PREDICTING RATINGS USING YELP DATA

Nakul Thakare

1. Main Idea

Most consumers today check a company's rating before utilising its services. On Yelp, consumers write reviews and rate companies on a scale of one to five. There are over 100 million reviews and ratings of almost every type of local business, from restaurants, boutiques and salons to dentists, mechanics, plumbers and more. Economic research has shown that star ratings are so central to the Yelp experience that an extra half star allows restaurants to sell out 19% more frequently. Ratings and reviews directly influence the revenue and result in more visibility of the business. A one-star increase in Yelp rating leads to a 5-9 percent increase in revenue. (Luca,2016)

One issue with these ratings however, is that a Yelp star rating for a particular business is the mean of all star ratings given to that business. It is necessary to consider the implications of representing an entire business by a single star rating. What if one user cares about only food, but a particular restaurant's page has a 1-star rating with reviewers complaining about poor service that ruined their delicious meal? The user may likely continue to search for other restaurants, when the 1- star restaurant may have been ideal. Our aim is to first find the relevant factors that influence a business's rating. Using this, we train a model to predict users' rating of a business. The motivation includes that if we can predict how a user is going to rate a business, then we can recommend the business that the user is more likely to rate higher than the others.

2. Description of Data

The Yelp Dataset is publicly available on its website https://www.yelp.com/dataset
The dataset is a subset of Yelp's businesses, reviews, and user data. We chose to work with the Yelp dataset as compared to Amazon and Netflix since has more information so we could try more models. It covers 10 metropolitan areas, 188 thousand businesses and 5m reviews. We use two files, the business attributes and checkin file.

Business attributes contains 15 variables such as

Name	**Description**
business_id	ID for each business (primary key)
categories	Types of business, such as pub, bakery, spa etc
attributes	Features of the business such as ambience, wifi, parking etc
stars	review scores in the number of stars (5 max)
review_counts	The number of review counts by Yelp users
city,state,	Address of the business

Checkin file is a smaller dataset consisting of one additional variable:

Name	**Description**
time	Checkin day, time and count of each business

We use training set of 75% and test set of 25% of the cleaned data.

3. Methodology and Results

Data Preprocessing: We took several steps for this purpose as given below:

1) Getting rid of the data points with null values
2) Back-filling data wherever possible: wherever data for geographical coordinates was available, we could easily fill up the corresponding columns with the city and state
3) Getting rid of nonsensical values: negative star ratings can be found in several instances
4) We also removed the data for banks and all the entities other than restaurants from our dataset
5) We applied LabelEncoder from sklearn.preprocessing package on state and city columns to identify all unique strings with a unique integer value

Summary Statistics: The table below gives the average number of review counts for different star categories

Stars	Review Count
1.0	7.30
1.5	16.10
2.0	19.19
2.5	27.40
3.0	38.89
3.5	51.13
4.0	67.06
4.5	48.85
5.0	14.55

Higher the star rating of a restaurant, higher is the average number of reviews it has received on Yelp.

Our feature set consists of the following variables: [review count, checkin count, state, city]. The label set consists of the star ratings of all the restaurants. We built a

series of machine learning models for the purposes of labeling each restaurant with its star ratings.

The data is divided for both training and testing purposes. We run the data on three models:

Random Forest

Training Parameters: Number of trees in the forest, the maximum depth of the tree, the seed used by random number generator

Results:

Number of trees in the forest	Maximum depth of the tree	Accuracy
400	**8**	57.41%
500	**8**	58.41%
600	**8**	56.23%
500	**6**	54.28%
500	**10**	51.21%

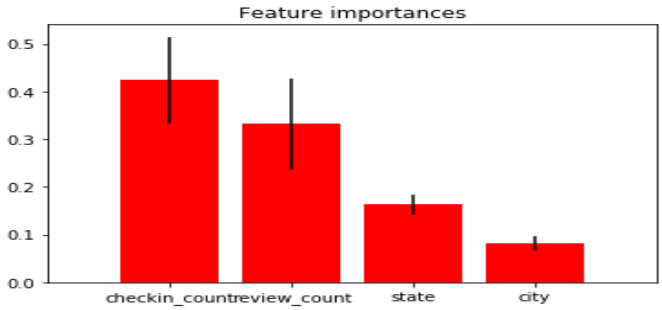
The best random forest model gave the following results:

Confusion Matrix:

	1	2	3	4	5
1	**0**	**2**	**0**	**167**	**27**
2	**0**	**2**	**0**	**3716**	**134**
3	**0**	**0**	**0**	**3693**	**81**
4	**0**	**2**	**0**	**15614**	**366**
5	**0**	**2**	**0**	**3288**	**507**

Accuracy %: 58.41%

Feature Ranking:



Discussion:

On keeping maximum depth constant, when the value of number of trees is low, accuracy achieved is good. If we increase value of number of trees up to 500, accuracy increases. However, beyond 500 the accuracy starts to decline

Logistic Regression

Training Parameters: Number of classes, tolerance for stopping criteria, algorithm to be used for optimization, maximum number of iterations taken for the solvers to converge

Number of classes can be either 'ovr' or 'multinomial'. If what is chosen is ovr, then the model does a binary classification.

Results:

Number of classes	Tolerance for stopping criteria	Algorithm to be used for optimization	Maximum number of iterations taken for the solvers to converge	Accuracy
5	1e-3	'newton-cg'	1842	51.19%
5	1e-3	'newton-cg'	1670	57.89%
5	1e-3	'newton-cg'	2000	55.23%

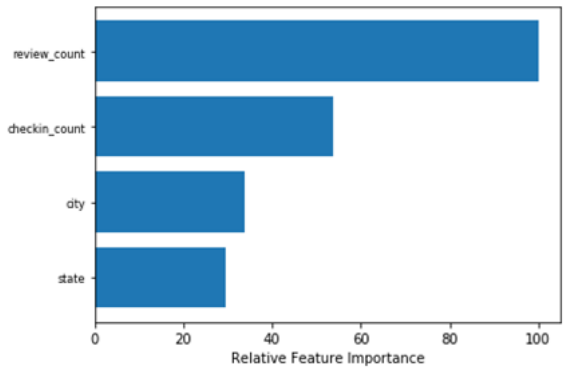
The best logistic regression model gave the following results:

Confusion Matrix:

	1	2	3	4	5
1	0	0	0	196	0
2	0	0	0	3852	0
3	0	0	0	3774	0
4	0	0	2	15980	0
5	0	0	0	3797	0

Accuracy %: 57.89%

Feature Ranking:



Discussion: We had a multiclass problem and hence used the 'newton-cg' optimizer. Accuracy for all different models we tried were lesser than the random forest model

k-Nearest Neighbors

Training Parameters: number of neighbors to use, algorithm to compute nearest neighbors, weight function used in prediction, power parameter for Minkowski metric (The Minkowski distance is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance)

Results:

Number of neighbors to use	Algorithm to compute nearest neighbors	Accuracy
3	**Ball-Tree**	45.33%
5	**Ball-Tree**	43.91%
3	**Brute**	43.89%
3	**KD-Tree**	44.2%

The best model gave the following results:

Confusion Matrix:

	1	2	3	4	5
1	**10**	**61**	**16**	**77**	**32**
2	**54**	**867**	**407**	**2208**	**316**
3	**31**	**726**	**432**	**2370**	**215**
4	**186**	**2747**	**1553**	**10484**	**1012**
5	**122**	**803**	**370**	**1783**	**719**

Accuracy %: 45.33%

Discussion: The best accuracy was for 3 nearest neighbors using Ball-Tree algorithm.

4. STARBUCKS MONTHLY RETURN PREDICTION

Can we use the ratings predicted by our models to predict the returns of big chains like Starbucks?

Starbucks, all over the country, received over 20,000 Yelp reviews from January, 2008 to December, 2017.

In order to predict returns, we take the following steps:

- Compute the monthly average for review counts
- Control for the effect of systematic effects from the returns of the restaurant
- X: 1 or 0 depending on whether or not the current month's number of reviews increased from the month before
- Y: 'cleaned' returns for the next month
- When we build a logistic regression model, we achieve accuracy rates of about 58%!

5. NEXT STEPS

Following are some of the steps which can be taken for research purposes in future:

1) Using lagged star ratings as an input X variable in our models
2) Building a full-blown trading strategy given that there is a relation between future returns and number of reviews (and hence star ratings) as we saw in the Starbucks example before