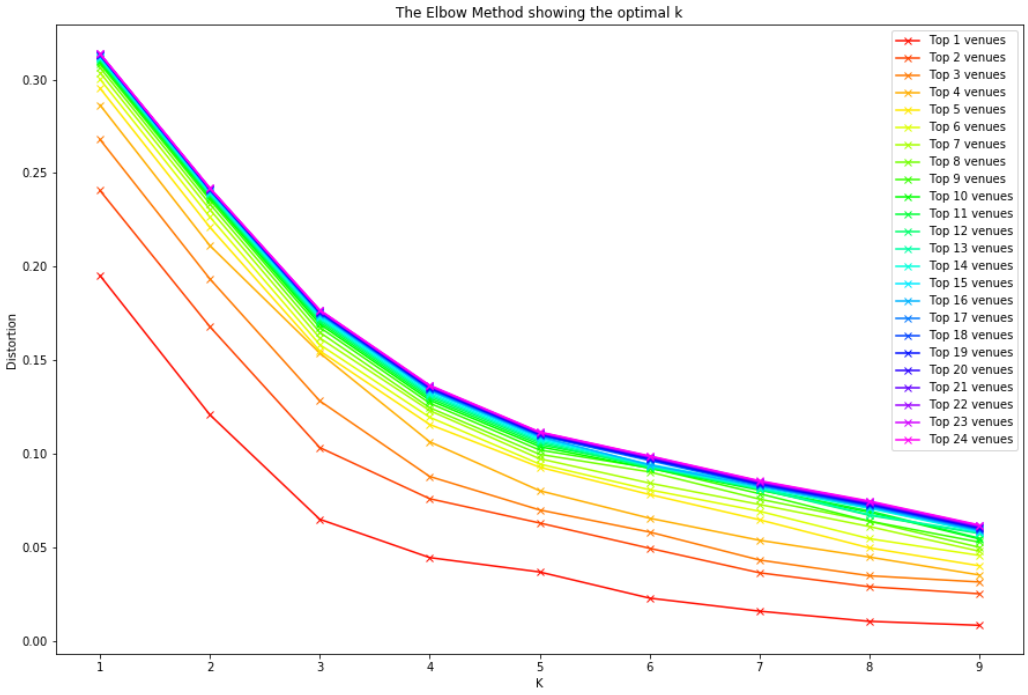

By the elbow method, we can clearly see that we have the elbow point at 2. Hence, we will proceed with 2 as the number of clusters we would be dividing our dataset into. Let us see how good the clusters are:

```
1    63
0     2
dtype: int64
```

Okay! Our algorithm did divide the dataset into two clusters, but the disparity is huge. This is clearly due to lack of data and we cannot efficiently say if we would be able to see a reliable correlation between these clusters and prices.

Let us take a step further and try to find out how we can get better clusters, and maybe more clusters with better distributions.

Since we have limited data, one approach would be to pass just a limited number of venues instead of all the 195 venues. Hence, we can just pick **top n** venues and run our algorithm on it. Further, we will plot the elbow curves to determine the optimum number of top n venues and the optimum number of clusters that we need to feed to our algorithm to get better results.




This looks great! On looking closely, we can see that we have two good elbow points:

- Top 2 venues and 3 clusters
- Top 3 venues and 4 clusters

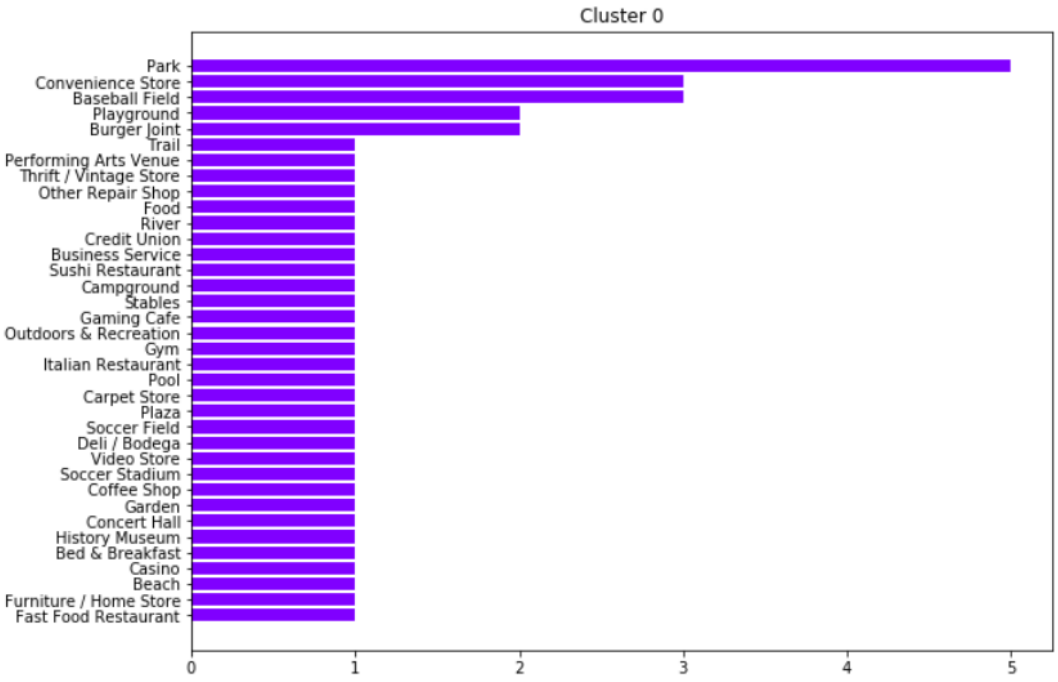
Because we have limited data, let's go with the first option (it won't hurt to select the second option too) and run our algorithm on these two parameters. The clusters we receive are as follows:

```
1    36
0    23
2     6
dtype: int64
```

Great! These clusters look better than the one that we generated using the entire dataset. Clearly, the disparity is low, and we are expecting the clusters to make some sense as well. Let us explore further to understand if these clusters do indicate a type of neighborhood or not.

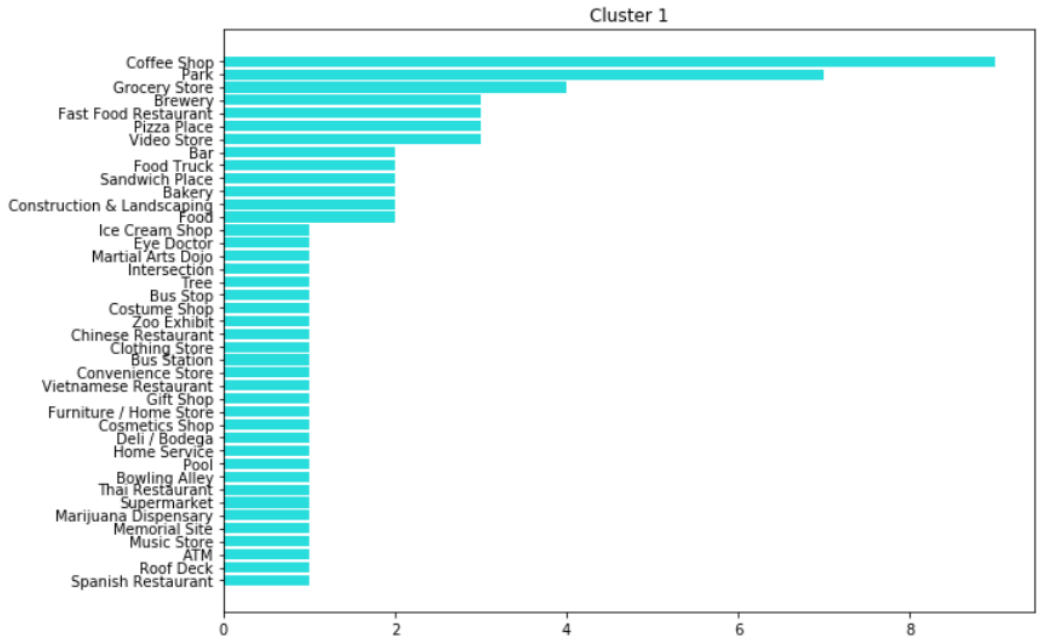
## Results

Now, when we have the clusters, let us create a bar graph for each of the clusters with frequencies of the type of venue that they contain and try to understand the clusters:



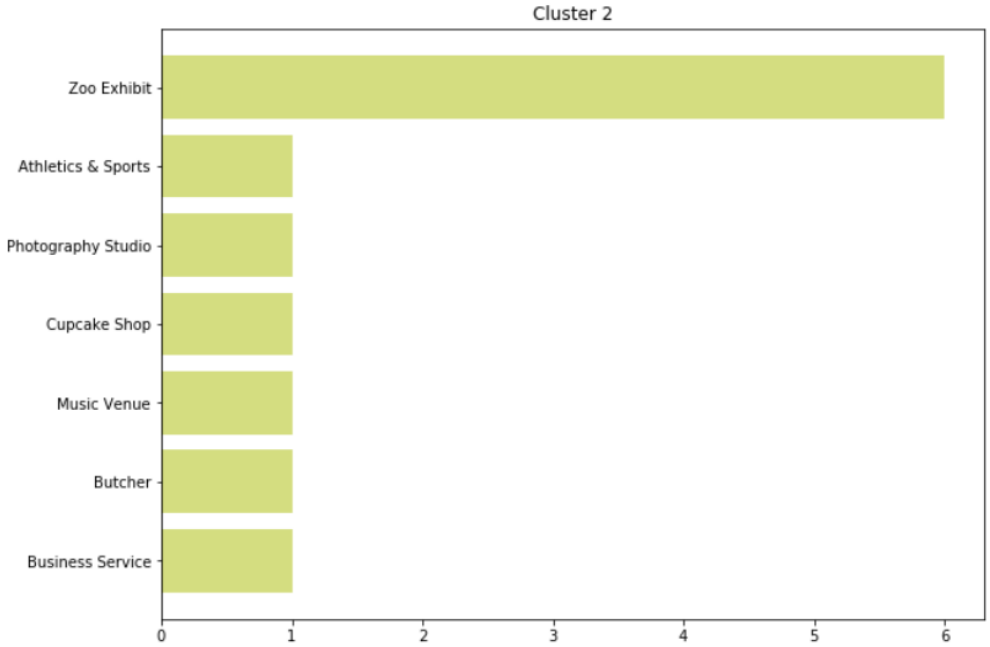

**Cluster 0**: Outdoor activities

In cluster 0, we can clearly see that parks, fields, tracks and playgrounds are predominant. Along with it, we do see some shops, casino, and museums as well. However, we do see some restaurants and other anomalies which might not fall within the category 'Outdoor activities', but most of the venues do fit into this description.




**Cluster 1**: Cafes and restaurants

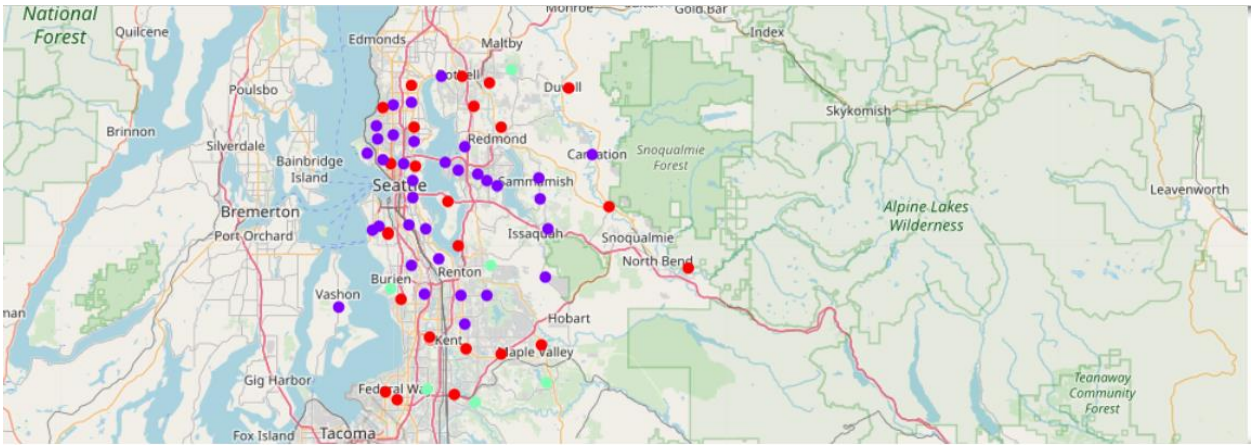
Cluster 1 mostly comprises of coffee shops and a variety of restaurants. However, we do see a significant anomaly - 'park', the frequency of which is relatively high. But since most of the places belong to the category of eateries, we can safely tag this category as 'Cafes and restaurants'.



**Cluster 2**: Zoo and misc.

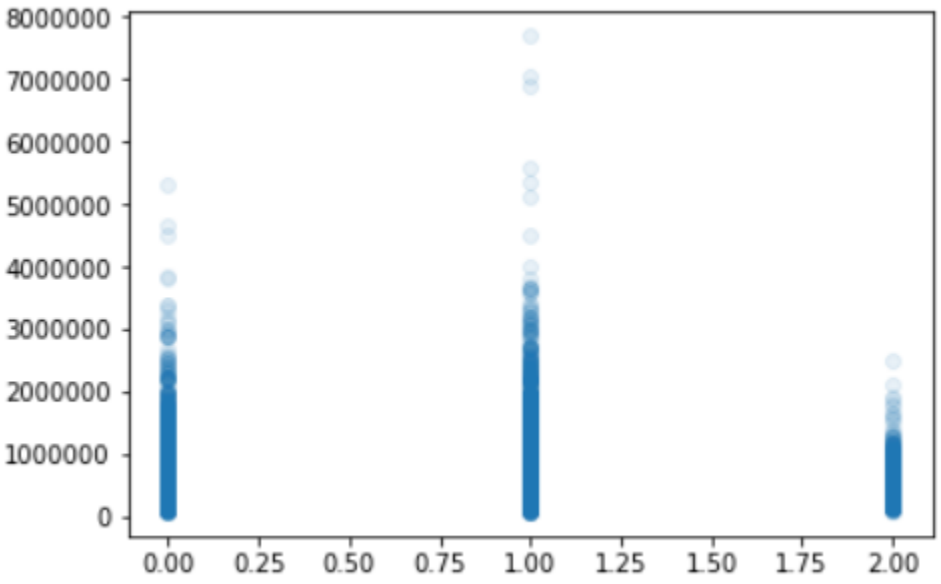
This cluster has a relatively high frequency of zoos. The other places are quite varied and do not seem to fit in a single category of venues. Probably they are built around the zoos.

Let us try to plot these clusters into a map to see how the distribution looks like:



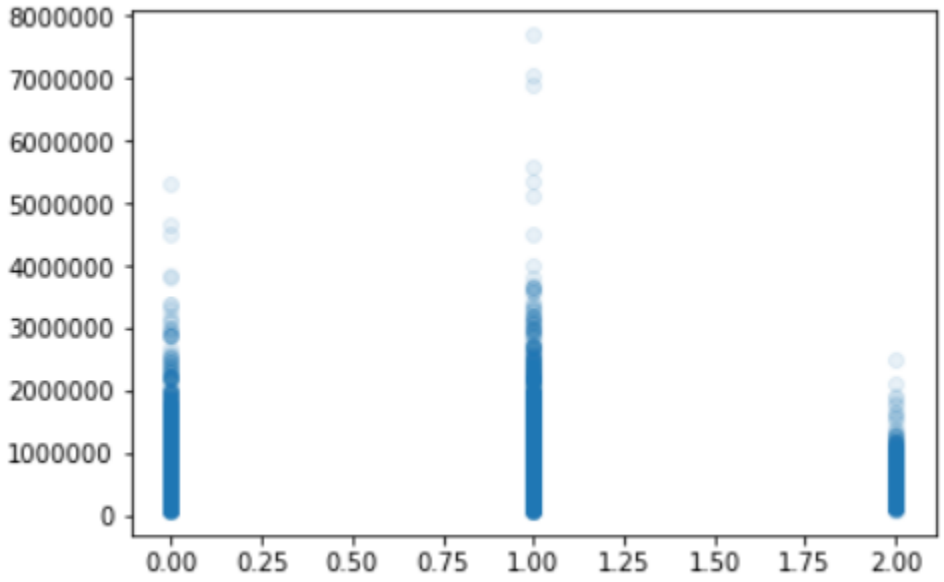

Red represents Cluster 0, purple represents Cluster 1 and green represents cluster 2.

Now, as our final analysis, let us see what impact these clusters has on housing prices. We will plot a scatter plot between these clusters and prices and try to find a correlation between them.



## Discussion

Let's take a glimpse of the graph for price vs clusters that we obtained in the previous section:




Interestingly, we can see a pattern emerging out of our clusters. It looks like houses in the neighborhoods which have more number of eateries have a greater range of house prices than the other two clusters. Even though the cluster seems to have outliers, we can say that the more expensive houses belong to this category.

Furthermore, it can be said that houses in neighborhoods which have zoos, do not have a bigger range of selling prices. However, this cannot be very accurately said as we have limited data for this cluster.

# Conclusion

In this project, we tried to find a relationship between similar neighborhoods based on the venues they contain and the house prices in those areas. The data sets we used for this project were retrieved from multiple sources such as Kaggle and Foursquare. Once we had the required datasets, we performed the required data cleaning to make it consumable for the machine learning algorithm.

To cluster our dataset into clusters of similar neighborhoods, we used an unsupervised machine learning approach – KMeans. Initially, the clusters that we received by feeding in the entire dataset were not uniformly distributed and hence we took a step further to find the top n venues that we need to feed in to the algorithm to fetch the best results.

Once the correct set of parameters were found, we analyzed the impact of these clusters on housing prices.

**After this analysis, we can conclude that the division of cluster can be a fair addition to the set of features which can influence housing prices. Since there is a correlation that emerges out of this division, once we have access to more data, it can be a good predictor in predicting the selling price of a house based on the venues around it.**