Final Project

Amazon Customer Review Analysis

**Preface**

Amazon Customer Reviews (a.ka. Product Reviews) is one of Amazon’s Iconic products. Which contains more than 130 million customer reviews, which are available to researchers for research purpose. Dataset contains multiple languages and for multiple countries. In the analysis we are only dealing with US English reviews. Trying to figure out if the sentiment of the reviews conforms the stars given to the product.

**Table of Content**

**1.** Introduction

**2.** Sentiment Analysis

**3.** Averaging Stars per Product

**4.** Averaging Sentiments Score per Product

**5**. Top N analysis

**6.** Results

**Introduction**

The hadoop ecosystem is widely used for finding solutions for bigdata. Ever growing data needs a powerful tool like hadoop. Customer’s reviews on amazon is one such example, it keeps on growing everyday as people write comments for a product every second. In order to analyze customer reviews on the amazon site we are using hadoop.

We are trying to find out if the the average stars given to a product is how much same as the average sentiment of the score given to a product. And check if they come up with the same result. The logic behind if are tend to say good words about a product in general will tend to have higher star rating.

**Sentiment Analysis**

This is the first part of the analysis where which was done on hadoop. The sentiment analysis on comment gives a number -1.0 to 1.0, know as Sentiment Score. -1.0 is very bad, 0.0 is neutral and 1.0 is very good. This is calculated by counting the number of positive words and negative words in the comment. And finding the score through the equation.

Sentiment Score = (Positive count – Negative Count)/(Positive Count + Negative Count)

In order to implement this the main work was done by mapper of the map reduce job. Mapper loaded the file containing positive and negative words in setup. And counted the score for each and very comment. And passed it to the rreducer of the map reduce job which simply writes it to the output.

In output of the mapper contained many things in form of composite key and composite value:

**CompositeKey**

1. Review Id

2. Product Id

3. Product Parent

4. Product Category

The following were sorted based on compreTo function(Secondary sorting was not used) in the following manner:

1. Product Category

2. Product Parent

3. Product id

4. Review Id

**Composite Value**

1. Star Rating

2. Word Count

3. Helpful votes

4. Total Votes

5. Sentiment Score

**Average Stars per Product**

This analysis counted the average of the stars given to a product by the users. This analysis was done on the result of sentiment analysis. This has two main parts mapper and reducer

1. Mapper: Sending one star per review.

2. Reducer: This was also used as combiner. It counted the average of the stars and send it to the output.

**Average Sentiment per Product**

This analysis counted the average of the sentiment score for a product. This analysis was done on the result of sentiment analysis. This has two main parts mapper and reducer just like average star per product.

1. Mapper: Sending sentiment score per review.

2. Reducer: This was also used as combiner. It counted the average of the sentiment score and send it to the output.

**Top N Analysis:**

This analysis was done to find out the top 10 results for average stars of products and average.

This analysis has two parts:

1. Mapper: The main work is done by the mapper here:

a. First initialize the SortedMap in setup of size 10.

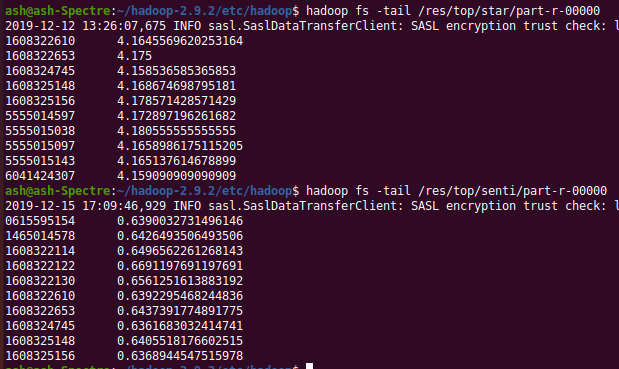
b. Second step was to go through the previous result and and put it inside the sortedMap and remove the first element when the size of the sortedMap grow bigger than 10.

c. Third, in the cleanup method empty the sortedMap and return all the value in it to the reducer.

**Result:**

What we hopped here was the rated product in the form of a star should also have highest sentiment score.

After checking we can see in top 10 there are 4 products which are same. From this we can say products with higher rating will tend to have higher sentiment score as well.

****