Yelp Dataset Challenge: Analysis of Rating Prediction from User Review

By Chiranjib Ghorai
655493675

Final Project Report
IDS 561: Big Data Analytics
University of Illinois at Chicago

Table of Contents

1. Introduction

Yelp is a popular business review and social network website. Local businesses can register on Yelp to increase their social presence. Any registered user can rate the business on a scale of 1 to 5 stars and write reviews of the services or products offered. Other registered users can view the reviews, comment on the reviews and rate the reviews as *useful*, *funny* and *cool* on a binary system. These online reviews function as the 'online word-of-mouth' advertisement and consumers use them as criteria to choose between similar services or products. The review and star rating play an important role in consumer behavior and decisions.

Because the nature of business of Yelp is such, a robust rating and recommendation system is key to their success. However, the relationship between the review text and the rating is not obvious because ratings are objective. It has often happened that multiple reviewers have written similar positive review but have rated it differently. A very positive review may come with a five-star rating while another similar review with a four-star rating. E.g.

An auto services is reviewed as- *"Mike"s auto body is fantastic. Not only do the do great work, but the experience is great to. The shop is clean and comfortable. They have fantastic customer service."* is rated five-star by one consumer where as a very similar review by another consumer on same service provider - *"Blown away at how Mike's turned an awful situation into a very unexpectedly pleasant experience! The customer service is stellar. The staff at Mike's genuinely love their jobs and it's evident pleasing their customers is priority one. I'll def be back to Mike's!"* is rated as four star.

Yelp has thousands of reviews, which make it very difficult for readers to go through each of them. The readers tend to just see the star rating rather than reading the whole review. Yelp must analyze and understand the sentiment of each review for the local businesses by mining the text to develop a better rating prediction system.

2. Project Goal

The goal of this project is to apply existing supervised learning algorithms to execute Review Rating Prediction on a scale of one to five based only on the review text. Review Rating Prediction can be defined as following:

Given the set $S = \{(r_1, s_1),\ldots,(r_N , s_N)\}$ for a product P, where r_i is the i'th text review of P and s_i is the i'th numeric rating for P, the goal is to learn the best algorithm for mapping a word vector r with numeric rating s.

In this paper, Review Rating Prediction Problem is treated as a multi-class classification problem in Machine Learning. Star ratings are the class labels and reviews are input vectors. I have combined two feature extraction methods – unigrams and bigrams with three supervised learning algorithms- Decision Tree, Random Forest and Naive Bayes to build six prediction models. The dataset provided by Yelp is trained on each of these models and evaluated on the performance based on the Root Mean Squared Error (RMSE). RMSE represents the sample standard deviation of the differences between predicted rating and actual ratings. RMSE can be calculated as:

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Where,

Y_i = Observed value
ˆY_i = Predicted value
n = number of observation

Each model is evaluated for all four different types of businesses – Restaurants, General Services, Travel & Hotels and Shopping.

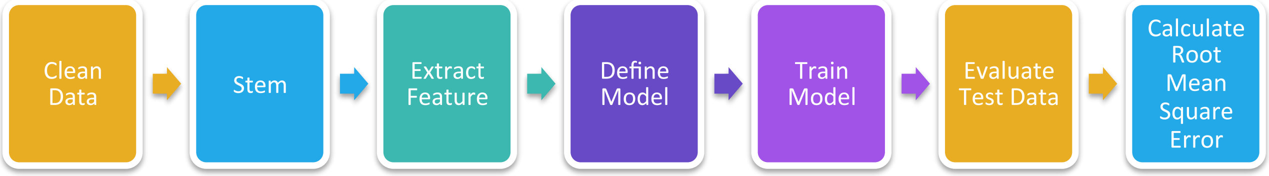
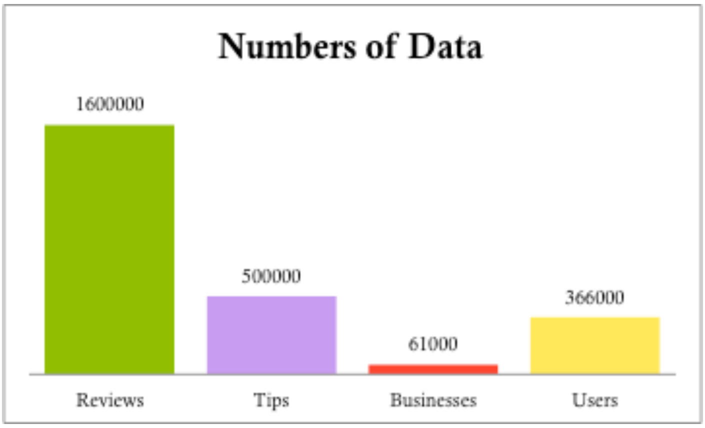


Fig 2.1 overall process of analysis

3. Dataset and Data Processing

The dataset is provided freely upon registration by Yelp as part of their Dataset Challenge 2015 (Link) for training and testing the machine learning models. The Yelp Dataset has 1.6M reviews and 500K tips by 366K users for 61K businesses with 481K business attributes.



The dataset consists of five files, one for each object type: business, review, user, check-in and tip. Each file consists of one json object per line. Thus, a business is represented in the 'business.json' and a review in 'review.json'. For the scope of this project, only business and review objects are considered for analysis, as the rest of the data are irrelevant for the model defined.

Business Objects: Business objects contain basic information about local businesses. The fields are as follows:

```
{
  'type': 'business',
  'business_id': (a unique identifier for this business),
  'name': (the full business name),
  'neighborhoods': (a list of neighborhood names, might be empty),
  'full_address': (localized address),
  'city': (city),
  'state': (state),
  'latitude': (latitude),
  'longitude': (longitude),
  'stars': (star rating, rounded to half-stars),
  'review_count': (review count),
  'photo_url': (photo url),
  'categories': [(localized category names)]
  'open': (is the business still open for business?),
  'schools': (nearby universities),
  'url': (yelp url)
}
```

Review Objects: Review objects contain the review text, the star rating, and information on votes on the review. business_id attribute is used to associate this review with others of the same business in the business data.

```
{
  'type': 'review',
  'business_id': (the identifier of the reviewed business),
  'user_id': (the identifier of the authoring user),
  'stars': (star rating, integer 1-5),
  'text': (review text),
  'date': (date, formatted like '2011-04-19'),
  'votes': {
    'useful': (count of useful votes),
    'funny': (count of funny votes),
    'cool': (count of cool votes)
  }
}
```

4. Pre Processing

The given data is converted from JSON to CSV format using the script provided by Yelp dataset challenge Github.

4.1. Categorization and Stemming

The CSV files are split based on four categories: **Restaurant, General Services, Travel\Hotels** and **Shopping** by merging *Review* and *Business* Objects on the *type* and *business_id* fields. The reason for doing so is that the text reviews for different business categories may be very different. For example, a typical hotel review may contain the words/phrases such as ‘bed’, ‘elevator’ and ‘fridge’, but these words would not occur in a restaurant review. So each model is trained individually on each category to perform Review Rating Prediction for each business category independently.

Each of the dataset is processed using PorterStemmer to normalize by removing unwanted words, letters and punctuations before executing feature extraction.

4.2. Feature Extraction

4.2.1 Unigrams

In the uni-grams of bag-of-words model each unique word in the pre-processed review is considered as a feature. This makes feature vector for a review is straightforward affair. Steps to create unigram –

1. Each text is stemmed using PorterStemmer method.
2. Dictionary of all the words occurring in the review corpus is created.
3. Word-review matrix M(i, j) is constructed, where (i, j) is the frequency of occurrence of word i in the j'th review.
4. Feature matrix is constructed by applying *Term Frequency - Inverse Document Frequency* (TF_IDF) weighting technique. This weighting technique assigns less weight to words that occur more frequently across reviews (e.g. "service" or "food"). These high frequency words are not significant to distinguish between different reviews. In TF-IDF, a word with lower frequency is given higher significance. The four datasets, each corresponding to the business types is processed using *DictionaryVectorizer* and *TFIDFConverter* class of apache Mahout library.

4.2.2 Unigrams and Bigrams

The primary drawback of unigram is lack of ability in capturing relationships between two inter-linked words such as "not bad" or "great disaster" because it treats each word individually. Bigrams are used to capture the effect of such phrases where one word may negate the effect of next word. In this method, the dictionary additionally consists of all pairs of consecutive words occurring in the reviews. The matrix is computed as explained in part 4.1.1 but now has more rows. Similar to previous method, TF-IDF weighting is applied to this matrix to give more importance to rare words and more importance to words/phrases higher frequency.

5. Naïve Bayes

Naïve Bayes is a popular algorithm for text classification. Naive Bayes algorithm in Apache Mahout library is used as one of the machine learning algorithms to predict star rating from reviews. The algorithm can be defined as follows –

Lets consider a multinomial $(p_1...p_n)$ where pi is the probability of occurring of event. In our case it is rating 1 to 5. Feature vector is $X = (x_i,...,x_n)$ where xi represents the occurrences of a word. The likelihood of observing a histogram x is given by:

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

The Review texts are referred to as the features and the rating is referred to as the classes. Following tables shows RMSE analysis of Naïve Bayes executed on Unigram and Unigram &Bigram.

Unigram	
Category	RMSE
Restaurant	1.849686294
Travel and Hotels	1.815248516
Shops	1.97117907
General Services	2.061158615

Unigram and bigram	
Category	RMSE
Restaurant	1.803722076
Travel and Hotels	1.814674701
Shops	1.887892253
General Services	2.06464919

Table 5.1 Room Mean Squared Error for each category using Naïve Bayes model and unigram/bigram feature extraction

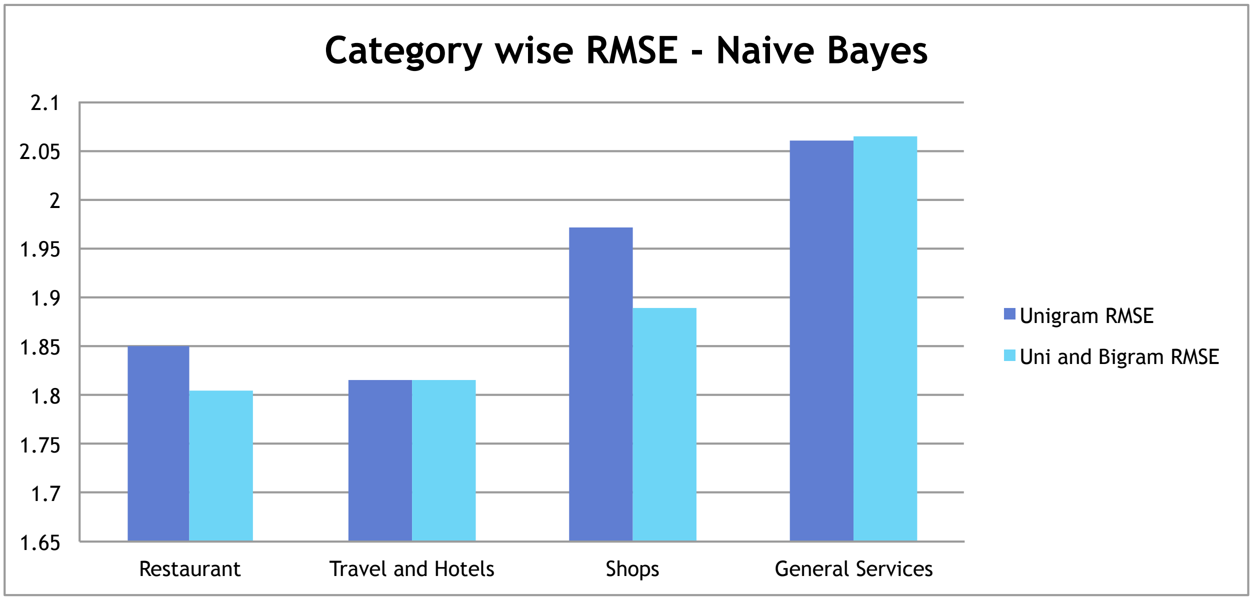


Fig 5.1 Chart of comparison - RMSE for each category using Naïve Bayes model and unigram/bigram feature extraction

6. Decision Tree

Decision Tree Classifier is a simple and widely used classification technique. Decision Tree Classifier applies a straightforward idea of posing a series of questions about the attributes of the test record. Each time an answer is received, a new possible question is chosen by selecting best split from the set of possible follow-up questions. This process recursively continues until a conclusion is achieved. Decision trees can handle categorical features in the multiclass classification setting. Variance impurity Decision Tree is used for this analysis. The algorithm is given below (source):

The variance reduction of a node N is defined as the total reduction of the variance of the target variable x due to the split at this node:

$$I_V(N) = \frac{1}{|S|}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(x_i-x_j)^2 - \left(\frac{1}{|S_t|}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(x_i-x_j)^2 + \frac{1}{|S_f|}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(x_i-x_j)^2\right)$$

S = set of presplit sample indices
S_t = set of sample indices for which the split test is true
S_f = set of sample indices for which the split test is false

Apache Spark MLlib library is used for training and testing the datasets on decision tree algorithm. The Spark MLlib library supports decision trees for binary and multiclass classification and for regression, using both continuous and categorical features.

Unigram	
Category	RMSE
Restaurant	1.304743138
Travel and Hotels	1.327311412
Shops	1.384391139
General Services	1.637388983

Unigram and bigram	
Category	RMSE
Restaurant	1.303728131
Travel and Hotels	1.326276377
Shops	1.409344061
General Services	1.643195012

Table 5.1 Room Mean Squared Error for each category using decision tree model and unigram/bigram feature extraction

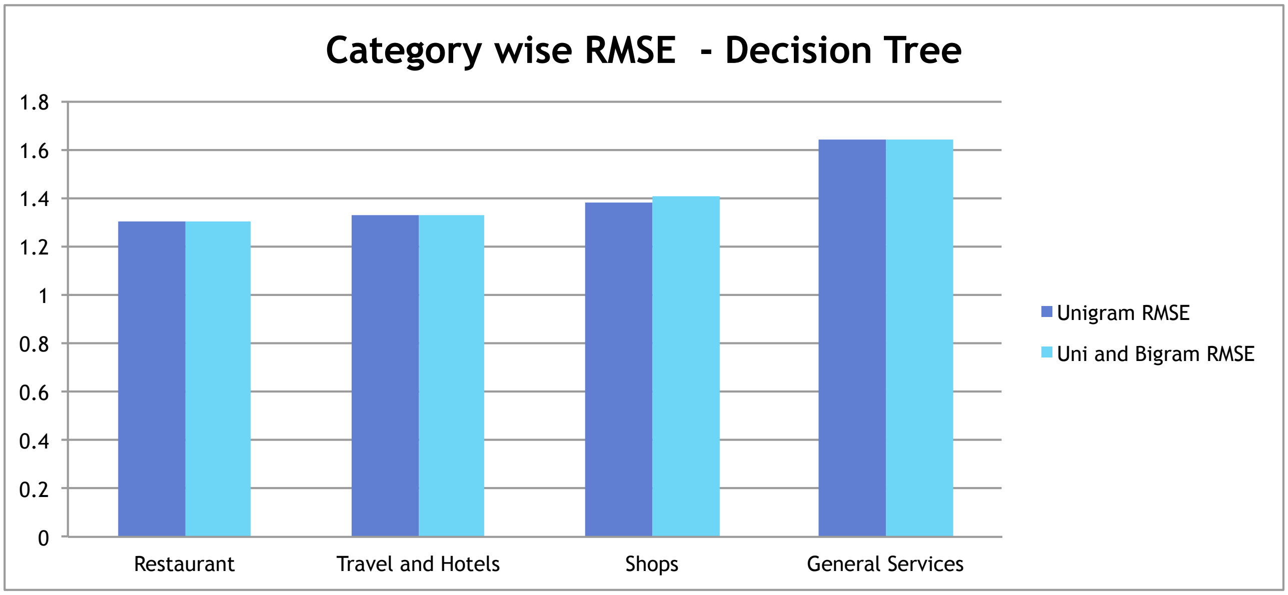


Fig 6.1 Chart of comparison - RMSE for each category using decision tree model and unigram/bigram feature extraction

7. Random Forest

Random forests are a collection of decision trees. They are one of the most popular machine learning models for classification and regression. They combine many decision trees in order to reduce the risk of over fitting. Random forests do not require feature scaling, and are able to capture nonlinearities. Multiclass classification random forest of Apache Spark MLlib library is used for training and testing Yelp dataset. MLlib implements random forests using the existing decision tree implementation. In random forests algorithm, a set of decision tree aimed separately and can be done in parallel. Each tree is created differently because of randomness. Following algorithm shows how such a system is trained (source):

For P number of trees T:

1. Sample N cases at random with replacement to create a subset of the data. The subset should be about 66% of the total set.
2. At each node: For some number m, m predictor variables are selected at random from all the predictor variables.
3. The variable that provides the best split, according to classification or regression, is used to do a binary split on that node.
4. Choose another m variables at random from all predictor variables and do the same.

Unigram	
Category	RMSE
Restaurant	1.6964369065
Travel and Hotels	1.7537580318
Shops	1.9100110934
General Services	2.6388165794

Unigram and bigram	
Category	RMSE
Restaurant	1.3655873773
Travel and Hotels	1.7519434889
Shops	1.8998295271
General Services	2.0378662293

Table 5.1 Room Mean Squared Error for each category using random forest model and unigram/bigram feature extraction

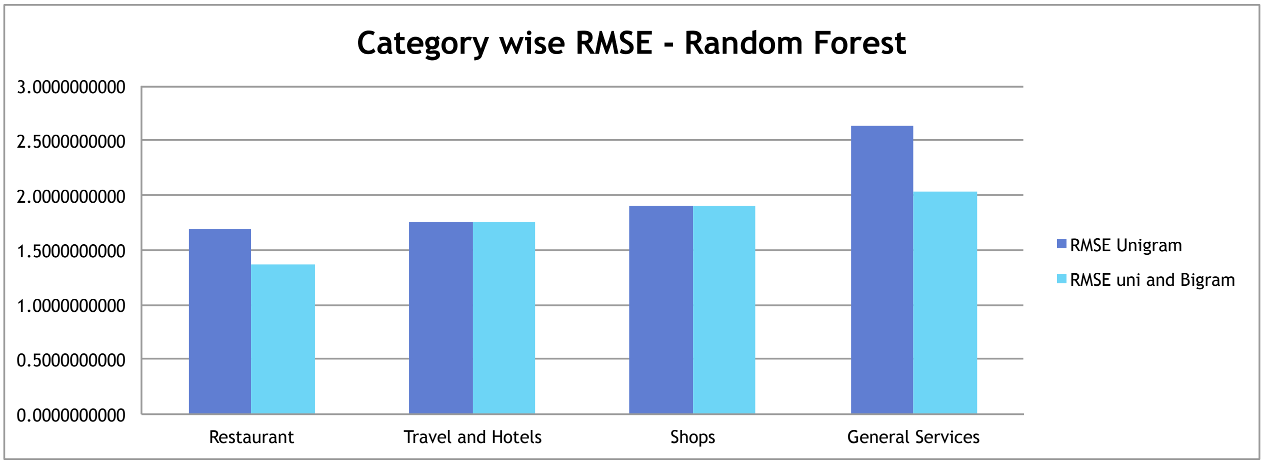


Fig5.1 Chart of comparison - RMSE for each category using random forest model and unigram/bigram feature extraction

8. Conclusion

The following charts show the comparison and analysis of the results:

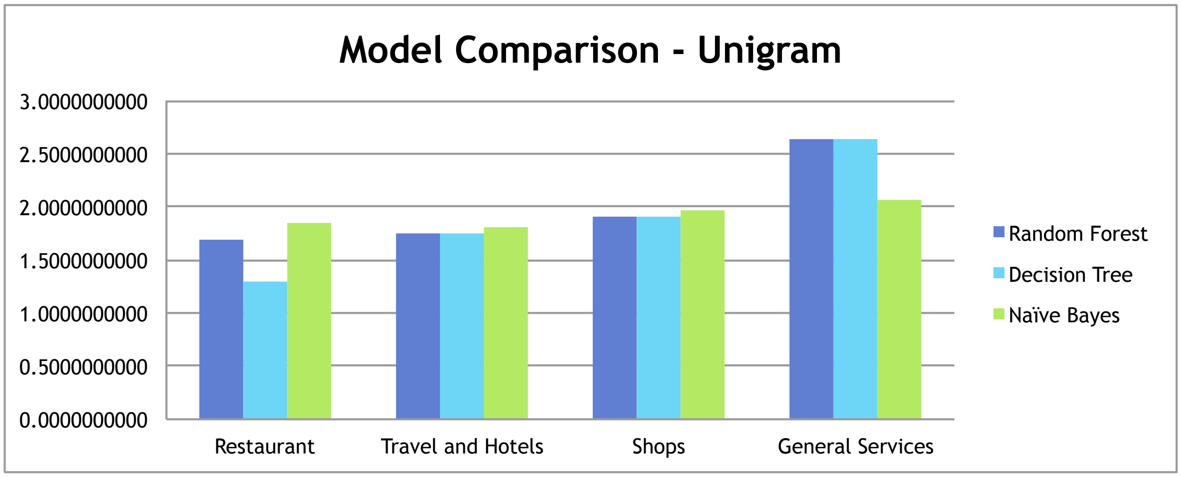


Fig 8.1 Comparison of three models on different category based on unigram

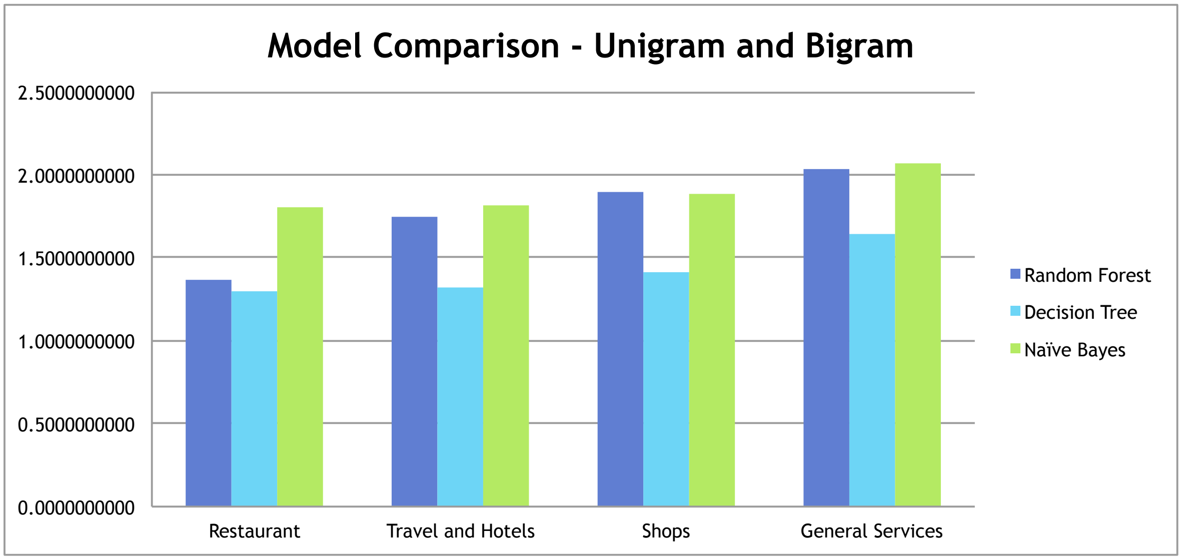


Fig 8.2 Comparison of three models on different category based on unigram and bigram

Following conclusions are derived:

1. The results show that using bigram there are some significant improvement in RMSE for Random forest and Naïve Bayes model.
2. All across the models, reviews of restaurants provide the least RMSE where as General services shows high error. As general services business can be from a varied domain, it is very difficult to train the model properly with given data because review text and topic varies a lot. This goes on to show that for classification algorithm, model has to be trained on a dataset pertaining to the test topic.
3. Overall, decision tree gives best result when comes to using unigram and bigrams where as for unigrams Naïve Bayes gives better result of the three.
4. In future, models are to be trained on part of speech to further fine-tune the prediction.

9. References

https://www.yelp.com/academic_dataset
http://www.yelp.com/dataset_challenge
http://spidr-ursa.rutgers.edu/resources/WebDB.pdf
http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf
http://skife.org/
http://blog.trifork.com/2014/02/04/an-introduction-to-mahouts-logistic-regression-sgd-classifier/
https://mahout.apache.org/users/algorithms/spark-naive-bayes.html
http://en.wikipedia.org/wiki/Root-mean-square_deviation
http://www.csie.ntu.edu.tw/~cjlin/talks/icmr2012.pdf
https://spark.apache.org/docs/latest/mllib-guide.html
https://mahout.apache.org/users/basics/algorithms.html
http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
http://www.amazon.com/Mahout-Action-Sean-Owen/dp/1935182684
http://www.statsoft.com/Textbook/Classification-and-Regression-Trees
https://github.com/Yelp