Identifying the best places to open a French restaurant in Nagoya

Jonhathan Pinon

2019-09-11

Business Approach

- ❑ Japan : French restaurants are popular
- ❑ Nagoya : one of the largest cities in Japan

  - ➢ Nagoya = attractive place to open new French restaurant

▶Where?

Popularity = Income = Interest

Data Description

Data Source

- ❑ **Data source = Foursquare API**
  - ➢ All venues in Nagoya with positions (including French restaurants)

- ❑ Problem : No data about venue popularity

- ❑ Data : Inside python pandas dataframes

Data Description

Data

- ❑ Data collected:
  - French restaurants
  - Food-related restaurants
  - Transport-related venues
  - Long stay venues (Residence, Work, Education related venue)
  - Short stay venues (Shops, Art, Nightlife, Recreation related venues)

- Data cleaning
  - Remove duplicates

Methodology

- Fundamental Hypothesis :
  - Past shop owners chose the best places to open French restaurants
  - French restaurants are situated in areas suited for them
  - Finding areas like areas with French restaurants = finding good places to open new French restaurants

Model

- Quantity to measure similarity = Similarity Distance:

$$similarity\ distance_i \ =$$

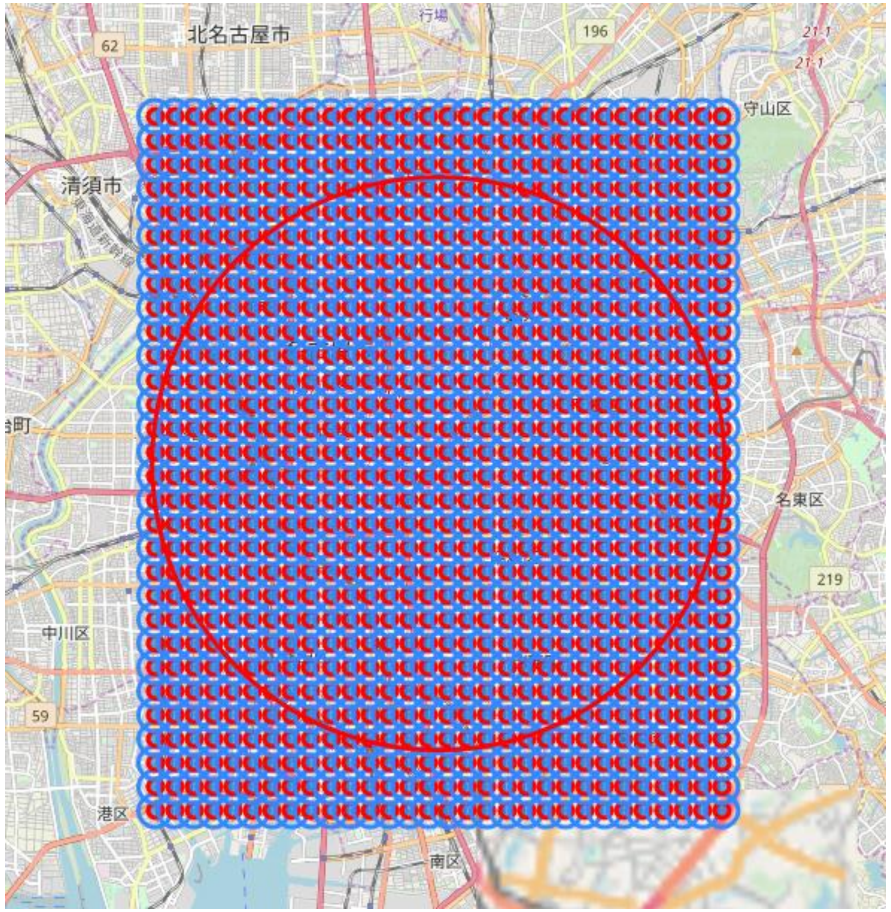
$$\sum_j a_1\,(pfood_{fr,j} - pfood_{a,i})^2 \qquad + a_2\,(ptrans_{fr,j} - pfood_{a,i})^2$$

$$+\ \ a_3\,(plgstay_{fr,j} - plgstay_{a,i})^2 + \ a_4\,(pshstay_{fr,j} - pshstay_{a,i})^2$$

$$+\ \ a_5\,(nvenues_{fr,j} - nvenues_{a,i})^2$$

Grid of Nagoya

- 30 * 30 grid
- One small circle ~ 330 meters
- Big red circle = encompasses all French restaurants (6 kilometers)



Coordinate System

- For more speed without losing accuracy:
  - New local cartesian coordinate system

$$x = R * (lgt - lgt_0) * \cos lat_0$$

$$y = R * (lat - lat_0)$$

- R = Earth’s radius
- lgt = longitude of the point
- lgt_0 = longitude of the center of the system
- lat = latitude of the point
- lat_0 = latitude of the center of the system

F1 Score

- For each area, calculate similarity distance
- Get number of areas with at least one French restaurant = *nb_areas_with_fr*
- Sort the areas by smallest similarity distance
- The *nb_areas_with_fr* areas with the smallest distances are predicted to have at least one French restaurants
- Compare reals and predicted French restaurants
- F1 score

Result

Vector

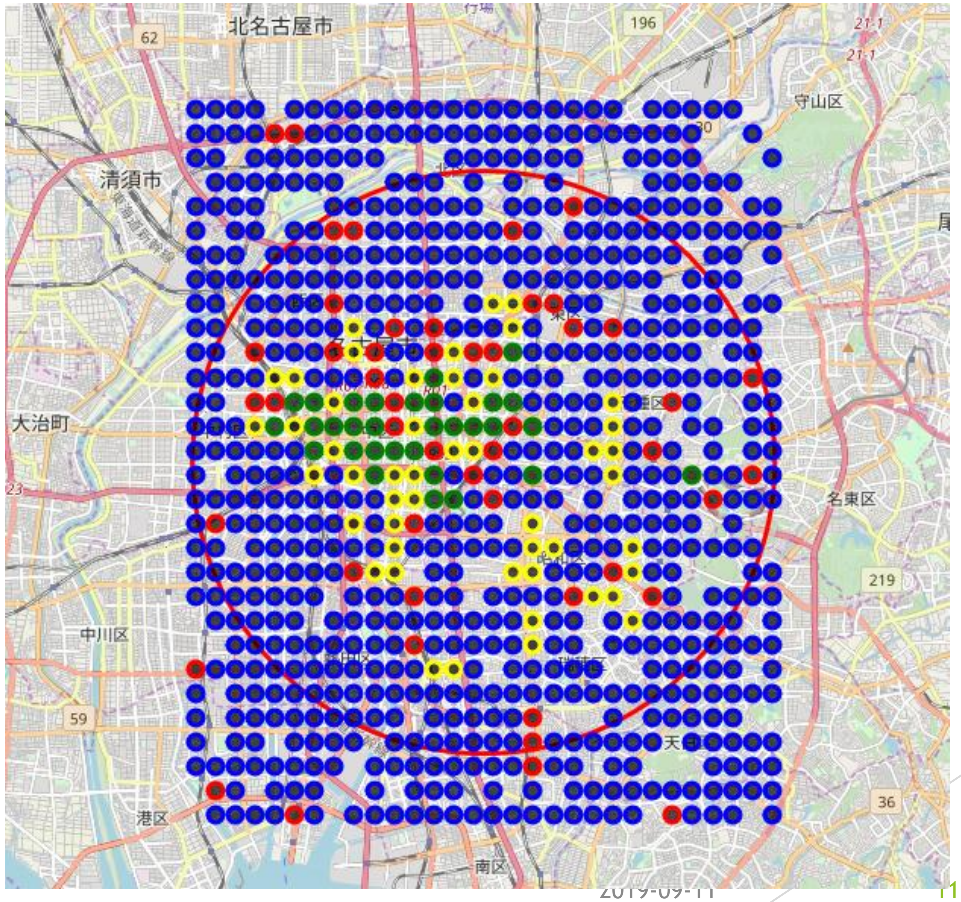
- Optimization algorithm to find best vector (a_1 , a_2, a_3, a_4, a_5)
  - Least square method
  - Broyden-Fletcher-Goldfarb-Shanno (BFGS) (scipy)

- Initial vector : (1, 1, 1, 1, 0.01)
  - Failed to obtain a better vector
  - Similarity distance function too complex (~80*5 + 900*5 independent variables)

Result

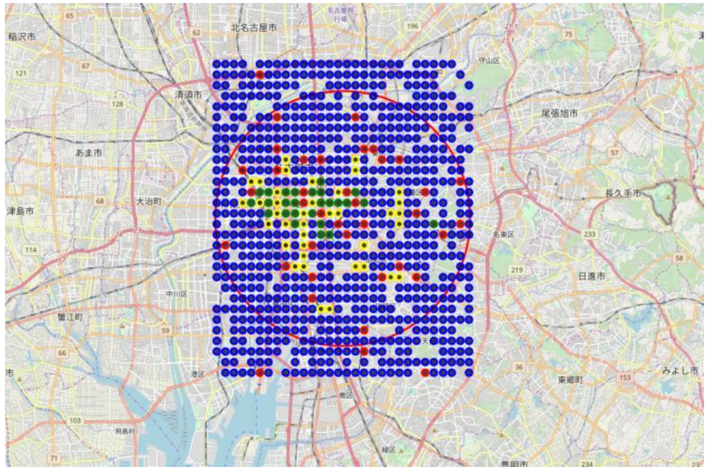
Complete set, zoom

- Green = True Positive
- Red = False Positive
- Yellow = False Negative
- Blue = True Negative

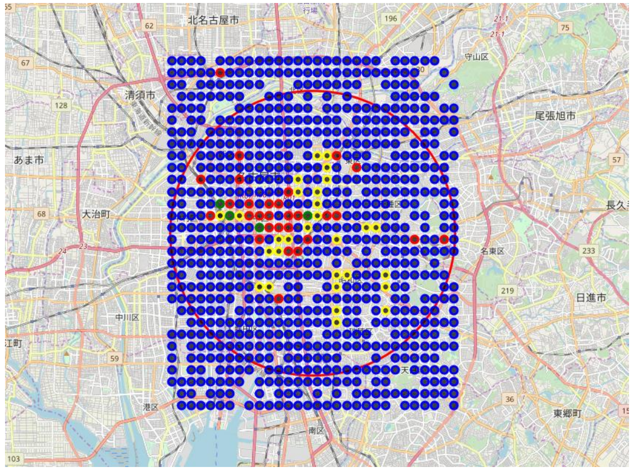


Result

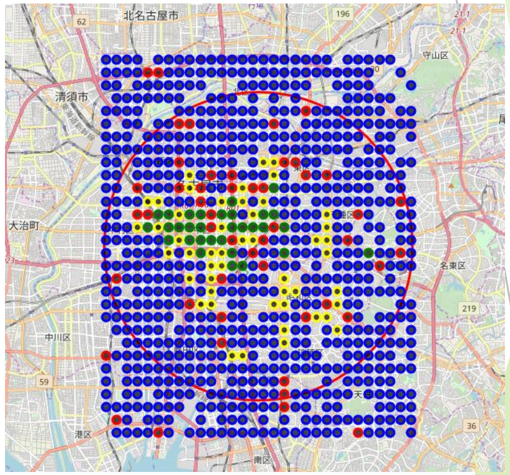
Maps



Training Set
F1 = 0.40625



Test Set
F1 = 0.1212



Complete Set
F1 = 0.3875

Discussion

- true positive + false positive = real number of areas with French restaurant
- true negative + false negative = real number of areas without French restaurant

- F1 score ~ 0.4
  - Model is inaccurate
- Average F1 score of random model ~ 0.1
  - Model still far better than random

- Recommended areas to open French restaurants = areas with consecutive False Positives and without True Positives or False Negatives.

Conclusion

- Best areas for new French restaurants
  - Look for areas like areas with French restaurants
- Data using Foursquare and pandas (python)
  - French restaurant, food, transport, long stay, short stay
- Similarity distance
  - Bad : F1 score = ~0.4
  - But good enough for pedagogic purposes
- Improvements
  - Data about popularity, visits
  - Different radius, different initial vector, model

Thank you for reading this presentation