CS 6350.002 Spring 2016

(Professor Anurag Nagar)

Predicting star rating using Sentiment Analysis (Yelp Dataset)

By

Marisundar Selvakani (mxs153130)

Mahesh Sankaranarayanan (mxs154930)

Soorya Prasanna Ravichandran (sxr152130)

Suraj Poojary (ssp151830)

DESIGN

Problem Definition:
Most yelp reviews are skewed towards higher ratings and do not represent a clear picture of the business quality. Also, reviews are subjective and hence make it difficult for new customers to gauge service satisfaction. We aim to create an accurate rating model by performing sentiment analysis on the review text and to generate ratings that are objective and uniform.

We also extend this solution to implement a recommendation and prediction model that takes the ratings calculated from part 1 of our process and uses machine learning algorithms to recommend businesses to users and to predict user ratings for a business.

Dataset Profile:
The data being used for the project is from the yelp dataset. The dataset consists of 3 JSON files containing business, review and user data. This set includes information regarding businesses in 10 cities across 4 countries as shown below.

- U.K.: Edinburgh
- Germany: Karlsruhe
- Canada: Montreal and Waterloo
- U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison

The data size, instances and the columns in focus are mentioned below.
User -> #instances 552K, size 230 MB, format (user_id, name, average_stars, etc)
Business -> #instances 55K, size 67 MB (business_id, name, stars, etc)
Reviews -> #instances 2.2M, size 1.8 GB (business_id, user_id, stars, etc)

Algorithm Design:
Part1:
The algorithm in part 1 of our project initially preprocesses the data into the desired format. It then applies sentiment analysis on the review text by making use of a positive and negative word dictionary. Ratings are calculated for each review by means of a custom formula. We set a baseline for neutral ratings i.e. 2.5. Any rating points greater than 2.5 moves closer to 5 based on the increasing positively

measure in the ratings. Similarly, ratings with lower than 2.5 points start moving closer to 0 on increasing negativity measure.

Formula:
2.5 + [(positive - negative) * 2.5 / (positive + negative)]

Positive and negative counts are gathered for each review. If both are same, then the ratings is 2.5 i.e. neutral. If positive> negative, then the formula above takes the difference between the two into consideration and calculates the final rating.

Pseudo Code:

```
1. Perform Pre-Processing
    1.1. Conversion from json to csv. (Python)
    1.2. Filter the text eliminate punctuations, digits and extra spaces. (Python)
    1.3. Extract only specific columns that are required. (Unix)
    1.4. Eliminate empty reviews. (Unix)
2. Calculate positive and negative ratings to come up with final ratings.
    2.1. Divide the review words into key-value pairs, with key being the word and
         the value being the word count.
    2.2. Join the reviews words to the words in the positive and negative dictionary,
         and take their neighbors into consideration to calculate the positive and
         negative ratings.
    2.3. Calculate the actual ratings by using the custom formula mentioned earlier.
3. Compare actual and predicted ratings
```

Part2:
We extend the idea to implement a recommendation and prediction model by using Mahout's ALS (Alternating Least Squares) machine learning algorithm.
The ratings from part 1 are divide into a training dataset and a testing dataset in the ratio of 80:20. The 80% data is fed to the ALS algorithm using Apache Spark to build and train the model.

Challenges:

- ALS does not work with alphanumeric ids, hence we needed to map the unique alphanumeric business and user ids to unique numeric ones.

- Data is sorted and hence cannot be easily divided into training and testing datasets.

Pseudo Code:

```
1. Divide the dataset from part 1 into training and testing datasets.
2. Train the ALS model (Apache Spark)
    2.1. Map the alphanumeric ids to their corresponding uniquely generated
         numeric ones to generate a modified training and testing dataset.
    2.2. Feed the modified training dataset to the ALS model and train it.
    2.3. Run the recommend function to generate the recommendation for a
         particular user.
    2.4. Run the predict function on the testing dataset to generate the list of
         predicted ratings.
    2.5. Remap the unique numeric ids to corresponding alphanumeric ones.
3. Calculate the mean square error for the predicted ratings.
```

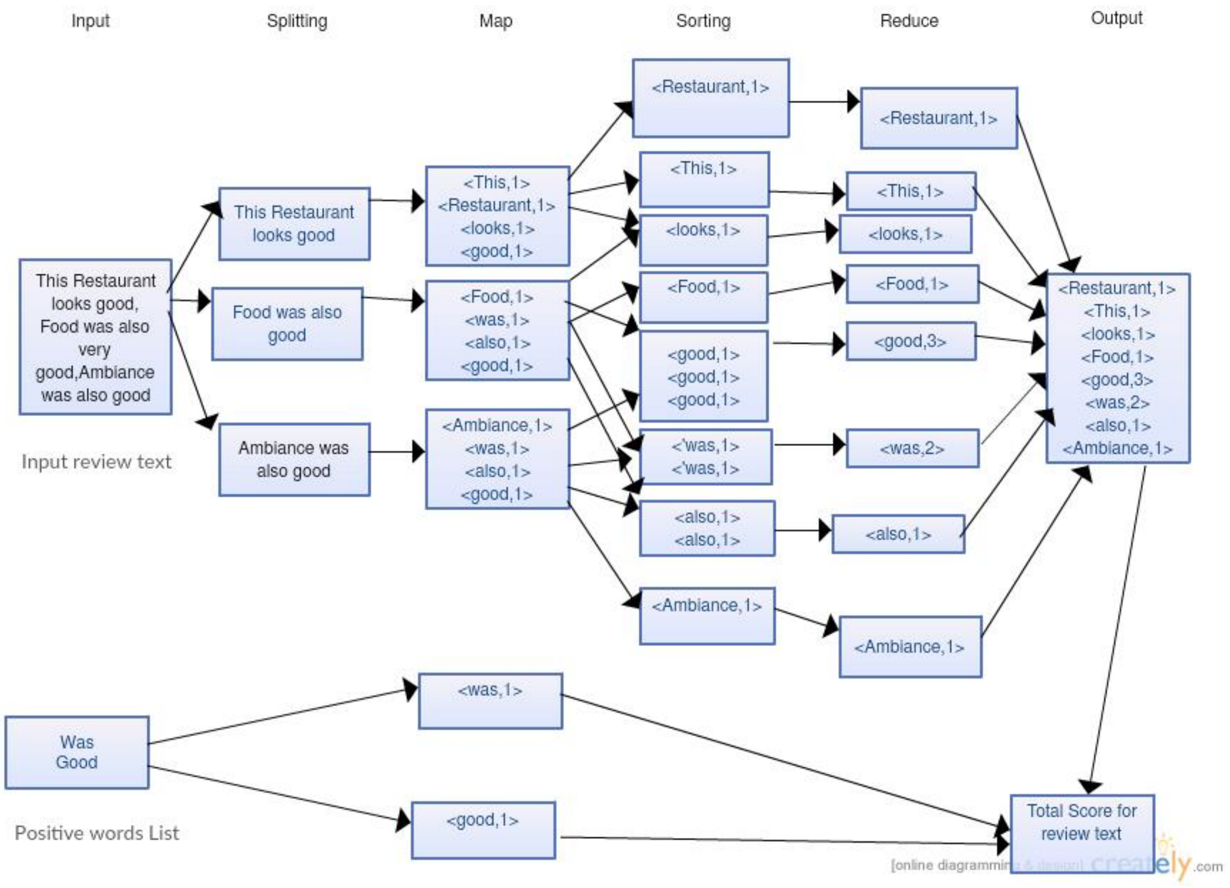
Big Data Strategies:

For part 1 of the project, we did a comparative study with 3 technologies namely Hadoop MapReduce, Apache Spark and Pig. Sentiment analysis works really well with Big Data as it is deals with a huge number of key value pairs and is based on the parallel processing paradigm. Apache Spark is our point of focus for the sentiment analysis process as it is faster when compared with the other 2 and also that it generated better results.

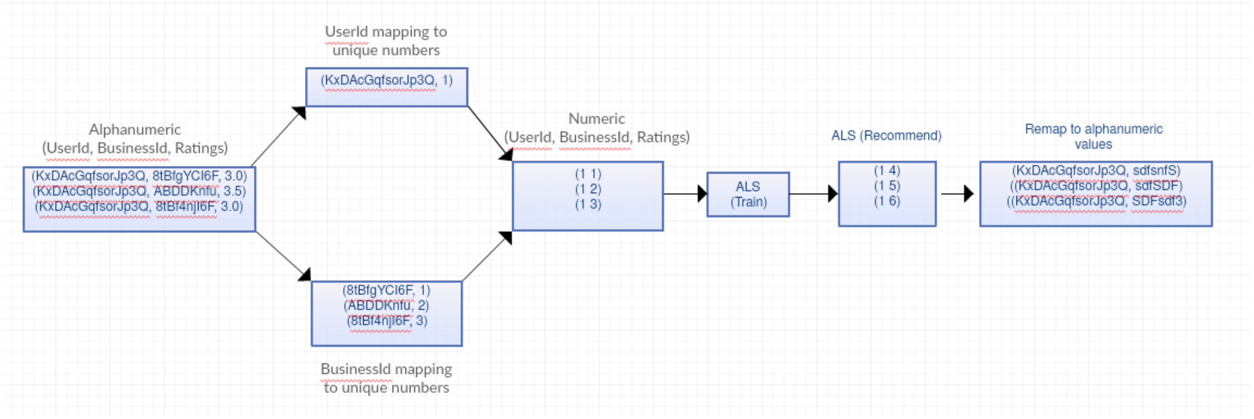
Part 2 of the project focuses on machine learning concepts as we use the ALS recommendation model to come up with user recommendations and prediction of user ratings. Recommendation models are common use cases for ecommerce businesses like amazon and service oriented businesses like yelp. Machine learning libraries and built in algorithms serve as a great tool for implementing these use cases.

Data flow diagrams:

Part 1:


Input
Splitting
Map
Sorting
Reduce
Output
This Restaurant looks good, Food was also very good,Ambiance was also good
Input review text
This Restaurant looks good
Food was also good
Ambiance was also good
<This,1> <Restaurant,1> <looks,1> <good,1>
<Food,1> <was,1> <also,1> <good,1>
<Ambiance,1> <was,1> <also,1> <good,1>
<Restaurant,1>
<This,1>
<looks,1>
<Food,1>
<good,1> <good,1> <good,1>
<'was,1> <'was,1>
<also,1> <also,1>
<Ambiance,1>
<Restaurant,1>
<This,1>
<looks,1>
<Food,1>
<good,3>
<was,2>
<also,1>
<Ambiance,1>
<Restaurant,1> <This,1> <looks,1> <Food,1> <good,3> <was,2> <also,1> <Ambiance,1>
Was Good
Positive words List
<was,1>
<good,1>
Total Score for review text
[online diagramming & design] creately.com

Part 2:

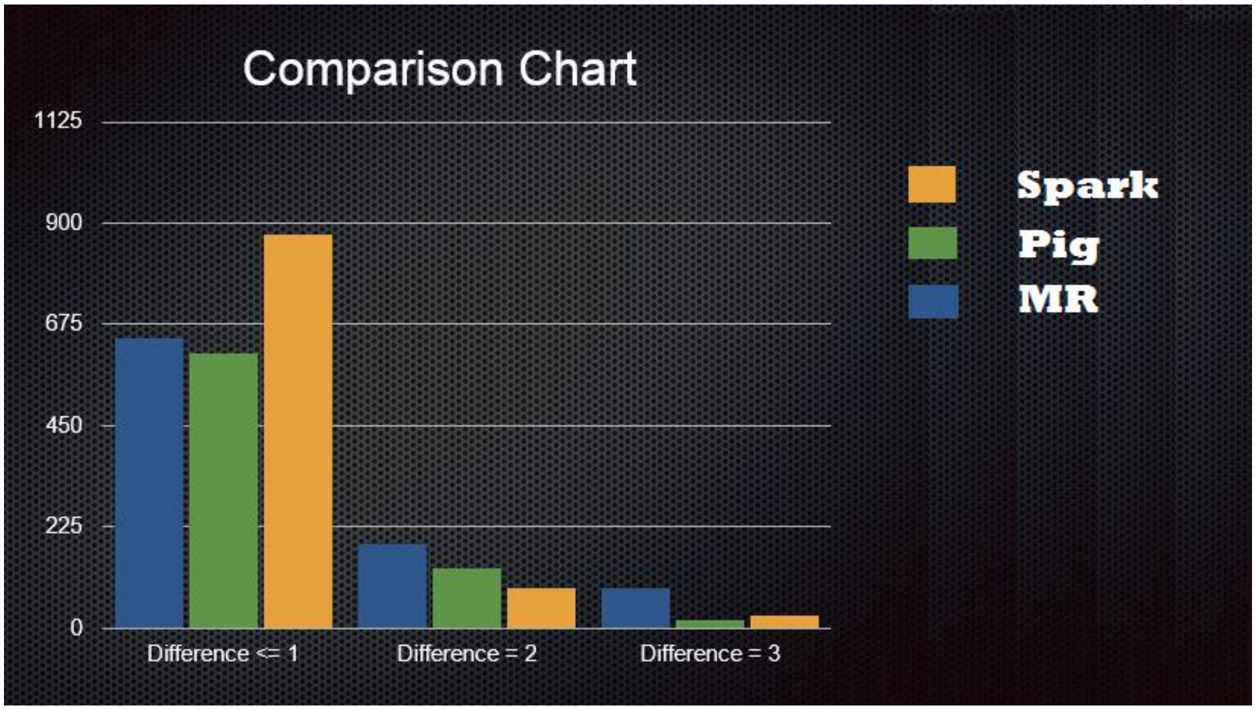

UserId mapping to unique numbers
(KxDAcGqfsorJp3Q, 1)
Alphanumeric (UserId, BusinessId, Ratings)
(KxDAcGqfsorJp3Q, 8tBfgYCI6F, 3.0)
(KxDAcGqfsorJp3Q, ABDDKnfu, 3.5)
(KxDAcGqfsorJp3Q, 8tBf4njI6F, 3.0)
Numeric (UserId, BusinessId, Ratings)
(1 1)
(1 2)
(1 3)
ALS (Train)
ALS (Recommend)
(1 4)
(1 5)
(1 6)
Remap to alphanumeric values
(KxDAcGqfsorJp3Q, sdfsnfS)
((KxDAcGqfsorJp3Q, sdfSDF)
((KxDAcGqfsorJp3Q, SDFsdf3)
(8tBfgYCI6F, 1)
(ABDDKnfu, 2)
(8tBf4njI6F, 3)
BusinessId mapping to unique numbers

Handing edge cases:

The preprocessing step takes care of the bad data as it filters out non-alphanumeric characters. The missing data is taken care of too through a unix script as part of the preprocessing/data cleansing step. We have used the standard lambda value of 0.01 that takes care of overfitting.

ANALYZING RESULTS

Part1:



We implemented the neighbor’s technique in Spark and the plot above is indicative of that. For Spark, 900 out of every 1000 reviews differed with respect to the actual rating by <=1 and the numbers reduced for larger differences, indicating that as error scope increased, the error rate decreased. MapReduce and Pig which did not have the neighbor’s technique implemented gave not so accurate results.
Also, Spark as expected gave faster run times than MR and Pig.

Part2:
We use 2 parameters namely the mean square error and accuracy for gauging the accuracy of our model. Below are the results of the validations.
Mean Square Error: 0.041
Accuracy: 87.4%

CONCLUSION

Big Data and machine learning technologies are designed for high volume parallel processing. Our project topic is one of the common use cases for sentiment analysis and recommendation models. Implementing the same using standard programming languages will not yield desired results in terms of efficiency and performance.
Apache Spark due to its in memory RDD based processing fastened the entire key value pair generation and joining operations.

Key Learnings:

- Better understanding of the various recommendation and prediction strategies.
- How recommendation systems work in real time and a brief idea on what goes in the background of let's say an amazon.com.
- Exposure to machine learning and statistical concepts and how they help in implementing and validating recommendation systems.
- Practical Application of Hadoop Map Reduce, Apache Spark and Pig on a large scale of data.

Improvements/Future work

We would like to revisit the sentiment analysis in part 1 and use a more sophisticated process like an NPL API to perform sentiment analysis for better results. As for the 2nd part, we currently feed only specific fields like userid, businessid and rating to the ALS model. We would like to use a more intuitive recommendation model that can take into consideration other parameters of the dataset like location, reviews from other users, etc to come up with better/relevant recommendations for the users.

TEAM ROLES

- Data Analysis (Soorya Prasanna Ravichandran)
- Pre-processing (Soorya Prasanna Ravichandran)
- Star Rating Prediction
    - Map Reduce (Marisundar Selvakani)

    - Apache Spark (Marisundar Selvakani)
    - Pig (Mahesh Sankaranarayanan)
- Recommendation model (Mahesh Sankaranarayanan and Suraj Poojary)
- Prediction model (Suraj Poojary)