Fordham University

**Research on Amazon Fine Food Reviews**

Group Member: Qinhuai Gong, Kefan Zha, Chongdi Fu, Bingyue Zeng

Big Data Analytics

Professor: Yilu, Zhou

Date: May 6, 2020

**Executive Summary**

This project work on Amazon fine food review. Our group do sentimental analysis, key words frequency to see insights of the reviews. Eventually, we build a recommendation system. The purpose of our group project is figure out the relationship between scores given by customer and the polarity score we extract from sentiment analysis and list the most frequent nouns in the reviews. Most important is build a recommendation system based on those reviews. The dataset is the Amazon fine food review dataset from 1999 to 2012 containing millions of reviews on food. It contains 10 features. We use python NLTK to do the sentimental analysis and key words frequency. We use pyspark ALS algorithm to build the recommendation system. We evaluate our recommendation system by popularity, which is a feature create by ourselves, From the sentiment analysis, we find the rating could be predicted by formula: rating = 3.76 + 3.76 \* positive polarity score – 6.88 \* negative polarity score. The recommendation system will recommend 10 products to each user, and within those recommendation, the most popular products are not in the list.

**Business problem**

The purpose of this project is to explore the relationship between review score and the review polarity score which is extracted from sentiment analysis. In specific, sentiment analysis will generate reviews’ positive polarity score, negative polarity score. We want to figure out if there is a relationship between review score and comment positive polarity score and negative polarity score respectively. Then extract most frequent words used in review comments. We focus on most frequent nouns. Finally build a product recommendation system for amazon using ALS algorithm.

This project will benefit the shopping website such as amazon, eBay and forum website like yelp, reddit. In online shopping website, customer will give a rating and write a brief review to the products they bought. If the products are good, generally, customers will give high ratings and good reviews. To some extent, the review and rating represent the customers’ attitude. We use sentiment analysis and extracted polarity scores, which measures how many good words in your comment and how many bad words in your comment. By seeing the correlation between score and polarity score will give seller an insight of their products. The key words frequency will also give seller an insight of their products.

We extract most frequent words from customer reviews. This might lead to best sell products. We also build a recommendation system for amazon. It predicts top 10 products recommendations for each user, top 10 user recommendations for each food using ALS algorithm. We will also use the popularity algorithm predicting the top 10 products for each customer and top 10 customers for each product. We calculate the overlapping rate of two algorithms. Then compare the two model. Building this recommendation system could help shopping website correctly show recommended products that customers want. Shopping website can keep showing recommended products to customers. This targeted advertisement could increase the chance that customers will buy the products. In terms of forum website like yelp, the concept is the same. The recommendation system could tell most popular restaurant, store, recreational place, etc. If we can narrow down the pool of selection options for customers to a few meaningful choices, they are more likely to make a purchase.

According to our research, retail giant like Amazon credits recommender systems with 35% of their revenue. They credit recommender systems with a 29% increase in total sales, bringing their yearly sales volume in 2016 up to (wait for it) 135.99 Billion.

**Dataset:**

The name of the dataset we use is *Amazon Fine Food Reviews*. This csv file consists of reviews of fine foods from amazon. The time length of the dataset is more than 10 years, including all reviews from Oct 1999 - Oct 2012. Reviews include row ID, product ID, user ID, profile name, helpfulness numerator, helpfulness denominator, score, time, text, summary. This dataset is 287 MB. It has 568,454 reviews, 256,059 users, 74,258 products.

In building our recommendation system, we don’t use user ID and product ID, instead we create new user ID and product ID. User ID is from 1 to 256058. The product ID is from 300000 to 374257. We don’t use their ID like AF1NQ188YO7A7 or B001EQ550E. We pair the new ID with original ID one to one, and same for product ID. We split our data into 80 percent for training and 20 percent for testing. The user column is new user ID, the item column is new product ID, and define score as rating column. The new product ID is the target variable and user ID, score is the input variables when predicting the top 10 products for each user. The new user ID is the target value and new product ID, score is the input variables when predicting the top 10 customers for each product.

A close up of text on a white background

Description automatically generated A close up of text on a white background

Description automatically generated

*Figure 1: Top 10 recommended products for each customer Figure 2: Top 10 user for each product*

We create a feature called popularity. We use group by product ID can calculated the average value of score for each product. The following is the popularity table after we sort by score. We also calculated the average rating of each user for all the products he purchased.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

*Figure 3: Popularity of each product Figure 4: Average rating for each user*

**System Design and data preprocessing**

Our problem is to explore the relationship between review polarity score and customer’s rating, and extra the most frequent noun in the text. Finally build the recommendation system using pyspark by following lab 8.

In order to run correlation between polarity score and rating, we have to do sentiment analysis and get the polarity score. We only want the review done by customer which is text column. We assign all the text review to a new list df1. Then we do sentiment analysis. We extra all the polarity score of each review to a new data frame.

A screenshot of a cell phone

Description automatically generated

*Figure 5: Polarity score for each review (Fist 10 rows)*

We define the origin data frame as left dataset, the polarity score dataset as right dataset. We use left join combing two dataset into one.

A screenshot of a cell phone

Description automatically generated

*Figure 6: Combined dataset*

Then we do correlation of positive polarity score, negative polarity score and rating.

In terms of key words frequency analysis, we use the text reviews as well. We eliminate all the stop words. Then we use porter stemmer and wordnet lemmatizer to see the overall most frequent words.

A close up of a logo

Description automatically generated A close up of a logo

Description automatically generated

*Figure 7: Frequent words in porter stemmer. Figure 8: Frequent words in wordnet lemmatizer*

Then we use pos tag to extra the nouns. We keep words that the tag is NN which means the nouns.

A close up of text on a white background

Description automatically generated

*Figure 9: Most frequent nouns using pos tag*

We build the recommendation system using ALS algorithm and evaluate by popularity algorithm.

In terms of ALS algorithm, we use new user ID, new product ID, score. As we talked before, we create new user ID from 1 to 256059 and assign each user ID, so for the product ID.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

*Figure 10: New User ID Figure 11: New Product ID*

Then combined this two data frame to the origin dataset using index of origin ID. Then we want to calculate popularity. If those high rating products would be recommended or not. We group by products ID and calculated the average score of each product. We choose top 10 popular products and see if they are in recommendation list and calculating overlap rate.

**Algorithm Development**

The first approach we used is keyword frequency by NLTK, and the input of this approach is reviews. However, we have to randomly choose 20000 reviews from the dataset because the raw dataset is too large to be processed on our computer. According this approach, we can get the keyword frequency list of reviews in datasets. Meanwhile, we use two different stemmers, which are porter stemmer and wordnet lemmatizer, to present the keyword frequency list of reviews. Meanwhile, we also focus on the nouns of keyword frequency list, so that we can get some useful information such as the most popular product customers mentioned and the point how they judge the product.

The second approach we used is sentiment analysis by NLTK, and the input of this approach is reviews, so that we can get positive score and negative score on each review in the dataset. After gathering the sentiment score on each review, we also build a regression and correlation to discover the relationship between sentiment score and review score.

On the recommendation system part, the first approach we used for recommendation is popularity-based recommendation system, and the inputs of this approach is Product ID and Score. Therefore, we can sort the score and get the 10 most popular products in Amazon Fine Food.

On the other hand, we also use Alternating Least Squares (ALS) algorithm, a popular algorithm on building recommendation system, to make fine food recommendation for each user on Amazon. However, we did meet a challenge on implementing ALS algorithm for recommendation system, and it shows that the datatype of User ID and Product ID should be numeric but the datatype of User ID and Product ID in raw datasets is string. Therefore, we used python to preprocess our raw dataset to transfer the datatype of User ID and Product ID to numeric, which means that we assigned a specific number to each user and product so that we can build the recommendation system with ALS algorithm successfully. The inputs of the model are User ID and Product ID, and we assign 10 products’ recommendation to each user in the result. Meanwhile, we also can get 10 users’ recommendation for each product.

**Results and Evaluation**

In the sentiment analysis, we found average positive polarity score is 0.19, average negative polarity score is 0.042. We do correlation between positive polarity score and score given; the correlation coefficient is 0.40. P value for both variables are significant. We conclude the higher positive polarity score leads to higher rating and higher negative polarity score leads to lower rating.

In the key word analysis, we extracted top 30 most frequent nouns from the customer reviews. This is a line chart displaying the frequency of top 30 nouns in customer reviews. As we can see coffee appears 5289 times, food appears 4419 times, tea appears 3933 times, etc. We find the drink such as tea, coffee and sweet food such as sugar, bit, appear most frequently.

A close up of a map

Description automatically generated

*Figure 12: Top 30 frequent nouns*

Finally, for the recommendation system, we figure out top 10 food for each user and top user for each food.

A picture containing building, window, large, white

Description automatically generated

*Figure 13: Recommendation of top 10 products for each user*

Then we use popularity to count the most popular products and see if those products are in the recommendation system using ALS algorithm. The overlapping rate is 0. Which means the recommendation will not recommend the most popular product. After we check those products, we found those products are rice, bottle water, milk, etc. that is necessity in our life. People will buy them if they use them out and don’t need to remind customer to buy those kinds of products. Therefore, they are not in the recommendation list. That’s why overlapping rate is 0.

According to the evaluation, the shopping website could recommend those products to the customer if their cart don’t have them. Giving them a coupon would be a good way to remind or lead customer to buy those products. Advertise those products in their home page would be a better way to narrow down customer’s choice.

**Conclusion and Lessons Learned**

From the sentimental analysis, we find there is correlation between polarity score and customer rating. The higher positive polarity score, the higher customer rating, and the higher negative polarity score, the lower customer rating. The formula is rating = 3.76 + 3.76 \* positive polarity score – 6.88 \* negative polarity score. We list top 30 most frequent words in key words frequency analysis. We build a recommendation system recommend top 10 products for each user. Within those products, most popular products are not in the list.

In the project, big size data is a challenge for use. It is hard for us to read and process in our computer. It would be huge computation burden in our laptop. We use google cloud platform to read and manipulate with those datasets. On the other hand, the challenge we met on building the recommendation system for Amazon Fine Food is that the data type of ALS algorithm only supported the numeric data, but all our datatype in input are string instead of numeric. Therefore, we need to pay attention on the datatype of inputs before we build the recommendation system by ALS algorithm in the future.

Reference:

1. <https://cloud.google.com/dataproc/docs/tutorials/bigquery-sparkml>
2. <https://github.com/boomsaka/spring2020-bigdata/blob/master/Rec%20spark%20on%20GCP%20DEMO.ipynb>