Cooked To Location

Chenlu Ji, Jiayi Wang, Sam Park
Department of Statistic, Department of Computer Science
Columbia University
cj2452@columbia.edu, jw3316@columbia.edu , shp2135@columbia.edu

***Abstract*— In this project, we aim to do three things. Firstly, predict the location of a restaurant type based off the description of the restaurant, which would yield success of that restaurant. Secondly, based off the location, predict the type of restaurant that would yield to the success of the restaurant. And lastly, based off the location and restaurant type, predict the yelp rating. For recommendation, we utilized SQLite for database and queries of different conditions. For rating classification, Naïve Bayes, Decision Tree and Tree Ensemble models are implemented.**

Keywords-component; resturant; Yelp; location; prediction; recommender

I. INTRODUCTION

The restaurant business collectively is quite massive. In 2016, there has been $782.7 billion in restaurant industry sales. In addition, there are over 1 million restaurant locations in the United States, with 14.4 million restaurant industry employees. All in all, the restaurant workforce make up 10% of the overall United States workforce. [1]

Although the restaurant business jointly may seem to be successful in the amount of sales, many restaurant business fails. In fact, according to a study by Ohio State University on failed restaurants, 60% do not make it past the first year, and 80% will fail within five years. [2]

This is where the motivation behind our application comes in. The success of a restaurant has many factors to it. But one factor that we wanted to focus on was location. Location is arguably one of the key components of the restaurants success. According to a study done by Cornell University on why restaurants fail, one of the key important parts is location. It relates to the market, demographics, possible business competitors, as well as partners. In addition, while a restaurant can benefit from close proximity to competition and often located in clusters to attract more traffic, it can also fail, due to not being able to differentiate itself in that cluster. [4]

With the advent of Yelp, and other rating systems, we can arguably quantify "success" by its rating from customers worldwide. A study done by a member of Harvard Business School, found that a one-star increase in Yelp ratings led to a 5-9 percent increase in revenue. In addition, a consumers' response to a restaurant's average rating was affected by the number of reviews and whether the reviewers were certified as "elite" by Yelp. [5]

In order to combat some of the failure issues restaurant owners face, we do three things to help address these issues.

Firstly, a restaurant owner can have a specific restaurant type in mind, for example a café, deli, or an Italian Restaurant, but doesn't know where to open the restaurant. In this instance, the restaurant owner would input the proper features that best describe the restaurant, and our application would output a set of locations that would be best for that restaurant description.

This application can also work in reverse. Suppose, you have a location in mind, but don't know which restaurant type would be ideal. In this scenario, the user would input the location, and the application would output the ideal restaurant type.

And lastly, given that fact that Yelp ratings have a considerable factor a customer's decision as to whether they should eat there or not, we also incorporated a prediction feature, where the application can predict what Yelp rating a restaurant would attain with a certain set of features. (location, restaurant type, etc.)

II. RELATED WORKS

With the information, rich dataset provided by Yelp, it made it possible for predicting a restaurant's star rating and success as well as possible location. Some similar analysis done was by Kong, Nguyen, and Xu where they predicted international restaurant success with Yelp. They have set to identify key features people in different counties looked for in their dining experience. They have performed feature selection to identify the business attributes that correspond to high star ratings for each country. Afterwards they classified the data using models such as Naive Bayes, support vector machines (SVM), decision trees, logistic regression, and Gaussian Discriminant Analysis (GDA) to evaluate the strength of the feature sets they have selected. Their focus was not only modeling a restaurant's success through textual analysis of user reviews, but analyzing which features are better predictors of success among

restaurants in different countries to provide data driven predictions of international trends. [3]

III. System Overview

A. Dataset Used

For the dataset, we mainly use the over 2 GB's Yelp challenge dataset, which contains 2.7M reviews and 649K tips by 687K users for 86K businesses. For each row in businesses, there are 566K business attributes, such as hours, parking availability and ambience. Social network of 687K users for a total of 4.2M social edges is also included, aggregating check-ins over time for each of the 86K businesses and 200,000 pictures from the included businesses.

All the datasets are in JSON template, requiring us to transfer to CSV and then to SQLite database. Since we are dealing with "Big Data", we spend plenty of time finding out the fastest and most efficient way to do calculation regarding recommendation system. To achieve this, we decided to create a SQLite database where two separate tables with different functionality were structured: one for location and cuisine recommendation, and another for review analysis and rating prediction. In this way, we make the decision process faster when user hits either recommendation engine or prediction system.

B. System Design

We implement several different models and methods on rating prediction, and integrate them with Flask in order to achieve our ultimate goal of providing wise investment information for those users (Figure 1). For the recommendation engine, the methods used are mainly filters, sort and queries. Here the SQLite database participate as the most important part, locating the best place or cuisine type based on certain user input. For rating classification, Naïve Bayes, Decision Tree and Tree Ensemble models are implemented. Regarding the "Big Data" purpose, we ran all three models using Spark, which also integrated with HDFS. Since there exists multiple functions for our project and the goal is to serve for investors with different demands, we choose Flask, a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine, to connect all the functions and make every piece works properly. PySpark and Hadoop environment also works well with Flask, making it easier for users to run all the function he/she wants. Except for the backend side, Flask is friendly interactive with HTML and JavaScript as well. We believe the final product for a good project would not only involve complex calculation with fancy model, but also something easy and efficient for target customers to use with friendly user interface. By using Ajax, we made our application beautiful and easy to manipulate.

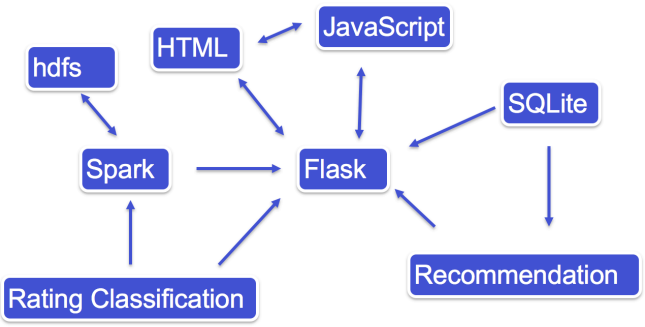

Figure 1. System Design

IV. Algorithm

A. Recommendation System

As is mentioned in previous part, our recommendation system aims at two goals: given cuisine type, recommend restaurant location and given location, recommend cuisine type. The realizing of the two goals depends on a SQLite database and queries of different conditions.

1. Filter the database with cuisine type (or restaurant location) and save the returned results.
2. Calculate average ratings within each cuisine type (or ZIP Code).
3. Sort the ratings in descending order.
4. Recommend the top 5 locations (or cuisine types) with highest average ratings.

B. Classification System – Data Manipulation

The classification model depends on another set of data features including "*Delivery*", "*Happy Hour*", "*Waiter Service*", "*Wi-Fi*", "*Price Range*", "*Cuisine Type*", "*Rating*", "*ZIP Code*" and "*State*". Types of the feature values are all categorical. For features that only have one value for each observation, we used dummy variables to vectorize it. Other features were vectorized through Bag of Words (BoW). BoW is a common technique for Natural Language Processing (NLP). For each feature mentioned above, we extracted all the values from our dataset. BoW is constructed as a set of unique values for each feature. The vectorized feature is then a sequence of binary values. The size of each feature equals to the size of the corresponding BoW. Here we illustrate the algorithm of BoW using a small subset of cuisine type feature.

The feature values before vectorization are

Restaurant id	Cuisine Type
1	Fast Food
2	Bars; American
3	Steakhouse

The feature values after vectorization are

id	Fast Food	Bars	American	Steakhouse
1	1	0	0	0
2	0	1	1	0
3	0	0	0	1

C. *Classification System – Naïve Bayesian Model*

Naïve Bayesian Classifiers are conditional probabilistic classifiers based on Bayes' theorem. Given an observation vector of n features, the probability of the response variable falling into class k is represented as

$$p_k = p(C_k|x_1, \dots, x_n).$$

Bayes' theorem decomposes the above probability as the product of prior and likelihood divided by evidence

$$p(C_k|\boldsymbol{x}) = \frac{p(C_k)p(\boldsymbol{x}|C_k)}{p(\boldsymbol{x})}.$$

The denominator $p(\boldsymbol{x})$ depends only on $\boldsymbol{x}$, so it's a scaling constant for a given $\boldsymbol{x}$. Thus we can ignore the denominator when comparing different p_k.

The likelihood $p(\boldsymbol{x}|C_k)$ and the priors $p(C_k)$ follow the empirical distribution of the training data.

To classify, we compare each p_k for all k. The assignment is the k with highest p_k.

$$\hat{y} = \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}}\, p(C_k)p(\boldsymbol{x}|C_k).$$

V. Software Package Description

Since our project means to serve for those investors who have the thoughts to open a restaurant, we decide to implement our recommendation engine and classification methods by integrating them into a web application. The thought behind was to let the investor choose location where he/she finds interested and suitable for running a restaurant. Simply by entering Zip code or clicking on the map, the recommended cuisine type and other related information would show up. Or by telling the app with certain cuisine type, it tells which location is most profitable. After the investors decide where to start business and what type of food should be chosen, the application also provide rating prediction for restaurants with certain features (cuisine type, price range, delivery, Wi-Fi, happy hour and waiter service).

This application serves as a smart restaurant investor, providing crucial information about location and cuisine type, which not only enabled users to choose the right place and food type but also give rating prediction for certain type of restaurant based on yelp dataset.

The app, entitled "Cooked to Location", is designed to help make easy plan for investor who is not familiar with the catering industry. Figure 2 is the introduction screen to app. We mainly divide the app into three parts: location recommendation, cuisine recommendation and rating prediction.

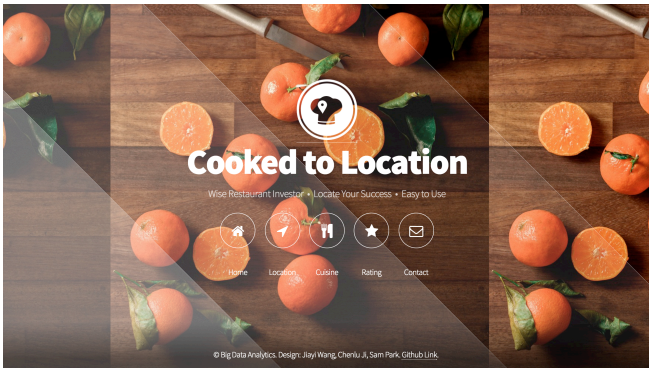

Figure 2. Homepage screen

For the location recommendation page, choosing the cuisine type the app offered, and type in the Zip code if you want to find the best place for restaurant within certain location range. Or it is also fine if you leave the Zip code box blank, the app will just search for the result nationwide. Figure 3 is the location recommendation shown for Chinese restaurant nationwide. The red markers shown on the map are the five recommended locations for opening a Chinese restaurant. Here the app also implements the Google Map API, making the visualization of locations straightforward and easy to read.

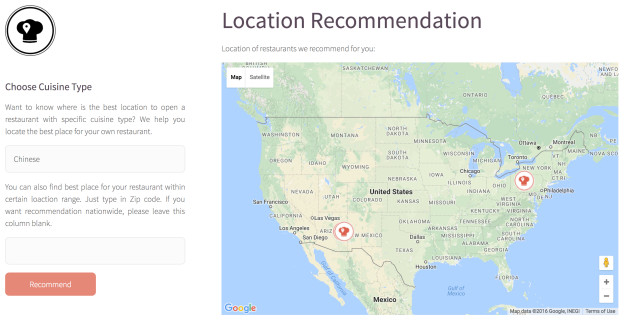

Figure 3. Location Recommendation Screen (1)

Figure 4 shows the detailed information related to the Chinese restaurants recommended. The left bar shows the most popular Chinese restaurants in the United States.

Business address and yelp rating are also provided. The corresponding location information and recommended level are listed in the middle. The application also gives some detailed statistics about the data.

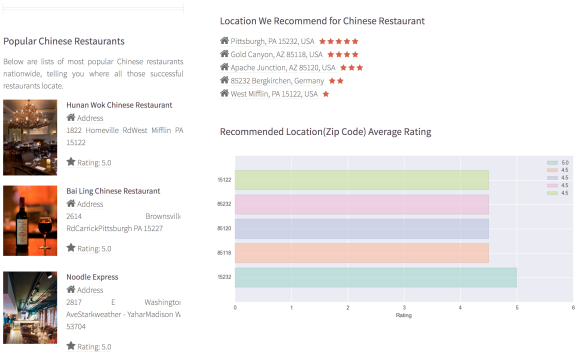

Figure 4. Location Recommendation Screen (2)

For the cuisine recommendation page (Figure 5), by typing in preferred location for opening a restaurant (zip code, coordinates, or simply click location on the map), the app offers the best choice of cuisine type and lead to the best chance to success.

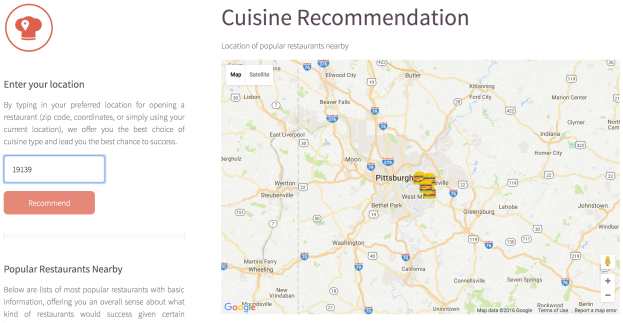

Figure 5. Cuisine Recommendation Screen (1)

In the Figure 6, the left side bar tells the most popular restaurants within the certain range of location typed in by users in the input box. The recommended cuisine types are listed in the middle.

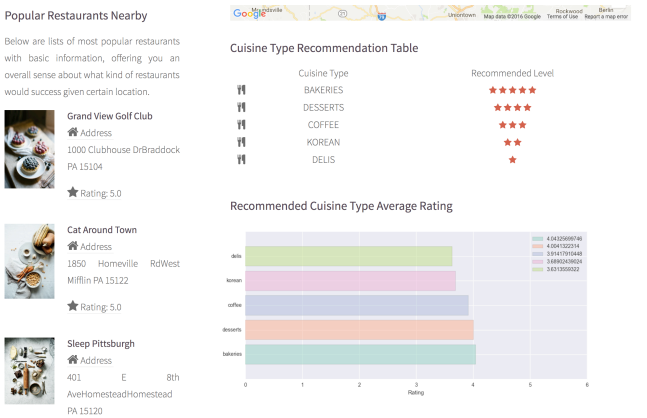

Figure 6. Cuisine Recommendation Screen (2)

For the rating prediction page (Figure 7), simply by choosing the cuisine type, price range, delivery, Wi-Fi, happy hour, waiter service, state and Zip code, the app can predict the rating based on above features. On the right side, the user restaurant interactive network is shown. The network gives a knowledge graph about the user reviews and restaurants.

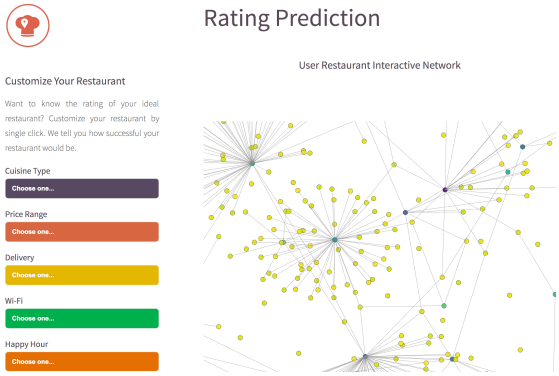

Figure 7. Rating Prediction Screen (1)

Figure 8 shows the predicted rating based on Naïve Bayes classification. The corresponding visualization helps better understand the features user selected.

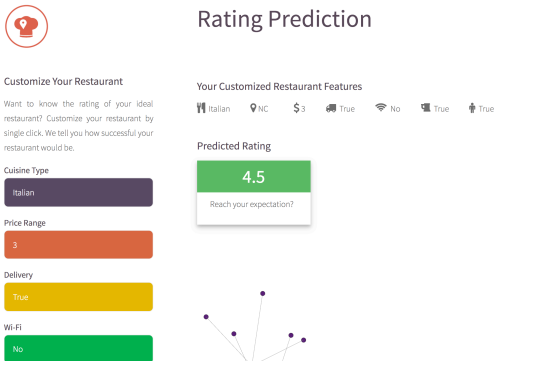

Figure 8. Rating Prediction Screen (2)

VI. EXPERIMENT RESULTS

A. *Recommendation Performance*

Since the Recommendation System is based on SQLite queries, there's no numerical way to evaluate the model. The performance depends strongly on the dataset composition, which is a limitation of our model. However, some experimental trials on the recommendation functions show reasonable results.

For example, the result of searching for Mexican food is shown in Figure 9.

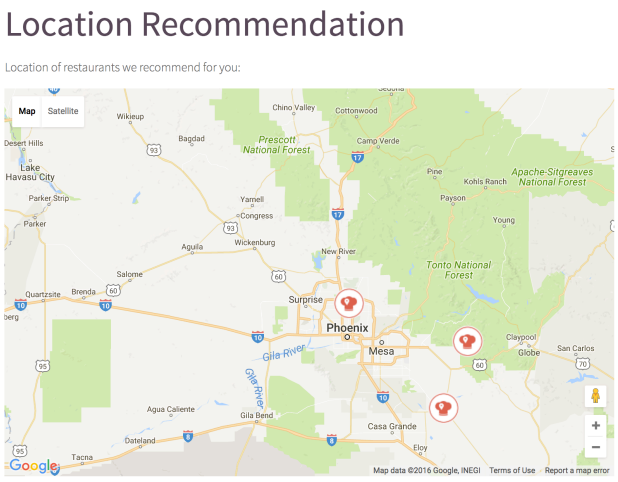

Figure 9 Location recommendation for Mexican food

Among the five locations we recommend, three are located around Phoenix, which is a large settlements for Mexican American.

Besides, the top 5 popular Mexican restaurants nationwide recommended by our app includes one in Phoenix, as is shown in Figure 10. It's consistent with the location recommendations.

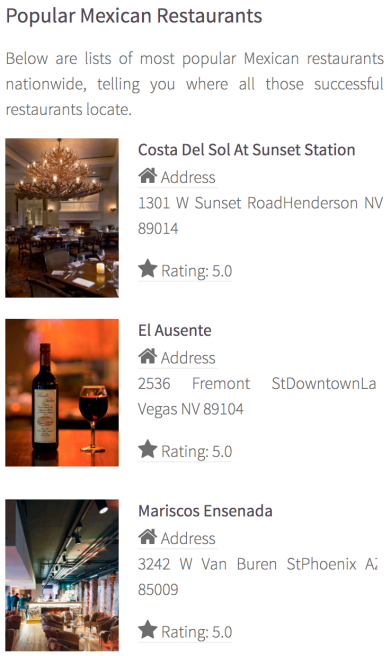

Figure 10 Popular Mexican restaurants nationwide

B. *Classification Performance*

Because the classes are ordered ratings, the normal classification accuracy criterion does not apply in our model. We use the Mean Squared Error (MSE) for evaluation of classification results.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

We tested 5-fold Cross Validation (CV) for the whole dataset. Three multiple-class classification models provided by PySpark:

1. Naïve Bayesian Model in pyspark.mllib.classification package
2. Decision Tree Model in pyspark.ml.classification package
3. Tree Ensemble Model in pyspark.ml.classification package

For each model, we ran three times of 5-fold CV. The comparison of different classifiers' performance is shown in Figure 11. It is clear that Decision Tree Model is the worst classifier and Naïve Bayesian is the most stable one with smallest average MSE.

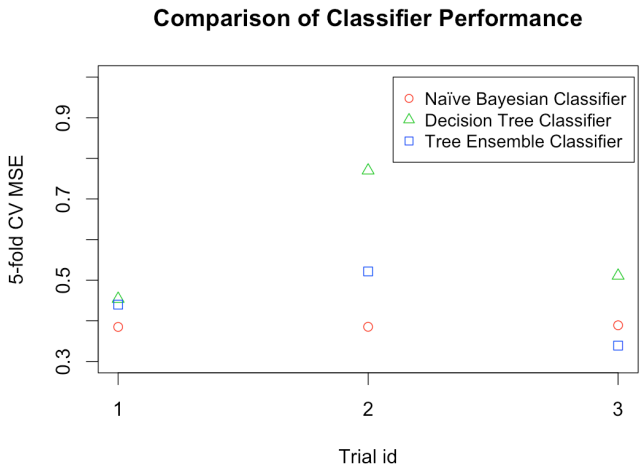

Figure 11 Comparison of Classifier Performance

VII. Conclusion

Based on outputs presented by our application, one improvements can be more data. From the demonstration, input location, output restaurant type, we have found that because we do not have enough data, we cannot recommend a restaurant type for that exact location, but rather a location nearby.

Possible future iterations of this project can be factoring other features that lead to the success of a restaurant. For example quality of food, entrepreneurial aptitude by the owner, surrounding changing culture, and seasonal

Acknowledgment

THE AUTHORS WOULD LIKE TO THANK TO PROFESSOR CHING-YUNG LIN FOR HIS DEDICATION IN TEACHING OF BIG DATA ANALYTICS COURSE IN FALL 2016. WE WOULD ALSO THANK TO TAS FOR THEIR HARDWORKING AND HELPFUL ASSISTANCE.

Appendix

References

[1] "Facts at a Glance." National Restaurant Association. N.p., n.d. Web. 22 Dec. 2016

[2] Feloni, Richard. "Food Network Chef Robert Irvine Shares The Top 5 Reasons Restaurants Fail." Business Insider. Business Insider, 2014. Web. 22 Dec. 2016. http://www.businessinsider.com/why-restaurants-fail-so-often-2014-2

[3] Kong, Angela, Vivian Nguyen, and Catherina Xu. "Predicting International Restaurant Success with Yelp." Stanford Uiversity (2016): n. pag. Web. http://cs229.stanford.edu/proj2016spr/report/062.pdf (related works)

[4] Parsa, H. G. "Why Restaurants Fail." Cornell Hotel and Restaurant Administration Quarterly 46.3 (2005): 304-22. Web. http://journals.sagepub.com/doi/abs/10.1177/0010880405275598

[5] Luca, Michael. "Reviews, Reputation, and Revenue: The Case of Yelp.com." Harvard Business School Working Paper, No. 12-016, September 2011. (Revised March 2016. Revise and resubmit at the *American Economic Journal - Applied Economics*.) http://www.hbs.edu/faculty/Pages/item.aspx?num=41233

[6] P. Cheeseman, & J. Stutz, "Bayesian classification (AutoClass): Theory and results," in *Advances in knowledge discovery and data mining*, CA: AAAI Press, 1996, pp. 153–180.

[7] S. Raschka, "A thorough guide to SQLite database operations in Python,"http://sebastianraschka.com/Articles/2014_sqlite_in_python_tutorial.html, 2014.

[8] PySpark, "PySpark 2.0.2 documentation," http://spark.apache.org/docs/latest/api/python/.