2.Related Works

Recently, the Internet is everywhere. Yelp is one of a service which publish crowd-sourced reviews about local businesses, as well as the online reservation service. We use the yelp data to analyze the popularity of different restaurant. According the data set, the visualization of some user and business behaviors in [1]. And the prediction of a restaurant’ Star and review are in [2] and [3]. With rating, how the online customer review affects restaurant demand is in [4].

To the best of our knowledge, none of the previous works addressed how to construct a model that based on data mining to predict how to open a popular restaurant and choose the restaurant style in different areas.

However, the existing works in the literature do not consider the comprehensive factors how to affect the popularity of restaurants. In addition, none of the works in the literature considers the problem of regional differences.

References:

[1] Zhang T, Pan Y. Yelp Challenge Project Report[J]. University of Washington Publication, 2014.

[2] Fan M, Khademi M. Predicting a business star in yelp from its reviews text alone[J]. arXiv preprint arXiv:1401.0864, 2014.

[3] Asghar N. Yelp Dataset Challenge: Review Rating Prediction[J]. arXiv preprint arXiv:1605.05362, 2016.

[4] Luca M. Reviews, reputation, and revenue: The case of Yelp. com[J]. 2016.

6.Algorithm Design

To train our prediction models, we use five different algorithms.

6.1 Decision Tree

There are many attributes in our data set. Based on these attributes, we can use the following pseudocode to express our ideas:

Method: Generate\_decision\_tree (samples, attribute\_list)

(1) Create node N;

(2) If training set are in the same class C then (class label attribute values ​​are C the candidate attribute value is not considered)

(3) Return N as a leaf node, marked with label C;

(4) If attribut\_list is empty then

(5) Return N as a leaf node, marked as the most common category of samples; (class label attribute value of the largest)

(6) Select attribute\_list with the highest information gain attribute best\_attribute; (to find the best partitioning attributes)

(7) Mark the node N as best\_attribute;

(8) For unknown value in each best\_attribute a i // divide the sample samples by best\_attribute

(9) Grows a branch of the best\_attribute = a i from node N;

(10) Assume si be the set of samples of best\_attribute = a i in samples; (a partition)

(11) If si is empty then

(12) Plus a leaf, marked as the most common class in samples; (from the sample to find the largest number of class labels, as the node of the mark)

(13) Else plus a node returned by Generate\_decision\_tree (si, attribute\_list-best\_attribute);

(14) Recursive call to the data subset si, where the selected attribute has been removed best\_attribute

According this algorithm, we could generate a decision tree to classify the objects in our data set.

6.2 Rule-based Classifier

We have 2 methods to define a rule: direct method and indirect method. According the 6.1, we could generate the rules, it is called indirect method. For accuracy, we use the direct method that we can verify the decision tree and make our model more reliable. There are 5 steps to generate rules base on the direct method. Firstly, we should grow a single rule. And then Remove Instances from rule. Thirdly, if necessary, prune the rule. After that we should add rule to Current Rule Set. Repeat the above steps. By this method, we could generate a Rule Set by our attributes from yelp.

6.3 KNN Algorithm

In our project, every location point is such as (x, y). To classify the restaurants in different locations, we choose KNN algorithm. Basically, some restaurants are popular because of the location. We use the following pseudocode to express our ideas:

Method: kNN (A [n], k)

# Enter: A [n] is the coordinate of N training samples in space, k is the number of neighbors

# Output: x belongs to the category

(X, A [i]), i = 1, 2, ..., k is taken as the initial distance between x [1] and A [k] as x,

Sort by d (x, A [i]) in ascending order;

The maximum distance of the sample is D = max {d (x, a [j]) | j = 1,2, ..., k};

# Continue to calculate the Euclidean distance of the remaining n-k data

for (i = k + 1; i <= n; i ++)

       Calculate the distance d (x, A [i]) between a [i] and x;

       if (d (x, A [i])) <D

                use A [i] instead of the farthest sample

                # Will be followed by the calculation of the data can be inserted directly

The probability of the category of the first k samples A [i], i = 1,2, .., k is calculated by the order of the K data, and then the statistics of the K samples are calculated. The category is the class of sample x.

6.4 Naive Bayes Classification

A Naive Bayes classifier is based on the Naive Bayes assumption to model the joint probability P(r,s) for any feature vector r and star rating s. Then, given a new feature vector r∗ for a new review r∗, the joint probability function is computed for all values of s, and the s value corresponding to the highest probability is output as the final class label for review r∗.

We use the Naive Bayes classification to help us calculate the conditional probability. The conditional probability could make a more objective analysis.

6.5 Support Vector Machines (SVM)

According our high-dimensional data, we use the SVM. It is rooted in statistical learning theory and works very well in this scenario. The SVM also uses the maximal margin hyperplane to linearly separate the data objects of different classes and represents the decision boundary by support vectors derived from a subset of training examples. In the future work, we will deeply study this algorithm.

7.Development tools and Environment

We use several development languages and tools in our project.

The development languages include: Java, Python and Matlab. We also use Git for better teamwork. In addition, we select some powerful libraries in different languages and we will not mention here.