Capstone Project - The Battle of Neighborhoods

1 Introduction

1.1 Background

For a restaurant business, there are some crucial factors determine whether the business is finally successful or not. For example, if the restaurant sits on a place with abundant guest resource, or if nearby restaurants are already in good reputation, as common sense, it inherits high grade mature clients from the very beginning (restaurant clustering theory); and if its menu offers more attractive foods, it also gives much better chances for a success.

1.2 Business problem

Assuming here we have a new investor, who is planning to open a new restaurant, how to get some early advices to avoid risk? Hence one on-line intelligent estimation system is expected. This system simply takes some easy inputs such as the location of your targeted venue and some of the foods you are planning to offer, then a score will be predicted as an estimation for how successful your new business will be.

It will be a free on-line service opened to the audience who are planning to open a restaurant in a certain venue in New York city. By clicking on the map and input some of your special food, the investor then gets a score as predicted success indicator. And the investor can always simply move mouse to change position or type in new food to get different scores continuously. Keep trying this service, the investor tests different combinations of better address and more attractive foods for their final success.

This prediction system is very useful to help investors because the risk of restaurant investment could be greatly decreased from beginning.

1.3 Problem solving

To implement this system, I will explore Foursquare's geographic location data and pertinent social network data, to train a machine learning model (classification), and finally to run a back-end prediction service.

2 Data exploration

2.1 Feature data definition

To train models, I explored pertinent information to select reasonable features to build the final data set.

First of all, I believe a good restaurant must own plenty of client source, means it can't be too far away from neighborhoods where popularity gathered, so that neighborhood is a good starting point. As the story happened in NYC, I got this location data for free on the web, here is one link to the dataset: https://geo.nyu.edu/catalog/nyu_2451_34572

The dumped neighborhood data looks like this way:

Tab-1: Neighborhood data for NY city.

Borough	Neighborhood	Latitude	Longitude
Bronx	Wakefield	**40.894705**	**-73.847201**
Bronx	Co-op City	**40.874294**	**-73.829939**
Bronx	Eastchester	**40.887556**	**-73.827806**

Here I essentially uploaded a dataset that contains the 5 boroughs and 306 neighborhoods that exist in each borough, and the latitude and longitude coordinates of each neighborhood is the focus data I need for the next step operation.

Restaurant is the focal entity in this case, now that I had all the neighborhoods uploaded, next step I started from pairs of latitude and longitude, to get all the restaurants from New York. These restaurants are representatively our studying goals, because they are close to resident people, in other words, they are close to potential clients.

I then moved to the true target data of our study, namely: the “restaurant”. Through Foursquare *"search"* API, I got all nearby restaurants for each of the above neighborhoods, to construct a full id list of all restaurants who sit near those 306 New York neighborhoods.

Following restful API will download all the venue IDs whose category is Restaurant or Coffee:

https://api.foursquare.com/v2/venues/search?&query=Restaurant,Coffee

The raw data set of restaurants in NY city looks like this:

Tab2: restaurant scanned through neighborhoods in NY city:

v_id	v_name	v_dist	v_cat
4db03c875da32cf2df4509f4	Big Daddy's Caribbean Taste Restaurant	1008	Caribbean Restaurant
4c66e0068e9120a15929d964	Kaieteur Restaurant & Bakery	1011	Caribbean Restaurant
508af256e4b0578944c87392	Cooler Runnings Jamaican Restaurant Inc	479	Caribbean Restaurant
4be5f0eacf200f47d1fa133c	McDonald's	904	Fast Food Restaurant

Here I have the basic restaurant information listed including:

- v_id: the venue id of the restaurant, I use this ID to approach more information around the restaurant
- v_name: the name of the restaurant
- c_dist: the distance from neighborhood to nearby restaurant
- v_cat: the restaurant's major category

On this table, the restaurant ID is the key to catch more detailed restaurant information in next step. The distance from neighborhood to restaurant could also be useful, as I believe, the closer a restaurant to neighborhood, the more convenient for residents to visit the restaurant.

However, the above dataset is not what I finally want, because for each restaurant, there must be multiple nearby neighborhoods, so there must be duplicated lines for a single restaurant, I need to transpose it and merge same restaurants by "*groupby*" operation. Then I got one additional transposed feature: the average distance from a restaurant to all its nearby neighborhoods.

Here is the new data set:

Tab 3. Basic restaurant data:

v_name	v_cat	avg_dist_2_neighborhood	cnt_near_neighborhood	restaurant_id
Boca Loca Restaurant	Latin American Restaurant	465	1	4feba1c9e4b0367e6123bf4c
Acapella Gourmet Pizza & Restaurant	Pizza Place	469	1	55906dbb498e4edbe4888785
Sorrento's Restaurant & Pizza	Italian Restaurant	1108.5	2	52004756498ed9485d3eabad
Tony & Cyndi's Pizzeria & Restaurants	Pizza Place	933.3333333	3	4f21c904e4b0831873dc6256

From above data sample, you can see two new columns named as *"avg_dist_2_neighborhood"* and *"cnt_near_neighborhood"*. The useless column *"v_dist"* has been dropped off, and those duplicated restaurants were also grouped together in a single line according as the restaurant ID.

From now on, I started to use this table as the base to extend more features, what I then added up including:

- Latitude and Longitude of the restaurant
- Average rating of clustered restaurants where this restaurant is the centroid
- How many recommended popular venues nearby this restaurant
- Rating, it is the target/label (binary-classification or regression) of the restaurant
- The menu items (food) offered from this restaurant

To get these basic restaurant information, such as latitude, longitude and rating, I need to invoke the Foursquare API “venue”:

`https://api.foursquare.com/v2/venues/{ restaurant id }`

For the recommended popular venues nearby this restaurant, I need to invoke Foursquare API “explore” by the restaurant ID centroid with the restaurant’s latitude& longitude as parameters, and the query condition as “food, drinks, coffee, shops, arts, outdoors, sights, trending”:

`https://api.foursquare.com/v2/venues/explore/ll{restaurant Latitude, Longitude}`

and based on the recommend venue ID, one further step to cascade the venue search again to get all of their ratings to count the average rating, it is almost the same invocation as above, but with the recommended venue ID this time:

https://api.foursquare.com/v2/venues/{ recommend venue id }

For the menu items (food) offered from this restaurant, I need to invoke the Foursquare API “menu”:

https://api.foursquare.com/v2/venues/{restaurant id}/menu

After all the iterative (or even cascaded) Foursquare APIs invocations, I got the feature list looks like this :

Tab 4:advanced restaurant data set

v_name	v_cat	avg_dist_2_neighborhood	cnt_near_neighborhood	restaurant_id	avg_rate	nearby_rec	rating	menu
Cooler Runnings Jamaican Restaurant Inc	Caribbean Restaurant	479.0	1	508af256e4b0578944c87392	6.416667	10	6.5	NaN
McDonald's	Fast Food Restaurant	904.0	1	4be5f0eacf200f47d1fa133c	6.400000	13	6.5	Big Mac??|Cheeseburger|Double Cheeseburger|Ham...
241 St Cafe & Restaurant	American Restaurant	1019.0	1	4c010e75cf3aa593825eccb0	6.400000	12	6.6	NaN
Ripe Kitchen & Bar	Caribbean Restaurant	798.0	1	4d375ce799fe8eec99fd2355	6.700000	14	8.7	Cuban Plantain Boat|Jerk Chicken Quesadilla|St...

Since I got all the raw information about the restaurant, it is almost ready to fit for the machine learning models, let's summarize a dictionary:

Tab 5: Restaurant data set explanation

attribute Name	Data description	Data Type	Potential Contribution
v_name	Restaurant name	String	n/a
v_cat	Category	Foursquare Category	certain Category maybe special popular, more easily catch eyes, will be encoded
avg_dist_2_neighborhood	Average distance from this restaurant to all its nearby neighborhoods	float	the lower means distance closer to potential clients
cnt_near_neighborhood	how many nearby neighborhoods are close to this restaurant, for example with 1 km	int	the more means more potential clients
avg_rate	Average rating of nearby popular sites(such as food, drinks, coffee, shops, arts, outdoors, etc.)	float	the higher, the more possibility for stable client source
nearby_rec	the total number of popular sites recommended from Foursquare which centriod by the restaurant	int	the higher, the more possibility for stable client source
menu	special food offered	list of food items	will do clustering , turn it to group(clustering) label before being used for classification model
rating	**the restaurant rating (how good the restaurant is)**	**float/classification**	**the label or target value(y)**

2.2 Data preprocessing

I dumped almost all the information from Foursquare, but the data quality is not stable. I met some issues which I have to handle before taking about feed these data to train the machine learning models.

First, Foursquare is a global portal support multiple access devices, somehow, some inputs were not stored correctly, so I met some UTF-8 encoding error when I parse the json response from Foursquare, I have to skip these lines, but lucky, the data loss not big, just about 10+ rows.

Second, a few restaurants exist without a rating, this maybe either data bug, or nobody cares it at all, so I simply remove them by leveraging the dataframe's "dropna" method.

Third, quite a few restaurants exist without menu data, this is truly an issue which I feel hard to handle, I will touch more in the discussion segment. I think it will impact the accuracy finally, but I have to leave it as is at this moment by leveraging the dataframe's "fillna" method to set a default value.

2.3 Feature selection

From above table 5, I selected following features as input of the model training:

1. Category
2. Average distance to neighborhoods
3. Number of nearby neighborhoods
4. Average nearby rating
5. Recommended nearby popular sites
6. Menu item

- Label or target variable: Rating (meanwhile also the output for prediction)

Current menus dataset is a string list, it needs further processing for quantification, this is a lightweight NLP text processing to label texts into digital groups by leveraging clustering algorithm, I will describe more in the next segment.

3 Methodology

In above segments, I have got the restaurant data, and made it almost ready for model training, then I stepped further to prepare the training data set.

3.1 Exploratory Data Analysis

For this scenario, the score prediction created from the input information turns out to be the goal of the proposal. Looking on the data set I got in above segment, obviously the *"Rating"* attribute is the target variable (label).

That means, the predicted value will be the potential "rating" base on the fact that to open a new restaurant on the location you specified and through the foods you are going to offer.

However, there are two types of models, regression and classification, so I still need to make a decision, shall I make the model to give a prediction as simply "bad or good" ? Or make a regression to predict the expected "rating" as a float value?

Now take a look on the rating column by invoking *pandas.DataFrame.describe()* on sampling data:

mean	**6.939024**
std	0.913203
min	5.200000
25%	6.200000
50%	6.700000
75%	7.750000
max	8.800000

Looks like the rating value is high precision float and the std deviation is not significant. That means, it needs relative more training data if adopting a regression prediction model which it very hard for my free developer Foursquare account. Thus I decided to simplify the problem by turning it to a binary classification. In that way, I need to label the restaurant data set by this

rule: take the 75% value 7.75 as a threshold, if rating larger than 7.75, then label it as “Good”(1); or else label it as “not good”(0).

Then here I got the label segment like:

restaurant_id	lat	lng	avg_rate	nearby_rec	rating	menu	label
508af256e4b0578944c87392	40.898276	-73.850381	4.530947	11.0	6.5	NaN	0
4be5f0eacf200f47d1fa133c	40.902645	-73.849485	7.401281	11.0	6.5	Big Mac®|Cheeseburger|Double Cheeseburger|Hamb...	0
4c010e75cf3aa593825eccb0	40.903573	-73.850228	7.411729	15.0	6.6	NaN	0
4d375ce799fe8eec99fd2355	40.898152	-73.838875	8.553371	4.0	8.7	Cuban Plantain Boat|Jerk Chicken Quesadilla|St...	1

3.2 Transform text attribute by leveraging clustering

Moving forward, look on the menu attribute. My gut feeling, the restaurant's menu definitely influences the rating. But how? It's hard to use it on classification model directly, so I need to vectorize it. Text clustering is one possible way allocate the menu content into a certain category per its similarity to other menus, so restaurants with similar menu may compete on the same court, thus they have more chances to have similar rating.

First I need to clean the messy strings by:1)remove punctuations 2) remove stop words 3) tokenize it and at last make tokenize the string into "words". These are typical NLP steps to clean up the text data, so I leveraged NLTK to do some of the data preprocessing. After the text processing, I got a list of BOW(bag of words) of the menus. I still need to transform the word lists into array of figures.

TF-IDF is a good choice, by scikit-learn lib, I turned the menu BOW into a quantified TF-IDF matrix (numpy array). I then invoked scikit-learn Kmeans model, clustered TF-IDF matrix(menus array) into groups labels for each the menu, it then looks like:

restaurant_id	lat	lng	avg_rate	nearby_rec	rating	menu	label	menu_group
4b0578944c87392	40.898276	-73.850381	4.530947	11.0	6.5	NoneUpload	0	1
ıcf200f47d1fa133c	40.902645	-73.849485	7.401281	11.0	6.5	Big Mac®|Cheeseburger|Double Cheeseburger|Hamb...	0	6
cf3aa593825eccb0	40.903573	-73.850228	7.411729	15.0	6.6	NoneUpload	0	1

3.3 Transform dummy variables

The last step is to work on the dummy variable (menu group and category). I used Panda's method 'get_dummies' to assign one-hot code to them. The original “Category” and “Menu Group” attributes were transformed to a set of 0/1 codes looks like:

ant	Sandwich Place	Seafood Restaurant	Spanish Restaurant	Sushi Restaurant	Thai Restaurant	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0

3.4 Classification model and result

The classification models of this proposal was much straightforward, with the location and food input features, I need to predict whether these choices combine together, will bring up one successful (1) or failed (0) business.

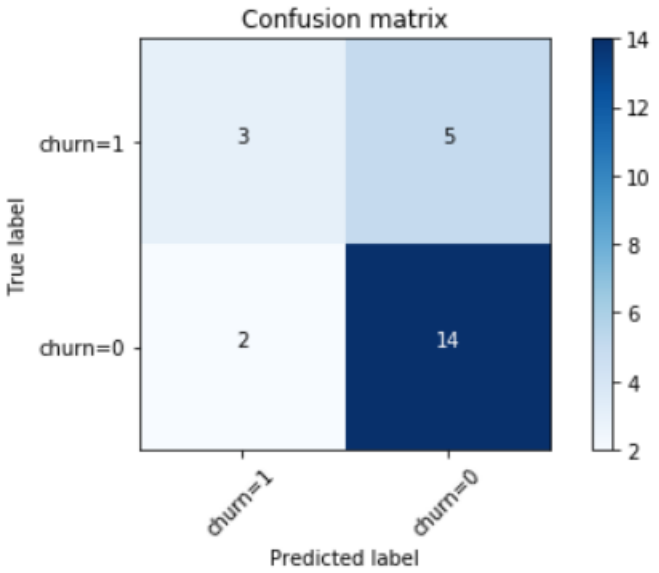
```
LR = LogisticRegression(C=0.1, solver='liblinear',class_weight={1:0.65,0:0.35}).fit(X_train,y_train)
```

Considering the bad/good ratio is not unbalanced, after identifying the class weight, the accuracy lifted.

I had a quick look on the validation data per *jaccard index*, the indicator is about 71%. Comparing with some other classification model such as SVM and decision tree, none of them higher than the LR model.

3.5 Evaluation

Now from a comprehensive angle, look at accuracy of classifier through confusion matrix. I created a confusion matrix here:



[[3 5]
 [2 14]]

The classifier correctly predicted 14 of 16 as 0, so, it has done a good job in predicting the higher risk of *negative* result. Meanwhile 5 of 8 good rating prediction incorrectly to risky, this is bit high, Now that avoiding risk is our major goal, it is acceptable.

4 Conclusion

In this study, I explored data of Neighborhood, restaurant, menu, and rating through invoking Foursquare API; to generated training and testing data, I selected features and did reconstruction and transformation for feature encoding and text clustering; at last I implemented Logistic Regression algorithm to predict the success or fail possibility of a new restaurant business in NYC.

The accuracy and recall rate of the algorithm implementation is acceptable considering the training data limited volume.

5 Discussion for future

The implementation of this story mainly focused on venue, and extended from venue, menu is attribute is also leveraged. However, there are still other entities such as the *users* who give the rating, and *tips* about the comments to the restaurant very possible also contribute to the prediction.

So in future I have more chances to improve the proposal by horizontal extension through leveraging more pertinent features.

Another pain is, the quality of Foursquare is not ideal. I got 5000+ restaurant data items from Foursquare in the beginning, but after exploration, only small part of restaurants (<30%) have a rating, and only small part of restaurants (<20%) are attached with menu. And more serious, the daily quota of Foursquare API access for free developer is very limited, that led to the amount of training data is pretty small. If I have enough time to accumulate more high quality data, the model training result could be much better.