ILS Z534 : Search

Final Project : Yelp Dataset Challenge

Project Report

Akash Sheth

Akshay Naik

Jigar Madia

Saniya Ambavanekar

Index

I. Task 1 : Recommend Business to Users

1. Problem Statement

For organizations like Yelp, the main problem while listing business is somehow finding out what the user may like and providing information which they are more prone to like. This is the problem we are trying to address as to how to recommend restaurants to users. For our solutions we have limited our methods to proposing just restaurants. Also since the original dataset was too large we have limited ourselves to data from 2 cities – Charlotte, North Carolina and Las Vegas, Nevada. Out of the 2 Charlotte has a mid-range data while Las Vegas has a large dataset.

The statistics for each are given below :

City	Charlotte	Las Vegas
Total Restaurants	247	4600
Total Users	300	1000
Total Reviews	20000	130000

Table 1: Dataset Statistics

2. Proposed Solution

- **Method 1 : Feature Based Approach**
  - **Overview**

After analyzing the data, we realized that each restaurant has 3 types of explicit features given in the information section of the restaurant to user – Neighborhood, Categories and Attributes

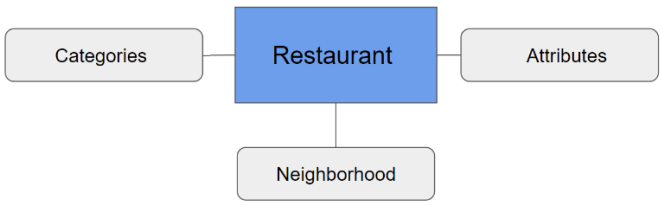

Figure 1: Restaurant Features

Neighborhood : Each restaurant belongs to a neighborhood. It's a common pattern that users are more prone to visit the neighborhoods which are close to their home or place of work.

Categories : Every restaurant has several categories which provide information on the type of establishment, cuisine etc. Usually a customer may visit different types of restaurants but when we see the overall data of customers with large number of reviews we may find that they always give preference to few select cuisines or may be more frequent visitors to fine dining or pubs

Attributes : Each restaurant has several parameters which tell us as to what facilities are available at the restaurant like Wi-Fi, Television, Alcohol, Card Payment, Reservations etc. along with additional information like cost for 2 people, best days to visit, happy hours etc.

Combining these 3 feature lists and finding out the establishments a user may like is the purpose of this method. We propose a probabilistic based approach here where for each user we maintain an index of features and the probability that the user will like this feature. The probabilities are

derived from the user's past visits. The more number of visits a user has in the system, a better chance of correct recommendation.

- **Indexes**

a. User Index

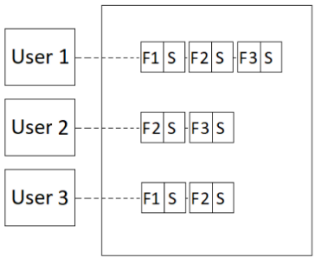

Figure 2: User Index

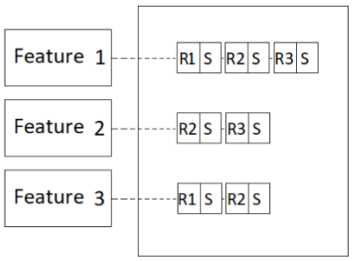

Figure 3: Feature Index

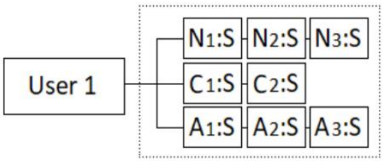

Figure 4: Feature

We create an inverted index of users and features where for each user we list features collected from all the restaurants he has visited in the past. These features are stored with a score which is the probability of user liking that feature. Also, since we want to make sure the type of features with less values don't over impose on the type with more values, we have segregated the user index into 3 parts as shown in Fig. 4.

b. Feature Index

When we get the features from users, we quickly scan the feature index and retrieve the restaurants which offer these features. The feature index is common to all users and the scores attached to these restaurants come from the feature score of the user at run time. A restaurant can be repeated across features and if that happens we sum up the probabilities.

Let's go through the algorithm now

- **Algorithm**

Step 1 : User Index Creation	**Step 2 : Feature Index Creation**
1. For each user U 2. Create neighborhood list N, category list C and attribute list A for U 3. For each restaurant R visited by user U 4. Retrieve neighborhood of R in list N 5. Retrieve category list of R in C 6. Retrieve attribute list of R in A 7. End for 8. Calculate probabilities of neighborhoods in list N 9. Calculate normalized probabilities of categories in list C 10. Calculate normalized probabilities of attributes in list A 11. End for	1. For each neighborhood n 2. Get restaurants with neighborhood n 3. End for 4. For each category c 5. Get restaurants with category c 6. End for 7. For each attribute a 8. Get restaurants with attribute a 9. End for 10. Merge lists of neighborhoods, categories and attributes list to form single feature index

Neighborhood probability : P(Neighborhood n | List N) = Count of n in N/Total count of N
Category probability : P (Category c | List C) = Count of c in C / Total count of C
Attribute probability : P (Attribute a | List A) = Count of a in A / Total count of A

Step 3 : User Recommendation

1. Get index list of user U
2. For neighborhoods N of user U
3. Get restaurant list R from feature index for feature N
4. Add restaurants from R to user recommendation list RL with score S of N
5. End for
6. For categories C of user U
7. Get restaurant list R from feature index for feature C
8. Add restaurants from R to user recommendation list RL with score S of C
9. End for
10. For Attributes A of user U
11. Get restaurant list R from feature index for feature A
12. Add restaurants from R to user recommendation list RL with score S of A
13. End for
14. Combine the entries in RL where R matches and add scores of those entries
15. Remove restaurants R from RL which user U has already visited
16. Sort RL in descending order of S
17. Recommend top k entries in RL

- **Method 2: Review Based Approach**
  - **Overview**

User review is the piece of text that tells us the most about user's likes and dislikes. Every user has patterns about his/her choices and discovering such patterns can be very informative and helpful in making business decisions. For this task, we operate on the Charlotte database. A recommender system requires user profile to provide recommendations. A user profile for this approach consists of the user query which can be generated using user's previous reviews. Generating user query at runtime can have high costs so it is imperative to make it an offline process.

- **Index**

For this approach, we created 2 indexes.

a. User Query Index

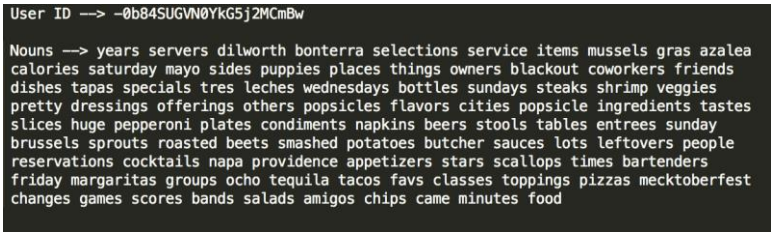

Figure 5: User Query Index

Using Stanford Core NLP, we extracted nouns from all the user's previous reviews to form the user query and created the index for faster access of the user query. To handle cold start

problem, if the index for a user does not exist, we assign some default keywords as query for that user.

b. Business Document Index

Figure 6: Business Document Index

We concatenated all the reviews for a business and created index as one document per business. The average rating denotes the sentiment toward the restaurant.

c. Test Data Index - This index is used to evaluate this methodology.

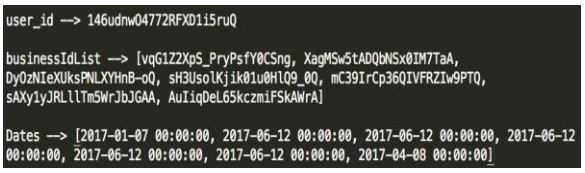

Figure 7: Test Data Index

- **Algorithm**

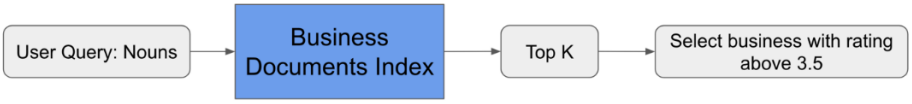

Figure 8: Algorithm Flow

1. We take the User ID and find the corresponding query from User Query Index.
2. Once we get the query, we find the top K documents from Business Document index using ClassicSimilarity and BM25Similaity, and recommend top 25 restaurants that have average rating above 3.5.
3. The average rating is used to quantify sentiment of the user regarding that a restaurant.

3. Evaluation

For evaluating our algorithm, we divided the dataset in training and testing data with a ration of approx. 3:1 The evaluation method we have selected is Mean Absolute Percentage Error.

- **Mean Absolute Percentage Error (MAPE)**

MAPE is a measure of evaluating accuracy in terms of percentage. It is especially useful when we need to evaluate predictions which are not on the same scale like in our case. We cannot always be sure the total reviews which we test our methods against are always the same for different

users. In this case it becomes important to bring them on same scale of evaluation and calculating error in terms of percentage does just that. We calculate the percentage of error for each prediction and take mean of those values to find out how much error prone our algorithm is.

$$\mathrm{M} = \frac{100}{\mathrm{n}} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

Figure 9: MAPE Formula[1]

In the formula A_t is the actual value while F_t is the predicted value. In our case actual value is the number of restaurants user visited in testing data and predicted value is the number of predictions we correctly guessed in our recommendations. The evaluation results using MAPE are as below for both the methods.

Method 1 Evaluation : We tested method 1 with both the datasets of Charlotte and Las Vegas and unfortunately the results we got were not very good. Below is the evaluation result for both the cities.

		Charlotte			Las Vegas		
Minimum Training	**Minimum Testing**	**Recommendations**	**Users**	**MAPE**	**Recommendations**	**Users**	**MAPE**
1	1	10	134	94.73	10	1052	98.31
		20		89.95	50		93.09
		25		87.6	100		88.03
10	5	10	63	95.59	10	794	98.29
		20		91.49	50		92.9
		25		89.12	100		87.69
25	10	10	34	95.38	10	396	98.45
		20		91.12	50		93.36
		25		88.18	100		88.55
45	20	10	15	95.76	10	232	98.63
		20		89.19	50		93.86
		25		85.15	100		89.81

Table 2 : Method 1 – Evaluation Results

Conclusions : From the above evaluation results, we can see that the results are very bad if we set the minimum review conditions to very low like 1 for each of them. This is an issue because it tries to even rank users which don't have sufficient data to be evaluated. Also, the results slowly start improving a bit once we set the minimum conditions a little higher. We can also see that the results improve with every increase in number of recommendations to user which is natural since the more chance we get to recommend options, higher the hit ratio. The results are not as promising as we expected but we feel that this method if given more time can be essential in finding patterns in users.

Method 2 Evaluation : As we mentioned we don't face a cold start problem for users in the IR approach since if we don't find a query for user we generate a default query based on most popular terms and provide recommendations. This gives us an advantage that it always gives

recommendations to users irrespective of the data it has on them or not. We cannot generate the evaluation for method 2 as we did for method 1 because it does not depend on the count of reviews and always gives recommendations for all users. We can only experiment with the k value of top k recommendations and we display the evaluation results below. Due to the POS approach of taking nouns from reviews, we unfortunately don't have the computational power to index data of Las Vegas, we only have evaluation results from Charlotte for method 2. We have also taken a similar evaluation from method 1 table and displayed below for comparison of 2 methods.

K Value	Method 1	Method 2
5	97.15	85.04
10	94.73	75.33
20	89.95	62.2
25	87.6	58.33

Table 3 : Comparison of Evaluation Results

Conclusions : As we can see, the results of IR approach are much more promising than Feature based approach. It proves the notion that a user review is much more informative on user likes and dislikes than the default features it likes. As in method 1 case, even in method 2 the accuracy goes on increasing as we increase the K value.

4. Limitations and Future Work

Over the course of this project we have faced several limitations which we would like to list down. Some of them are on our part which we feel we can improve given more time. We also mention some prospects of these solutions and the direction we would like to take them in.

- We could not run these methods on the entire data set due to computational limitations. We would have liked to run these methods on the entire dataset and not just 2 cities. It would have given us better insights on how the methods fare across entire US and Canada for which we had data available.
- In Method 1 we feel the next step should be trying to recognize patterns in the dataset where we are getting better results. We also would like to analyze the user data where our predictions were incorrect to identify what exactly went wrong.
- In Method 2 we have included all the nouns for now. But it's possible it may not be required and all we may have to do is extract nouns related to restaurant features and its descriptions. It would really help in terms of performance if we can single out those type of nouns.
- Also in Method 2 we would like to try and relate nouns to adjectives and adverb and find out the dependencies. This will help us include sentiment analysis of which features the restaurant is getting praised for and which feature the users don't like for a restaurant. It will really increase the effectiveness of the algorithms in predicting correct restaurants.
- In future we would like to combine both the methods to include default features and user mentioned nouns as features to provide a common index for both. This will try and include all possible domains at finding out the restaurants for recommendations.

II. Task 2 Predicting Rating from Review

1. Problem Statement

Almost all people prefer restaurants or hotels with higher rating which determines their choice most of the time. However, we know that not all user ratings are objective all the time. It is possible that a very positive review may come with a five-star rating while another similar one just has a three-star rating. Rating standards of people are different. This will in-turn adversely affect the overall restaurant or hotel rating. So, we can instead predict the rating for a review text if the customer's review deviates from the overall restaurant rating. The under underlying goal here is to attenuate the effect of subjective reviews by learning rating from a large number of examples.

There are two ways to analyse the subjectivity of rating from reviews and predict appropriate ratings:

- Sentiment Analysis of reviews
- Machine Learning prediction models

We are focusing on the second approach here

2. Proposed Solution

Let's go through each step in our solution one by one.

o Data Extraction and Data Pre-processing

The primary dataset used for analysis is the Yelp Dataset from the official challenge. There are multiple associated data files as part of the entire dataset. All the data is publicly available and can be downloaded from the official website. The data is in the form of SQL file and JSON file format. We considered the SQL file and used MySQL for all the manipulations. After carefully extracting all the data, it was filtered for Charlotte city. We consider business and review table here which were joined on Business ID field.

o Experiment Design

For feature selection, we use part-of-speech (openNLP in R) tagging to filter the keywords (i.e. nouns, advantages, nouns & adjectives) from reviews to obtain more informative features and compare how informative these features are for rating prediction. We only considered those nouns, adjectives, nouns & adjectives having frequencies greater than 50, 100, 50 respectively in all the reviews considered. We can consider higher frequency threshold of word to reduce the feature space, but it has some adverse effects.

We constructed a review-feature matrix based on different types of features extracted.

	able	amazing	attentive	awesome	bad	bests	betterr	big	busy	cool	decent	delicious	different	excellent	fantastic
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	1	0	0	0	0	0	0	0	0	0	0	2	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10: Review-Feature (Adjective) Matrix

We consider three hypotheses here:

1. Considering nouns as features give us the best accuracy for predicting the rating from reviews.
2. Considering adjectives as features give us the best accuracy for predicting the rating from reviews.
3. Considering nouns & adjectives as features give us the best accuracy for predicting the rating from reviews.

We even tried different lower frequency limit for words to be considered as features and checked which one gives us the best accuracy in prediction.

- **Modelling**

We considered different supervised machine learning models to predict the rating from review and tried to find out the one which gives us the best accuracy.

Regression models used are:

- Linear Regression
- Support Vector Machine for Regression (SVR)
- Random Forest
- Classification and Regression Trees (CART - ANOVA)

We have used R inbuilt function to run each of these models.

3. Evaluation

Since our goal is to predict star rating from reviews of users using regression models we have used RMSE as evaluation metric to quantify error instead of accuracy. We also implemented k-fold cross validation technique in order to check the decrease or increase in error rate for each fold of each model. Since we have 3 hypotheses we developed graphs to compare and analyze how the regression models behave for each hypothesis.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n}(y - y_j)^2}{n}}$$

Here y_j is predicted labels and y are actual labels.

1. Considering nouns as features for regression models the error rates got plot as follows

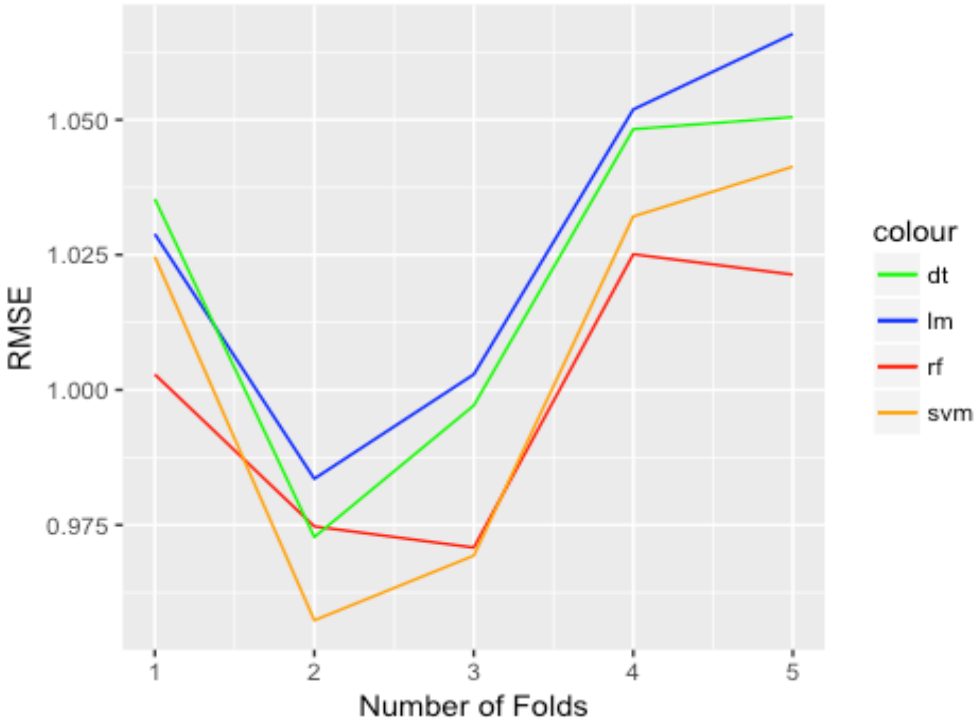

Figure 11: Error Rates - Nouns

2. Considering adjectives as features for regression models the error rates got plot as follows:

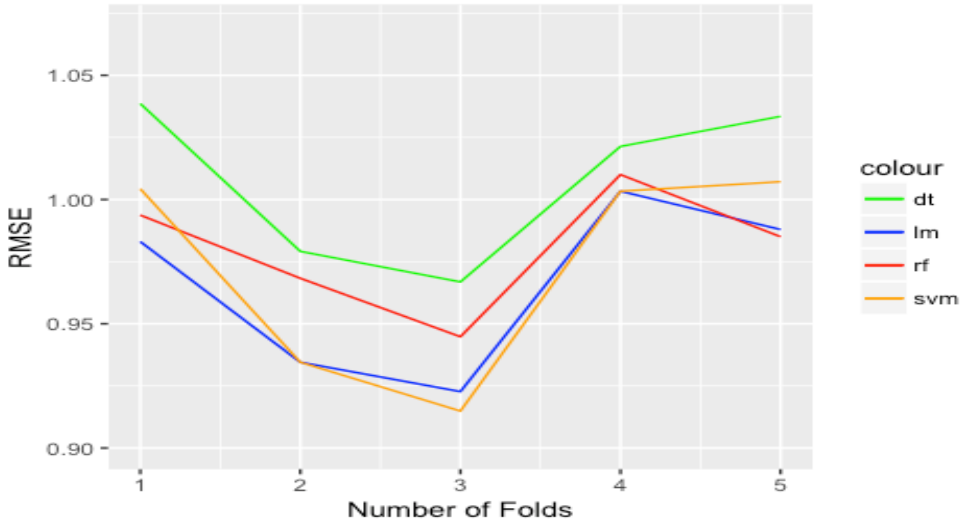

Figure 12: Error Rates - Adjectives

3. Considering adjective and noun combination as features for regression models the error rates got plot as follows:

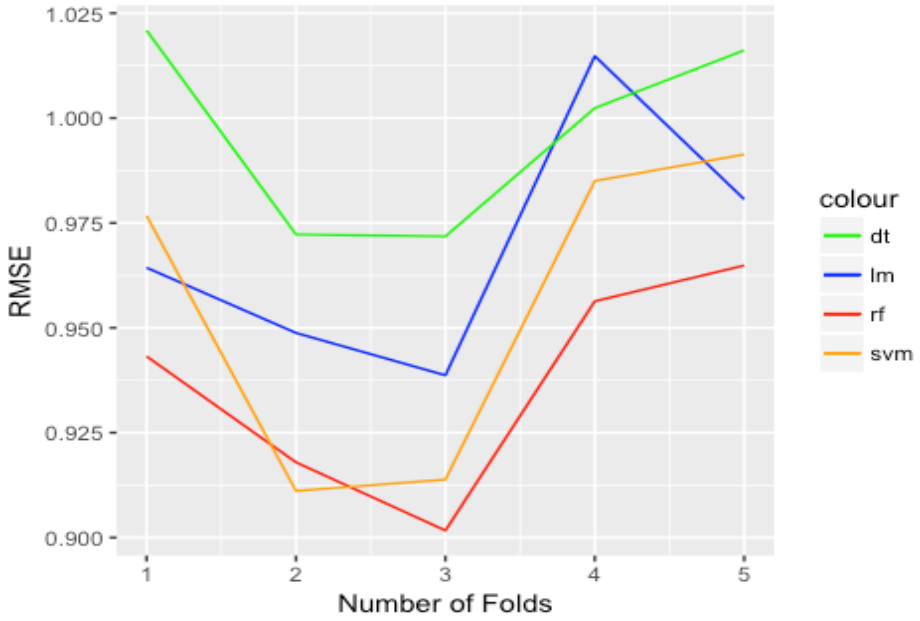

Figure 13: Error Rates – Nouns & Adjectives

We observed that for all cases decision trees performance was worst. Also, Support Vector Machine was notices as a better model especially for adjectives. For adjective noun combination, we observed two models performed good that is random forest and Support vector machine for regression.

4. Limitations and Future work

- The performance of the model appears to be good if we keep the frequency threshold between 50 to 100. If we increase the threshold to 200, the RMSE increases.
- About Space complexity, we only focused on restaurant category business reviews. Therefore, it reduced the total number of records to be processed in memory at each step.
- The data of Yelp is unbalanced dataset that is more reviews with rating 4 and 5 are present, so chance is there of our model predicting more number of 4 and 5 stars.
- Future work is the ratings predicted from reviews will help to have an unbiased overall rating of the business. Also, we have considered only single business category which can be extended to many like Fashion, Beauty etc.