Building a Recommender System for Similar Holiday Destinations

MORGAN THOMAS

Introduction

- Travel and tourism is vital for the success of many economies around the world
- Many holiday-goers have specific tastes in terms of holiday destinations that they enjoy and do not enjoy
- we aim to build a model to cluster world-cities using cultural, climate, cost and geographical data, in order to build a recommender system to suggest new holiday destinations.
- Audiences:
  - Consumer
  - Travel groups
  - Countries wanting to boost their travel and tourism industry

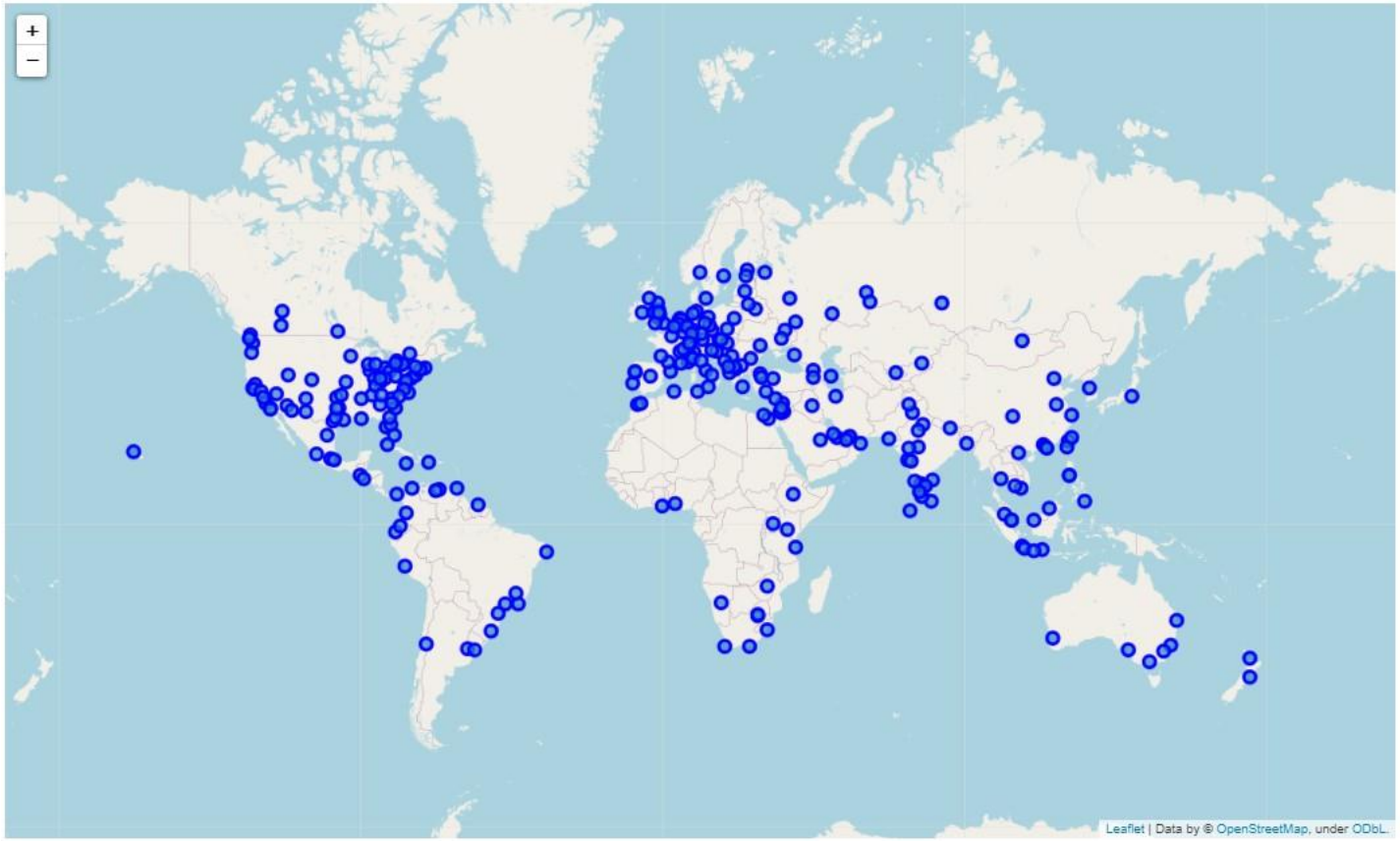
Data

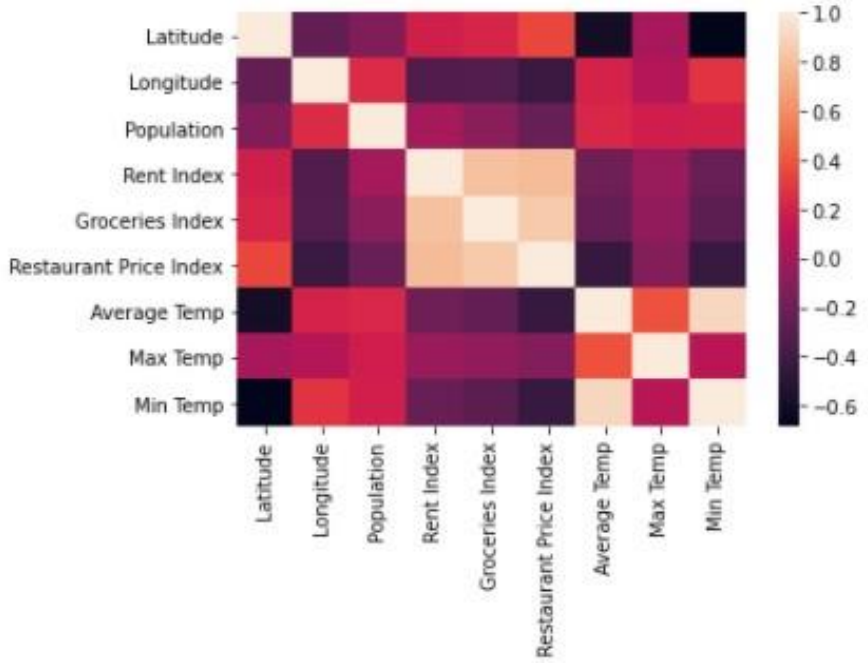
- City data:
  - City name, co-ordinated, population
- Climate data:
  - Average, min and max temperature
- Cost data:
  - Common indexes of cost of living
- Foursqaure data:
  - City venue data

	City	Latitude	Longitude	Population	Rent Index	Groceries Index	Restaurant Price Index	Average Temp	Max Temp	Min Temp
0	Tokyo	35.6897	139.6922	37977000.0	36.41	81.72	54.41	15.748201	32.39	-5.11
1	Jakarta	-6.2146	106.8451	34540000.0	15.80	41.08	24.79	26.945059	29.56	24.32
2	Delhi	28.6600	77.2300	29617000.0	8.73	25.80	25.30	24.774463	40.34	5.92
4	Mumbai	18.9667	72.8333	23355000.0	20.31	26.63	26.36	26.548575	31.46	18.60
5	Manila	14.5958	120.9772	23088000.0	15.80	31.73	24.11	27.631514	32.23	22.12
...	...	...	...	...	...	...	...	...	...	...
3012	Porto	41.1495	-8.6108	237591.0	21.87	39.50	42.58	15.221466	29.28	2.96
3055	Lille	50.6278	3.0583	232787.0	22.98	52.70	66.87	10.797798	30.10	-13.75
3292	Geneva	46.2000	6.1500	201818.0	69.16	112.08	119.78	9.242442	27.48	-16.67
3608	Basel	47.5606	7.5906	177595.0	46.14	120.44	129.10	9.885712	28.22	-16.77
3667	Braga	41.5333	-8.4167	181494.0	12.92	38.22	34.80	14.174088	31.00	-0.40

287 rows × 10 columns

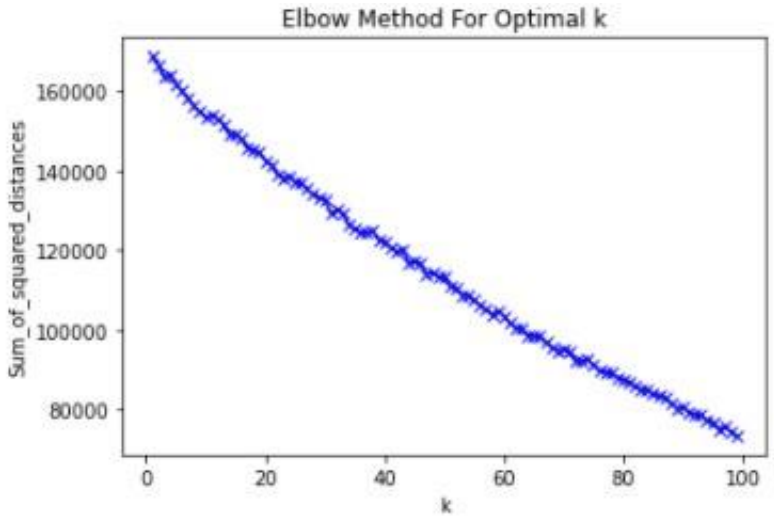
Data


+
−
Leaflet | Data by © OpenStreetMap, under ODbL.

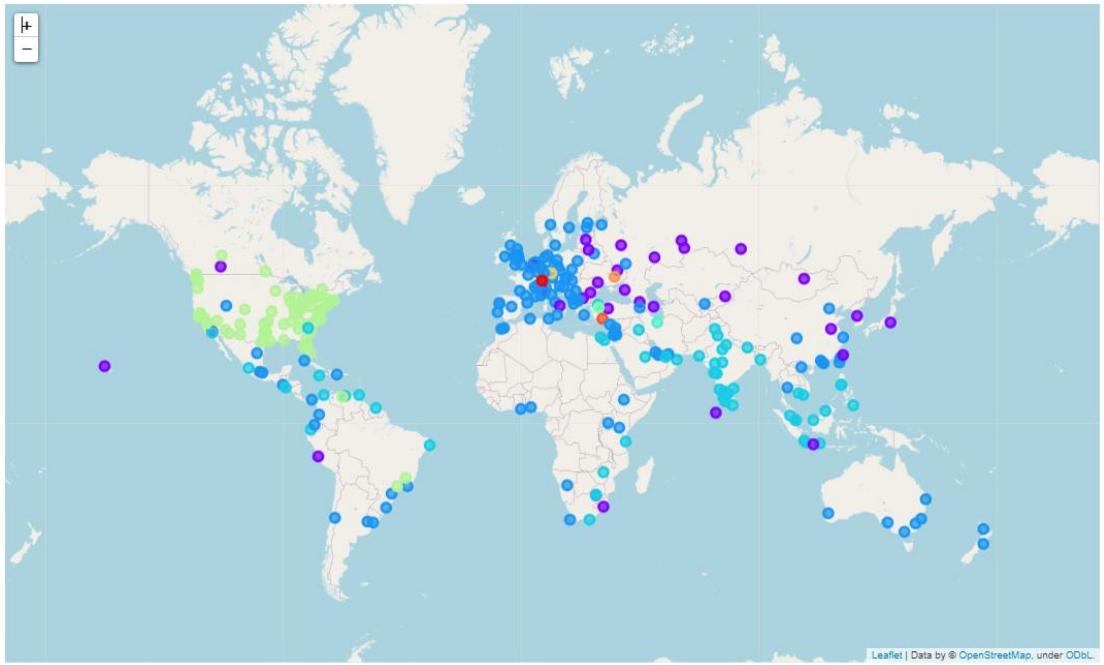

Latitude
Longitude
Population
Rent Index
Groceries Index
Restaurant Price Index
Average Temp
Max Temp
Min Temp
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Methodology

- Data standardized prior to machine learning
- Clustering with K-means
- K-determined via elbow method
- K-nearest neighbours used to determine recommended cities



Results



- Cities coloured according toa ssigned clusters
- K = 11
- Cities clustered geographically without explicit geographical data.

Results

Finding Nearest-Neighbor(s)

```
#finding the most similar datapoint....

def nearest_city(City, n):
    X = np.array(scaled_data)
    knn = NearestNeighbors(n_neighbors=(n+1))
    NearestNeighbors(algorithm='auto', leaf_size=30, n_neighbors=5, p=2, radius=1.0)
    knn.fit(X)
    neighbors = knn.kneighbors(X, return_distance=False)
    index = cities_complete.index[cities['City'] == City].tolist()

    Nearest_cities = []

    for i in neighbors[index]:
        Nearest_cities.append(cities.iloc[i].City)

    print(Nearest_cities)
```

```
nearest_city('Tokyo', 3)
```

```
[0                 Tokyo
547               Leeds
32      Ho Chi Minh City
1553           Stuttgart
Name: City, dtype: object]
```

- Simple function takes in a city (e.g Tokyo) and a number (n) and churns out the n closest related cities
- So, if you fancy a trip to tokyo just go to Leeds.

Conclusion

In conclusion we have built a recommender system for holiday destinations by clustering cities together based on climate, cost, population and venue information. Although there is obvious error in the model its a proof of concept that this may be possible given better and more relevant data.