Capstone Project

Recommending Rental Properties

Aron Shirazi August 2020

Contents

1

Introduction

Problem definition and description Case study explaination Data sources

2

Methodology

Neighbourhood scoring Clustering properties

3

Results

Map Clusters Conclusion

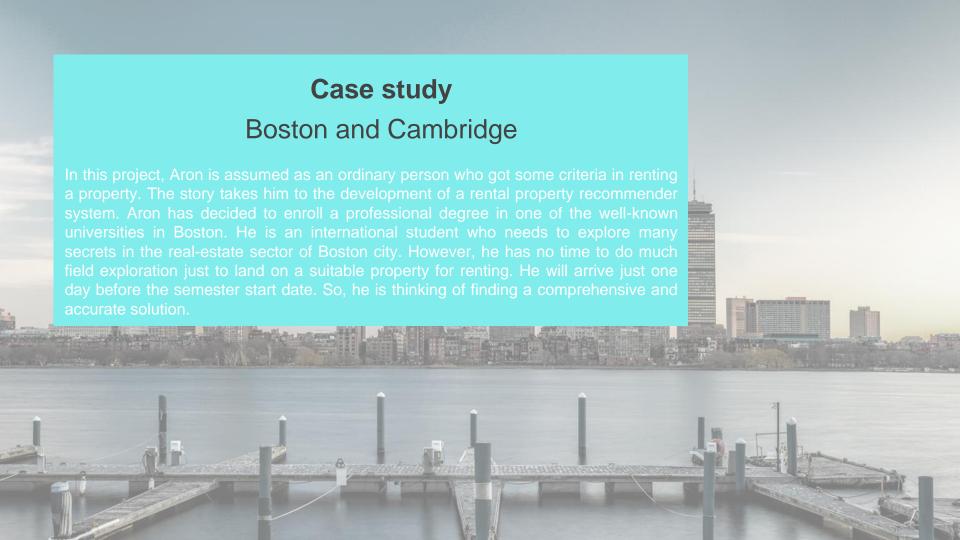
Problem!



- Many people struggle in the time of renting a property
- a great deal of time and money wasted through finding an appropriate property
- Not always ends to a suitable option
- Search engines could evolve the market, but not effective when floods home seekers with thousands of rental cases
- The question is how AI can assist tenants

Description

Having a city name, a recommending system is needed to explore specifications of neighborhoods and search for available rental properties to match them with the given criteria. Main parameters are the distance from a desired location, and important venues which define the appropriateness of each rental option.



Data sources

Neighbourhoods' Coordinates

- List of neighborhoods extracted from public reports
- Coordinates of neighborhoods extracted by Geopy from Nominatim

Foursquare

for retrieving data of venues https://foursquare.com/

Realtor API

for exploring available rental options https://rapidapi.com/blog/be st-real-estate-apis/

Methodology

STEP 1

Downloading the neighbourhoods' reports

STEP 3

Finding the coordinates of neighborhoods

STEP 5

Finding and analyzing venues in the selected neighborhoods

STEP 7

Finding, filtering, and plotting rental properties

STEP 2

Importing, cleaning, and forming the dataset

STEP 4

Filtering neighbourhoods by distance

STEP 6

Developing scoring frame and ranking neighborhoods

STEP 8

Clustering and visualization of results

Neighborhoods Scoring!

Aron has put these scores out of 10 for each

class of venue:

Gym: 10 / 10
Coffee: 8 / 10

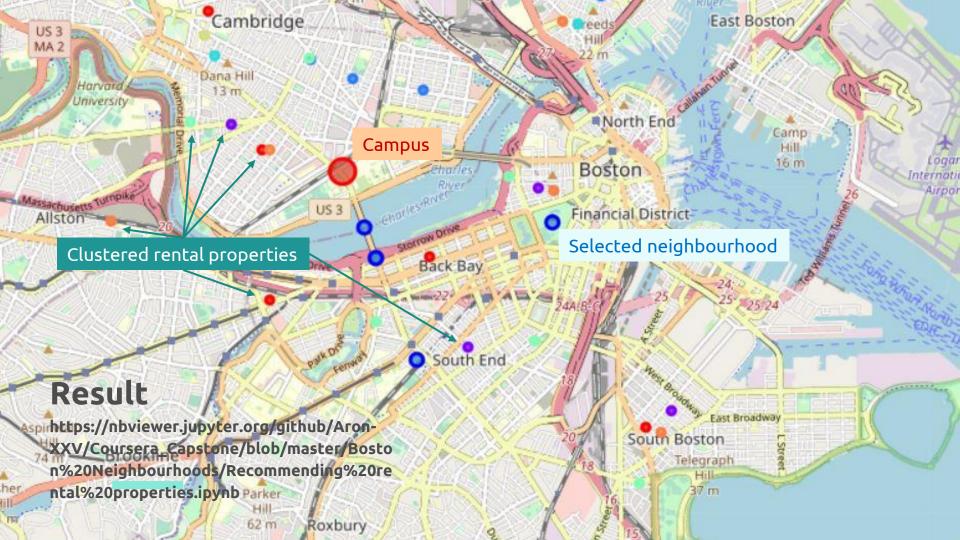
3. Restaurant: 5 / 10

4. Bar: 5 / 10

al Min					10 mm 110
	Restaurant	Bar	Coffee	Gym	Score
neighbourhood					
Cambridgeport	0.320000	0.483871	0.653061	0.847458	2.304390
East Cambridge	0.320000	0.483871	0.653061	0.847458	2.304390
Downtown	0.280000	0.645161	0.489796	0.847458	2.262415
South End	0.440000	0.645161	0.489796	0.677966	2.252923
MIT	0.320000	0.322581	0.489796	0.847458	1.979834
Wellington-Harrington	0.386667	0.645161	0.326531	0.508475	1.866833
Longwood	0.373333	0.322581	0.489796	0.677966	1.863676
Mid-Cambridge	0.453333	0.483871	0.326531	0.508475	1.772209
Riverside	0.266667	0.161290	0.816327	0.508475	1.752758
Strawberry Hill	0.266667	0.161290	0.816327	0.508475	1.752758
West End	0.266667	0.161290	0.816327	0.508475	1.752758
Neighborhood Nine	0.360000	0.161290	0.489796	0.677966	1.689052
Beacon Hill	0.266667	0.000000	0.489796	0.847458	1.603920
Fenway	0.320000	0.161290	0.326531	0.677966	1.485787
Deel Dee	0.000000	0.404000	0.000504	0.500475	4.050000







Conclusion

- Promising results
- Wide range of clustering
- User can easily compare few clusters (7 versus 63) then decide which one he should go for
- A great deal of time and money saved
- User is getting trained through the process
- The project would give better results if there was no API calls limitation
- Future development, online interface

