2.Related Work:

As we all know, the Yelp has a very big dataset. It is also a valuable resource to predict a restaurant’s star ratings and make prediction. Actually, Bochkov and Gingerich have already conducted similar analysis on restaurants based on text analysis and word vectors [1]. In another paper, Yun, Wu, and Wang explored using Part-of-Speech analysis to predict a restaurant’ rating based on user-generated text alone. Therefore, while the accuracy of these sentiment based predictions are high, they may not necessarily provide recommendation to a beginner who wants to open a restaurant. In our project, we focus on not only modeling a restaurant’s popularity through different attributes analysis, but recommending which type of restaurant is proper in different cities.

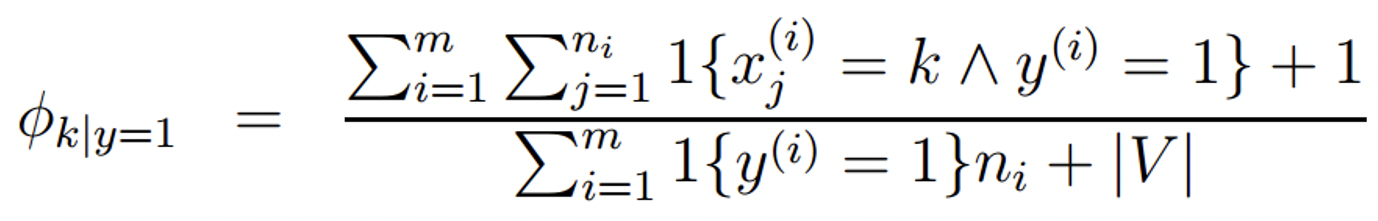
[1] Gingerich, Travis, and Yevhen Bochkov. ”Predicting Business Ratings on Yelp.” Stanford University. 2015.

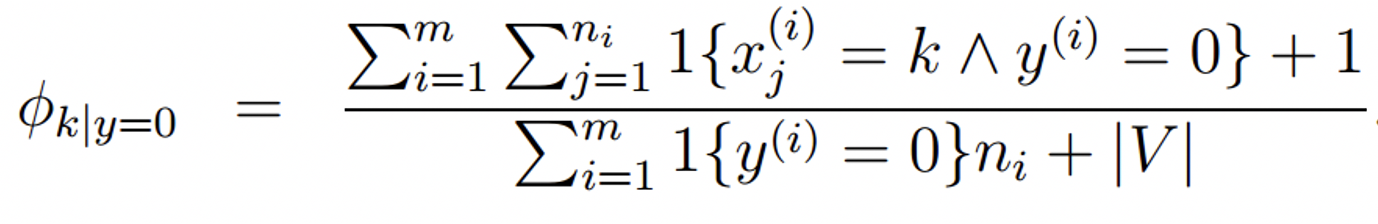
6.Algorithm Design:

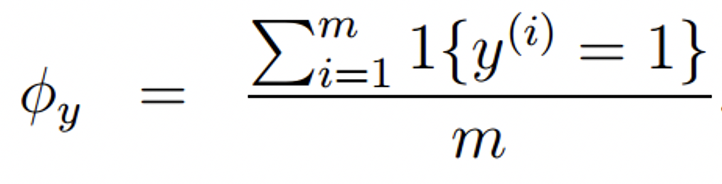
We use 2 modes of restaurant’ classification that based on star ratings: binary and multiclass. In binary mode, a restaurant which with a star rating below 4.0 is classified as 0, and the restaurant with a star rating above 4.0 is classified as 1. The algorithm we used to train and predict the data are Naive Bayes, support vector machine (SVM), decision tree and kNN. In multiclass mode, the restaurants are classified from 0-5 based on the integer value their star rating. The algorithm selection is our future works.

6.1. Naive Bayes

Our project uses the Multinomial Naive Bayes classifier with a Laplace smoothing value of α = 1.0. Using all of the countries data, the test accuracy was 0.5517. The results for each separate city are included in the results section of our paper. The test accuracy ranges from 0.4725 for the State College to 0.5846 for Pittsburgh. The equations used are included below:







For logistic regression, the test accuracies varied depending on the regularization. We used logistic regression by scikit, where the inverse of regularization strength is represented by parameter C. Improving or reducing C by factors of 10 does not yield significant differences in test accuracy, but we noticed that a higher C-value, results in higher test accuracy. Finally, our binary test accuracy as shown in the table below and for the multi-class, we will continue study in the future.

|  |  |
| --- | --- |
| C | Binary Test Accuracy |
| 0.01 | 0.5398 |
| 0.1 | 0.5485 |
| 1.0 | 0.5503 |
| 10.0 | 0.5505 |
| 100.0 | 0.5506 |

6.2. Decision tree:

Up to now, our team just finished the decision trees of both binary classification. Basically, the decision tree is simple and convenient to accurate model. The simplest version of a decision tree we refer to the [7]. Benefits of decision trees include logarithmic growth in the cost of predicting data as the number of data points increase, and statistical validation of its reliability. Although decision trees are very susceptible to minor changes in the data and tend to overfitting, we use it in order to predict the data. For binary classification, the decision tree test accuracy is 0.5514. For the multi-class classification, we will study in the future work.

6.3. 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.

6.4. 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;

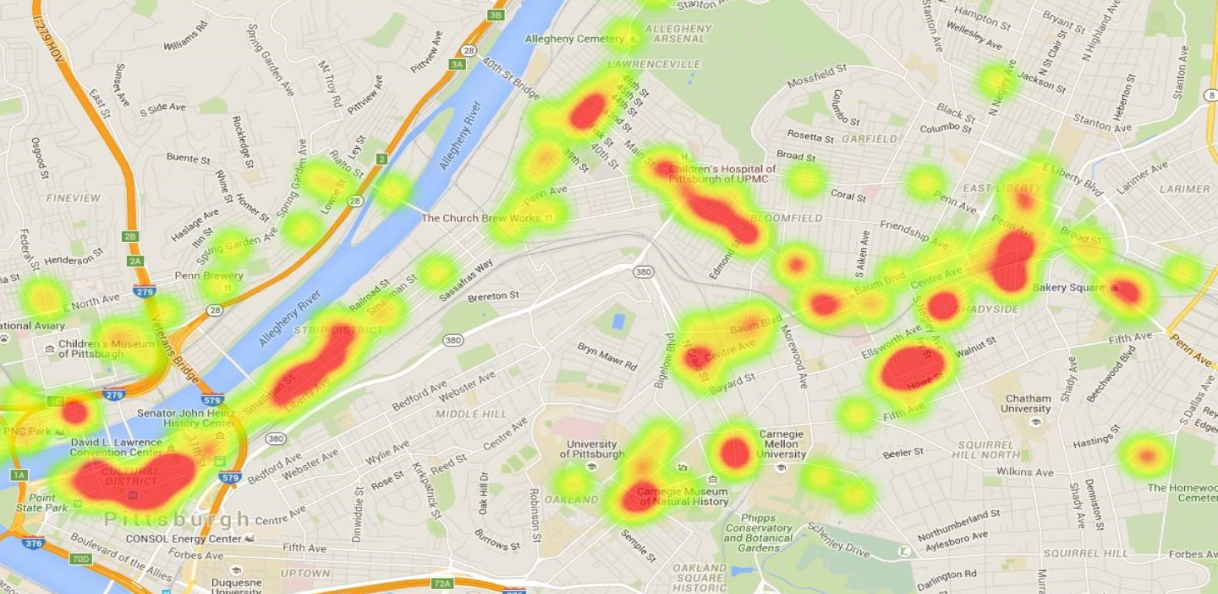
       if (d (x, A [i])) <D

                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.

By this algorithm, we can calculate the distances between any two points and draw a heat-map as below:



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.