**Analytics System**

**Extra features**

Front end-integration: Our analytics system is integrated with our front-end so that users can seamlessly click on the analytic tab on the website and receive the output

**Pearson Correlation**

**How to use**

1. Navigate to Analytics tab from the front end and click on the correlation drop-down.
2. Click on the calculate button
3. Wait until you are brought to the next page and it will show the Pearson Correlation between price and average review length

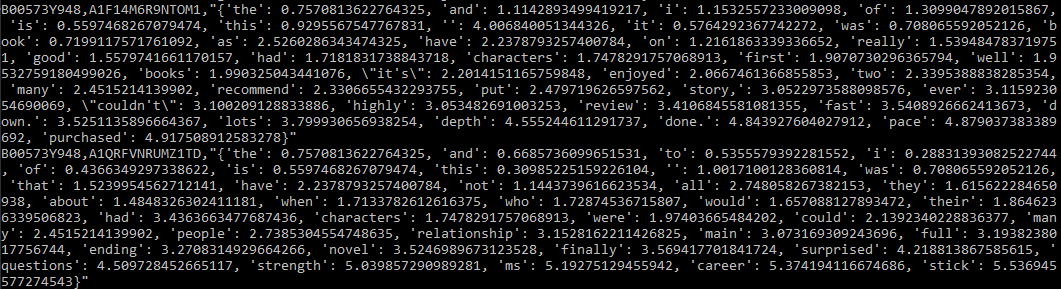
**Implementation**

1. During the automation process, a kindle\_reviews.csv and mongo\_price\_asin.csv is placed in the /project folder of the HDFS
2. Using spark, we can read both of this csv and do some pre-processing:
   1. We calculate the length of each review
   2. Use a groupBy function to calculate the mean of the review length
   3. Rename this column to be average\_reviewLength
3. We then join the two data by its ASIN and filter columns whereby price and average\_reviewLength is not NULL
4. Calculate the values of X,Y,XY,XX,YY in a map reduce fashion using rdd.map and using the pearson correlation formula for the final result

**TF-IDF**

**How to use**

1. Navigate to Analytics tab from the front end and click on the tf-idf drop-down.
2. Input word in text box to get tf-idf score of that word in all documents OR leave text box blank to get tf-idf score of all words in all the documents.
   1. If word does not exist in vocabulary, all tf-idf for all the words will be shown instead
   2. If text box is left empty, all tf-idf for all the words will be shown instead



1. Check the file outputs by typing <hdfs dfs -ls /result2> in the terminal
2. Read the file output by typing <hdfs dfs -cat /result2 [filename]> in the terminal

**Implementation**

1. During the automation process, a kindle\_reviews.csv is placed in the /project folder of the HDFS
2. Using spark, we can read this csv and do some pre-processing on the dataset such as dropping any NULL “reviewText” columns
3. We then tokenize the reviewText and build a vocabulary of known words. (our vocabSize is set to 2000)
4. We can then calculate the td-idf of each word in the vocabulary that appears in each document
5. Finally return the ASIN and a dictionary of all the tf-idf score as a CSV and save in HDFS