Visit Prediction For Yelp's Restaurants

collaborative filtering and binary classification methods

Mao Li
CSE, UCSD
San Diego, CA, USA
mal131@ucsd.edu

Zhaoyi Huang
CSE, UCSD
San Diego, CA, USA
zhh083@ucsd.edu

Yilin Xie
CSE, UCSD
San Diego, CA, USA
yix176@ucsd.edu

Haoming Zhang
MATH, UCSD
San Diego, CA, USA
haz260@ucsd.edu

ABSTRACT

Understanding whether a user would like, rate, or visit a local restaurant is a rather challenging task because multiple factors need to be taken into account. The key factors to be considered include users and restaurants' geographical location, users' categories preference, restaurants' popularity, etc. As part of this assignment, we are provided with a subset of Yelp's business, reviews and users data. Our objective is to gain insight into the user's preference and predict whether a given user would visit a given restaurant.

We performed exploratory analysis on the provided dataset. Based on our analysis, we extracted features from restaurants, users, and users' reviews on these restaurants that represent user-restaurant interaction. With these features, we tried a variety of methods and build collaborative filtering models and binary classifiers for this predictive task. Our best performing model outperforms the baseline models.

1 INTRODUCTION

Nowadays, understanding and predicting users' preference from their past behaviors is in high demand for businesses. As part of Yelp's Academic Data Challenge[9], Yelp provides a subset of its businesses, users, reviews and other related data for the public to study and build predictive models. In the past rounds of the competition, many contributions have been made for the fields of hidden factors in review text, restaurants recommendation, stars rating prediction and etc. One of the key fields is understanding a given user's preference towards a business. For example, whether a given user would visit, recommend, like a business and how would the user rate the business.

Taking inspirations from past researches, we aimed to build a predictive model for determining whether a user would visit a given restaurant. We tried to incorporate user, business and review feature to build collaborative filtering models and binary classifier with logistic regression and support vector machines.

2 DATASET Specifications

2.1 Data Collection

The dataset we used was from Yelp Open Dataset[9]. This dataset was collected by Yelp, containing the data of business, users, and reviews for use in personal, educational, and academic purpose. The files we used in this predictive task were

yelp_academic_dataset_business.json:

- contains **188,593** unique businesses across USA and Canada ;

The key features of this datasets are:

- business_id - The unique business id
- city - The string of city a business located in
- categories - The categories of the business
- is_open - The binary value indicating if the business is open. 1 if yes, 0 not.

yelp_academic_dataset_user.json:

- contains **1,518,169** unique users;

The key features of this dataset are:

- user_id: The unique user id identify users on yelp
- friends: The id of friends for the user
- elite: The years the user is credited by Yelp as an elite user

yelp_academic_dataset_review.json:

- contains **5,996,996** reviews.

The key features of this dataset are:

- user_id: The id of users
- business_id: reviews of business
- text: The text of the review

Each file is composed of a single object type, one JSON-object per line.

2.2 Data Properties

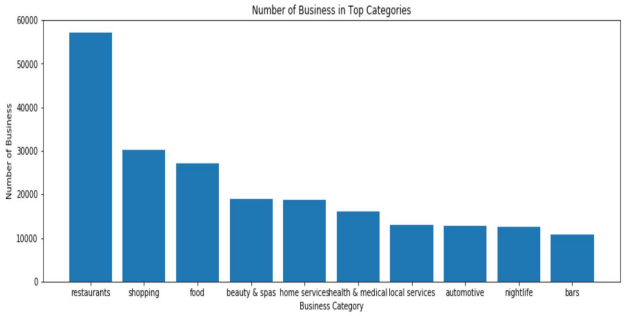
Each business in this dataset has a unique bussiness_id. It also includes precise location, rating, review count, cuisine categories, and working hours related to the business.

Each user in this dataset has a unique user_id, a list of their friends user_ids, review count, the helpfulness of his or her reviews, the average rating of all reviews, and evaluation of user by other users.

Each review in this dataset has a unique review_id, together with a user_id and a business_id. It also contains the rating and review text given by the user to the business. Other users' evaluation of this review is also included.

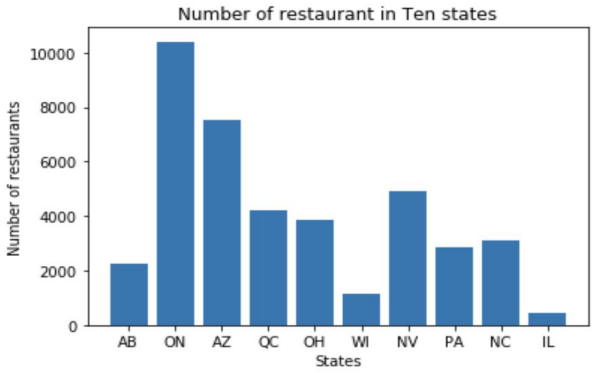
2.2.1 Business

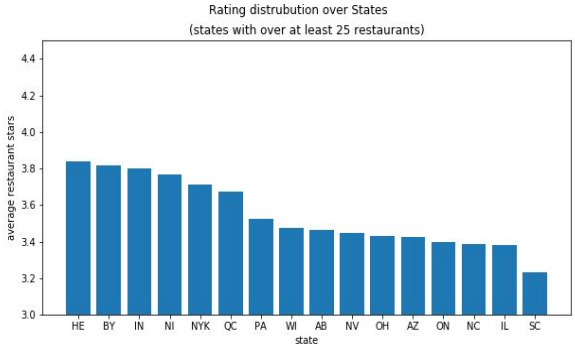
The analysis of the dataset noted that over 55,000 businesses are in the category "restaurants", while "shopping", the second largest category, only consists approximately half of the size (around 30,000).



Businesses of major categories like "restaurants" and "shopping" are also related to other more detailed sub-categories such as "burritos" or "Chinese". Therefore, to reduce the training time and focus on our objective, we decided to only use the businesses categorized as "restaurants".

The geographical distribution of businesses spreads over 100 cities across two countries. Most businesses in the dataset are located in Ontario, Canda. We found the average ratings for all restaurants in each state ranged from 3.2 to 4.0 and most of the averages are around 3.5. And the restaurant with the highest stars rating, "Mon Ami Gabi" is located in Arizona.

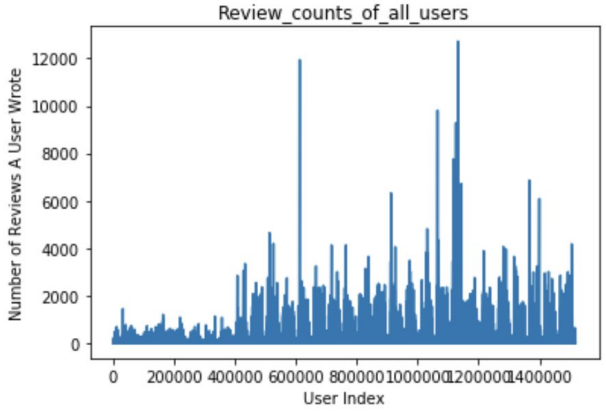




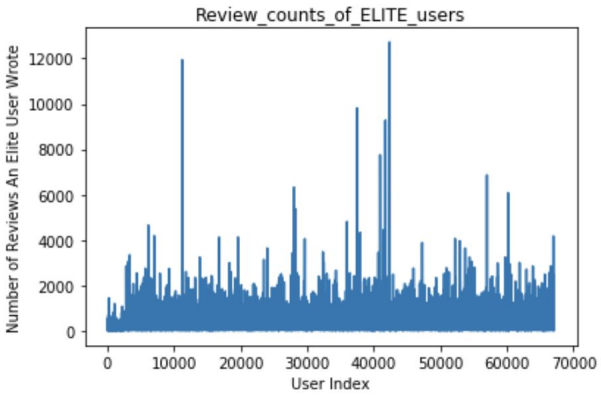
2.2.2 Users

Our analysis indicated that although the total number of users is over 1 million, the proportion of elite user credited by yelp is less than 5% (67,109 / 1,518,169).

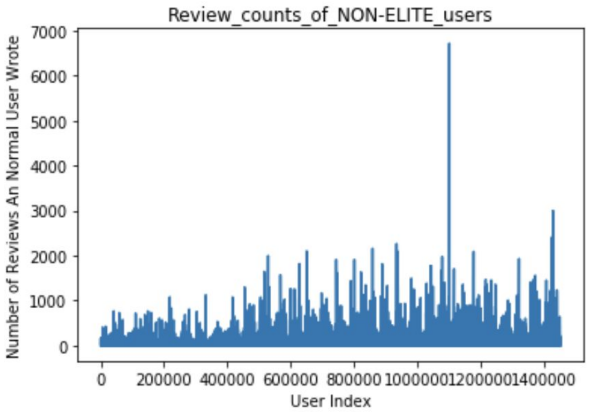
In terms of reviews counts, we found the average review counts of all users was 22.432, that of elite users was 224.69, and non-elite users was only 13.08.



Not surprisingly, the user with the most number of reviews was from the elite group.

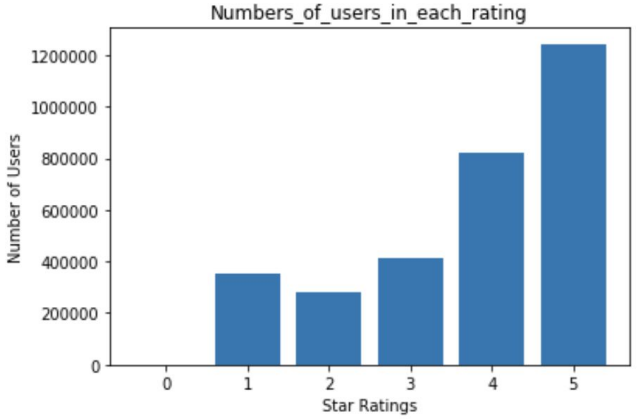


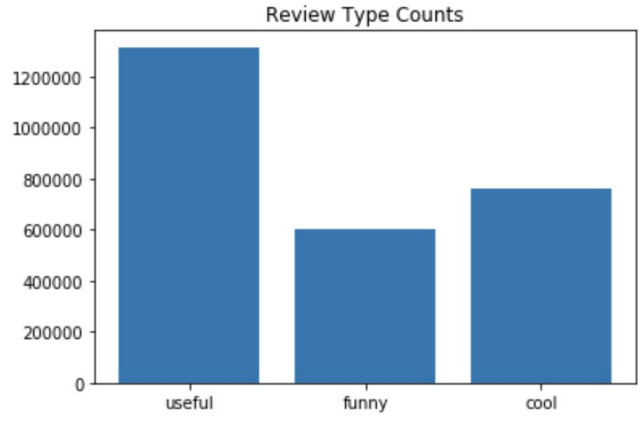
Nevertheless, there were a few users from the non-elite group whose review counts were also over thousands. The user with the most number of reviews from the non-elite group had over 6,000 reviews.



2.2.3 Reviews

The review data gives us huge corpus of user and restaurant pairs which the user has visited the restaurants. Our analysis indicated that there were 5,996,996 reviews in total, and 3,115,009 of them were restaurant-related. For the reviews related to restaurants, the majority of users (1,243,004) rated 5 stars. Most reviews were considered as "useful".





Our textual analysis on the review text also provides interesting insights. In five stars reviews, which we considered as positive review, words like "excellent", "reasonable", "friends", "flavour", and "comfy" appear more frequently than others. In reviews rated below 3 stars, which we considered as negative reviews, words like "madness", "dead", "wonton", "unacceptable", and "infestation" appear more frequent.

3 PREDICTIVE TASKS

Based on the dataset, we chose to predict whether a given user would visit a given restaurant in the future. (By visiting a restaurant, we really mean that the user write a review for the restaurant.) In particular, our predictive task is a binary classification problem: true if the user would visit/have visited the restaurant, and false otherwise.

3.1 Baseline Model

As suggested by exploratory analysis, we have huge number of data points across many different cities over U.S. and Canada. It's rather intuitive to assume users from a certain city would have a higher chance of visiting local restaurants in the city due to geographical reachability. Therefore, we have the baseline model to predict true if the given user has been to any restaurants in the same city as the given restaurant, and false otherwise. The accuracy of baseline model on the testing set is 0.741.

3.2 Model Evaluation

Reviews data contains user and restaurant pairs where the user visited the restaurant in the past. Those are all positive label data points. In order to complete our dataset, we randomly sampled user and restaurant to construct non-visited user and restaurant pairs with the size as positive pairs.

Then we evaluated our model's performance by computing its accuracy, the number of correct predictions (both positive and negative) divided by the total number of data points to predict, on the testing set.

3.3 Data Processing

For the task predicting whether a user would visit a specific restaurant, we only focused on the business data (from the original business dataset) that is categorized as "restaurants" and is currently open. In total we have 41,342 open restaurant data. For the users, we only considered the users who had been to

at least one of the 41,342 restaurants (i.e., had posted review). In total, we have 953,395 related users.

From the original review dataset, we had 5,996,996 pairs of user and business, which represented the action of a user visiting a business. Considering the open restaurants only, we had a dataset of users visiting restaurants with size of 3,115,009.

After that, we randomly generated 3,115,009 pairs of data which were not in the review dataset to represent the non-visited data.

Finally, we combined the first 1,000,000 visited data and 1,000,000 non-visited data to be our training set. Similarly, the second 1,000,000 visited data and 1,000,000 non-visited data as the validation set. The 1,115,009 visited and unvisited data left as the testing set.

3.4 Feature Extraction

There were multiple features we considered as appropriate for our predictive task. They are listed as following:

3.4.1 City Features:

Users are more likely to visit the restaurants in their own cities or some cities they visited before. Therefore, the city where the business is located in could affect if a certain user would visit the business. If the user visits the city the business is located in and posted a review for other businesses, then it is possible that the user would visit the business we want to predict as well. There are totally 1,111 cities in the dataset. To handle this feature, we simply stored the related cities of the businesses the user visited and tried to see if the business we want to predict is in the same city that user visited before.

3.4.2 Category Features:

The categories of the business are worth noticing. Users might repetitively go to restaurants of the same category. Known a user repeatedly visited businesses of certain categories, we assumed the user to be more likely to visit businesses of the same or similar kind. We used category features as a list of strings. Users categories are considered as the union of all categories of the restaurants the user has been. Then for every user and business pair, we compared their list of categories. If the categories of business are all in the user's categories list, the user might have higher chance to visit the business before. We also used the category features to find out similar users to the given user in prediction. If the similar users, who not only have similar taste (visit the restaurants with similar categories) but also visited the same cities, has visited the given business before, the given user may be more likely to visit the restaurants.

3.4.3 Review Text Features:

The review text gives us the information about the sentiment of users demonstrate to the restaurants. In particular, certain words may appear more frequent in positive reviews than usual and vice versa. To extract such feature, we compute Term Frequency - Inverse Document Frequency (TF-IDF) for the words in review text. Let t be term, d be document and D be corpus, the formula for TF-IDF is given as:

$tf(t,d)$ = number of times the term *t* appears in the document *d*

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D) \quad [10]$$

Higher TF-IDF scores of a word implies that this word appears more frequently in this document compared to others. Thus, TF-IDF of review words are fair indicator for whether the review is positive or not.

3.4.4 Friends Features:

Users' social network also provides insights for users' preference. Known a given user's friends have visited a certain business, the user would be more likely to visit the same business. We could count how many friends of a user visited the business before and take it

as feature. The more friends the user have visited the business the more likely the user visited the same place.

3.4.5 Similarity Features:

Considering restaurants and user features separately would provide half of the information for the predictive tasks. The interactions between users and business could give insights for user activities. Various methods can help us to get such insights. Certain similarity features could be useful for this by using different similarity calculations(Jaccard, Cosine, or Pearson similarity). Several similarity features can be constructed.

1. user - user similarity based on categories of business they have visited
2. business - business similarity based on categories

To determine user-user similarity based on categories, first create a dictionary of users containing a list of all categories a specific user visited before (that is, visit the business with that certain categories). Then we calculated the similarity between users with either Jaccard similarity or Cosine Similarity. We repeated the same steps to compute similarity between restaurants.

3.4.6 Popularity Features:

Based on our exploratory analysis, users tend to visit the restaurants that already have many customers. Thus, popularity of a restaurant is a very important indicator for whether a user would visit a restaurant. In our models, we compute the popularity score of a restaurant as the percentage of customers in the city who visit this restaurant. In a given city, the more customers the restaurants attract, the higher its popularity score. We also integrated the average rating of each restaurants as another feature of its popularity.

4 MODELS

To predict whether a given user would visit a given business, we used the idea of collaborative filtering and built logistic regression and support vector machine binary classifiers.

4.1 Similarity Based Models

Collaborative filtering model is a methods to predict a user's taste based on the tastes of many other users (which is the collaborative part come from). To do this, we should compare the data for many users and try to find out the similarities between them and make predictions based on such similarities. The two commonly used similarity methods includes the Jaccard similarity and Cosine similarity, we use these two similarities measures to produce different models based on collaborative clustering.

4.1.1: Naive Jaccard Similarity model

This model makes use of Jaccard similarity to search the most similar users to the given user and check if those users visited the given restaurant before. First, our model found the similarity between the categories the user has and the category of the given business using a very naive method. We constructed the categories of user by unioning all categories of the restaurants which this user visited before. Then computed user-restaurant similarity scores based on intersection of list of categories of the business and that of the user. Larger similarity score means user is categorically more similar to the restaurant. Second, our model computed the similarity scores between the given user and other users who visited this restaurant before. In order to do some, we need a threshold that distinguish "similar users" and "different users". In other words, if their similarity exceeds certain threshold, our model regarded two users are similar. Otherwise, it regarded them as different. Then the model predicted true if similar users have visited this restaurant before. We carefully chose the way to compute user-user and user-restaurant similarity

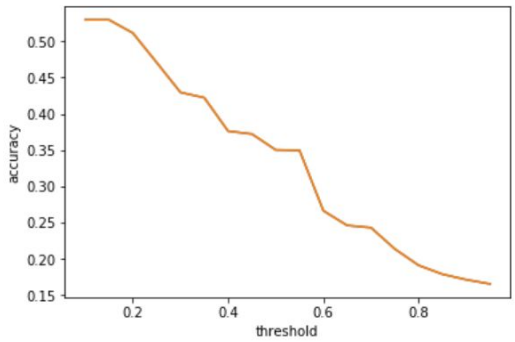
scores and selected threshold that distinguishes similar data points with non-similar ones.

In this naive model, we use Jaccard similarity to compute user-user and user-restaurant's categorical similarity[7]:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad [12]$$

Jaccard similarity is given by the size of intersections of categories between users divided by size of the union of users' categories.

To select the threshold of user-business similarity, the naive model took an unsupervised learning approach. For each different threshold, we trained model which returns true if user-business similarity score is greater than threshold. Otherwise, it returned false. The result shows that accuracy is inversely correlated with threshold. The max accuracy for the validation set occurs when threshold is rather small. In particular, threshold of 0.1 and 0.15 yields the accuracy of 0.54. It's reasonable because user may tend to visit a certain restaurant if a few categories of this restaurant match user's interest.



To select the threshold of user-user similarity, we repeat the above steps. For each different thresholds, we built model which predicted true if users who visited a given restaurant have similarity scores over a certain threshold and false otherwise. To reduce runtime, we subsampled 100,000 users. The threshold is also inversely correlated with accuracy. The threshold of 0.25 yielded the best accuracy of 0.7902 in the validation set. Smaller threshold results in better accuracy because users tend to have larger number of different categories and even a few similar categories between any user pairs can indicator they have similar taste.

After choosing the way to compute Jaccard similarity and selecting the thresholds for similarity scores, we trained our model on 100,000 data points and check its performance on test set of the same size. The resulting accuracy is around 0.77. We noticed that the model worked equally well if we discarded the user-restaurant similarity scores and we believed one possible reason was that it's less influential a factor than other factors of a restaurant, such as its popularity or rating.

This model in general performed better compared to the baseline model because it incorporated not only the city features but also the similarity between users based on categories. Yet, this naive model utilized unsupervised learning by manually choosing similarity threshold and we could possibly overfit our model by selecting the best thresholds on validation sets.

4.1.2 Cosine Similarity with SVM

We further built a model based on cosine similarity between users and trained a SVM with this similarity feature. Unlike the naive model, this time we use supervised learning to avoid selecting the thresholds manually on validation set. The cos similarity was calculated by the simple formula as follow:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

We built feature vectors for each user with the number of restaurants in each city and category where the user visited before. To compute the similarity between user A and user B, we calculated the dot

product of A's and B's feature vectors and divided by the products of their magnitudes.

For each (user, restaurant) pair in the dataset, we used the cosine similarities scores between the 10 most similar users who have been to the given restaurant and the given user as the feature vector for the data point. We used the Linear SVM from the sklearn library to implement our support vector machine. Due to runtime concern, we used 10% random sample of training set to train our model, and then tuned our regularization parameter C for the SVM based on validation set accuracy. C as 0.01 gives the best accuracy of 0.8211 on validation set, and 0.7967 on test set.

This is a fairly good model considering the fact that we only used a fraction of data to train this model. Given more affluent computation resources, this model could possibly perform better.

4.2 Support Vector Machine with Multiple Features

Features other than categorical similarities are also important in user's decision process of whether to visit a business. Thus, we incorporated different and key features of users and restaurants into our model. For each pair of user and restaurant, we built a feature vector that included popularity score, categories based similarity score, geographical location, number of friends who visit the restaurant and review text TF-IDF features. We aimed to use as many features as we can to capture all the properties of users and restaurants.

We used Linear SVM from sklearn library to train our binary classifier, based on the feature vectors. We set the max iteration parameter of Linear SVM to be 10000 for it to converge. We chose Linear SVM because it has better runtime on large dataset while still have fair performance, compared to SVM.SVC. Our binary classifier will be a hyperplane that classifies the given user and restaurant points into "would visit" and "won't visit". Based on our classifier's performance on the validation set, we chose regularization parameter C = 0.01, which results an accuracy of 0.76 on the test set. Unfortunately it did not have much improvement from baseline model. One possible reason would be that the features we chose are not descriptive enough for user-restaurant interaction.

Therefore, we repeated the process above with different combinations of features. As a result, we found that the feature vector with restaurant's popularity score, categories-based similarity, location, , stars rating and friend count yields the best overall performance. It achieved an accuracy of 0.8331 on validation set and 0.8024 on test set. Excluding TF-IDF feature leads to higher accuracy because it only indicates the whether the user tend to give positive reviews or not in the past and has little to do with whether the user would visit a new restaurant. Features like popularity, location and star rating are very good indicators because they are the decisive factors users take into account when choosing a restaurant.

4.3 Logistic Regression with Multiple Features

We also applied logistic regressions, which used sigmoid functions over linear combinations of features. We used the same vector features as the best performing one in previous section.

For binary classifications, logistic regression returned a value between 0 and 1 and we would asset the value to see if it is closer to which boundary. The logistic regression is more runtime efficient and less memory consuming than SVM, especially for the dataset of this size. We used the LogisticRegression from sklearn to implement the model. Then we tuned the hyperparameters such as solver and tolerance value based on validation set performance. We took the "sag" solver as it has relative better performance for

large dataset.. The result from logistic regression was lower than that of SVM, which was about 0.8101 on test set. One possible explanation for the poorer logistic regression performance may be that it failed to directly optimize the misclassification errors. We also tried other feature combinations for this model, but the performance improvement is trivial.

5 LITERATURE

The dataset we are using is from Yelp Data Challenge. In this challenge, Yelp provided a subset of its business, user, reviews, tips and check-in data and encouraged students to conduct researches and analysis on it to gain valuable insights.

Since the first round of Yelp's Data Challenge in 2014, hundreds of academic papers have been published to study multiple dimensions of the dataset with topics ranging from social recommender network to hidden factors and hidden topics in review text. Some research studied the latent subtopics from Yelp's restaurant reviews[1], while others related review text topics with star ratings[2]. From this dataset, many build predictive models with collaborative filtering, latent factors, neural network and hybrid of these models. In 2017, Ruirui Li and her team introduced a customer recommendation model that considers personal preferences of customers, geographical influence and business reputation[3]. In 2015, Kevin Hung and his team built a bigram multinomial naive Bayes to predict yelp review star classes[4].

Datasets similar to the Yelp's dataset including Amazon's product reviews, TMDB movie dataset from Kaggle, Goodreads books ratings and reviews, KKbox user-song data and many others. In the aspect of recommendation system, methods proposed for these dataset range from latent factor and logistic regression to neural networks and more sophisticated hybrid models. For example, Sparsh Gupta and his team built latent factor model for song recommendation using WSD-KKbox's user-song dataset[5]. Y. Koren and R. Bell also suggests matrix factorization techniques for building recommendation system for such data corpus[8].

One state-of-art methods studying this type of datasets was introduced by Julian McAuley and his team in 2013. They proposed to fuse latent rating dimensions with latent review topics and build "Hidden Factors as Topics" model[6]. Their model outperformed most baselines and could be scaled to many different datasets. It also facilitated tasks including but not limited to rating prediction, recommending new products and genre discovery.

Existing studies on this type of datasets and predictive task suggest that collaborative filtering based on user and business/book/movie/music is the most common methods and has a fair performance. This finding is consistent with our experiments. Additionally, previous studies have shown that hybrid features yields better prediction result in general. In our assignment, we found that hybriding some features such as geographical and categorical features could improve accuracy, while other features like review text had little effect. One possible reason may be that our TF-IDF text features are too naive to be as descriptive as advanced latent text features found by Latent Dirichlet Allocation algorithm[11].

6 RESULTS

The performance of our models evaluated in terms of their accuracy on test set is shown as follows:

Model	Validation Acc	Test Acc
Naive Jaccard Similarity	0.7902	0.7743
Cosine Similarity with Support Vector	0.8211	0.7967

Machine		
Support Vector Machine with Multiple Features	0.8331	0.8024
Logistic Regressions with Multiple Features	0.8323	0.8101

From the table, our best performing model was the classical binary classifier support vector machine with multiple features. The logistic regression with same multiple features works also better than naive similarity based approach, but performs worse than SVM model. It was also worth mentioning that the unsupervised learning naive similarity model worked fairly well, but it was probably a result of overfitting.

The final features we chose included restaurants' geographical information, restaurant popularity score, user-user categories-based similarity, and number of user's friends that visited the same restaurant. Our experiments have shown that these features lead to increase in both validation and test set performance. Meanwhile, the review text TF-IDF feature and user-restaurant similarity feature only lead to similar or worse performance than baseline. The reason could be that these features failed to capture the properties of the user or the restaurant. For instance, review's sentiment (as implied by TF-IDF) might have little to do with whether user would like a new restaurant.

As for parameters of our models, we had to use large max iterations for linear SVM for it to converge, due to the size of our training data. Also we select smaller regularization parameter because it creates classifier with a larger-margin hyperplane while avoiding overfitting the training set.

In summary, our best performing models utilize the idea of collaborative filtering. Our models extracted the most descriptive features of user and restaurant then used these features to train binary classifier via support vector machine or logistic regression.

7 REFERENCE

[1] James Huang, et al. *Improving Restaurants by Extracting Subtopics from Yelp Reviews*.

https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_ImprovingRestaurants.pdf

[2] Jack Linshi. *Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach*.

https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_PersonalizingRatings.pdf

[3] Ruirui Li, et al. *CORALS: Who Are My Potential New Customers? Tapping into the Wisdom of Customers' Decisions*.

https://s3-media3.fl.yelpcdn.com/assets/srv0/engineering_pages/f63a086ef2a3/assets/vendor/pdf/DSC_R09_CORALSWhoAreMyPotentialNewCustomers.pdf

[4] Kevin Huang, Henry Qiu. *Oversampling with Bigram Multinomial Naive Bayes to Predict Yelp Review Star Classes*.

https://kevin11h.github.io/YelpDatasetChallengeDataScienceAndMachineLearningUCSD/

[5] Sparsh Gupta, et al. *Metadata Based Collaborative Filtering for Music Recommendation.*

[6] Julian McAuley, Jure Leskovec. *Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text.*

https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_HiddenFactors.pdf

[7] Jaccard, Paul (1912), "The distribution of the flora in the alpine zone", New Phytologist, 11: 37–50,

[8] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 2009

[9] Yelp Academic Dataset Challenge.

https://www.yelp.com/dataset/challenge

[10] TF-IDF, wikipedia.

https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[11] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.

[12] Julian McAuley, Lecture 8, CSE 158 Fall 2018. http://cseweb.ucsd.edu/classes/fa18/cse158-a/slides/lecture7.pdf