Gyms Data Analysis

Group 6

1 Introduction and Description

Keeping fit is becoming increasingly popular, so as its related business – Gymnasium. As a popular website to recommend business, Yelp and its rating about a business influence consumers' choice. In order to give some recommendations for gyms' owners to improve, we use Yelp datasets that contains reviews text and other attributes of the each gym to generate recommendations. After data cleaning, we constructed a random forest model to select aspects which mostly effect the stars of a gym. Then, we assign the weight of these aspects according to attributes of users and reviewing time. Finally, we give suggestions to gyms' owners in two aspects: First, overall suggestion to all gyms according to the whole dataset analysis. Second, the specific suggestion to a specific gym (and its owner) according to its reviews. The final suggestion would be given in our Shiny APP.

2 Data Preprocessing

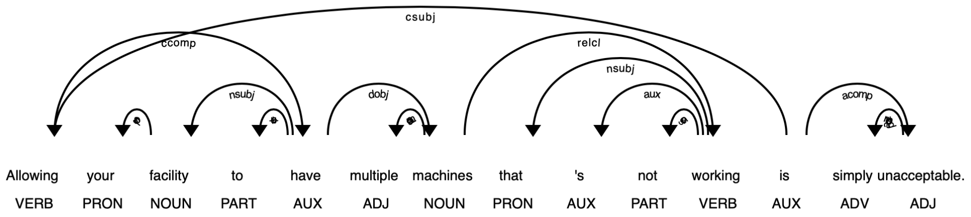
First we combined our data and arrange them to make them simple and clear. In this step we merge "review", "business" and "user" sets with review_id, business_id and user_id, leaving "tips" not used. Then we deal with text and non-text features.

2.1 Text Processing

2.1.1 Generate aspects for improvement based on words frequency

Firstly, we try to abstract some aspects which may make a difference to review stars from review texts and provide recommendations according to how these aspects are mentioned in the texts. For the sentence "**Needed to wash my hands, bathroom was not only gross there was no soap and no paper towels toilet paper anything.**", we want to abstract **bathroom** and **towels** as aspects to improve. According to words frequency of the whole review texts, we pick up 206 words such as staff, contract, treadmill, and locker as improving aspects.

Some words refer to the same aspect, for example, equipment, machine, and facility describe similar aspect, so we merge them into one. The number of aspects we abstracted is 121, which includes some words like class, distance, counter, staff, membership, price, sign, parking, kid, cardio, locker, pool, wait, open, massage, contract, yoga, maintenance, towel, floor, train, woman, space, card, 24/7, contact.



dependency tree

2.1.2 Extract describing word using dependency parser

Dependency parser in sPacy Dependency is a kind of analysis for a sentence in linguistics, which returns a tree to reflect the structure of a sentence. The following tree draw with Spacy is an example to show the relationship between words in a sentence. To be more specific, "**is**" is connected to "**allowing**" and "**unacceptable**", which means that they are directly related. Then, "**allowing**" is connected to "**have**" and "**have**" is connected to "**facility**".

Rules to extract describing words According to the tree structure, we extract adjective verb and adverb which are most closely connected to the aspect we consider. Therefore, we extract "**unacceptable**" to describe "**facility**". Actually, only considering the distance between words is an easier way to extract describing word. However, there is probably no dependency between them. Also, far distance may dismiss the important describing word, like the sentence mentioned above.

2.1.3 Sentiment scores for describing words

To make the describing word into scores, our group use sentiment dictionary which contains 8000 words to determine the exact score to use. Positive words are considered as 3. Negative words are considered as -3, and neutral words are considered as 0.5. We multiply scores if one aspect has multiple describing words.

2.2 Non-Text Processing

There are a few types of non-text features. We deleted useless features (e.g. latitude and longitude). For some similar features we summed them up and merged them to a new feature (e.g. review's useful/cool/funny -> new_useful). Complex features are transformed to a suitable form (e.g. "friends" were transformed from a list to friend number). Finally we extracted the features in "attribute" and transformed them properly (e.g. GoodForKids T/F-> 0/1 feature).

After data preprocessing, we have four types of features: Information, record the information of review, business or user; Variables, which are used as variables in our model; Weights, which are used as weights in our model; Star (rating), as our model's response.

3 Modeling

3.1 Motivation

After merging reviews, there're still a lot of features that may related to the business ratings. Thus, it's very naturally to consider feature selection. Among various methods, random forest is picked as it's powerful and efficient based on experience.

3.2 Input and output

To train a model, all features are treated as independent variables and review ratings which take integer from 1 to 5 are treated as dependent variables. Features and ratings together are the in-put of the model. As for the output, the most important outcomes are the feature importance sequence. Typically, the higher feature importance a certain feature is of, the more it contributes to distinguish ratings.

3.3 Model selection

As random forest gets a lot of parameters to tune, model selection is also necessary. While doing cross validation to perform model selection, it's very interesting that Both the out of bag scores (a rule to evaluate model ability of genelization) and the feature importance sequence don't change significantly while the balanced accuracy (a way to evaluate model perfermance) gets higher and higher. Thus, it's a wise choice to build an simpler but with lower out of bag scores random forest model to improve efficiency. The final random forest model comes with 100 trees. Each tree is built from "Gini" index and at most 12 features (as there're 146 features in total). The maximum depth allowed is 100, and minimum samples to conduct another split is 5, the minimum samples of each leaf is 2.

3.4 Model result

Due to page limits, only top 20 most important features and their feature importance are shown below.

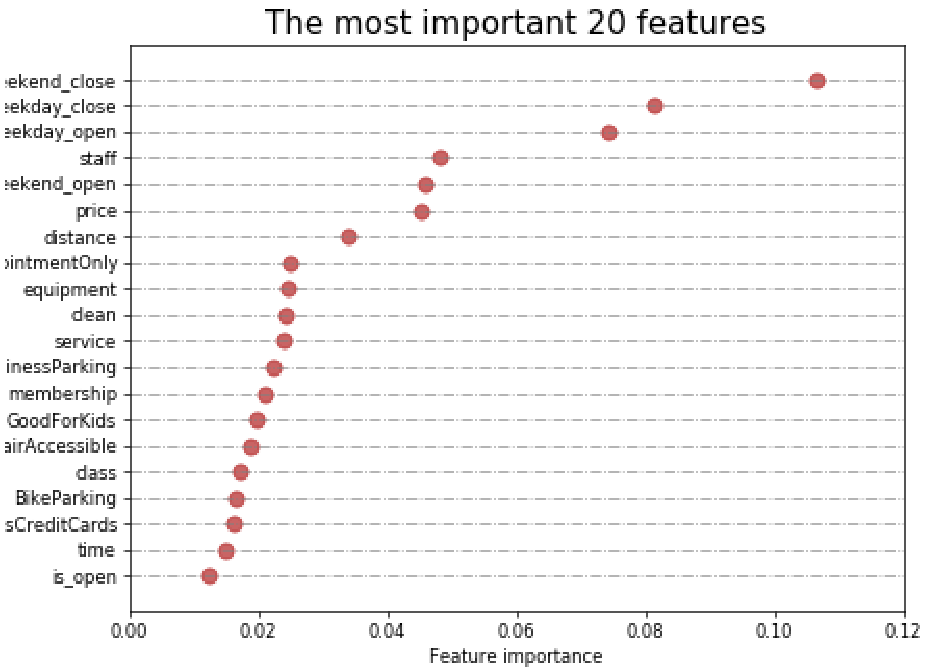
3.5 Model evaluation

Out of bag score is computed to evaluation the model ability of genelization. For this model, the out of bag score is about 56%, which is much better than random guess (20%).

Then, after the first-round of model fitting, another round of model fitting with top 100 most important features is conducted to test if the "most important" features are robust. It turned out that, they're very robust so the first-round result is trustworthy. Finally, the top 50 most important features are picked to serve as the aspects that gyms can improve.

4 Result Suggestions and Findings

According to our model, we give overall suggestions to all gyms and specific suggestions to an individual gym. Specific suggestions are based on the gym's text and non-text variables.



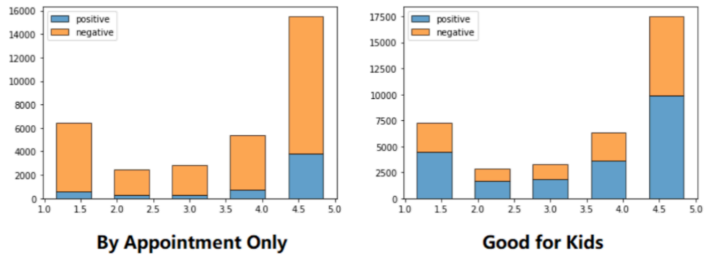
feature_importance

4.1 General Suggestions

According to the model's result, there are some general suggestions for all gym's owners.

For non-text part, the variables usually contain only few levels, so our suggestion direction is obvious. For example, the attribute: **"By Appointment Only"** and **"Good for Kids"** influence the ratings of a gym. If a gym's "By Appointment Only" is True(Positive), then their ratings is likely to be higher than those with False(negative). On the contrary, if a gym's "Good for Kids" is True, their ratings are likely to be lower. And their correlation coefficient between rating (1-5) and attribute positive proportion (True/(True+False)) is apparent. For "By Appointment Only", its positive value proportion is [0.086, 0.106, 0.111, 0.135, 0.243] for rating 1-5, so the correlation coefficient is 0.872. For "Good for Kids", positive value proportion is [0.698, 0.678, 0.672, 0.664, 0.637], the correlation value is -0.879. We make some reasonable guess for these results. The reason why appointment only helps improving ratings is that, customers will be satisfied if gyms provide appointment service such as trainers or some classes. While people are doing exercise, kids sometimes trouble them, then permission for kids might cause some trouble to other customers and they complain about it.

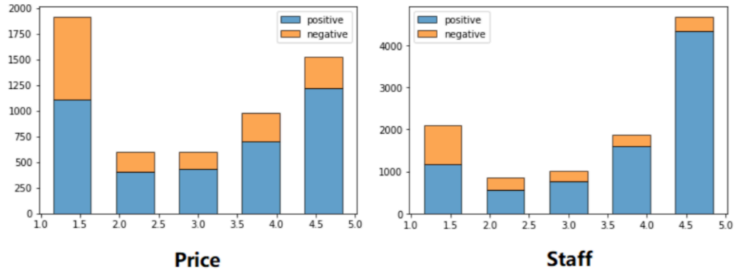
For text part, the information hidden in sentences have been extracted. What people mostly concern about are **"staff"**, **"open hours"** and **"price"**. It is proved that they are also highly related with ratings. For example, variable "staff"'s positive proportion has a correlation coefficient 0.998 with ratings, for "price" the coefficient is 0.954. According to these words, their related sentiment words and corresponding sentiment scores, we made such crucial suggestions. First, they should prolong their opening time, because it is shown that those open longer tend to have higher scores. Second, gym's owners should make their gym's fee more reasonable and affordable. Third, we recommend them enhance their staff's professional skills, so that staffs could serve customers



By Appointment Only

Good for Kids

Proportion of Positive or Negative (Non-Text)



Price

Staff

Proportion of Positive and Negative (Text)

better.

4.2 Suggestion for specific business

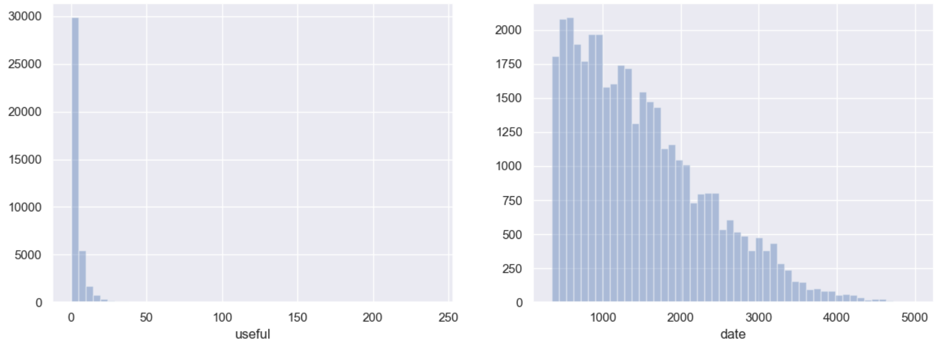
4.2.1 Assign weight for each review

For each review, we use seven feature to construct weight. They are useful and date of the review, review count, number of friends, fans and compliment of the review user. From the distribution of features, we use log to get the weight. We use $f(x) = log(x+1) + 1$ and sum them to get the final weight score. For data feature, since it's the number of days from when the review was written, it's weight should be different from other features: $f(x) = log(\max(x)/(x+1)) + 1$.

4.2.2 Extract important features

For specific business, we calculate the weighted sum of sentiment score for each feature. As for text features, we select the first three features with the smallest weighted sum. As for nontext features, we select first three features which the business does not have according to the feature importance of the random forest. Finally, we assign each extracted feature with corresponding suggestions.

It is necessary to mention our suggestion on gyms location. If the chosen gym has some reviews complaining about its location, we will recommend it to move or start a branch at another location. The suggested location is the disctrict that have many high ratings gyms in the same city. We use postal code to divide districts in a city, so our location related suggestion looks like "Customers sometimes complain about your gym's location. You could open branches at or move



weighted_features

your gym to some popular district such as places near postal code M4K 2P7.” (It is for LA Fitness, 1970 Eglinton Avenue E,Toronto)

5 Shiny APP

https://chrisqian.shinyapps.io/gym_review_recommand/

6 Contribution

Chen Qian: Web-based App and Cleaning Nontext Data.
Haoxiang Wei: Adjust Weight of Reviews and Generate Suggestions.
Chunyuan Jin: Review Text Cleaning with sPacy and Generate Sentiment Scores.
Hao Pan: Establishing Random Forest to Extracting Feature Importance.