Business Data Analysis On Yelp Dataset: Feature Selection, Prediction, Sentiment Analysis

Nikhil Patil

May 22, 2017

1 Introduction

Big Data analysis could impact business decisions. Starting a new business always comes with challenges and risks of failure. Analysis of existing business data and using this analysis for taking decision for new business minimizes the failure chances. Analysis of business meta-data reveals details about trends in existing businesses in a given area and time period with current set of customers. This analysis helps the existing businesses to find important business features, customers, trends, business categories etc. Moreover, for a new business, the analysis might help to decide its portfolio that will fetch it higher overall ratings.

The objective of this project is to use Yelp dataset [1] to provide insightful analytics on existing business meta data helping the businesses, existing and upcoming, to make important decisions leading to higher ratings and more success in business. In particularly, we are doing business feature selection and rating prediction based on given existing features. Moreover, we also done the sentiments analysis to find the sentiments, associated with every business, that indicates their place in the view of their customers. Using sentiment analysis on every customers review, we found the top-n customers who are valuable, and creating positive popularity of the business. We also done analysis on the features businesses are using to attract more customers, from that we suggested features to new business, which are most frequently occurred features in all businesses. Finally, for visualization we see how the ratings vary as per the location of a business in a given city.

2 Problem Formulation and Description

With arrival of mobile application (Yelp, Foursquare, etc.) that allow users to quickly rate and review the business, particularly restaurants, the data collected by these applications gets richer and larger with time. Big data analysis gives a methodology to handle these massive data using Map Reduce techniques and extract the useful information out of it [2]. Particularly, we could use distributed pre-processing of the data to extract useful data that could be used for analysis using different appropriate machine learning techniques. We formulated the analysis into following sections viz., feature selection, ratings prediction based on different attributes of business, sentiment analysis of review comments posted by user, finding top-N users of business considering following parameters, star rating they have given to business, sentiment analysis on their review comments, the useful votes they received for their review comments, finding top-N categories of business and geo-spacial visualization. The Yelp dataset of businesses is voluminous. A solution to approach the problem of handling such massive data is to use big data techniques. The underlying framework that we use is **Hadoop MapReduce**.

2.1 Feature selection:

Yelp business dataset has large number of features in the `attributes` and `categories` columns. These features are mostly categorical with every business having some unique and some common features. So, in order to form a feature space for further analysis, we find the top N frequently occurring attributes in businesses and top N frequently occurring categories in business. These feature space information we used for the ratting prediction of business and attributes, categories suggestion for new businesses.

2.2 Ratings prediction:

Given a set of features per business, we train models on the training data and use this model to predict the star ratings of any business from the testing data. The models that we implemented are described in details in the methodology section. The feature spaces that we use for this analysis are: a) categorical features from earlier section b) counts based feature space (explained later in the 'Dataset' section)

2.3 Sentiment analysis:

Based on the set of reviews associated with every business_id (retrieved from the reviews dataset using Map Reduce), we find the overall sentiment for a given business following the sentiment analysis method described in the article [5]. This natural language processing method tells whether the reviews that a business is getting is either positive, negative, or neutral sentiment considering the user community reviews as a whole. This analysis helps the business owner in determining the good or bad reputation of business and whether they need to improve their service or features knowing their average review sentiment value between -1 to +1.

2.4 Finding top-N users:

In this section, firstly we find the list of top-N users that are important to a particular business in promoting their business and increase ratings. For every business, these users are ordered in terms of the rating that they gave to that business followed by the positiveness of the sentiment value of their review followed by the usefulness (useful votes) of the review that they received from other users.

Secondly, we process the Yelp users data to find the top-N users in the entire Yelp user community based on a simple formula that orders the users in terms of their importance, from which we selected top-N users. Both the methods are described in details in methodology section. These user are top important users from the business point of view, as they are are giving positive popularity to the business. This data can be used for implementing different marketing technique, to achieve more success in business.

2.5 Visualization:

The best way to see the trends in any dataset is visualization. Therefore, in this section we do the simplest form of visualization of the ratings data for the businesses in a given city. For this we use the Google Fusion Table API [11] to understand the trends of ratings with respect to geographical area where the business is located. Even in the most primitive form, this visualization reveals many insightful geographical trends.

3 Dataset

The dataset we are using for analysis is the **Yelp** Business dataset [1] freely provided for academic research on their website which has a total size of **5GB**. It covers the businesses from the major cities like Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland from USA and few more cities from other countries. The original data is subdivided into five different sub-datasets viz., business, review, user, check-in, and tip which are described in details below in terms of their characteristics. Our primary focus is to study the first four sub-datasets. Each of these sub-datasets is a JSON file with one JSON-object per line,which contain nested JSON arrays and objects.

3.1 What was challenging?

3.1.1 Preprocessing on JSON:

Initial challenge that we faced was parsing a JSON object inside a Mapper class using external jar files. We had to spend a some time in order to parse a JSON object into a mapper. For parsing we are using `json-20160212.jar` [10] library available, that we put in the distributed cache using the method `DistributedCache.addFileToClassPath` in the main class. Then, we added this external dependencies into class path using `DistributedCache.addFileToClassPath` method in the main class. This approach allowed us to read the JSON objects directly into the mapper on per line basis using external dependencies.

The other challenge we faced while parsing the JSON data, was handling the nested JSON array and JSON objects. The occurrence of nested JSON object in data set is inconsistent, so we implemented different checks and coding techniques in Java to parse data. This allowed us to filter the required features and extract useful information using n number of map-reduce jobs.

3.1.2 Huge feature space with categorical features:

Yelp dataset has numerous categories in the features like business attributes, categories, zip-codes, etc. These features are qualitative features that does not have a numerical value associated with them. For example, attribute `BusinessAcceptsCreditCards:True` does not have a numerical value but it has a categorical value as `True`. In order to convert these textual, categorical features into an appropriate numerical value we used the techniques like `LabelEncoder` or `OneHotEncoder` described in details in [6] [7].

Using LabelEncoder method is not a good practice [6] since it results in false correlation between the numerical value associated with the assigned label. Therefore, for the classification algorithms that we are using, it is a good practice to use OneHotEncoder to encode, for example, True as 1 and False as 0. These numerical labels (1 or 0) can then be processed by using the classifier algorithms.

3.1.3 Integration of sub-datasets:

Another challenge that we faced was the integration of four different sub-datasets using the n number of map-reduce jobs. For example, in one of the methodology we require the dataset in the form of `business_id, attributes_count, categories_count, check-in_counts, review_counts, stars`. To generate this sub-dataset, we had to run 3 map-reduce jobs. The first job was to operate on business dataset to get the first three features sorted by `business_id` as the primary key. The second job was to count the number of check ins for a particular `business_id`. Finally, another map-reduce job combined the outputs from the previous jobs with the same `business_id` as the primary key. Similarly, we had to integrate the business and review dataset for sentiment analysis. The main challenge for integration was the caching of files during map-reduce jobs to combine them based on the primary key.

Following sections describe the sub-datasets in details.

3.2 Business data:

This data has a rich set of features such as categories, attributes, location, etc.

```
{   "business_id":"encrypted business id",
    "name":"business name",
    "neighborhood":"hood name",
    "address":"full address",
    "city":"city",
    "state":"state -- if applicable --",
    "postal code":"postal code",
    "latitude":latitude,
    "longitude":longitude,
    "stars":star rating, rounded to half-stars,
    "review_count":number of reviews,
    "is_open":0/1 (closed/open),
    "attributes":["an array of strings: each array element is an attribute"],
    "categories":["an array of strings of business categories"],
    "hours":["an array of strings of business hours"],
    "type": "business" }
```

3.3 Review data:

It consist of a set of reviews associated with a business_id. We use these review for sentiment analysis for every business.

```
{   "review_id":"encrypted review id",
    "user_id":"encrypted user id",
    "business_id":"encrypted business id",
    "stars":star rating, rounded to half-stars,
    "date":"date formatted like 2009-12-19",
    "text":"review text",
    "useful":number of useful votes received,
    "funny":number of funny votes received,
    "cool": number of cool review votes received,
    "type": "review"}
```

3.4 User data:

We use this data to find the top N users in the Yelp community based on various features that they have.

```
{   "user_id":"encrypted user id",
    "name":"first name",
    "review_count":number of reviews,
    "yelping_since": date formatted like "2009-12-19",
    "friends":["an array of encrypted ids of friends"],
    "useful":"number of useful votes sent by the user",
    "funny":"number of funny votes sent by the user",
    "cool":"number of cool votes sent by the user",
    "fans":"number of fans the user has",
    "elite":["an array of years the user was elite"],
    "average_stars":floating point average like 4.31,
    "compliment_hot":number of hot compliments received by the user,
    "compliment_more":number of more compliments received by the user,
    "compliment_profile": number of profile compliments received by the user,
    "compliment_cute": number of cute compliments received by the user,
    "compliment_list": number of list compliments received by the user,
    "compliment_note": number of note compliments received by the user,
    "compliment_plain": number of plain compliments received by the user,
    "compliment_cool": number of cool compliments received by the user,
    "compliment_funny": number of funny compliments received by the user,
    "compliment_writer": number of writer compliments received by the user,
    "compliment_photos": number of photo compliments received by the user,
    "type":"user"}
```

4 Methodology: Description of Algorithms, System Architecture, and MapReduce Jobs

4.1 Features Selection:

We select the most frequently occurring features (from attributes and the categories column) considering the entire set of attributes from all the businesses. Knowing that these features are used frequently, we can create a feature space with these set of features. Although this method is purely based on frequency (and not on inverse-frequency), it gives a list of top-N features that are prevalent in the businesses. Therefore, incorporating these features in a business will be beneficial in that they will have the most desirable features.

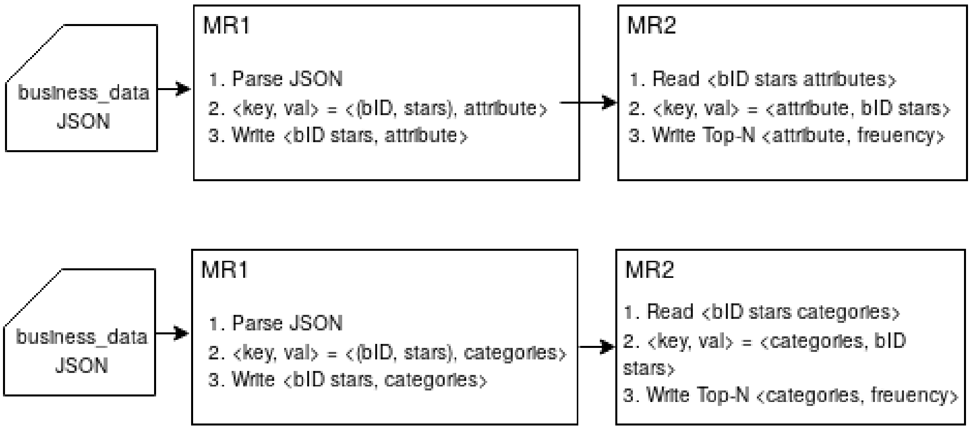
business_data JSON
MR1
1. Parse JSON
2. <key, val> = <(bID, stars), attribute>
3. Write <bID stars, attribute>
MR2
1. Read <bID stars attributes>
2. <key, val> = <attribute, bID stars>
3. Write Top-N <attribute, freuency>
business_data JSON
MR1
1. Parse JSON
2. <key, val> = <(bID, stars), categories>
3. Write <bID stars, categories>
MR2
1. Read <bID stars categories>
2. <key, val> = <categories, bID stars>
3. Write Top-N <categories, freuency>

Figure 1: Architecture of Top-N Feature Selection

`attributes` **features in business data:** These are a set of features (described below) that represent the attributes of a particular business such as 'business parking' along with their frequency. The following data shows the attribute and the count of businesses which have that attribute.

{ BusinessAcceptsCreditCards	103200;	BikeParking	51914
BusinessParking	49767;	RestaurantsTakeOut	45544
GoodForKids	44115;	RestaurantsGoodForGroups	40887
WheelchairAccessible	34799;	GoodForMeal	25093
RestaurantsTableService	24423;	Ambience	21965
HasTV	21233;	OutdoorSeating	19510
RestaurantsReservations	17378;	Caters	14938
ByAppointmentOnly	14201;	RestaurantsDelivery	10162
BestNights	5632;	HappyHour	5473
AcceptsInsurance	5255;	Music	4850
DogsAllowed	3089;	DriveThru	2154
GoodForDancing	1940;	HairSpecializesIn	1079
CoatCheck	1011;	RestaurantsCounterService	246
BusinessAcceptsBitcoin	178;	DietaryRestrictions	155
Corkage	140;	BYOB	47
Open24Hours	29}		

`Categories` **features in business data:** These features represent the categories in which the business falls, e.g., 'restaurant'. The following data shows the top 30 category and the count of businesses which have that category.

{ Restaurants	47588	Shopping	21442
Food	20718	Beauty & Spas	13322
Nightlife	10301	Home Services	9842
Health & Medical	9811	Bars	8932
Local Services	7265	Event Planning	6778
Active Life	6337	Fashion	5593
Automotive	5457	American (Traditional)	5269
Fast Food	5137	Sandwiches	5130
Pizza	5110	Coffee & Tea	5020
Hair Salons	4712	Arts & Entertainment	4439
Italian	4067	Home & Garden	3899
Burgers	3822	Mexican	3637
American (New)	3596	Chinese	3550
Breakfast & Brunch	3531	Nail Salons	3437
Doctors	3322	Specialty Food	3233}

The algorithm that we used is a simple sorting algorithm (embedded within the map-reduce job) that sorts all the features according to their frequency. Then we select top-N features with highest

Model	Feature Space (n_features)
Linear Regression	count-based (4)
SVM_rbf	categorical (31)
SVM_linear	categorical (31)
SGD	categorical (31)
SGD_kernel	categorical (31)

Table 1: Models with respective feature space

frequencies. Figure 1 shows the architecture of the map-reduce jobs for feature selection. As show, the first map-reduce job parses the JSON object for the business data to produce `<bID, stars features>`. The second map-reduce job reads this file to order the features by their frequency.

4.2 Ratings Predictions:

Ratings prediction is a methodology where based on the given set of training samples (i.e., features) and their respective labels (i.e., stars rating) we train a set of different models. Using these models we predicted the ratings of the test samples. We used two sets of feature spaces viz., categorical feature space and count-based feature space as described earlier. Following table gives the models that we trained and the feature space used in that model; these models are described in details in the following sub-sections. A general architecture of 'ratings prediction' methodology is shown in 2. As shown, we used map-reduce jobs to create features such as review counts, check-in counts, attribute counts, category counts and stars for each business. that were arranged based on business_id as the key. Using these features, we trained the machine learning models in Python with the help of the standard machine learning package `scikit-learn` [4] since it has stable and optimized algorithms.

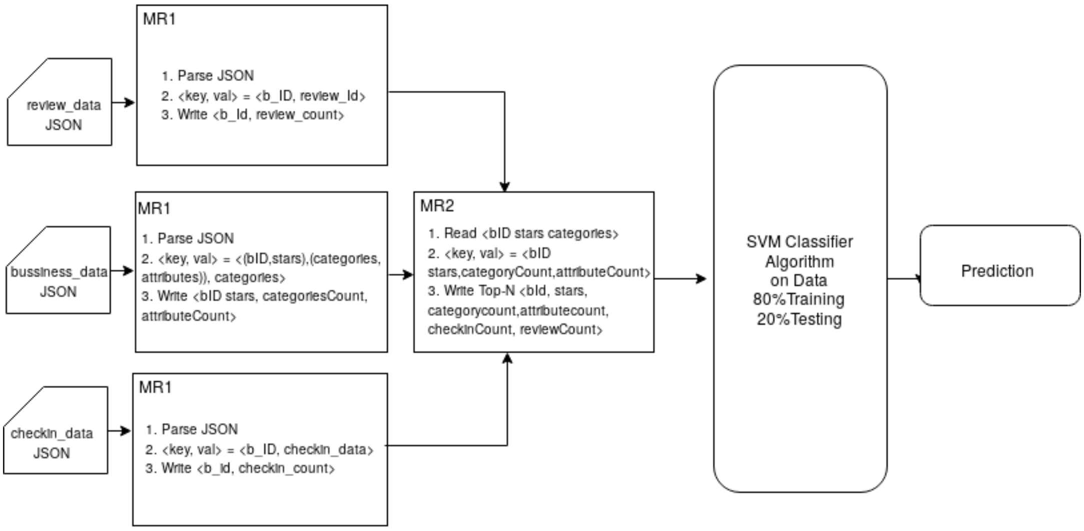
review_data
JSON
MR1
1. Parse JSON
2. <key, val> = <b_ID, review_Id>
3. Write <b_Id, review_count>
bussiness_data
JSON
MR1
1. Parse JSON
2. <key, val> = <(bID,stars),(categories, attributes)), categories>
3. Write <bID stars, categoriesCount, attributeCount>
checkin_data
JSON
MR1
1. Parse JSON
2. <key, val> = <b_ID, checkin_data>
3. Write <b_Id, checkin_count>
MR2
1. Read <bID stars categories>
2. <key, val> = <bID stars,categoryCount,attributeCount>
3. Write Top-N <bId, stars, categorycount,attributecount, checkinCount, reviewCount>
SVM Classifier
Algorithm
on Data
80%Training
20%Testing
Prediction

Figure 2: Architecture of Ratings Prediction

4.2.1 Linear Regression:

We initially formulated this as a simple linear regression problem on the count-based feature space. However, given the discrete nature of the labels, the model did not work properly. For example, consider the plot shown in ??. The graph shows the labels (stars) versus the check-in counts of a business. We fit a linear model through 4 such count-based features. Given the discrete nature of output labels, the model fails to predict the labels correctly with a 'score' of approximately 0 (0.0078). Therefore, we decided to model the problem of ratings prediction as a classification problem using linear as well as nonlinear classifier from `sk-learn`.

4.2.2 Classification:

We used multiple set of models on the categorical feature space, viz., Support Vector Machine (SVM) with linear kernel, SVM RBF kernel, Stochastic Gradient Descent (SGD), SGD with kernel. We formulated the classification problem as a multi-class classification problem with 11 output classes corresponding to the 11 values of ratings viz., `[0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5]`.The results (discussed in evaluation) were better as compared to linear regression. However, they were not up to the mark. The classifiers work well for few of the classes, for example class 8 class 10. This suggest that we need to extract better features from the datasets.

4.3 Sentiment Analysis on Reviews:

In order to understand the overall sentiment about every business in the dataset, we parse the 'user review dataset' JSON file to get the individual review. The architecture of the map-reduce jobs is given in figure 3. At the end of the first map-reduce job, we get the data in the format `<reviewID, stars [review text]>`. We apply sentiment analysis [5] on each review i.e., for all the review_id. Finally, we integrated the individual user reviews with their sentiment value with the 'business dataset' using 'business_id' as the primary key. Then, we find the mean value of the sentiment for each business based on the sentiments of all the review for that business.

We use two text files `pos-words.txt` and `neg-words.txt` [5] to get a set of positive and negative words. Then, we find the total number of positive and negative words that are mentioned in a review. Now, using following formula described in [5] we calculate the total sentiment value associated with the review.

$$sentimentValue = (nPositive - nNegative)/(nPositive + nNegative) \tag{1}$$

Algorithm 1 sentiment analysis

```
procedure SentimentCalculator($reviewtext$) ▷ returns sentiment value between (-1, +1)
2
    while $EndOfReview$ do                          ▷ for every word in the review
        if Word Present In pos-words.txt then
        $nPositive \leftarrow nPositive + 1$
        if Word Present In neg-words.txt then
        $nNegative \leftarrow nNegative + 1$
    $sentimentVal \leftarrow (nPos - nNeg)/(nPos + nNeg)$
    return $sentimentVal$                           ▷ .
```

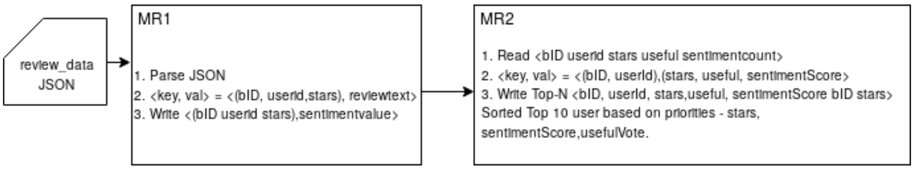
review_data
JSON
MR1
1. Parse JSON
2. <key, val> = <(bID, userid,stars), reviewtext>
3. Write <(bID userid stars),sentimentvalue>
MR2
1. Read <bID userid stars useful sentimentcount>
2. <key, val> = <(bID, userId),(stars, useful, sentimentScore>
3. Write Top-N <bID, userId, stars,useful, sentimentScore bID stars>
Sorted Top 10 user based on priorities - stars,
sentimentScore,usefulVote.

Figure 3: Architecture of Sentiment Analysis

Table 2 gives a list of 3 businesses and their 3 different users and sentiment value business received from respective user comment.

business_id	user_id	Sentiment Value
bid1	uid1	+1
bid2	uid2	+0.8666667
bid3	uid3	+0.71428573

Table 2: example mean sentiment value per business

4.4 Finding top-N business promoting users:

In this section we find the top-10 users that promoted a given business by providing positive sentiment reviews and higher stars to business as well as their review have higher useful vote by other users. The priority used while sorting and selecting these users is the star rating, sentiment score, and useful vote respectively

These user are top important users from the business point of view, as they are are giving positive popularity to the business. This data can be used for implementing different marketing technique, to achieve more success in business.

4.5 Visualization

Visualization facilitates a better understanding of the trends in a data. In order to understand how the stars are related to the location within a city. For instance, as we observe in the 4, Old Toronto (lower red blob) and Yonge & Eglinton (upper red blob) regions in the Toronto has more popular and highly rated (close to 5) businesses depicted by the red color in the heatmap. So these kinds of map help to decide what are the probable locations within a city where a business will get higher ratings assuming the precondition that business provides good service in terms of its features.

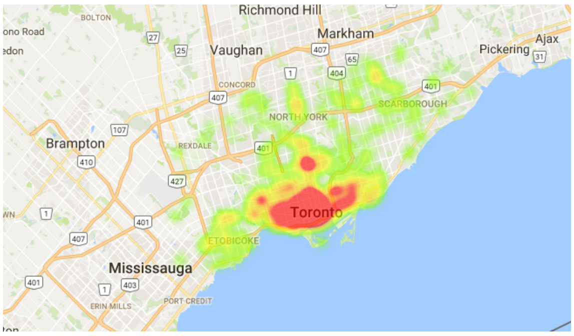

Figure 4: Stars variation in Toronto

5 Discussion of Analysis and Conclusions

5.1 Evaluations:

5.1.1 Evaluation of Classification Algorithms:

The dataset we use on categorical feature space (i.e., `attributes`) has 144072 samples (i.e, `business_id`). We divide the dataset into **80-20 train-test** split. Ground truth is the available ratings in terms of stars (out of 5). For the purpose of classification we convert the class values from floating values to integer values by multiplying by 2. Therefore classes `[0,1,2,3,4,5,6,7,8,9,10]` represent the actual rating of `[0,0.5,1,1.5,2,2.5,3,3.5,4,4.5,5]`.

Table 3 shows the **Precision, Recall, and f1 scores** scores for the classification methods that were described in the methodology section. The precision and recall should be high as possible and close to +1. The confusion matrices for SVM_linear, SVM_RBF, SGD, and SGD_kernel, are shown in the figure. The scale represents the number of business. So diagonal elements are the correctly classified businesses and the non-diagonal elements are either false positives or false negatives. As we can see, the classification methods performed relatively well for the final class (class 10 = 5 stars). However, their performance deteriorates for lower classes.

Model	Mean Precision	Mean Recall	Mean F1 score
SVM_linear (C=1)	0.24	0.26	0.19
SVM_rbf (C=1,g=1/31)	0.19	0.25	0.18
SVM_rbf (C=1000,g=0.3)	0.23	0.25	0.20
SGD	0.21	0.20	0.12

Table 3: Precision and Recall score for the models

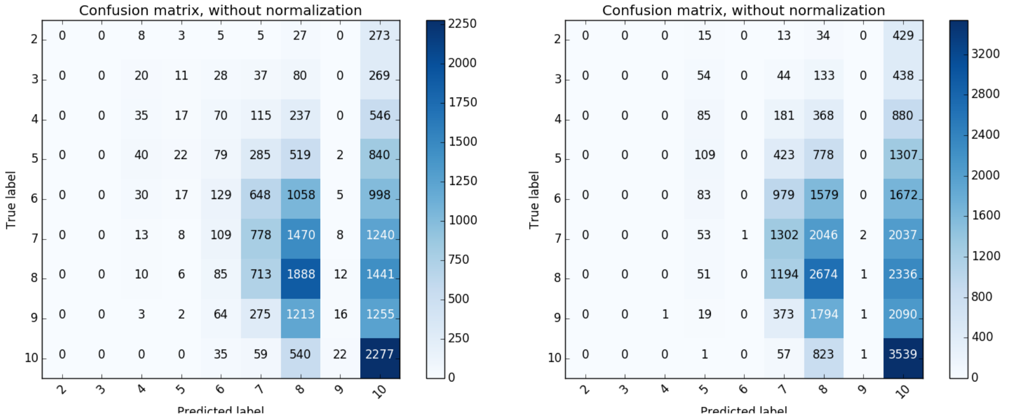

Figure 5: Confusion Matrix: SVM_linear (left) and SVM_RBF (right)

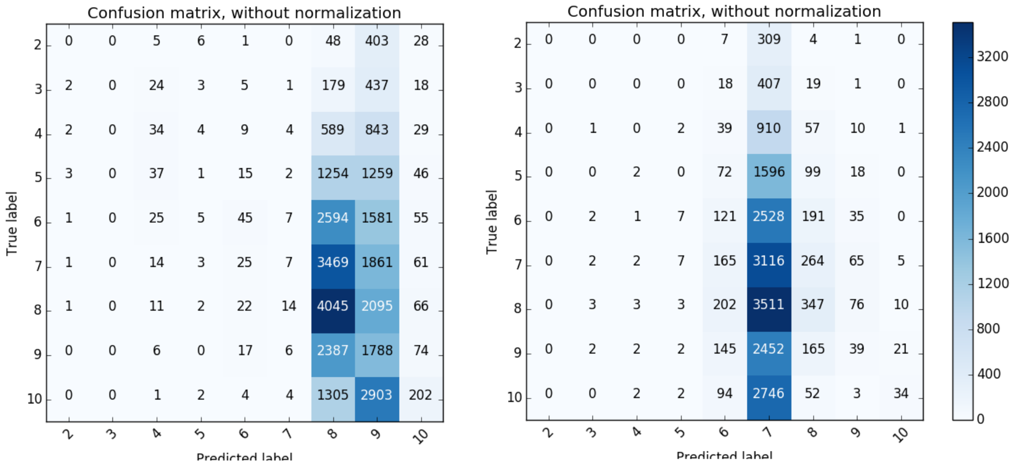

Figure 6: Confusion Matrix: SGD (left) and SGD_kernel (right)

5.2 Conclusions:

Yelp data can be mined using a large variety of approaches and methods. We initially performed feature selection based on a frequency of occurrence approach on the attributes and categories in the business dataset giving us 31 attribute features and 30 category features. We used these features to predict the star ratings on the test dataset. We used regression approach that did not work well. Classification models like SVM, SGD worked better than linear regression. However, the results suggest that these approaches must be tweaked or modified in order to get better results on our feature space.

Finally, we performed sentiment analysis on all the reviews received by every business. The sentiment values showed that higher star correspond to higher and positive sentiment value while the lower stars correspond to lower and negative sentiment value. The heatmap visualization facilitates to understand what regions in a city could fetch higher star ratings.

6 Project Timeline

- **Preprocessing:** Collecting, pruning, and pre-processing the dataset to suit our analysis. Set up the Hadoop clusters according to our dataset.
- **Feature selection:** Find important or frequently occurring features for a particular business. Extract important features from dataset using Map Reduce.
- **Ratings Prediction: :** Train machine learning models to predict the rating of businesses
- **Sentiment analysis:** Design and Run algorithm for sentiment analysis. Top N user determination based on sentiment analysis on review comments.
- **Visualization** Geo-spacial heatmap visualization of the ratings data .

References

[1] Yelp Dataset https://www.yelp.com/dataset_challenge

[2] Anand Rajaraman, Jure Leskovec, and Jeffrey Ullman,"Mining of Massive Dataset", Cambridge University Press, 2012

[3] Bryan Hood, Victor Hwang, Jennifer King, 'Inferring Future Business Attention'

[4] http://scikit-learn.org/stable/tutorial/machine_learning_map/

[5] https://www.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_example_4_sentiment_analysis.html

[6] http://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features

[7] https://www.kaggle.com/c/titanic/discussion/5379

[8] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[9] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[10] https://mvnrepository.com/artifact/org.json/json/20160212

[11] https://support.google.com/fusiontables/answer/1152262?hl=en