Project Final Paper - Popularity Forecast with Yelp Dataset

Group: Yelp Help

Name: Chiao-ya Chang, Chen Song, Tony Chu, Xinyi Yang

I. Introduction

Yelp is a public company that develops, hosts, and markets Yelp.com and Yelp mobile app, which publish crowd-sourced reviews about businesses. As a great platform choosing consumer activities, Yelp provides pages devoted to information about restaurants and stores, allowing users to use a one-to-five-star rating system to evaluate products and services and to provide meaningful and detailed reviews about their experience (Chafkin, 2010). On one hand, users can make their decisions between similar restaurants or services based on the ratings and reviews shown on the Yelp pages. On the other hand, business owners can get an overall assessment of their restaurant's performance based on the reviews and ratings on Yelp and they can also further leverage the same massive dataset to analyze users to generate business insights for wiser decisions. Our project goal is to predict the popularity of restaurants. We used several machine learning methods including logistic regression, Naive Bayes, xGBoost, CNN, and LSTM to make predictions of popularity.

I-1. Motivation

Consider the following scenario: you just opened a restaurant a few months ago, and luckily it already had quite an amount of customers. However, since there are still a few seats left empty during mealtime, you wonder if your restaurant will continue to grow. Therefore, it would be great for you, the restaurant owner, to know how popular the restaurant will be in the future. If the restaurant cannot increase its popularity in the future and is still not making any profit, it might be better to halt the business now. We believe such a scenario happens quite often for many local restaurant owners, and the information garnered from Yelp should help business owners make those difficult business decisions. Besides, customers can choose a restaurant that is not only popular right now but also in the future, and investors can use data from yelp as reference to know whether a restaurant is worth an investment or not.

I-2. Research Question

Our main research question is: *how popular will a restaurant be in the future?* Please note that this question is formed with caution. The popularity of a restaurant is not an unbiased estimator for the profitability of a restaurant. However, since (1) it is almost impossible to retrieve financial data from local restaurants (they are very unlikely to have standard income statements) and (2) financial status may be affected by several other issues that have no direct link to the service that a restaurant provides, we feel that it is more appropriate to use popularity instead of profitability to estimate how well a restaurant operates. We are more interested in researching the relationship between the growth of a restaurant provided by its current services. Given our popularity forecast, restaurant owners should be able to further

calculate their own profitability easily. The following lists some sub-questions that we are interested in or we must deal with during our research:

- How popular will a restaurant be in the future?
- How to define popularity? That is, what is the appropriate metric for popularity?
- What features can be used to predict popularity?
- How do different features impact the performance of predictions?

I-3. Previous Work

Yelp not only wants to make sure that consumers are making informed decisions but also concerns about how to help restaurants become popular and make more profits. Since 2012, they started to try a lot of methods to understand consumer behaviors and restaurant performance from the reviews. For example, they developed the "consumer alert" to avoid the "significant attempts" to pay for reviews. To help users decide what to order, they also used complex Natural Language Processing (NLP) models (Anna F., 2019) to discover popular dishes. It focused on matching photos and reviews given by customers and the dishes on the menu provided by the owners.

Though many researchers have conducted research on ratings or review text respectively, only a few works combined the two sources to develop models. The most-cited paper is "Hidden factors and hidden topics: understanding rating dimensions with review text (McAuley, J., 2013)", which aimed to develop statistical models titled 'Hidden Factors as Topics', or HFT for short to combine latent rating dimensions (such as those of latent-factor recommender systems) with latent review topics (such as those learned by topic models like LDA). Based on their models, they found the features to suggest informative reviews. Another top-cited paper is "review, reputation, and revenue relationship research (Luca, 2016)" which researched the impact of Yelp. For example, they investigated whether consumers used Yelp to learn about restaurants. They used a regression discontinuity approach to support the hypothesis that Yelp has a causal impact. They also applied fixed effect regressions to estimate the heterogeneous effects of Yelp ratings. Finally, they certified the quality of a review by comparing the response to Yelp with the Bayesian hypothesis. Another project used review data to predict the closure of restaurants (Alifierakis, 2018), which explored reasons for success or failure. Alifierakis established a complicated model to predict the results. The precision of open restaurants is 91%, which is rather high and referable to our project. Except for research methods, there are also different calculation bases about popularity. For example, in the paper "Predicting Restaurants' Rating And Popularity Based On Yelp Dataset", the popularity trend is used to evaluate whether a restaurant will be more popular next year.

$$trend_i = \frac{\sum_{j=1}^{j\leq(len_i+1)/2} \frac{rev_{j,i}}{\text{total \# of reviews in year j}}}{\sum_{j\geq(len_i+1)/2}^{len_i} \frac{rev_{j,i}}{\text{total \# of reviews in year j}}}$$

(Yiwen Guo, ICME, Anran Lu, ICME, and Zeyu Wang, "Predicting Restaurants' Rating And Popularity Based On Yelp Dataset")

In spite of the similarity between our work and the above research, here is one critical difference: our goal is to predict the popularity of restaurants from the reviews while other works may only focus on the

relationship between the reviews and star rating. We also define the popularity for our project based on the project goal and features we use. These projects inspired us to explore the reasons for the popularity of these restaurants. The prediction of popularity can both benefit restaurants and customers. It would be a big step forward!

II. Data Description

II-1. Data Source

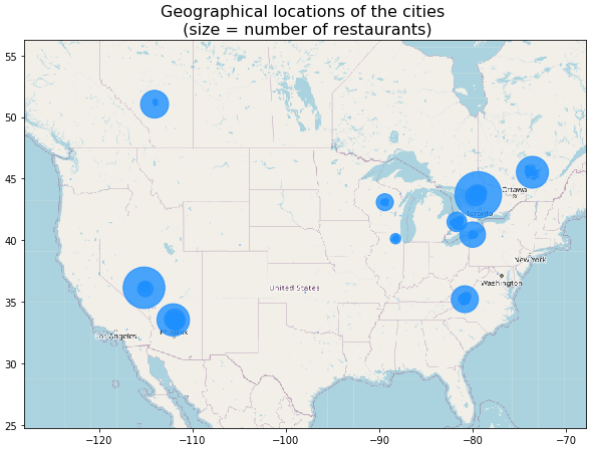
The Yelp open dataset (link: https://www.yelp.com/dataset) is our primary source of data, which provides a large amount of data regarding restaurants, customers, and their reviews. This dataset is a subset of Yelp's database, provided by Yelp for use in personal educational, and academic purposes. It is strictly linked to the topic of restaurants, easing our tasks regarding data collection and data cleaning.

The Yelp dataset includes 5 json files, containing over 8 million user reviews on 200,000 businesses from 1,250 cities (Yelp, 2020). The detailed description of each json file is listed in Appendix A.

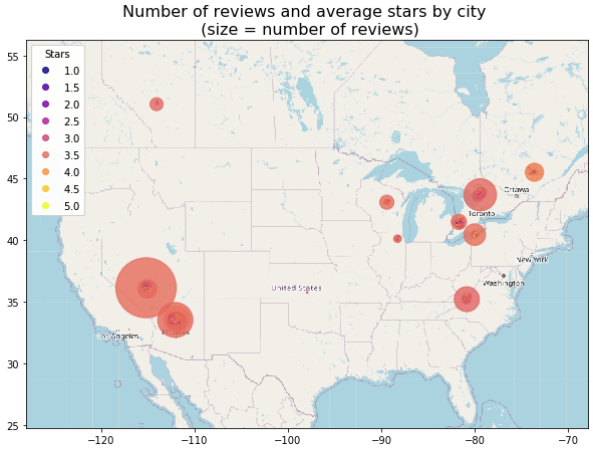
In our project, we use three out of the five json files, which are business.json, review.json and user.json, because the features included in these datasets are highly related to our topic.

II-2. Exploratory Data Analysis

Number of restaurants and reviews of each city



(Figure 1) Number of restaurants by city



(Figure 2) Number of reviews and average stars by city

In this dataset, there are lots of Canadian restaurants, including restaurants in Toronto, Montreal, and Calgary. As for restaurants in the US, their locations include Las Vegas, Phoenix, Charlotte, and Pittsburgh. It is noticeable that our research domain is limited to these metropolitan areas. In addition, from the number of reviews, we may guess that Americans seem to be more passionate about writing reviews, as the number of reviews in Canadian cities is less than that in American cities on average. We may see that the dot sizes of Canadian cities shrink significantly in Figure 2 compared to Figure 1.

Popular restaurant categories

(Figure 3) Word cloud of restaurant categories

From the word cloud, we may see that American restaurants are the predominant type of restaurant in the dataset. Some other noticeable types include Asian Fusion, Mexican, Middle Eastern, and Italian. In addition, a significant element in the categories is bars; there are multiple keywords relevant to bars: nightlife bars, sports bars, and cocktail bars. Another significant element is fast foods with keywords such as chicken wings, fast, pizza, and burgers. Some other important elements include breakfast/brunch and coffee/tea, which are offered by many American restaurants.

Ⅲ. Data Analysis

Ⅲ-1. Restaurant Popularity

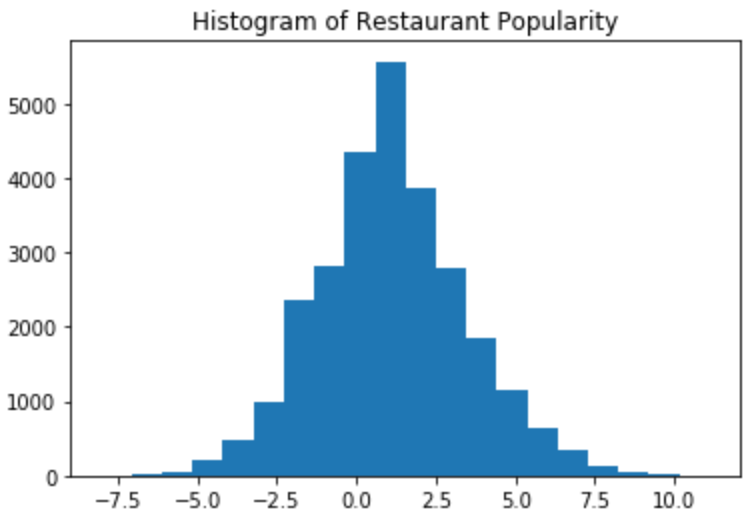
Before discussing how reviews impact a restaurant's popularity, we must first define what is popularity.

Definition

To define popularity, we use the following variables: average stars and the number of restaurant reviews. We subtract average stars by three for normalization and take natural log to reduce the effect of the high number of reviews and to minimize computational complexity. The following shows our formula for popularity:

$$popularity = (average\ star - 3) \times log\ (number\ of\ reviews)$$

In this project, we calculate the popularity of restaurants in 2019 on a yearly basis. That is, we use data in 2018 to predict how popular a restaurant will be in the year of 2019 (in a time interval manner). The following figure shows the distribution of restaurant popularity, which follows Gaussian distribution.



(Figure 4) Histogram of restaurant popularity in 2019

Label of Restaurant Popularity

We converted continuous popularity scores to 5-class ordinal labels to help us make predictions easier since we faced major failures while predicting continuous outcomes. Our cut points are 20-quantiles, 40-quantiles, 60-quantiles, and 80-quantiles. We ranked 5 as the highest popularity (i.e., greater than 80 quantiles) and 1 as the lowest popularity (i.e., less than 20 quantiles).

Ⅲ-2. Features engineering

In the business dataset, we calculated the average star rating and the total number of reviews received of each restaurant. We also aggregated the number of reviews for each city to avoid bias caused by regional differences.

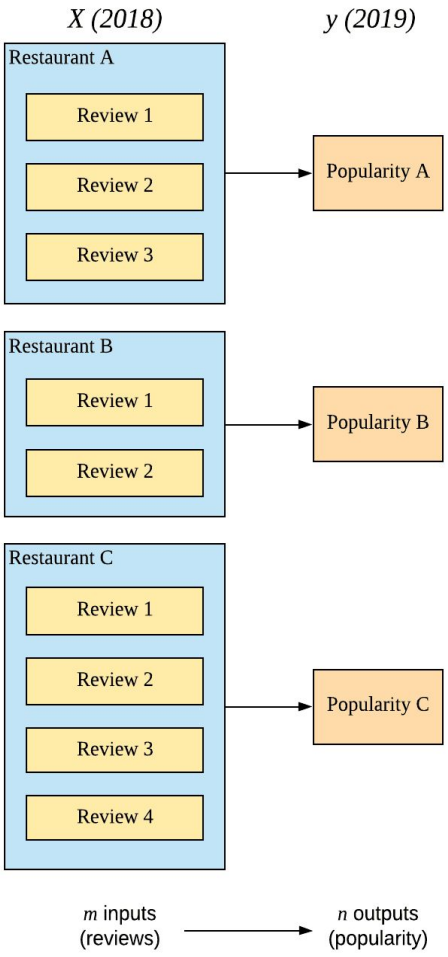
In the review dataset, we extracted stars, likes, text length and text contents of each review. Considering that users apply distinct criteria to rate stars based on their presumably similar experience, there might be bias if we only include star ratings to evaluate how much the restaurant is liked by its customers. Therefore, we did sentiment analysis on the text reviews to extract users' polarity and subjectivity contained in the reviews they wrote. With these two features involved in the overall analysis, we can more or less weaken the influences of star rating bias. We also used text processing methods, like TF-IDF and Word Embedding (GloVe) to process review contents.

In the user dataset, we calculated the average star rating, the total number of reviews each user gives, the number of fans each user has, and the length of the period the user has been using Yelp. We aggregated all the votes, such as "cool", "funny" and "cute", and compliments each user has received. Moreover, we calculated the Eigenvector Centrality of users to consider the level of influence of a user within the Yelp network. The reason for extracting users' features is to take the influence of users on the popularity of the restaurant into consideration.

After selecting features, we combined all features together into one dataset. We randomly selected 17,071 restaurants as our training set and 5,335 restaurants as the test set. More details are in Appendix B.

Ⅲ-3. Data Modeling

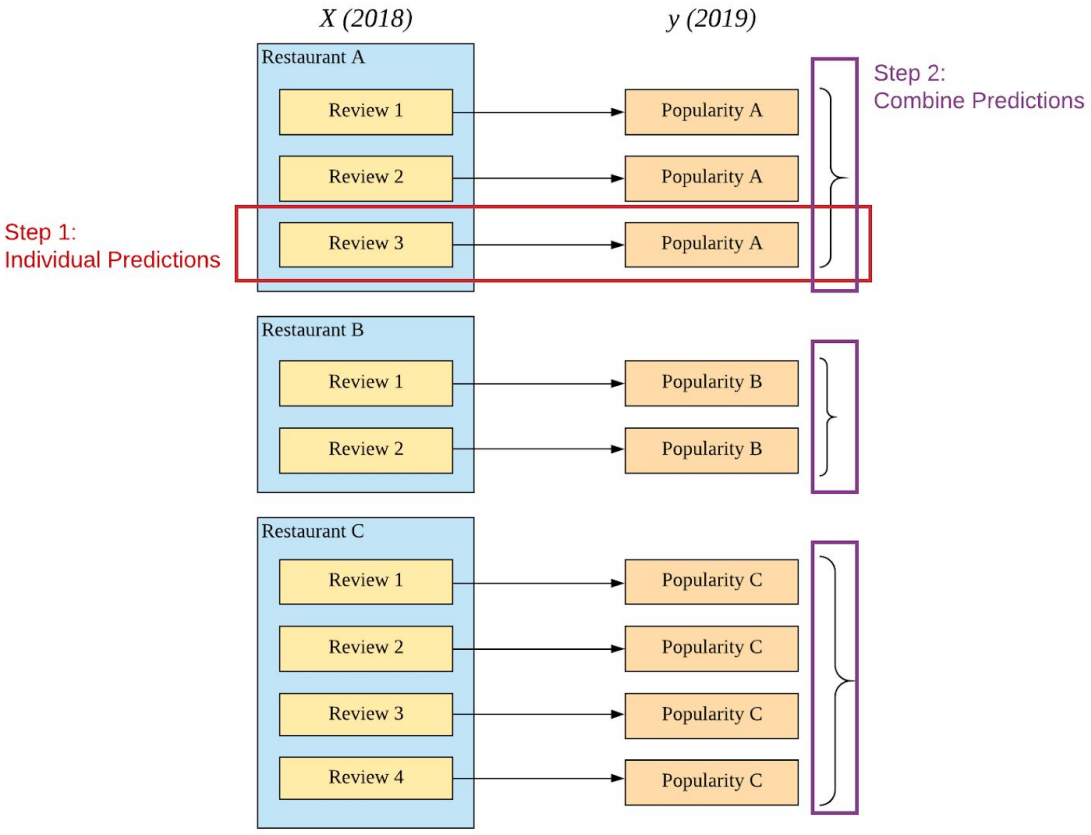
Since we want to predict restaurant popularities with customer reviews, we will inevitably face a problem regarding data modeling given the features we have. To elaborate, consider the normal data model as a bijective function in which every X corresponds to a y. That is, take our dataset for example, n restaurants will correspond to exactly n popularity scores. However, this is not the case for our data. Since every restaurant may have multiple reviews, our model can not be a simple one-to-one correspondence relationship but the following:



(Figure 5) Data model with surjective representation

We may see that our model should be a surjective function in which multiple reviews can correspond to one popularity score. That is, we will have m customer reviews correspond to n restaurant popularity

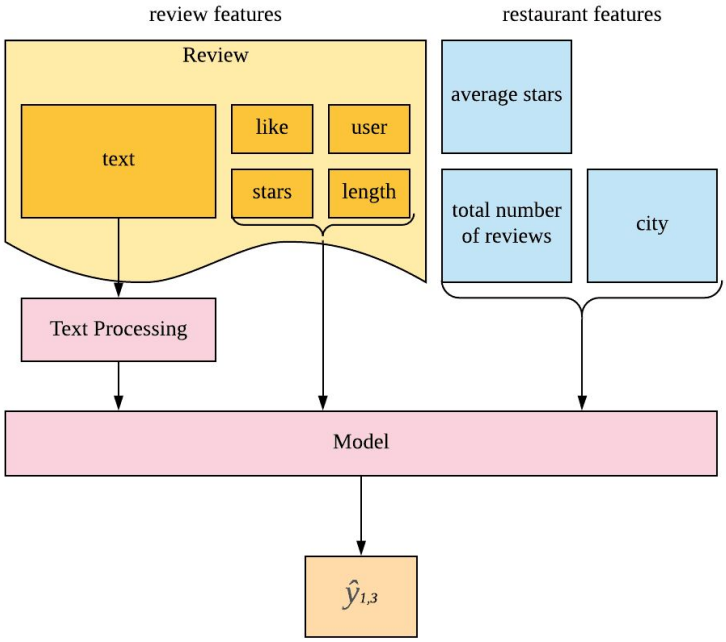
scores, with m much larger than n. Then, in this case, we cannot apply normal data models to predict popularity. To solve this problem, we applied the following two-step model:



(Figure 6) Two-step data model

First, we let every customer review predict the popularity score of its restaurant. Then, we will find a method to aggregate all these individual predictions by restaurant. The aggregated prediction will be our final popularity prediction for the restaurant.

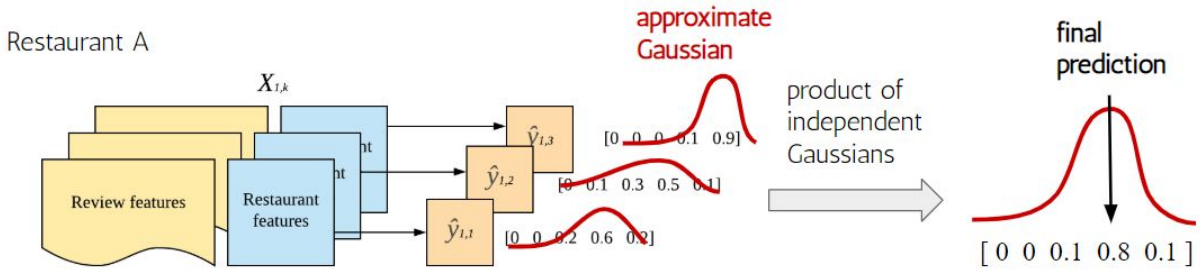
First, let's dive deep into Step 1.



(Figure 7) Step 1: Individual Predictions

We used features of a review, including the text of the review, stars of the review, and the user of the review, and features of the restaurant, including the average stars of the restaurant and the number of reviews received in 2018, to make individual predictions. Before streaming all features into a model, we will first apply some text processing methods to the text. Then, the processed text and the rest of the features will be inputs of the model to produce prediction $\hat{y}$, which is one-hot encoded labels with five classes, following categorical distribution. The results of the text processing methods and the models we chose would be summarized in Part III-4.

Now, let's proceed to Step 2, in which we will combine the individual predictions made by Step 1:



(Figure 8) Step 2: Combine Predictions

Since our labels are in fact ordinal, we assume that these $\hat{y}$, which follow categorical distribution, would approximate Gaussian distribution. The product of these assumed independent Gaussian distributions, which will also be a Gaussian distribution, will be our final prediction for the popularity of the restaurant,

with the mean of the combined Gaussian distribution rounded to the nearest integer as the predicted class. Please note that this Gaussian combination method is not the only method to combine predictions; some other combination methods are listed in Appendix C.

Thus, through this two-step method, which combines individual predictions, we can now deploy normal classification models to predict popularity of restaurants. The following part will show the result of text processing methods and models we chose in Step 1.

Ⅲ-4. Results and Findings

Model Selection

We first built a simple logistic regression using average review stars and total review counts as our baseline model. Then, based on the previous work, we knew that the Naive Bayes model can have better performance on text classification. The xGBoost (eXtreme Gradient Boosting) model can work efficiently with the large scale dataset and is popular since many winning teams of the competitions chose to use it. Deep learning models can help us prevent overfitting. The CNN model focuses on small patterns and LSTM models can take input as a sequence vector. Thus, we decided to deploy Naive Bayes, Xgboost, CNN, and LSTM models with the same numeric features and different text features.

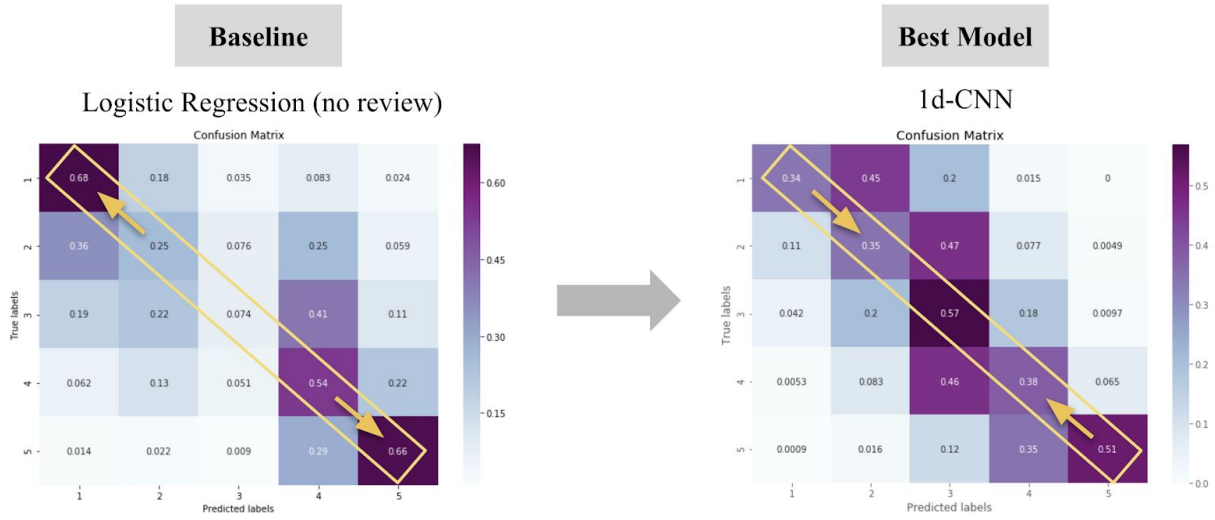
Evaluation

After combining predictions by restaurants, we calculated root mean square errors (RMSE) instead of classification accuracy to evaluate our models. The reason is that though our problem is a classification problem, our labels are ordinal, and we need to consider the distances among them. RMSE makes a lot of sense for us to interpret the results. Then, we also compared the models, using confusion matrices.

Results

We tested the models with diverse parameters and remained one with the best performance in each supervised model. The RMSE is 1.2063 in the baseline model. The highest RMSE is 1.2833 in the Gaussian Naive Bayes model while the lowest RMSE is 0.9807 which happened in the CNN model with one 1d-convolutional layer and three fully-connected layers. More details are shown in Appendix D.

Since the popularity is Gaussian distribution, if the middle labels can perform better, the prediction is more accurate. From figure 9, the left matrix is from the baseline model and the right one is from the CNN model. We can see the trend that higher recall values are in labels 2, 3 and 4 in the right plot.



(Figure 9) Confusion matrices of the baseline model and the CNN model

Conclusion

Both text and non-text features can impact on the performance of predictions. Compared to the baseline model, we successfully built models to predict the next year's popularity, using features like the reviews and users. Our best model is the CNN model with RMSE = 0.9807.

Ⅳ. Future Scope

Due to time constraints, our current analysis does not consider various issues. The following issues will be discussed in our future analysis if time permits.

more data	The Yelp dataset collected more than 10 years of data, allowing us to make predictions recursively and increase the dimension of our analysis. Moreover, we can combine the Yelp dataset with other relevant datasets, like demographic datasets and macroeconomic data to find more features.
anomaly detection	It is possible that there are anomalous reviews in our dataset. If we can conduct anomaly detection to exclude these data points, we can decrease bias. Some common approaches for anomaly detection include cluster-based estimation and isolation forest (Li, 2019).
more models	Currently we have considered only a few models for classification, such as logistic regression, xGboost, and neural networks, and a few techniques for natural language processing, such as TF-IDF and word embedding. We will continue researching more feature engineering methods and models for our analysis in the future.

Ⅴ. Division of Work

Process	Work	Name
Introduction	Motivation, research questions, and previous work	Chiaoya, Chen, Tony, Xinyi
Data preparation	Data cleaning	Chiaoya, Tony
	Features engineering	Chiaoya, Tony
	Exploratory data analysis	Chen, Xinyi
Data modeling	Model selection	Chiaoya, Chen, Tony, Xinyi
	Model architecture	Tony
	Model building	Chiaoya, Chen, Tony, Xinyi
	Model evaluation	Tony
Future work	--	Chiaoya, Chen, Tony, Xinyi

* Github for our code: https://github.com/cyac15/IMT575_yelpHelp)

References

Algorithmia. (2020, March). Using machine learning for sentiment analysis: a deep dive. Retrieved from https://algorithmia.com/blog/using-machine-learning-for-sentiment-analysis-a-deep-dive

Alifierakis, M. (2018, January). Using Yelp Data to Predict Restaurant Closure. Retrieved from: https://towardsdatascience.com/using-yelp-data-to-predict-restaurant-closure-8aafa4f72ad6

Anna F., Parthasarathy G. (2019, October) Discovering Popular Dishes with Deep Learning. Retrieved from https://engineeringblog.yelp.com/2019/10/discovering-popular-dishes-with-deep-learning.html

Chafkin, Max (February 1, 2010). "You've Been Yelped". Inc. Magazine. Retrieved January 6, 2013.

Disney, A. (January 2, 2020). Social network analysis 101: centrality measures explained. Retrieved from: https://cambridge-intelligence.com/keylines-faqs-social-network-analysis/

Luca, M. (2016). Reviews, reputation, and revenue: The case of Yelp. com. *Com (March 15, 2016). Harvard Business School NOM Unit Working Paper,* (12-016).

McAuley, J., & Leskovec, J. (2013, October). Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems (pp. 165-172).

Yelp, Inc. (2020). Yelp Dataset JSON. Retrieved from https://www.yelp.com/dataset/documentation/main

Yelp, Inc. (2020). Yelp Dataset [Data file]. Retrieved from https://www.kaggle.com/yelp-dataset/Yelp-dataset

Yiwen Guo, ICME, Anran Lu, ICME, and Zeyu Wang, “Predicting Restaurants Rating And Popularity Based On Yelp Dataset”, 2017

Appendix A: Details of the Yelp dataset

1. business.json (209,393 observations of 14 variables): contains data of businesses including name, location, and basic information such as average stars, number of reviews, and operating hours.

Column Name	Data Type	Content
business_id	string	22 character unique business id
name	string	the name of the business
address	string	the full address of the business
city	string	the city where the business is located
state	string	2 character state code of the business is located in, if applicable
postal code	string	the postal code of the business
latitude	float	latitude of the business location in decimal degrees
longitude	float	longitude of the business location in decimal degrees
stars	float	the average star rating of the business, rounded to half-stars
review_count	integer	the number of reviews that the business has
is_open	integer	boolean indicator of whether the business is closed or open (0: closed, 1: open)
attributes	object	an object of key business attributes to values boolean indicators; for instance: {“RestaurantTakeOut”: True}
categories	array	an array of strings of business categories; for instance: [“Mexican”, “Burgers”, “Gastropubs”]
hours	object	an object of key day to value hours, hours using a 24hr clock; for instance: {“Monday”: “10:00-21:00”}

2. review.json (8,021,122 observations of 9 variables): contains data of reviews including the user who wrote the review, the business being reviewed, and text content of the review.

Column Name	Data Type	Content
review_id	string	22 character unique review id

user_id	string	22 character unique user id of the user posted the review; maps to the user in user.json
business_id	string	22 character unique business id of the business being reviewed; maps to the business in business.json
stars	integer	star rating given in the review
date	string	when the review was written, with format YYYY-MM-DD
text	string	full text of the review
useful	integer	the number of useful votes received
funny	integer	the number of funny votes received
cool	integer	the number of cool votes received

3. user.json (1,968,703 observations of 22 variables): contains data of users including user names, the number of reviews that the user has posted, when the user joined Yelp, and compliments that the user has received.

Column Name	**Data Type**	**Content**
user_id	string	22 character unique user id
name	string	the user's first name
review_count	integer	the number of reviews that the user has written
yelping_since	string	when the user joined Yelp, with format YYYY-MM-DD
friends	array	an array of strings; containing user's friends as user_ids
useful	integer	the number of useful votes sent by the user
funny	integer	the number of funny votes sent by the user
cool	integer	the number of cool votes sent by the user
fans	integer	the number of fans that the user has
elite	array	an array of integers; containing years that the user was elite
average_stars	float	average rating of all reviews that the user posted

compliment_hot	integer	the number of hot compliments received by the user
compliment_more	integer	the number of more compliments received by the user
compliment_profile	integer	the number of profile compliments received by the user
compliment_cute	integer	the number of cute compliments received by the user
compliment_list	integer	the number of list compliments received by the user
compliment_note	integer	the number of note compliments received by the user
compliment_plain	integer	the number of plain compliments received by the user
compliment_cool	integer	the number of cool compliments received by the user
compliment_funny	integer	the number of funny compliments received by the user
compliment_writer	integer	the number of writer compliments received by the user
compliment_photos	integer	the number of photo compliments received by the user

4. checkin.json (175,187 observations of 2 variables): contains data of check-ins, a feature of Yelp by which users can check-in a restaurant when they visit the restaurant, including the business being checked-in and the date and time of check-ins.

Column Name	**Data Type**	**Content**
business_id	string	22 character unique business id of the business being checked-in; maps to the business in business.json
date	string	a comma-separated list of timestamps for each check-in, each with format YYYY-MM-DD HH:MM:SS

5. tip.json (1,320,761 observations of 5 variables): contains data of tips, which resemble short reviews that tend to convey quick suggestions, including the user who wrote the review, the business that the user left the tip on, and text content of the tip.

Column Name	**Data Type**	**Content**
text	string	full text of the tip
date	string	when the tip was written, with format YYYY-MM-DD

compliment_count	integer	the number of compliments the tip has
business_id	string	22 character unique business id of the business that the user left the tip on; maps to the business in business.json
user_id	string	22 character unique user id of the user posted the tip; maps to the user in user.json

Appendix B: Train-validation-test split

X (2018)	Restaurants	reviews	users
(A)Train	17,071	428,724	235,107
(A1)Validation	4,268	102,135	76,143
(B)Test	5,335	129,141	93,472

Y (2019) popularity

Appendix C: Possible combination methods for Step 2

The combination of predictions is similar to the concept of weighted sum, that is, we need to find a way to determine the weights of each prediction. There are several methods to determine weights:

Equal Weights:
All predictions have equal weights.

- **Simple Average**: simply calculate the arithmetic mean of predictions.
- **Simple Voting:** simply count which class has the most votes, i.e., predictions.

Unequal Weights:
Predictions would have different weights based on the "confidence" of each prediction.

- **Variance**: weights are the reciprocal of the variance of the prediction. That is, predictions with lower confidence, i.e., larger variance, will have lower weights, and vice versa.
- **Bayesian:** weights are based on Bayesian probability. For instance, if P(true = 3 | predict = 3) = 80% and P(true = 5 | predict = 5) = 60%, prediction = 3 would have a higher weight than prediction = 5.
- **Max-rule (winner-take-all)**: the final prediction solely depends on only one prediction which has the highest confidence, i.e., the most concentrated prediction.

Some more advanced methods include Mixture of Experts (MoE), of which the weight of each prediction is trainable. For reference, the following article summarized various combination methods:

Xu, L., & Amari, S. I. (2009). Combining classifiers and learning mixture-of-experts. In *Encyclopedia of artificial intelligence* (pp. 318-326). IGI Global.

Appendix D: Results of different models

Models	Text Processing Method	Parameters	RMSE
Logistic Regression	-	Default values	1.2063
Gaussian Naive Bayes	TF-IDF	Default values	1.2833
xGBoost	TF-IDF & SVD (Singular Value Decomposition)	learning rate = 0.1, max_depth=8, early_stopping_rounds=10, eval_metric="mlogloss"	1.0035
CNN	Word Embedding (GloVe) & Conv1D(7) + Global Max Pooling	3 fully-connected layers (64, 64, 8), optimizer = adam with learning rate = 5e-4, batch size = 64	0.9807
LSTM	Word Embedding (GloVe) & LSTM(100)	2 fully-connected layers (256, 8), optimizer = adam with learning rate = 2e-4, batch size = 64	0.9813