Nightlife in Las Vegas

Junxia Zhu, Yiqun Jiang, Yingjing

April 3, 2019

1 Introduction

In this project, we analyse yelp data focusing on nightlife and bars in Las Vegas. We aim to explore which factors influence ratings of reviews and furthermore, give advice to owners for improving the ratings. Our work can be mainly divided into two parts: feature extraction from attributes and key words analysis from reviews. For the first part, we apply random forest algorithm and use ANOVA. For the latter, we use Python to do nature language provessing finding meaningful key words. We interpret the factors and make suggesstions due to outcomes. Besides, we also construct prediction models to achieve a "bonus" goal for predicting ratings.

2 Background

The Yelp data includes 4 json files: review_train.json, review_test.json, business_train.json, business_test.json, which contains 5364626 reviews, 1321274 reviews, 154606 businesses and 38000 businesses respectively.

3 Goal1--Analysis

3.1 Data Filtering

For the analysis part, we focus on the two train data files which has 5364626 reviews for 154606 business. To fit our thesis, we filter data by "bars", "nightlife" in categories and "Las Vagas" in city, and the total data involved in this part includes about 265847 reviews for 1201 businesses.

3.2 Business Analysis

3.2.1 Data Cleaning

1. There are nested dictionaries in business attributes, so we first extract all dictionaries as a new attribute list.

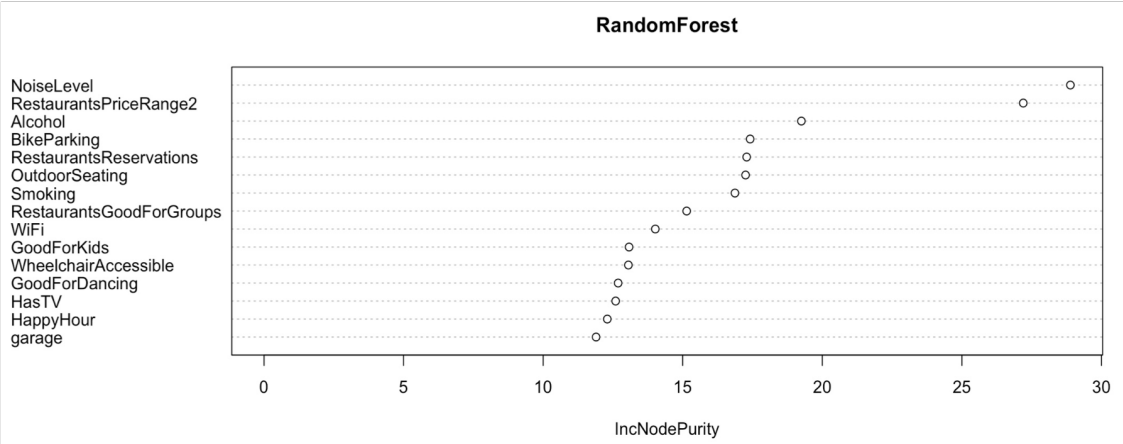
2. Then we calculate average star of all reviews for each business and set the average stars as our response variable.

3. One point needs to be metioned is that many attributes have missing values so we mark the blank with "unknown", which is treated as a new level.

3.2.2 Important Attributes Analysis

Here we use random forest computing variable importance scores to select useful attributes. "NoiseLevel" and "RestaurantPriceRange" are of the top importance, which means these two attributes are highly related to ratings.

```
In [29]: from IPython.display import display, Image
         display(Image(filename='rf.png'))
```



We do one-way ANOVA for both attributes, the model and outcomes are as below:

Model	P-value	result
stars~NoiseLevel	4.36*10e-8	reject H_0
stars~RestaurantPriceRange	2.44*10e-3	reject H_0

According to the results, we reject H_0--there is no differences between different levels of attributes, which means there does exist discrepancy between different levels of NoiseLevel and RestaurantPriceRange.

We also want to check interaction between these two attributes, so we construct full model with interaction term. The outcome is shown in below:

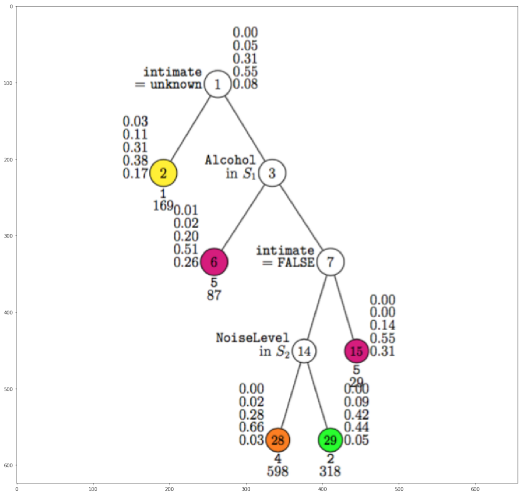
terms	P-value
NoiseLevel	3.02*e-8
RestaurantPriceRange	3.50*e-3
NoiseLevel*RestaurantPriceRange	3.58*e-2

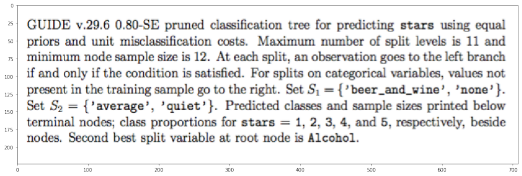
From the table above, we know that all terms in the full model are significant, so except for NoiseLevel and RestaurantPriceRange, their interaction also relates to ratings. Besides, we compare full model with all reduced models and find it had the lowest RSS, which means it is the best model. This result also testified that the outcome of two-way ANOVA is reliable.

3.2.3 Missing Value Analysis

From the random forest results, we can not identify the influence of missing values, so we apply decision tree method using GUIDE-a software for machine learning.

```
In [28]: import matplotlib.pyplot as plt; import matplotlib.image as mpimg; from matplotlib impo
         rcParams['figure.figsize'] = 40 ,40
         img_A = mpimg.imread('tree.png'); img_B = mpimg.imread('explain.png')
         fig, ax = plt.subplots(1,2)
         ax[0].imshow(img_A);ax[1].imshow(img_B);
```





The plot gives us insights about how missing value relates to ratings. We find missing value of intimate would somehow lead to low ratings and for alcohol, if it is "beer_and_wine" or "none", it tends to have high stars while it is "full_bars" or just NA, its ratings would depend more on NoiseLevel. When noise level is low, like "average" and "quiet", bars are likely to get high stars but when this attribute is missing or loud, they may get low stars.

3.3 Review Analysis

3.3.1 Data Cleaning

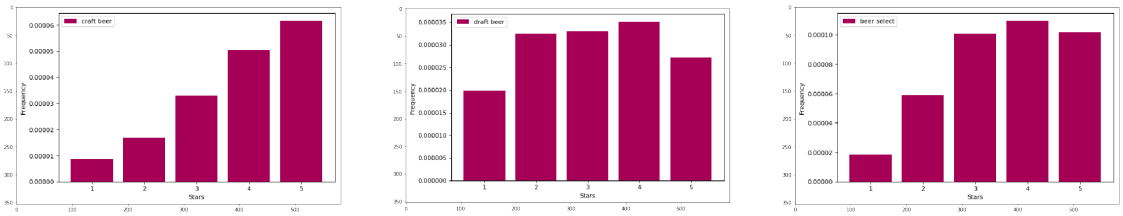
1. Tokenize each reviews, which means break paragraphs to sentences.
2. To deal with negative tone in reviews, we check each sentences. We transform those "n't" to not and then add "NOT_" to each word after words like "not" and "never" in those sentences. By the way in some .py scripts and .ipynb we use "addnot" instead of "NOT_" because some functions are not able to handle "_" punctuation.(Mainly in prediction part because of function "CountVectorizer()")
3. As what we care about is text, so we remove punctuations and meaningless symbols.
4. For each word, we do stemming which could avoid different word forms caused by tense, singular and plural. Then we are able to break each reviews to words and count frequency.
5. In the word lists, some words like "is", "the" actually make no sense so we construct a stop-word list to remove all these useless word.

3.3.2 Key Word Analysis

We first divide words into four aspects: food, beverage, entertainment and service, and then choose high-rate words from these four aspects respectively. Here, to deviod baseline differences' influence of star distributions, we divide word frequency by the number of words in reviews of each star level, which changes the frequency to rates. To be more specific for our word analysis, we pick words exclusive to bars like "beer", "cocktail", "strip", "casino", and plot histograms of each stars to see if there exists special patterns.

And when we try to interpret the histograms, we find the pattern for the word "beer" is really hard to explain since the rates of each star level are nearly the same. So upon previous unigram analysis, we develop bigrams to reach deeper exploration for beers. We find "beer selection", "draft beer" and "craft beer" are of top rates.

```
In [26]: %matplotlib inline
         rcParams['figure.figsize'] = 40 ,40
         img_A = mpimg.imread('craft_beer.png'); img_B = mpimg.imread('draft_beer.png'); img_C =
         fig, ax = plt.subplots(1,3)
         ax[0].imshow(img_A);ax[1].imshow(img_B);ax[2].imshow(img_C);
```



We plot the bigram frequency of each star level trying to find meaningful pattern, which are shown as above. It is clear that "craft beer" appears more frequently with higher star level, which fits our expectation since craft beer is generally more tasty than draft beer. But for "draft beer" and "beer select", there are not such clear pattern so we search for the adjective next to these bigrams to get more information to assist our analysis. And with analysis for the adjectives, we are able to give corresponding advice.

To attract adjective near specific terms, we use pos_tag(part-of-speech tagging) in NLTK to obtain adjective near "beer","cocktail","draft beer","beer select" and "craft beer". Here we only focus on general type of adjective instead of comparative and superlative. Then we manully classify them as "positive" or "negative" and drop those neutral adjective. The reason why we do this is because we don't want to waste time matching words with a large adjective dictionary where most of them don't appear in our review data. And we want to make sure every adjective near specific term is in our pos/neg corpus so that it will be counted. Because of our strict requirement(we search two words before term and two words after term and regard word as an adjective if it is in our pos/neg corpus) so we actually don't have very high occurance of adjective. Below is an example output for "beer" in star level 1:

top5 positive adj near 'beer' is ['great', 'good', 'warm', 'special', 'free']

top5 negative adj near 'beer' is ['cold', 'small', 'flat', 'damn', 'poor']

count of positive adj near 'beer' is 313

count of negative adj near 'beer' is 132

We also extract adjective based on every business id to give corresponding advice. To make our conclusions easily understandable, we made a shiny app to visualize our results. Here is the link for the shiny app: https://yingjingjiang.shinyapps.io/shiny_app/

4 Goal2--Prediction

we combine features extracted from both reviews and attributes. We make dataframe with columns of key words and attibutes and then construct several models to predict. For reviews, we apply TF-IDF algorithm to get frequency matrix and then combine the attributes columns to get our final data frame. After that, we try logistic regression and several machine learning methods to predict ratings. Our best prediction gives RMSE 0.825.

5 Conclusions and Our Advice

5.1 Conclusion:

We find NoiseLevel and RestaurantPriceRange are highly related to ratings in all attributes. For reviews, key words that can be used to improve bars' ratings are food-"salty", service-"rude" and "crowd", beverage-"beer" and "cocktail", entertainment-"casino" and "strip". And after deeper exploration of these words, we are able to give advice.

5.2 Advice:

1. For food, we recommend chef to ask customers when ordering about flavour to adjust salty level.

2. For service, bars better control attendences since crowded environment would give customers bad experience. And make sure waiters are well-trained and be nice to customers.

3. For entertainments, provide casino or strip could somehow help improve customers' satisfactory.

4. For beverage, which may be the most important part for bars, we would suggest that if possible, provide some craft beers but not only draft beer. And be careful to select beers.

5. Noise Level really influences bars' rating a lot, we understand that when people are high, they tend to make loud noises but if bars could make some control of that, it would be awesome.

6 Strength and Weakness

6.1 Strength:

1. Carefully deal with the reviews and take care of different baselines.

2. Use several methods to confirm our results, which makes our conclusion robust.
3. Consider NA as a level to see whether it makes differences but not simply omit them.

6.2 Weakness:

1. Fail to find patterns in time and hour analysis, which may need more careful inspectation.
2. Hard to interpret tree method outcome objectively more out of subjective analysis.

7 Contribution:

7.1 Goal1:

- Business Analysis:
  Data cleaning: Junxia Zhu, Yiqun Jiang, Yingjing Jiang
  Model: Yiqun Jiang, Yingjing Jiang
  Plots: Junxia Zhu

- Word Analysis:
  Data cleaning: Yiqun Jiang, Junxia Zhu
  Ngrams generating: Yiqun Jiang
  Adj analysis: Junxia Zhu
  Plots: Junxia Zhu
  Shiny app: Yingjing Jiang

7.2 Goal2:

- Kaggle: Junxia Zhu, Yiqun Jiang, Yingjing Jiang

8 Reference

[1] GUIDE Manual:http://www.stat.wisc.edu/~loh/treeprogs/guide/guideman.pdf