ID2221: Project Report

Haonan Liu
haonan@kth.se

Kaidi Xu
kaix@kth.se

September 2021

1 Problem Description

With increase in the number of users in the digital environment, e-commerce today is expanding rapidly and has noticed a massive growth. As rivalry in the market is very strong, e-commerce companies face significant challenges to keep their clients and improve their user experience. The cure would be using big data analysis to explore user behaviors so that to improve the performance of personalization, pricing dynamics, recommendation systems, and the trust base of the consumers in the e-commerce industry.

In this project, we plan to build a real-time analyzer for an e-commerce system. The analyzer can receive and process two types of data:

- orders, i.e. purchased product, purchased time, price, payment info
- reviews, i.e. order, content, time

Based on these two types of data, we will be able to utilize Kafka and Spark to analyze them and produce valuable information. The goal of the project is to answer the following questions:

- What are the top 10 rated products?
- What are the top 10 purchased products?

2 Tools

- Language: Python, Scala
- Streaming Processing: Spark Structured Streaming
- Messaging System: Kafka
- Storage: CSV files
- Sentiment Analyzer: SparkML

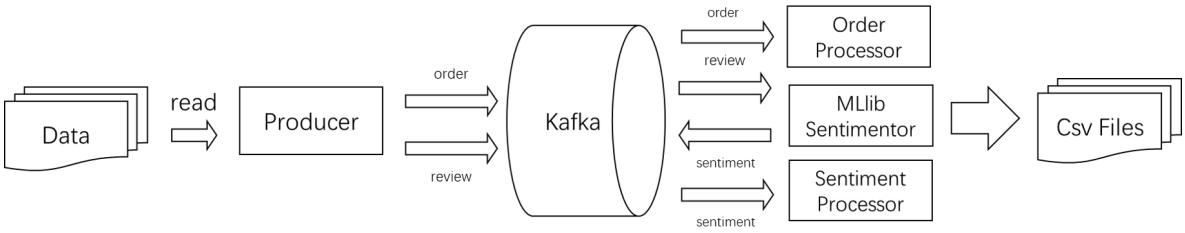

Figure 1: System overview

3 Data

We plan to use a Brazilian E-Commerce Public Dataset by Olist [1]. The dataset has 100k purchase records from 2016 to 2018 and part of them have review records, anonymized customer information, seller information, and product information. In our project, although the dataset are static files, we will read the purchase and review records in stream to simulate those behaviors.

4 Methodology

To achieve the goals of our project, we developed a data pipeline as in the Figure 1 including the following steps:

1. A producer that reads from files and produces data in a streaming based on the timestamps is implemented.
2. A Kafka with 5 topics(order, review, sentiment, product, payment) is set up.
3. A sentimentator model based on MLlib is trained with 80 percents of the review text data.
4. A sentimentator module that accepts review topic from Kafka and produces sentiment topic to Kafka is implemented.
5. A sentiment processor module that accepts sentiment topic from Kafka and produces top 10 rated products to CSV files is implemented.
6. An order processor module that accepts product topic from Kafka and produces top 10 purchased products to CSV files is implemented.

5 How to run the code

In short, by runing the following steps, you can run the project:

1. Start the zookeeper, kafka, hadoop services

2. Start all the consumers

3. Start the producer and the visualization script

6 Descriptions

- In producer.ipynb, we read from the data files and create new Dataframes including orders, reviews, products and by join() method to combine the information we need. We separate the data by group-by-date respectively to produce streaming data based on the implemented timestamps. Then we use producer.send() to respectively put the (key, values) pairs into the topics we set up in Kafka, which include 'order', 'review'.

- In consumer_review.py, we explore sentiment analysis using Spark Machine learning Data pipelines. We first work with reviews data and build a machine learning model to classify reviews as positive or negative. In the model training, we use the StopWordsRemover to filter out words which should be excluded, because the words appear frequently and don't carry as much meaning. Then we implement TF-IDF(Term Frequency–Inverse Document Frequency) feature extractor in SparkMLlib to convert text words into feature vectors. Here a CountVectorizer is used to convert the array of word tokens from the previous step to feature vectors of word token counts. IDF takes feature vectors created from the CountVectorizer and down-weights features which appear frequently(at least 3 times) in a collection of texts. A logistic regression classifier will train on the vector of labels(i.e. review score) and features and return a model.

  Finally, we put the processors and model together into a Pipeline to specify an ML workflow for training and using the model. Then we use pipeline.fit() method to return a fitted pipeline model.

  Next, we use the saved sentiment analysis model(i.e. ML Pipeline) with streaming data from review topic of Kafka to do real-time analysis and produce a sentiment topic to Kafka. Noted that here we turn the label index of the prediction Dataframe to string and label the reviews as 'pos' or 'neg'(i.e. positive or negative).

- In consumer_sent.scala, a sentiment processor module that accepts sentiment topic from Kafka and produces top rated products to top_rated.CSV file is implemented.

- In visualization.ipynb, we can read data of top 10 rated products of everyday from the top_rated.CSV file.

- Similar to the sentiment analysis above, we implement an order processor module to accept topic from Kafka and produce top purchased products to top_products.csv file. Then a visualization of the top 10 purchased products is implemented.

	product_id	pos_cot
99a4788cb24856965c36a24e339b6058		84
53759a2ecddad2bb87a079a1f1519f73		50
aca2eb7d00ea1a7b8ebd4e68314663af		45
368c6c730842d78016ad823897a372db		44
7c1bd920dbdf22470b68bde975dd3ccf		42
e53e557d5a159f5aa2c5e995dfdf244b		42
422879e10f46682990de24d770e7f83d		41
e0d64dcfaa3b6db5c54ca298ae101d05		40
389d119b48cf3043d311335e499d9c6b		38
89b121bee266dcd25688a1ba72eefb61		37

Figure 2: Top 10 rated products

	product_id	cot
99a4788cb24856965c36a24e339b6058		456
aca2eb7d00ea1a7b8ebd4e68314663af		425
422879e10f46682990de24d770e7f83d		352
d1c427060a0f73f6b889a5c7c61f2ac4		313
389d119b48cf3043d311335e499d9c6b		309
53b36df67ebb7c41585e8d54d6772e08		304
368c6c730842d78016ad823897a372db		291
53759a2ecddad2bb87a079a1f1519f73		287
154e7e31ebfa092203795c972e5804a6		262
2b4609f8948be18874494203496bc318		254

Figure 3: Top 10 sold products

7 Results

- The prediction accuracy of the trained sentimentator model on test-set is 0.864.
- The visualization of top 10 rated products is shown in Figure 2, in which we randomly select output of a day as an example.
- The visualization of top 10 purchased products is shown in Figure 3, in which we randomly select output of a day as an example.

References

[1] E-commerce public dataset by olist. https://www.kaggle.com/olistbr/brazilian-ecommerce?select=olist_orders_dataset.csv. Accessed: 2021-09-30.