

Clustering Hong Kong's neighborhoods

Find your next apartment in similar yet cheaper locations

Introduction

- Hong Kong – financial center
- Large expat population
- Communication problem
- High rents
- Ideal neighborhood

Data

- Dataset from Spacious

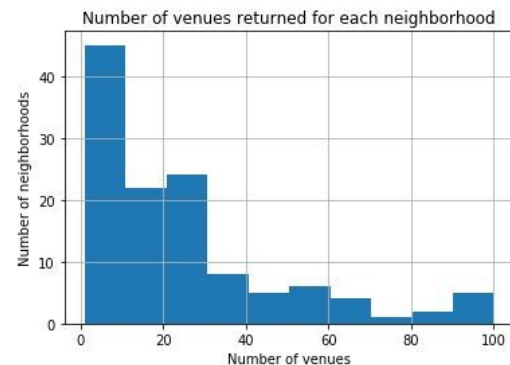
	Neighborhood	Latitude	Longitude	District	Rent per square foot
0	Kennedy Town	22.2824348	114.1284168	Central and Western	63
1	Shek Tong Tsui	22.287735	114.1345987	Central and Western	62
2	Sai Ying Pun	22.286121	114.1420862	Central and Western	69

- Venue data from Foursquare

	Neighborhood	District	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Kennedy Town	Central and Western	22.282435	114.128417	Little Creatures	22.283950	114.128264	Gastropub
1	Kennedy Town	Central and Western	22.282435	114.128417	Winstons Coffee	22.281374	114.127172	Coffee Shop
2	Kennedy Town	Central and Western	22.282435	114.128417	Catch.	22.283152	114.126988	Breakfast Spot

Methodology

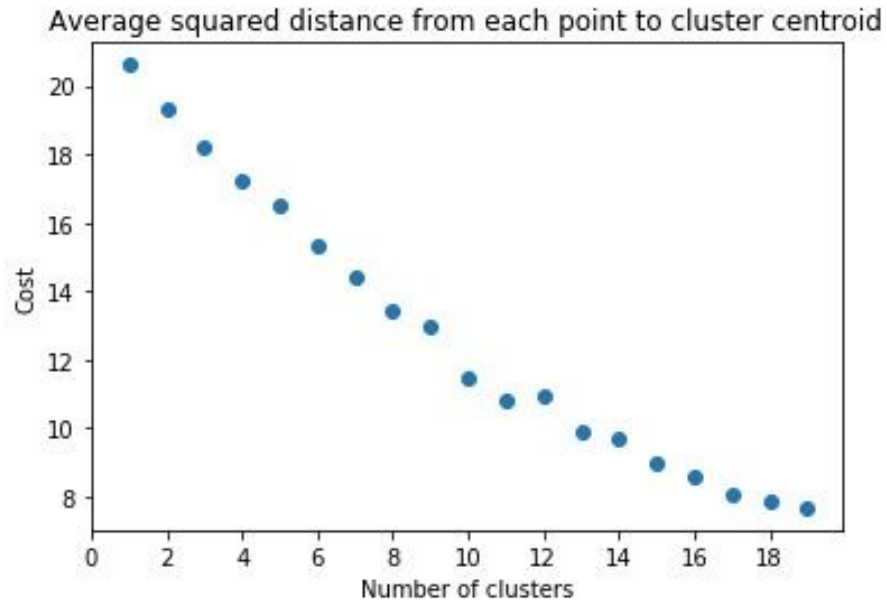
- Right-skewed Foursquare venue data
- Data processing including one-hot encoding and standardization



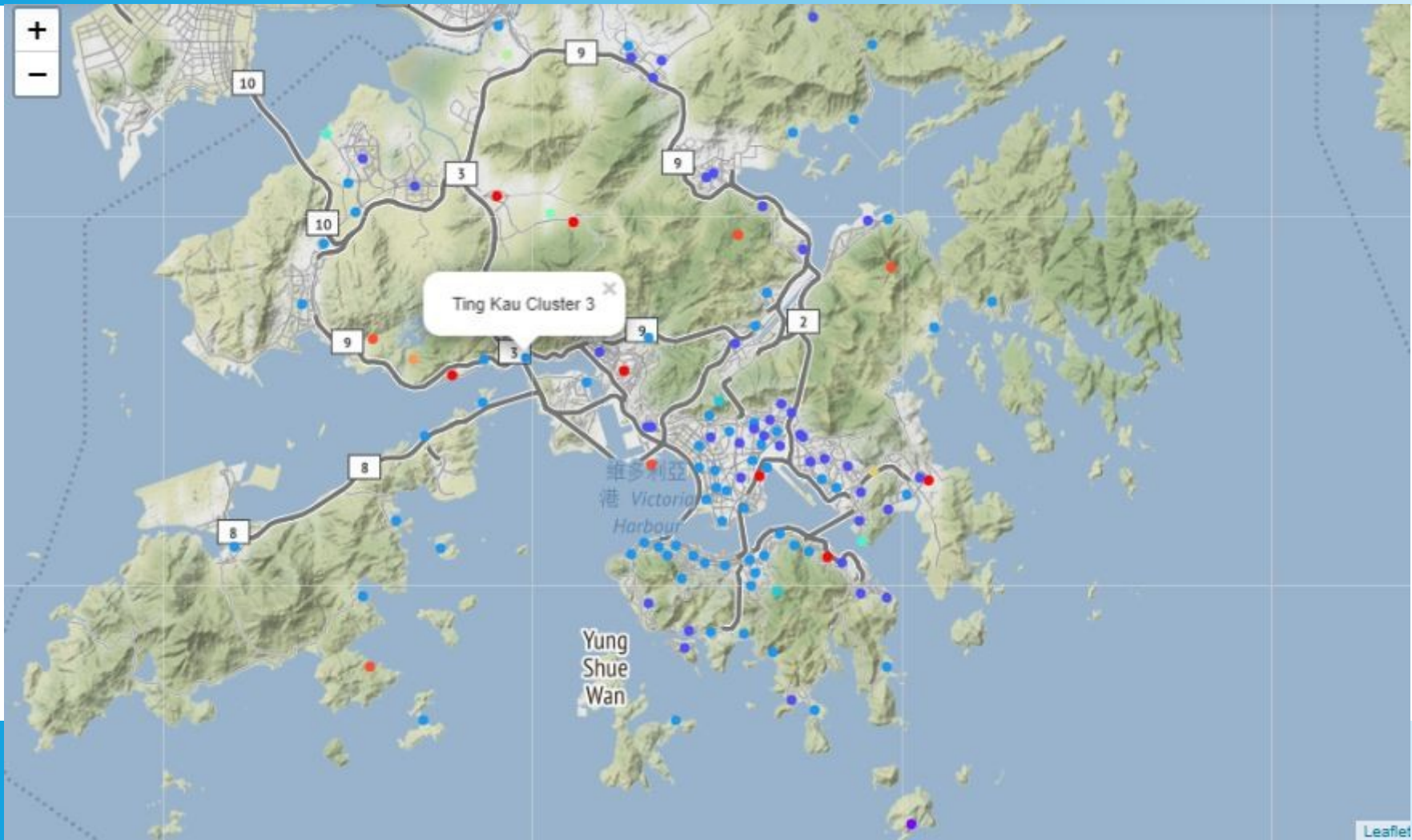
	Neighborhood	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue
0	Aberdeen	Sushi Restaurant	Athletics & Sports	Convenience Store	Shopping Mall	Bus Station
1	Admiralty	Café	Hotel	Steakhouse	Park	Cantonese Restaurant
2	Ap Lei Chau	Fast Food Restaurant	Seafood Restaurant	Shopping Mall	Chinese Restaurant	Furniture / Home Store
3	Beacon Hill	Mountain	Scenic Lookout	Fast Food Restaurant	German Restaurant	Gastropub
4	Braemar Hill	Japanese Restaurant	Taiwanese Restaurant	Ramen Restaurant	Chinese Restaurant	Zoo

Methodology (continued)

- Optimal number of clusters is 10 using elbow method

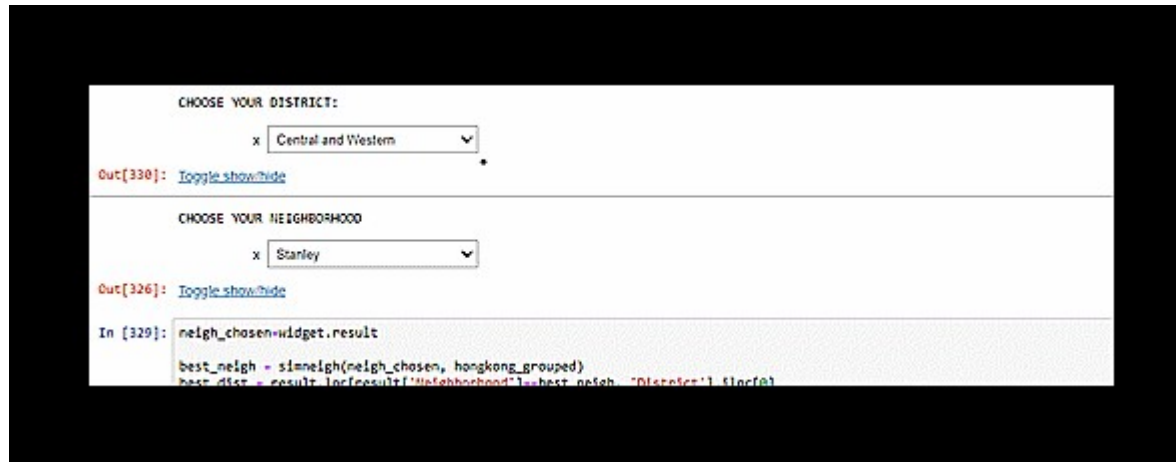


Results



Results (continued)

- Tool to identify most similar neighborhood using Euclidean distance



The screenshot shows a Jupyter Notebook interface with a web application. The application has two sections: "CHOOSE YOUR DISTRICT:" and "CHOOSE YOUR NEIGHBORHOOD". In the first section, a dropdown menu is set to "Central and Western". Below it, the output of a Jupyter cell is shown: "Out[330]: Toggle show/hide". In the second section, a dropdown menu is set to "Stanley". Below it, the output of a Jupyter cell is shown: "Out[326]: Toggle show/hide". At the bottom, a Jupyter cell is shown with the following code:

```
In [329]: neigh_chosen=widget.result
best_neigh = simneigh(neigh_chosen, hongkong_grouped)
best_dist = result.log(result['Neighborhood'] == best_neigh, 'Distance') / len(result)
```

Result (continued)

- Tool to identify cheapest neighborhood in the same cluster



The screenshot shows a Jupyter Notebook cell with the following content:

```
CHOOSE YOUR DISTRICT:
x Central and Western ▼

Out[344]: Toggle show/hide

In [340]: distr_chosen=widget.result
neigh_list = result.loc[result['District']==distr_chosen, ['neighborhood']][['neighborhood']].tolist()
widget = Interactive(my_function, x=neigh_list)
print('CHOOSE YOUR NEIGHBORHOOD')
display(widget)

hide_toggle()
CHOOSE YOUR NEIGHBORHOOD
x Yau Ma Tei ▼

Out[340]: Toggle show/hide
```

The code defines an interactive widget that allows the user to select a district and then a neighborhood. The first dropdown menu is labeled "CHOOSE YOUR DISTRICT:" and has "Central and Western" selected. The second dropdown menu is labeled "CHOOSE YOUR NEIGHBORHOOD" and has "Yau Ma Tei" selected. The code uses the `Interactive` widget from the `interact` library to create these dropdown menus. The `my_function` function is defined in the previous cell and takes the selected district and neighborhood as input to find the cheapest neighborhood in the same cluster.

Discussion

- Cluster overfitting?
- Data quality?
- Other data sources?
- Other K-means optimization methods

Neighborhood	
Cluster Labels	
0	8
1	1
2	40
3	65
4	2
5	2
6	1
7	1
8	1
9	1
10	5

Conclusion

- Useful tools have been created
- Better data is needed

Thank you!

If you have any questions, feel free to send me a message on Github :)