Yelp Rating Prediction Based on User Review

Final Report

Overview

Over the semester, the team focused on data preprocessing, data exploration and learning java Opennlp library. As a result, we have built a java program to parse raw JSON file from data set and then create features and write it into a CSV file. Moreover, we also built a MySQL database to store the entire data set.

Data Pre-processing

The data preprocessing is much harder than we expected. There are two major challenges that we have in data preprocessing. Firstly, the data is in json format not in the CSV files that we usually use in R. Secondly, given that the data set is a big we need to find a way to deal with this data efficiently.

The Raw Data

1. Data format

   The data set contains reviews data in Json format show as fellows:

```
{
      'type': 'review',
      'business_id': (encrypted business id),
      'user_id': (encrypted user id),
      'stars': (star rating, rounded to half-stars),
      'text': (review text),
      'date': (date, formatted like '2012-03-14'),
      'votes': {(vote type): (count)},
}
```

2. Data Size

   The data set is 2.29GB in size containing more than one million entries.

Using rjson Library to Convert into R Data Frame

The first ideal come to our mind was to directly convert the data into the Data Frame that R uses. After we investigate how to use JSON file in R we find a library might fit our use case. But as we try to use this library we find out that we have no control of how the Json Array is handled. Ideally we want to make each object in the array as one column but instead the library convert it directly into a list in R. Moreover, this library is slow and requires a lot of memory. One of our team mate has a 8G ram in his laptop but could not load the data into R. As a result, we give up on this library.

Using Java to Convert JSON file into CSV

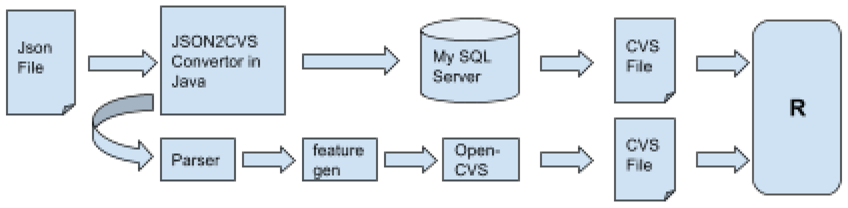
After we tried rjson and it does not work, we figure that we can use other tools to help us convert the JSON file to CSV. We build a simple app which could convert JSON into CSV in java. We use Simple-JSON library to parse the JSON files and use Apache-OpenCSV to write the data into CSV format.

There are several advantages of this, firstly we can use multithreading to parse and convert files which dramatically increased the speed of conversion. Moreover, we can use Opennlp library to create text features in java rather than in R so that we use R for analysis only but Java for feature creation.

Using MySQL Database to Store Data Set

We also managed to store the data into MySQL database, the advantage is that it can help use get a better subset of the data. We have Business table, User table and Review table. By doing this we can select a subset of the data set easily. For example, we can select all the reviews of san francisco in restaurant. And MySQL provide GUI interface to export the selected data into a CSV file which couldn't be any more helpful.

Data Preprocessing Summary



Feature Creation

We use AFINN Dictionary to evaluate the review text and generate the Positiveness and Negativeness of a review. The AFINN Dictionary is a sentiment lexicon with 2477 English words (including a few phrases) each labeled with a sentiment strength and targeted towards sentiment analysis on short text as one finds in social media to be specific twitter.

Example Reviews Text:

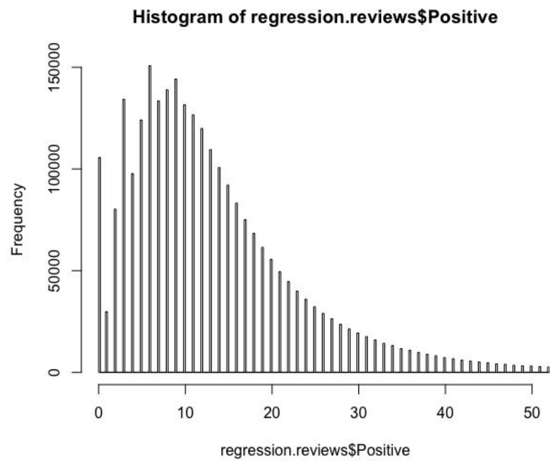
1. This resturant is amazing! The food is superb.
2. This shop is awful, the front door is broken.
3. I think the food here is superb here but the dining table is broken.

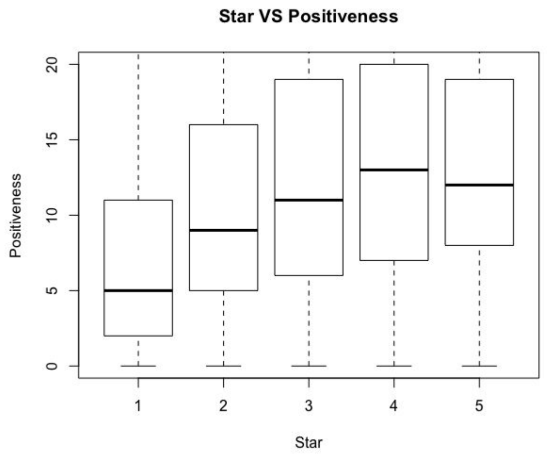
Example Words in AFINN Dictionary

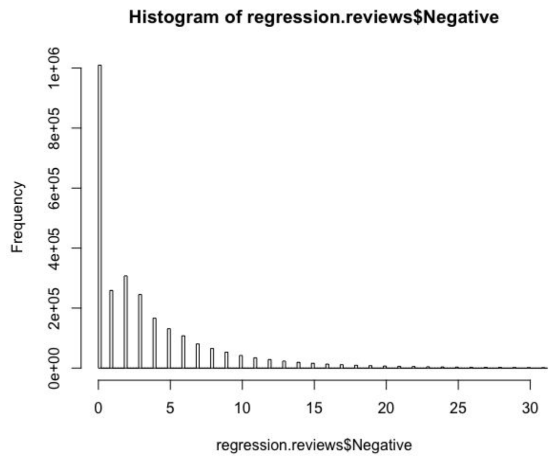
Word	Positiveness	Word	Negativeness
superb	5	awful	3
amazing	4	broken	3

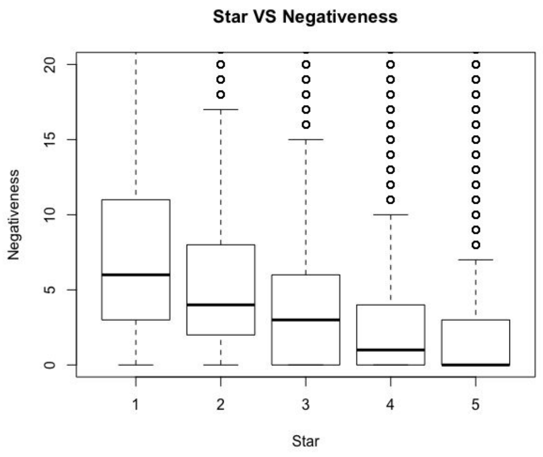
Example Reviews Text:

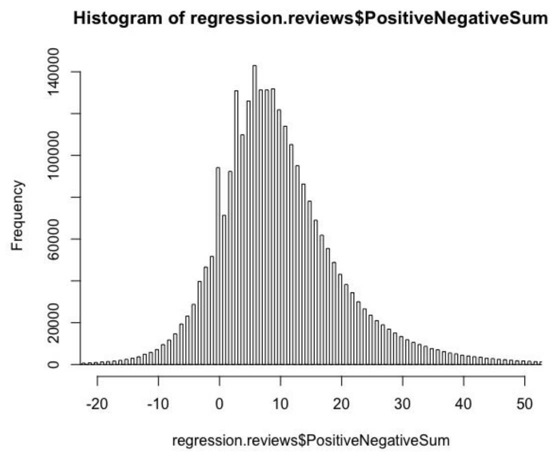
Review	Positiveness	Negativeness	Sum
1	9	0	9
2	0	6	-6
3	5	3	2

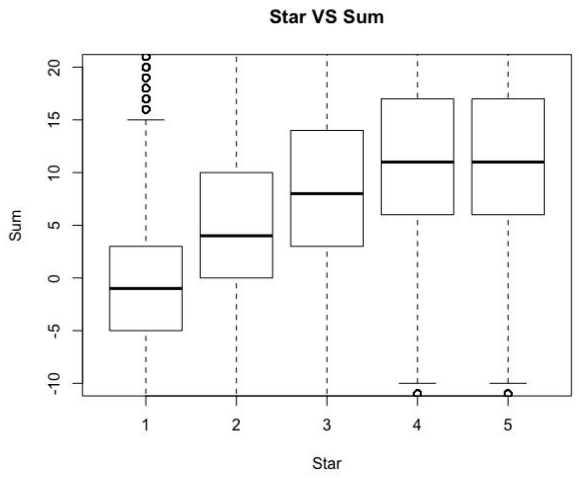

Histogram of regression.reviews$Positive
Frequency
regression.reviews$Positive


Star VS Positiveness
Positiveness
Star

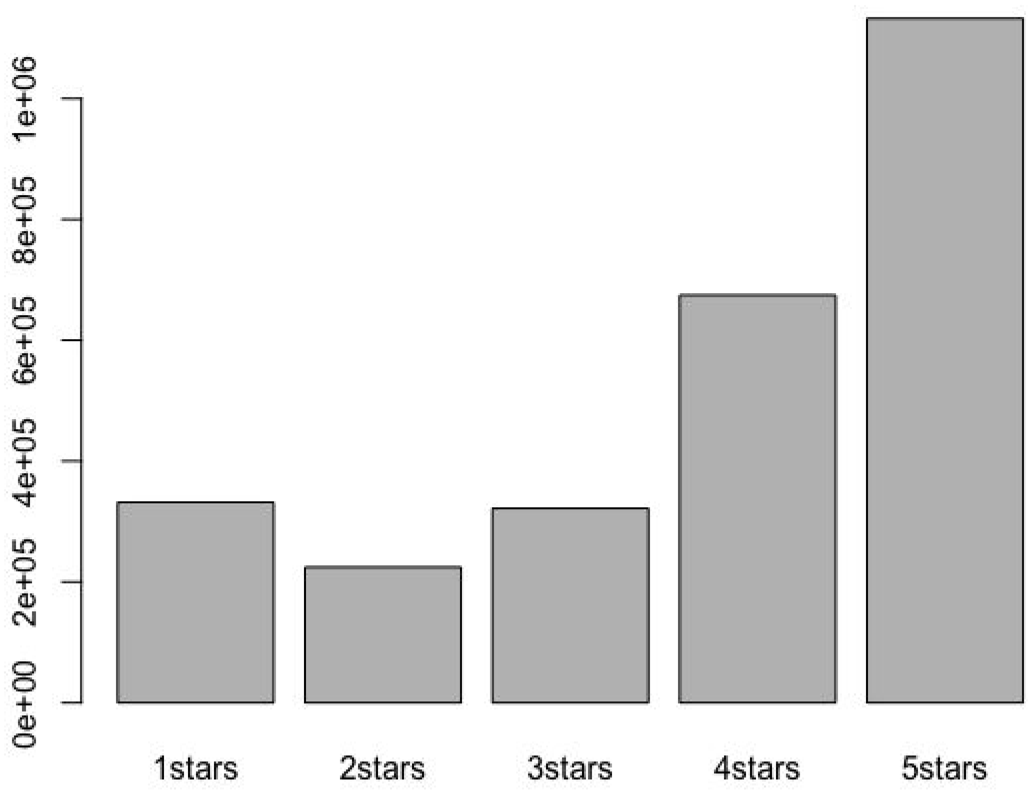

Histogram of regression.reviews$Negative
Frequency
regression.reviews$Negative


Star VS Negativeness
Negativeness
Star


Histogram of regression.reviews$PositiveNegativeSum
Frequency
regression.reviews$PositiveNegativeSum


Star VS Sum
Sum
Star

Data Skewness



Our data is Positive Skewed we should do class balance before we do the actual prediction, that said we sample from 1, 3, 4, and 5 stars review from the data set and make it as the same number of 2 stars.

Data Analysis

Features that was included in our prediction are { "Positive", "Negative", "PositiveNegativeSum", "PositiveWordsCount", "NegativeWordsCount", "TotalWordsCount", "Cool" ,"Funny" ,"Useful"}. We did the Analysis using all data in Cross Validation with 5 folds.

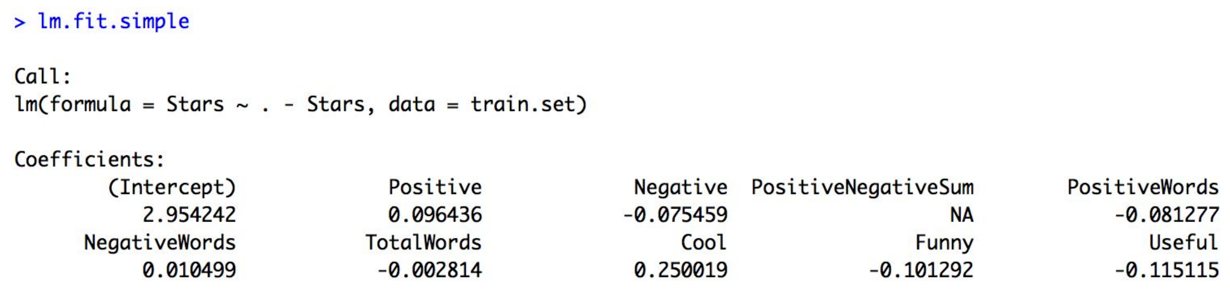
Regression Approach

First we tried to use regression to prediction, we round up the predicted response to do the prediction. For an example we have predict a review to be 4.55 star we just use the ceiling of the value which is 5 and give a 5star response.

R Code Snippet:

```
lm.simple.correct.rate <- c(0,0,0,0,0)

for(i in 1:5){

  testIndexes <- which(regression.reviews.folds==i,arr.ind=TRUE)

  test.set  <- regression.reviews[testIndexes, ]

  train.set <- regression.reviews[-testIndexes, ]

  lm.fit.simple = lm(Stars~.-Stars, data=train.set)

  lm.simple.probs=predict(lm.fit.simple, test.set)

  lm.simple.pred=ifelse(lm.simple.probs>=5,5,ifelse(lm.simple.probs<=1,
1,ceiling(lm.simple.probs)))

  lm.simple.correct.rate[i] = mean(lm.simple.pred==test.set$Stars)

}

mean(lm.simple.correct.rate)
```

Result:



```
> lm.fit.simple

Call:
lm(formula = Stars ~ . - Stars, data = train.set)

Coefficients:
        (Intercept)             Positive             Negative  PositiveNegativeSum        PositiveWords
           2.954242             0.096436            -0.075459                   NA            -0.081277
      NegativeWords           TotalWords                 Cool                Funny               Useful
           0.010499            -0.002814             0.250019            -0.101292            -0.115115
```

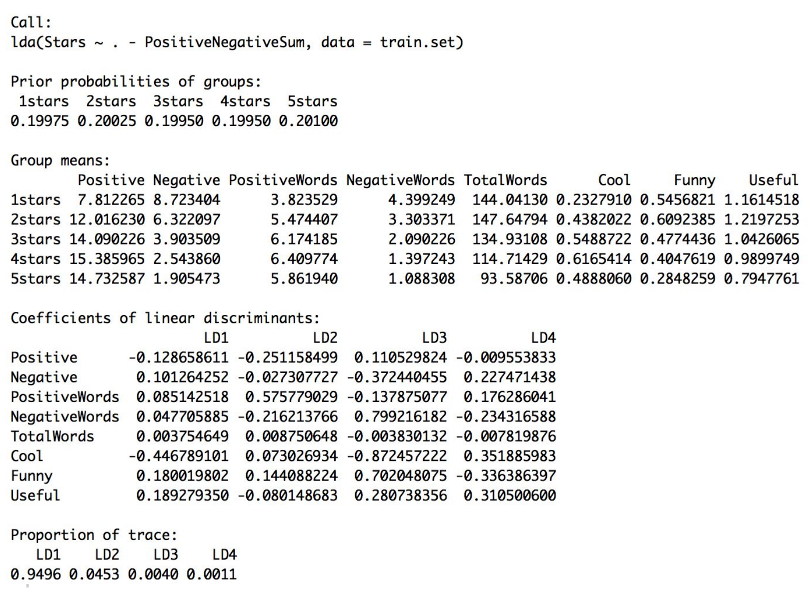
Classify Approach

We then try to do the prediction using the Classify Approach classify all the reviews into 5 responses form 1 star to 5 stars. We tried LDA, QDA, Decision Trees, Random Forest, Naïve Bayes and SVM.

R Code Snippet:

```
# try lda

lda.correct.rate <- c(0,0,0,0,0)

for(i in 1:5){

  testIndexes <- which(classification.reviews.folds==i,arr.ind=TRUE)

  test.set  <- classification.reviews[testIndexes, ]

  train.set <- classification.reviews[-testIndexes, ]

  lda.fit=lda(Stars~.-PositiveNegativeSum, data = train.set)

  lda.pred=predict(lda.fit, test.set)$class

  lda.correct.rate[i] = mean(lda.pred==test.set$Stars)

}

mean(lda.correct.rate)

# we have 0.3934 much better
```

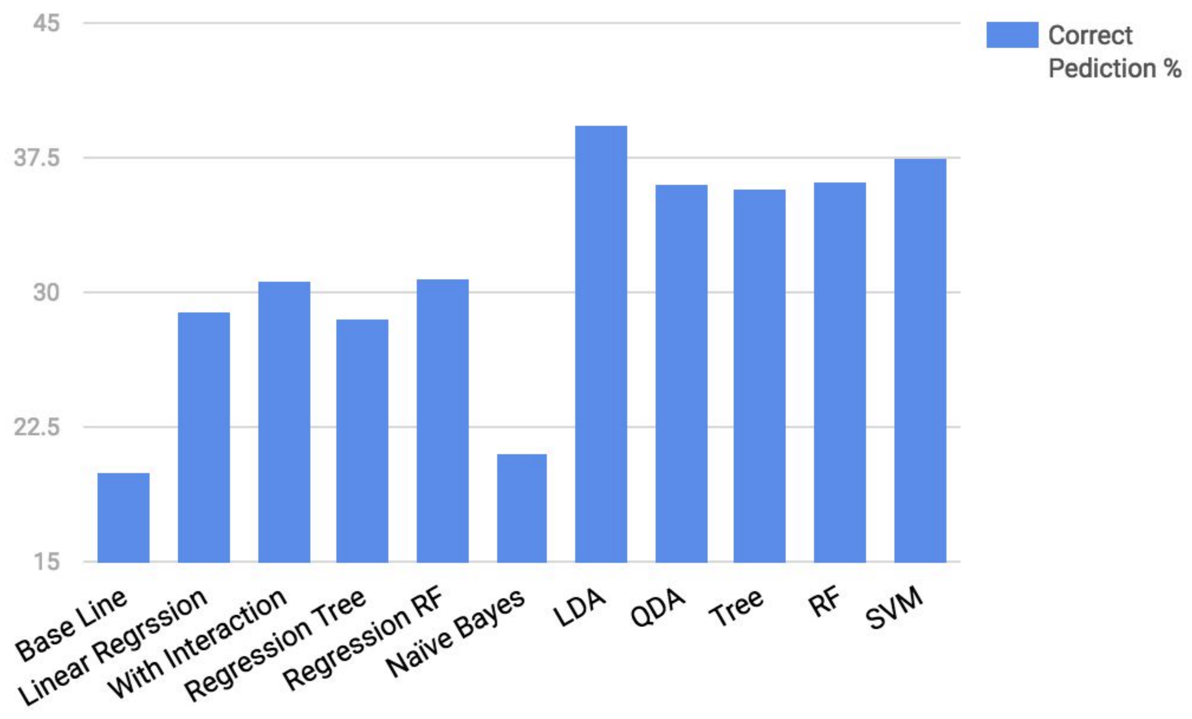
Result



```
Call:
lda(Stars ~ . - PositiveNegativeSum, data = train.set)

Prior probabilities of groups:
 1stars  2stars  3stars  4stars  5stars 
0.19975 0.20025 0.19950 0.19950 0.20100 

Group means:
        Positive Negative PositiveWords NegativeWords TotalWords      Cool     Funny    Useful
1stars  7.812265 8.723404      3.823529      4.399249  144.04130 0.2327910 0.5456821 1.1614518
2stars 12.016230 6.322097      5.474407      3.303371  147.64794 0.4382022 0.6092385 1.2197253
3stars 14.090226 3.903509      6.174185      2.090226  134.93108 0.5488722 0.4774436 1.0426065
4stars 15.385965 2.543860      6.409774      1.397243  114.71429 0.6165414 0.4047619 0.9899749
5stars 14.732587 1.905473      5.861940      1.088308   93.58706 0.4888060 0.2848259 0.7947761

Coefficients of linear discriminants:
                      LD1          LD2          LD3          LD4
Positive      -0.128658611 -0.251158499  0.110529824 -0.009553833
Negative       0.101264252 -0.027307727 -0.372440455  0.227471438
PositiveWords  0.085142518  0.575779029 -0.137875077  0.176286041
NegativeWords  0.047705885 -0.216213766  0.799216182 -0.234316588
TotalWords     0.003754649  0.008750648 -0.003830132 -0.007819876
Cool          -0.446789101  0.073026934 -0.872457222  0.351885983
Funny          0.180019802  0.144088224  0.702048075 -0.336386397
Useful         0.189279350 -0.080148683  0.280738356  0.310500600

Proportion of trace:
   LD1    LD2    LD3    LD4 
0.9496 0.0453 0.0040 0.0011 
```

Prediction Accuracy

We measure the accuracy of our models using Positive predictive value, that is the percentage of correct prediction in our PPT.



In conclusion LDA does best with a nearly 40% correct prediction rate and then SVM Classifier. In terms of MSE it works for regression approach.

Method	MSE
Linear Regression	1.36078
Linear Regression with interaction	1.251987
Regression Tree	1.5972
Regression Random Forest	1.6762

Challenges

- Data Preprocessing, try to implement converting JSON to CSV tooks a long time.
- Feature Creation, we first brainstorm 200 word with positive and negative with ourself but the result was not ideal more than 70% of the data don’t have neither word but after we use the AFINN Dictionary we have less than 1% of the data don’t have neither word form Dictionary.
- Dealing with large data sets.

Timeline

Item	Description	Due Date
1	Project Proposal Due	10/03/2016
2	Meet with team, decide on topic, features and target variable(s), split tasks	10/06/2016
3	Download data, setup git repo with team, install R packages	10/09/2016
4	Data exploration / preprocessing convert json files into CSV	10/12/2016
5	Feature selection / creation (implement features in R)	10/15/2016
7	Run Regression models, evaluate accuracy, plot accuracy results	10/21/2016
9	Update models, run cross validation, evaluate accuracy, plot accuracy results	10/27/2016
11	Project Progress Report Due	11/02/2016
12	Feature Transformations (try AFINN Dictionary)	11/05/2016
13	Dim invalid data	11/08/2016
14	Run models, evaluate accuracy, plot accuracy results	11/11/2016
16	Rerun Regression models, evaluate accuracy, plot accuracy results	11/17/2016
17	Run Classification models, evaluate accuracy, plot accuracy results	11/20/2016
18	Plot comparisons: all models, baselines (random), compute % improvements	11/23/2016
19	Prepare for final presentation	11/26/2016
20	Final Presentation	12/05/2016
21	Project Report and Code Due	12/07/2016