High Performance Computing and Big Data Assignment

Bo Kwok

Declaration

No part of this project has been submitted in support of an application for any other degree or qualification at this or any other institute of learning. Apart from those parts of the project containing citations to the work of others, this project is my own unaided work. This work has been carried out in accordance with the Manchester Metropolitan University research ethics procedures.

Signed: Bo

Date: 5th May 2025

Step 2

This report analyzes consumer reviews of Amazon products. The original hypothesis is "In the second quarter of 2014, products given a review rating of 3 or more are significantly different compared to other products". as requested "to make the hypothesis more precise and testable from the data available in the dataset", the hypothesis is refined to "In Q2 2014 (April 1 to June 30, 2014), products with an average rating of 3 or higher received significantly more reviews than products with an average rating below 3." The project is conducted in Google Colab by PySpark to process this dataset.

The Amazon dataset has four columns: userid, productid, Rating and timestamp, which represents the id of every user, id of every product, rating of the corresponding product by the corresponding user and Unix Time of the rating respectively. As the time range of the hypothesis is within the second quarter of 2014, which is April 1 to June 30, 2014. The Unix time of April 1 and June 30 of 2014 are defined as 1396310400 and 1404172799 respectively. The dataset will be filtered using this time range and create a new dataframe df_20140401_20140630, the products are grouped with their product_id to calculate the average rating received from customers by each product and the number of reviews for each product received in the time period. The dataset has been split into 2 parts, which is products with an average rating of 3 or more and products with an average rating of less than 3.

To find out whether the number of reviews for the high rated group is significantly more than low rated group, a one-tailed, two-sample t-test is performed using ttest_ind function from scipy.stats package. The statistical outputs of the t-test are t_statistic and p-value, t_statistic represents the size of the difference between the groups, while p-value is the probability that the difference of results occurred by random chance, if the null hypothesis is true. P-value =0.05 is considered as commonly used statistical significance in most hypothesis tests. The mean of two groups are calculated by agg(mean()) function of PySpark. As this is a two-sample one-tailed t-test, the p-value should be divided by 2 and the mean number of reviews of high-rated product is more than low-rated products to determine whether the hypothesis is accepted.

For the results of the statistical analysis, there are 2 values, which are p-value and mean number of reviews of both high-rated and low-rated groups. As the p-value = 7.5590900327766065e-168, and the mean number of high-rated group reviews are larger than low rated group, the hypothesis "In Q2 2014 (April 1 - June 30, 2014), products that received an average rating of 3 or higher had significantly more reviews compared to products with an average rating below 3" is accepted. To recheck whether the hypothesis is correct, statistical analysis is done in ipynb. The average number of high and low rated reviews in Q2 2014 has been calculated (5.978 for high rated and 2.691 for low rated) as in figure 1. The number of reviews distribution for high rated and low rated product in Q2 0214 in figure 2 also support the hypotheses. In figure 3, the mean of average rating for high and low rated product for April, May and June in 2014 does not have significant difference (High rated: 4.428-4.434 and Low rated: 1.532-1.543), further confirming the hypothesis.

There are lots of wisdom gained from the data, as the hypothesis above is accepted, product with high ratings may encourage users to fill in the product review. When handling with big data, utilizing cloud-based platforms such as Azure Labs and Google Colab are good choices because they supports various techniques, it is also convenient to process huge data in a distributed environment. UNIX timestamps are widely used in database and systems, as they are universally standardized, storing in UTC and representing in seconds so that it avoids confusion with different time zones. They are easy to filter as they use a bunch of number to stores time, they are also efficient to store as they do not contain any punctuations instead of using numbers only.

There are some measures to obtain a conclusive result in the future, such as extending the time frame such as multiple quarters or years to test whether the trend is consistent. Enlarging the dataset can obtain more information about the product reviews of Amazon products, adding product categories, brands and prices to the dataset allows further analysis by grouping products by their features. Adding user information to the dataset, such as age, gender and location helps finding out which types of users have a better impression of Amazon products. It helps other departments such as marketing, product designing to target those customers. Trying other hypotheses is also a good option to learn thoroughly about the dataset.

Step 3:

This task is using Hadoop, Spark and Kafka to design and implement a big data pipeline for OpenWeatherAPI as the data source in Google Colab. The pipeline is to simulate the data processing scenario in reality. As this task is executed in Google Colab instead of Azure Lab virtual machine, Scala programming language is not applicable in the task.

The big data pipeline in mini version of my code is to concatenate the techniques of Kafka, Spark and Hadoop to use a real-world dataset like OpenWeatherAPI. Kafka ingests streaming data, Spark provides a distributed computing platform for processing machine learning task, while Hadoop stores the dataset. They are processed in environments like Google Colab.

Kafka's responsibility is to deliver messages in the pipeline, it is configured with Zookeeper. The weatherAssignment topic is created to act as a channel to receiving and sending messages, the API_KEY and city are defined in advance, so it can put into the url, it would be convenient to change to another cities or API_KEYs. After that, the Kafka producer uses OpenWeatherAPI to collect real-time information weather of Hong Kong. The responses are converted into JSON format and sent to the Kafka weatherAssignment topic for every 60 seconds for 60 loops. Kafka ingests real-time data continuously and streams to the pipeline using the procedure above.

Spark is used to process the streamed weather data from Kafka. A PySpark session is developed for Apache Spark 3.44 and connected to Kafka using the spark-sql-kafka connector. The schema is created to receive the structure field which will be read later, then Spark reads the data from Kafka topic and decodes the JSON message to convert into a structure dataframe. Then Spark provide a platform for further processing of data such as machine learning, Linear regression model is applied to the data for the machine learning. As the linear regression model from pyspark.mllib has a FutureWarning: Deprecated in 2.0.0., the linear regression model from pyspark.ml is also used in the ipynb.

Hadoop is used to provide the storage for the processed data. Hadoop 3.3.6 is installed in the Google Colab. After Spark processed the data, the dataframe is transferred to Hadoop in Parquet format with append mode. Parquet allows efficient storage of big data as it is a columnar storage format, this format is optimized for the workloads of big data.

Appendix

Google drive link:

Bo Kwok Big Data Step 1.ipynb

https://drive.google.com/file/d/12qxtWDXkHL0-byB9Ao3RazOa_Qqe7C8b/view?usp=sharing

Bo Kwok Big Data Step 3.ipynb

https://colab.research.google.com/drive/1VxhWX90SBNwVBwta5N4JOT3Aua0WJA4N?usp=sharing

Appendix (Graph)

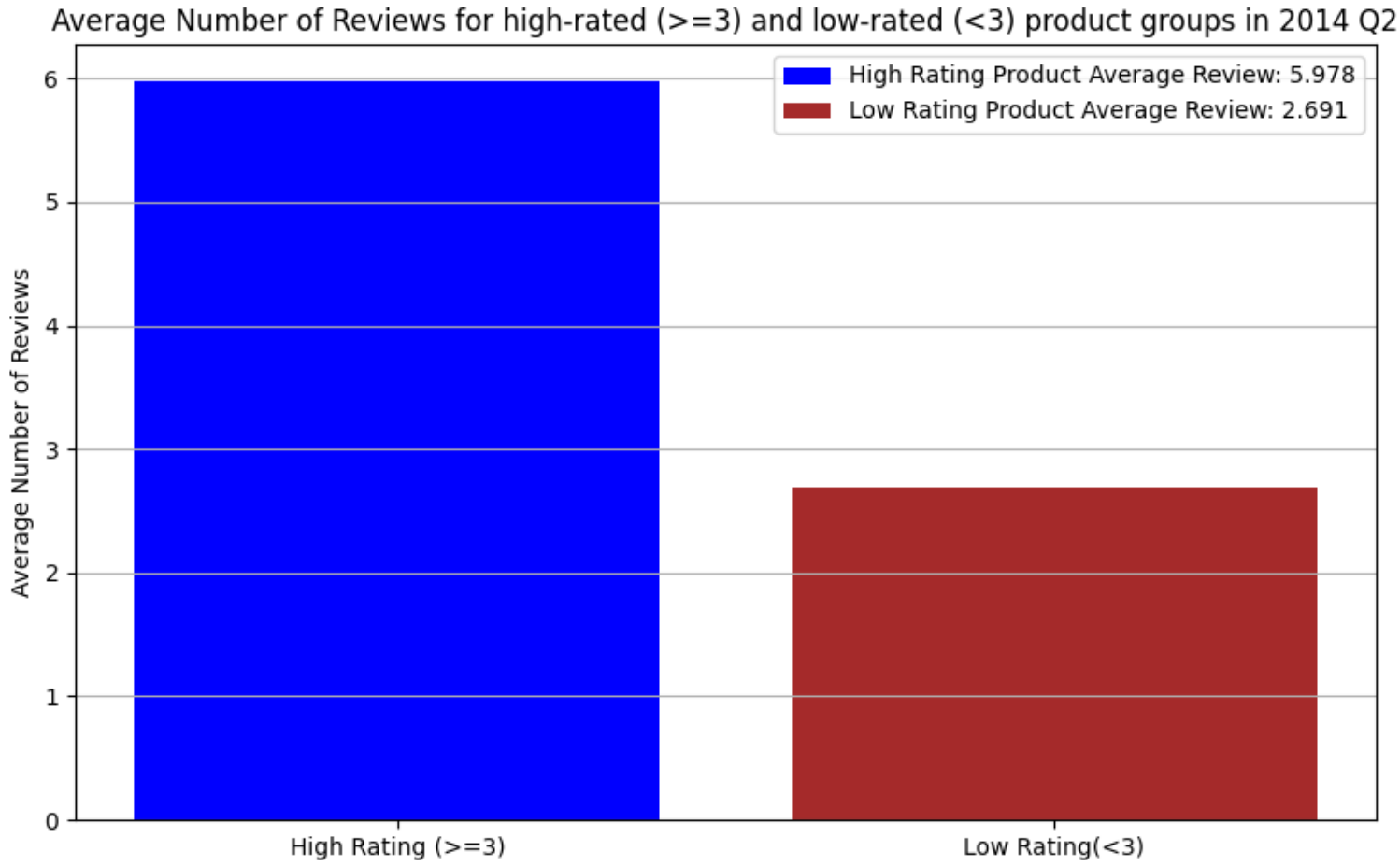

Figure 1 : Average number of reviews for high-rated (>=3) and low-rated (<3) product groups in 2014 Q2

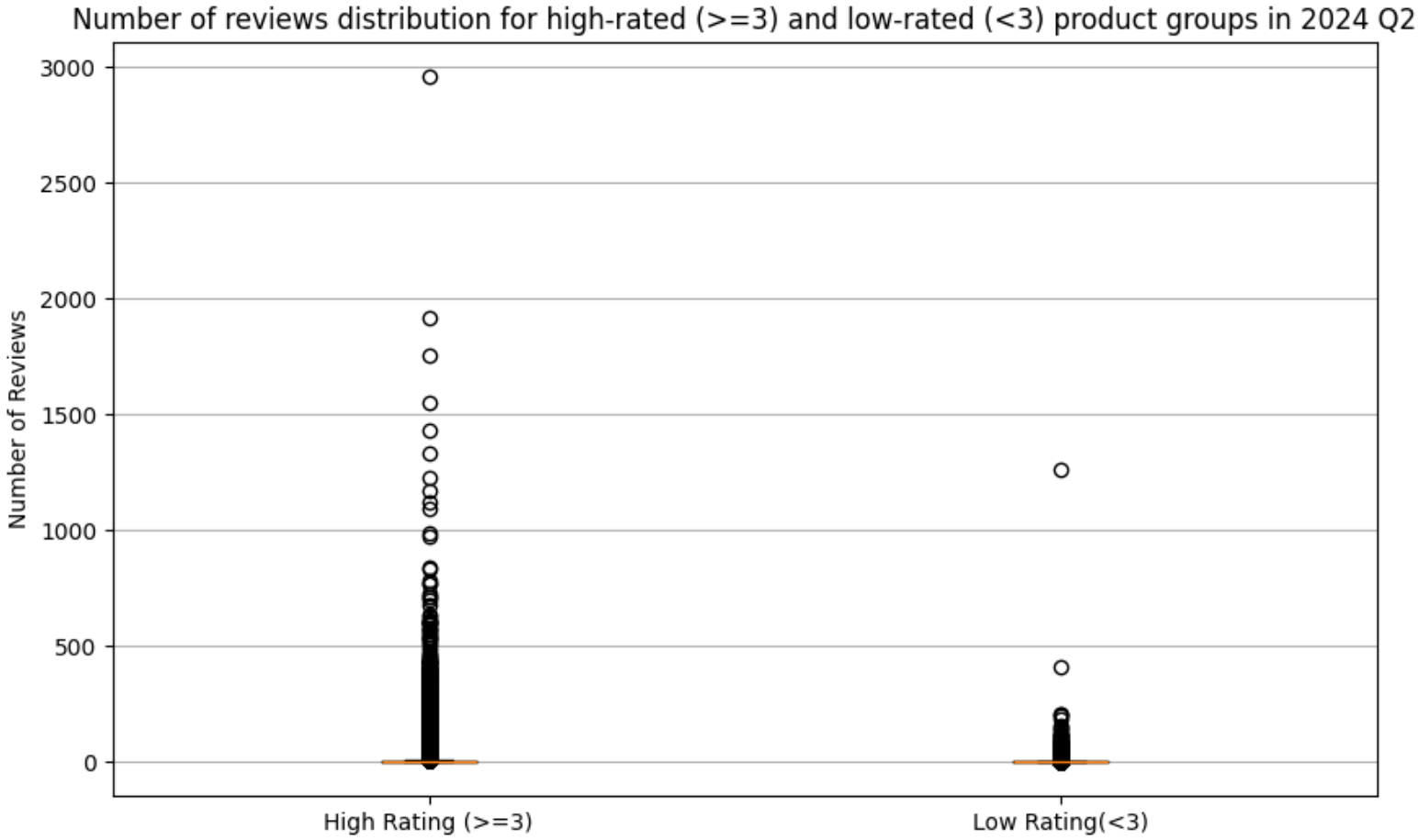

Figure 2 : Number of reviews distribution for high-rated (>=3) and low-rated (<3) product groups in 2024 Q2

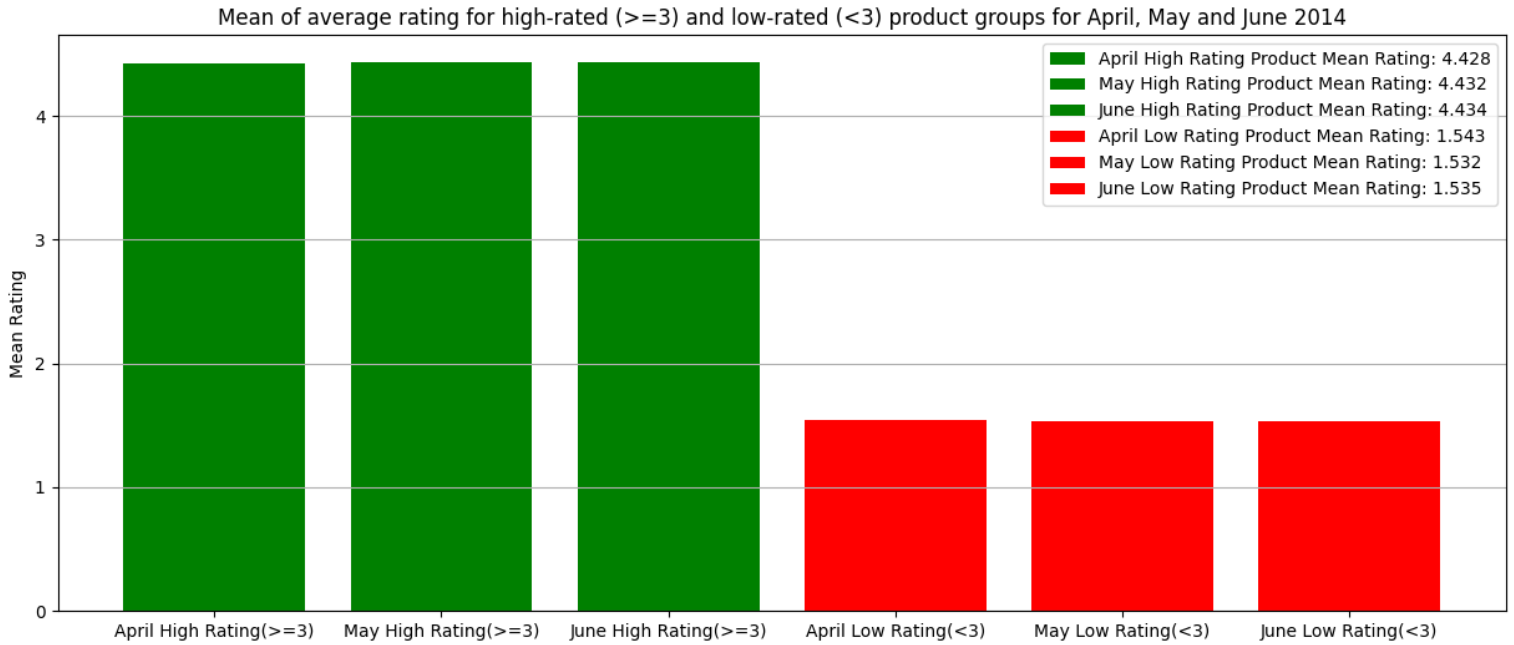

Figure 3 : Mean of average rating for high-rated (>=3) and low-rated (<3) product groups for April, May and June 2014

References:

Ajao, S. (2025) 'Introduction to Big Data', *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 12 April.

Ajao, S. (2025) 'Real-Time Data Streaming with Kafka– Week 4, *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 13 April.

Ajao, S. (2025) 'Spark2 Architecture, *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 13 April.

Ajao, S. (2025) 'Spark MLlib – Week 3, *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 12 April.

Allen, L. (2015) 'Spark Dataframes and MLlib'. Available at: https://www.lucasallen.io/spark-dataframes-mllib-tutorial/ (Accessed: 16 April 2025).

Anand, S. (2020) *Amazon Product Reviews*. Kaggle. Available at: https://www.kaggle.com/datasets/saurav9786/amazon-product-reviews (Accessed 5 May 2025).

Apache Software Foundation (2019). *Cluster Setup*. Available at: https://hadoop.apache.org/docs/r3.1.2/hadoop-project-dist/hadoop-common/ClusterSetup.html (Accessed: 12 April 2025).

Bane, M. (2025) 'GPU programming with CUDA, *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 19 April.

Bane, M. (2025) 'Message Passing Interface (MPI), *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 19 April.

Bane, M. (2025) 'Timing Methodology & Scalabity, *High Performance Computing & Big Data*. Manchester: Manchester Metropolitan University, 18 April.

Chen, B. (2024). *Building a real-time weather dashboard using Apache Kafka and Python*. Medium, 21 December. Available at: https://medium.com/@devcharlie2698619/building-a-real-time-weather-dashboard-using-apache-kafka-and-python-7de2d09ab9f8 (Accessed: 14 April 2025).

Chaudhary, V. (2020). *Pyspark Kafka Structured Streaming Data Pipeline*. Towards AI. Available at: https://pub.towardsai.net/pyspark-kafka-structured-streaming-data-pipeline-803095b7398a (Accessed 15 April 2025).

Epoch Converter (2025). *Epoch & Unix Timestamp Conversion Tools*. Available at: https://www.epochconverter.com/ (Accessed: 6 April 2025).

GeeksforGeeks. (2023). *Dynamic memory allocation in C using malloc(), calloc(), free() and realloc()*. Available at: https://www.geeksforgeeks.org/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/?ref=shm (Accessed: 26 April 2025).

GeeksforGeeks (2021). *PySpark – Linear Regression (Get coefficients)*. [online] Available at: https://www.geeksforgeeks.org/pyspark-linear-regression-get-coefficients/ [Accessed 16 April 2025].

Piatra (2024). *Highlight selected text in a web page with JavaScript*. [online] GitHub Gist. Available at: https://gist.github.com/piatra/0d6f7ad1435fa7aa790a [Accessed 14 April 2025].

SciPy (2024). *scipy.stats.ttest_ind — SciPy v1.13.0 Manual*. [online] Available at: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html [Accessed 11 April 2025]

Stack Overflow (2017). *PySpark: passing list/tuple to toDF function*. Available at: https://stackoverflow.com/questions/43747723/pyspark-passing-list-tuple-to-todf-function (Accessed: 8 April 2025).

Stack Overflow (2018). *How to convert timestamp column to epoch seconds?* [online] Available at: https://stackoverflow.com/questions/51270784/how-to-convert-timestamp-column-to-epoch-seconds/51270785 [Accessed 9 April 2025].