CS6350: Big Data Management and Analysis

Classification of Yelp Reviews

Project Report

Fall 2017

December 1st, 2017

By:

Arpita Mothukuri (axm163631)

Dileep Gudena (dxg161730)

Lakshmi Priyanka Parimi (lxp160730)

Keertan Dakarapu (kxd160830)

1. Introduction and problem description

Yelp has huge repositories of data pertaining to restaurant reviews and ratings of customers. The data collected is user entered and is not validated to any predefined standard. The ratings given are subjective measure of customers experience and can't be considered as an absolute measure of restaurant review. The review written by customers is un-validated and could contain complex contextual metaphors and non-standard expressions. Hence, simply looking for some positive words or negative words in the review may not help us get the actual essence of the review. Pertaining to this situation, analysing the reviews in huge number requires us to use big data techniques and analysis.

Problem Description:

A complete analysis of Yelp reviews and classifying user reviews as positive and negative using sentiment analysis and classification techniques.

Our Understanding

The Yelp dataset contains huge data about the business and the reviews it received. In our case, we consider only the business review data. We analyse the reviews and classify it either as positive or negative depending upon the sentiment of the review. Considering the no of reviews, running a sentimental analysis function on each row of review data, would result in very high latency. Hence we have to convert the problem into logical modules and convert into map reduce format.

2. Related work

https://www.ideals.illinois.edu/handle/2142/48832

http://www.nowpublishers.com/article/Details/INR-011

3. Dataset description

Name: Yelp Dataset Challenge

Link: https://www.yelp.com/dataset/challenge

Number of Instances: 4736897

Number of features: 6

List of Attributes:

- **User_id** The identity of the user
- **Review_id** identity of each review
- **Text** Review given by each user
- **Business_id** The identity of the restaurant
- **Stars** rating given by the user
- **Date** The date when the review was given

Snapshot of Data:

reviews

Filter and Sort | Query Builder | Where | Data | Describe | Graph | Analyze | Export | Send To

	user_id	review_id	text	business_id	stars	date
1	cjpdDjZyprfyDG...	VfBHSwC5Vz_...	My girlfriend an...	uYHaNptLzDLo...	5	2016-07-12
2	bjTcT8Ty4cJZh...	3zRpneRKDsO...	If you need an i...	uYHaNptLzDLo...	3	2016-10-02
3	AXqRULmWcM...	ne5WhI1jUFOc...	Mittlerweile gibt...	uYHaNptLzDLo...	3	2015-09-17
4	oU2SSOmsp_A...	IImdwOqDReuc...	Location is ever...	uYHaNptLzDLo...	4	2016-08-21
5	0xtbPEna2Kei1...	DuffS87NaSM...	gute lage im sta...	uYHaNptLzDLo...	5	2013-11-20
6	rW8q706dz5-N...	GvLmUkjUrOyF...	Erstklassige La...	uYHaNptLzDLo...	5	2016-06-05
7	yx8vNXUL0D0...	IGEI24NGj2HV...	Beautiful space...	uYHaNptLzDLo...	4	2015-02-21
8	zXnH6W74FAJ...	cUqvEy5wj7zY...	This is a fairly n...	uYHaNptLzDLo...	4	2013-07-07
9	c5yp5hxwC1N9...	FSB_BnvysBq...	First time at this...	uYHaNptLzDLo...	4	2013-04-27
10	xJisL5w4wOqiY...	dhI3ZW9aAEX...	Location locatio...	uYHaNptLzDLo...	4	2015-04-13
11	tqV6tsYQ66DZ...	JQJvnM3p-3eM...	A hotel that has...	uYHaNptLzDLo...	4	2016-11-08
12	Q-3YCVywc03...	6JF4WfHqwYrr...	Stayed here for...	uYHaNptLzDLo...	3	2015-07-27
13	Cx4UCow0zQq...	fbVYETRuWD...	Well, i like the i...	uYHaNptLzDLo...	4	2014-05-07
14	eqWEqMH-DC...	lobj38NqaokqV...	I really do love t...	uYHaNptLzDLo...	4	2015-02-26
15	d0DGZRp6lHX...	ysfjtAWreLoy7u...	Motel One sets...	uYHaNptLzDLo...	5	2015-08-21
16	IpLZ7RevQrFP...	OF1ToqGAubs...	Had Continenta...	uYHaNptLzDLo...	3	2013-12-07
17	kzyLOqiJvyw_F...	ByRzJ8rF2KJW...	This place is ho...	jQsNFOzDpxP...	1	2017-06-03
18	WZXp9-V2dqR...	i5UwUPIQFPLc...	For being fairly...	jQsNFOzDpxP...	4	2015-03-26
19	XylT12exfdLiI_...	EyQyvTTq2jX4...	I decided to try i...	jQsNFOzDpxP...	5	2012-12-30
20	Ji9PeffxjwqPLO...	G-EFA005besj...	I'm not saying P...	jQsNFOzDpxP...	3	2009-01-12
21	TLIWzAJPrET0...	6PcJSGUBSLjt...	Sometimes the...	jQsNFOzDpxP...	3	2015-07-11
22	JZEiTNWBwmv...	PFJmyZD_INB...	Decent custom...	jQsNFOzDpxP...	1	2015-05-27
23	E56sVQT5-OW...	_Qv1FQUToLr...	Super clean res...	jQsNFOzDpxP...	5	2015-02-28
24	4WYICo4emec...	s2mlqrFNaPEG...	Found this the...	jQsNFOzDpxP...	4	2010-04-05
25	P8mVj7AZwJT...	oiSzZRrbi3y01...	The staff here i...	jQsNFOzDpxP...	1	2015-05-22
26	7Y4NEBQqWq...	4BPjRE9VI0Hh...	I had the garlic...	jQsNFOzDpxP...	2	2011-06-15
27	vqZqQqe8cj6S...	kznHtw1Qido_9...	This review is b...	jQsNFOzDpxP...	5	2017-03-12
28	O7G_c6wFXSy...	HWRTVn3Lc-R...	I love this place...	jQsNFOzDpxP...	5	2016-12-19
29	UG4EKu13JRw...	GiEB_A-m9Hu...	1st! Place is not...	jQsNFOzDpxP...	4	2011-08-10
30	ZZG6yR27Ily3x...	GKi4i6qoclqaY...	Definitely not a...	jQsNFOzDpxP...	2	2013-06-17
31	1YorWW0Z-YD...	OrhWq2MmCz...	Pretty good, not...	jQsNFOzDpxP...	3	2015-11-03
32	ujOPJEz_KxzA...	QXWku_OB3F...	I wish I could gi...	jQsNFOzDpxP...	1	2017-07-08
33	6aEUn50d3Ts7...	5NtaW5EwXK5...	Disappointed th...	jQsNFOzDpxP...	2	2015-09-22
34	R6vb0FtmClhf...	ai6O4UqqDqnj...	1st visit had the...	jQsNFOzDpxP...	3	2012-10-08
35	CPuUaqT2rfUJ...	ZrvsD7PSyPolll...	As a vegetarian...	jQsNFOzDpxP...	5	2015-12-28
36	OYRBjBWy1uO...	p7OqbXTjwmIN...	Typical big busi...	jQsNFOzDpxP...	2	2009-11-18
37	PKZLwAGqBtQ...	ukpjwnetF5wG...	I love Pei Wei s...	jQsNFOzDpxP...	4	2012-07-29
38	9bJ6j0zrV1XSi...	mT6U5lujK_zlcl...	Great fresh foo...	jQsNFOzDpxP...	4	2016-06-19
39	8nCmV4RMwf4...	YxAxExTdWtd...	This is pretty go...	jQsNFOzDpxP...	3	2012-08-23
40	tbAQMMVlhxvX...	ue6ts-qA9khyw...	Food is good a...	jQsNFOzDpxP...	4	2009-06-25
41	1s0Q1KwGpJl...	WsTYqsyNyUd...	The hubs and I...	jQsNFOzDpxP...	4	2008-10-06

Figure 1: Snapshot of Data

Programming Languages used:

- Scala
- python
- R

Tools Used:

- Databricks
- SAS
- R Studio

Techniques Used:

- **Supervised Learning:**
  - Logistic regression
- **Unsupervised Learning:**
  - K Means

4. Pre-processing techniques

- **Converting JSON file to CSV file**
  The raw file downloaded at the Yelp repository was in Json format and was illegible for applying our bigdata techniques directly on it. To convert it into csv, we have used SAS software and loaded the Json data and converted it into CSV.

- **Checking null and missing values**
  The preliminary analysis of random samples of data revealed some missing and null values, which were conveniently removed after loading it up in Databricks.

- **Filtering unused columns**
  For our part of analysis and problem statement, we could relieve a lot of processing and memory load by reducing the data columns to only 3 columns which were to be used – Review_id, Text, Stars.

- **Resolving csv issue with review text**
  The reviews written by users contained ',' (comma) in the text, which induced errors and incorrect data into our dataframe. To resolve this issue, we used '*struct*', to impart datatype based separation.

  Code Snippet:

```
import org.apache.spark.sql.SQLContext
import org.apache.spark.sql.types.{StructType, StructField, StringType,
  IntegerType}
val customSchema = StructType(Array(
    StructField("review_id", StringType, true),
    StructField("text", StringType, true),
    StructField("stars", IntegerType, true)))
```

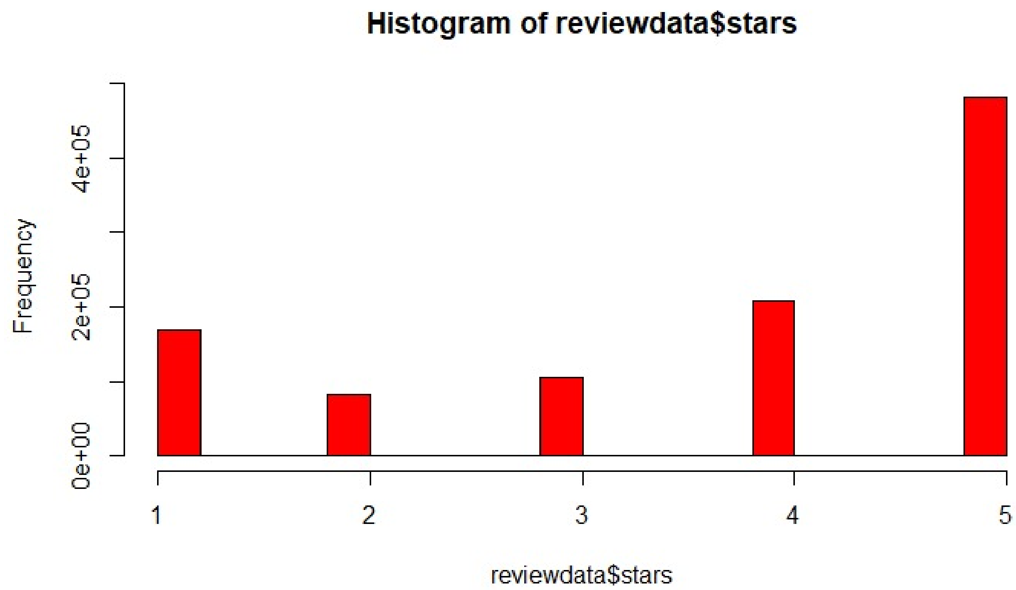

Figure 2: Histogram of ratings (stars)

- **Resolving additional '.' In review text**
  The review text is split based on '.' And each sentence is passed onto a sentiment function. But, reviews contained '…' and additional dots which resulted in empty strings and errors. We replaced all the dots with a '. ' (dot space) to include non null strings, which return a sentiment score of 0.

Packages Used:

Logistic Regression	org.apache.spark.ml.classification.LogisticRegression
KMeans Clustering	org.apache.spark.mllib.clustering.{KMeans, KMeansModel}
Stanford coreNLP	edu.stanford.nlp.pipeline.StanfordCoreNLP edu.stanford.nlp.*;

5. Proposed solution, and methods

Sentimental analysis

We use Stanford NLP libraries to get the sentiment score of each sentence in a review. We take each review and split it by "." and pass it on to *classify()* function which would use the Stanford apk to get a sentiment score for the sentence. The function is called by the map method, which takes in (*review_id, text*) as input and gives out (*review_id, 'sentiment score of each sentence in review'*). The reduce method aggregates the tuples with respective to *review_id* and sums all the sentiment values.

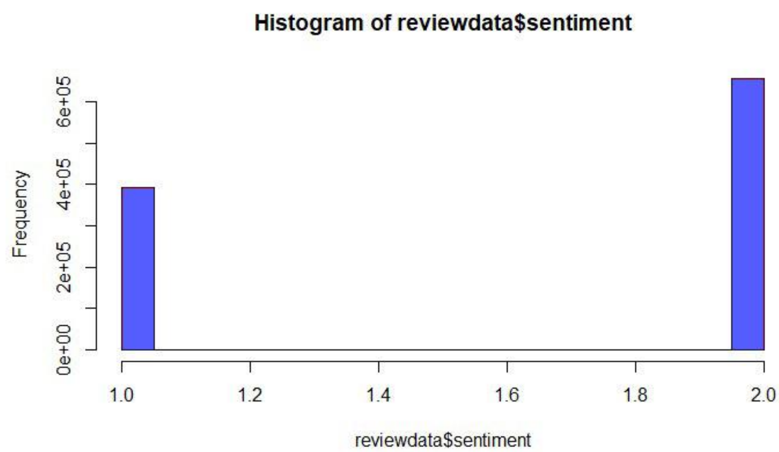

Figure 3: Histogram of sentiment scores

Map - Reduce

Map – Reduce concepts are integrated into our solution while calculating the final sentiment of entire review. When each review is split up into sentences and sentiment scores are calculated, aggregation of all the sentences of a review is a processor costly process. Hence, we use map-reduce methods to tackle this problem and aggregation is done at the reducers. [Figure 4]

MLlib for clustering

In our initial proposed solution, rather than clustering, we went ahead with logistic regression. We used parameter grid which helped us to choose the best parameter for regression and cross fold validation for efficient and more accurate model generation. We used pipelining to convert our output classification to 0 and 1 (normalize). We even continued the process till evaluation metrics like precision, recall and accuracy of the model.

We have evaluated our solution, and came to a decision that this case comes under unsupervised learning and applying a clustering on the aggregated sentiment values would result in a better solution.

```
1  finalReviewsdf.show()
```

▸ (1) Spark Jobs

```
+--------------------+--------------------+-----+---------+
|           review_id|                text|stars|sentiment|
+--------------------+--------------------+-----+---------+
|VfBHSwC5Vz_pbFluy...|My girlfriend and...|    5| positive|
|3zRpneRKDsOPq92tq...|If you need an in...|    3| negative|
|ne5WhI1jUFOcRn-b-...|Mittlerweile gibt...|    3| negative|
|llmdwOgDReucVoWEr...|Location is every...|    4| positive|
|GvLmUkjUrOyFH8KFn...|Erstklassige Lage...|    5| negative|
|lGEl24NGj2HVBJrod...|Beautiful space, ...|    4| negative|
|dhl3ZW9aAEX_T7_um...|Location location...|    4| positive|
|JQJvnM3p-3eML05eK...|A hotel that has ...|    4| positive|
|6JF4WfHgwYrrdZ2Ve...|Stayed here for t...|    3| negative|
|fbVYETRuWDw8Qnpim...|Well, i like the ...|    4| positive|
|lobj38NgaokqVseN8...|I really do love ...|    4| positive|
|ysfjtAWreLoy7um8W...|Motel One sets th...|    5| negative|
|ByRzJ8rF2KJWLr-cU...|This place is hor...|    1| negative|
|6PcJSGUBSLjt4VLXo...|Sometimes the foo...|    3| positive|
|PFJmyZD_lNBa_Y3kb...|Decent customer s...|    1| negative|
|_Qv1FQUToLrKMuG6p...|Super clean resta...|    5| positive|
|oiSzZRrbi3y01_wqU...|The staff here is...|    1| negative|
|kznHtw1Qido_9GX6s...|This review is ba...|    5| negative|
```

Figure 4: Merged dataframe including sentimental values after Map-reduce

Final output using clustering would be this

No. of Clusters = 3

Within Set Sum of Squared Errors for 3 no. of clusters = 17.483333333333377

6. Experimental results and analysis

Our approach for implementing this project is as follows:

1. **Pre-processing the dataset:**
   - Convert JSON file to CSV file.
   - Removing null and missing values.
   - Removing Unused Columns
2. **On the dataset:**
   - Implement all the above-mentioned classifiers on data.
   - Also, find the best set of parameters for which the techniques performed best.
   - Using Stanford Core NLP for sentimental analysis
   - Using ML lib for clustering, and logistic regression.
3. **Evaluation the techniques:**
   - The techniques are evaluated using Accuracy, Precision, Recall and WSSE metrics.

Experiment 1:

Initially, we tried splitting the words in the text and use a dictionary of words with score. We then mapped these scores to the words that we have split. In the reduce phase we sum the scores on each review_id and combine the total score to check if it is a positive sentiment or a negative sentiment.

Logistic regression parameter table

No of cross folds	Max Iterations	Reg Param	threshold	precision	recall	Accuracy
10	20	0.01	0.6	0.759	0.727	0.737
30	20	0.01	0.6	0.878	0.818	0.821
30	50	0.001	0.6	0.906	0.875	0.87
30	15	0.001	0.4	0.925	0.888	0.895

Experiment 2: (Final implementation)

We tried to create a classification based on aggregated sentiment scores of review using unsupervised machine learning. We tried to split our final sentiment scores using clustering. We have experimented with 2 clusters and 3 clusters, if we split into 2 clusters, then one of them should be positive and the other would be negative. When we split the data into 3 clusters, we assumed that the outlier data points between the clusters could form another cluster of neutral reviews, thus increasing out classification accuracy.

The Squared Error for clusters when using k =2, ad k = 3

No. of Clusters = 2

Within Set Sum of Squared Errors for 2 no. of clusters = 30.947619047619092

No. of Clusters = 3

Within Set Sum of Squared Errors for 3 no. of clusters = 17.483333333333377

7. Conclusion

In summary, we develop an algorithmic procedure for analysing and classifying the review for each review_id. We use Stanford NLP libraries to get the sentiment score of each sentence in a review and aggregate it for each review_id. We then apply logistic regression on the reviews based on the ratings given to develop a prediction model that predicts if the review is either positive or negative. We then compare the results to find the accuracy, which in this context is the percentage of matching results, which compares our sentimental output to a well-defined machine learning model. In our case, we received 89.5% accuracy rate, which depicts that our sentimental classification is accordance with logistic regression model. But, we decided to go ahead with unsupervised learning technique and performed clustering on resulting sentimental scores with k =2, and obtained a squared error = 17.48

Further development in our model could include context based sentimental score, curved aggregation of word score instead of logistic summation and unstructured spoken English analysis.

8. Contribution of team members

We have divided our work into logical blocks like pre-processing, sentimental analysis, Map-Reduce implementation, Logistic regression, clustering, integration and documentation. We followed pair programming and have distributed our work as follows.

Dilip and Arpita:

sentimental analysis, Map-Reduce, Integration, documentation

Keertan and Priyanka:

Pre-processing, Logistic regression, clustering, documentation

9. References

- https://spark.apache.org/docs/latest/quick-start.html
- https://spark.apache.org/docs/latest/sql-programming-guide.html
- https://spark.apache.org/docs/1.2.0/mllib-guide.html