RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Assignment

Course Code	CSC409A
Course Name	Data Analytics
Programme	B.Tech
Department	CSE
Faculty	FET

Name of the Student	Satyajit Ghana
Reg. No.	17ETCS002159
Semester/Year	VIII/2021
Course Leader(s)	E. Ami Rai

Declaration Sheet

Student Name	Satyajit Ghana		
Reg. No	17ETCS002159		
Programme	B.Tech	Semester/Year	08/2021
Course Code	CSC409A		
Course Title	Data Analytics		
Course Date		to	
Course Leader	E. Ami. Rai		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

List of Figures

1 Question 1

Solution to Question No. 1 Part A

1.1 Introduction to Data Analytics and its applications

The analysis of data to extract knowledge is the subject of a vibrant area known as data analytics, or simply "analytics". The definition adopted here is:

> **Analytics** The science that analyze crude data to extract useful knowledge (patterns) from them

In 'Competing on analytics', Thomas Davenport defines analytics as "the extensive use of data, statistical and quantitative analysis, exploratory, predictive models, and fact-based management to drive decisions and actions."

This process can also include data collection, organization, pre-processing, transformation, modeling and interpretation. Analytics as a knowledge area involves input from many different areas. The idea of generalizing knowledge from a data sample comes from a branch of statistics known as inductive learning, an area of research with a long history. (Moreira, J, 2019)

Taxonomy of Data Analytics

- Descriptive analytics: summarize or condense data to extract patterns
- Predictive analytics: extract models from data to be used for future predictions.

For example, a sales report of a company, say Pepsi. This report will tell you how many units of Pepsi were sold, where they were sold, what price and a lot of other things. All of this is information coming from the data. All you are doing is slicing and dicing the data in different ways, looking at it from different angles, along different dimensions etc. There is very little statistics involved in descriptive analytics and so you don't really need to be a statistical wiz to be able to do effective descriptive analytics.

While descriptive analytics is a very powerful tool, it is still giving us information about the past. Whereas, a business owner's primary concern is the future. If I run a hotel, I want to be able to predict how many of my rooms will be occupied next week. If I am a drug company, I want to know which of my under-test drugs is most likely to succeed. This is where predictive analytics comes in.

Coming to Data Deluge, *WE'RE DROWNING IN DATA*. Supermarkets, credit cards, Amazon and Facebook. Electronic medical records, digital television, cell phones. The universe has gone wild with the chirps, clicks, whirs and hums of feral information. And it truly is feral: According to a 2008 white paper from the market research firm International Data Corp., the

amount of data generated surpassed our ability to store it back in 2008. The cat is out of the bag. In 2010, the amount of digital information — from highdefinition television signals to Internet browsing information to credit card purchases and more — created and shared exceeded ***1 zettabyte*** for the first time. In 2011, it approached 2. The amount has grown by a factor of nine in five years, according to IDC, which pointed out in its 2011 report that there are "nearly as many bits of information in the digital universe as stars in our physical universe." (Stanford Medicine, Summer 2012)

1.2 Illustration with real world examples

We'll look into various real-world examples of how Data Deluge happens and affects starves information retrieval.

1. "Impressions are down. What's going on?"

This power company was concerned about impressions, believing that the quantity of people who viewed their ads was a crucial metric. To maximize impressions, they had cast a wide net with their keywords, hoping to reach as many people as possible. So, they were dismayed to see that their ads weren't getting as wide of an audience as they had hoped. That said, the company was quite happy with the click-through rate (CTR) of some of their keywords. While their average CTR was only .09%, the CTR for many of their low-impression keywords averaged .60% - almost 7 times the aggregate CTR. After examination, many of their low-impression/high-CTR keywords had nothing to do with what the company offered:

- Keywords related to gas prices had high CTRs, but were used almost exclusively by people seeking gasoline for their cars, not gas heating for their homes
- "Transformers" as a keyword attracted fans of the movies and toys, not people interested in electrical transformers
- More mystical keyword choices included "Power Rangers," "Power Ball numbers," "Monster Energy Drink," and "juicers"

Yes, these keywords were delivering clicks and impressions, but they weren't the impressions and clicks that would turn into customers and revenue. Worse, the company was spending almost 25% of their paid search budget on these non-relevant keywords.

2. "We don't have keyword-level data."

After working with members of this company's digital marketing team to build out keyword-level data and connect it to revenue, so it was astonishing to hear this. However, the two people we had worked with—the company's "stewards of data"—had left the company. When they left, not only did they take all their data-related knowledge with them, they left a void in the company for data-driven conversations. None of the remaining members of the digital marketing team knew about:

- The systems the two stewards had been using

- How to access those systems and harvest that data
- How to drill down in the data to get keyword-level information and tie it back to revenue
- The importance of regular conversations about data

(Elizabeth et al, 2018)

1.3 Discussion on the barriers for adoption

We are living in an age of information. Staggering amounts of information are collected, stored, and widely disseminated. Yet, we may be less informed and less knowledgeable than ever. This paradox of increasing information, yet decreasing knowledge and insight, has many possible causes, some of which are subtle and difficult to identify, and even more difficult to remedy. The fundamental issue is quantity crowding out quality, leading to an abundance of poor-quality information which may not be a good substitute for scarce but high-quality information. Information is not unique in exhibiting this paradox.

There are three fundamental reasons why quantity may crowd out quality. The most obvious is the production cost problem where the emphasis on quantity shifts the emphasis and resources away from quality. It is costly to produce quality information, and it is difficult to do both quality and quantity. When quality does not pay in proportion to its high cost, quantity wins over. This is also the most common explanation for non-information examples, but explanation for information products involves two other reasons. The second reason is the obsolescence problem. Information is not neutral with respect to the physical world, but it is an agent of change. Information is useful precisely because it is used to change the environment and subjugate nature and society to our purposes. But as information is used to change the environment to take advantage of new opportunities, our existing information about the environment becomes obsolete, leading to a loss of information. The net effect may be positive or negative, but it is increasingly negative as we will show, in a fast-changing information-intensive society. The third reason is the competition problem when information is used as a competitive weapon against others, to mislead and confuse others, leading to a loss of knowledge on their part. Information is power, because it can be used to control others and exploit them, by controlling their information sources, and consequently their behavior. (Orman, 2015)

Real-time expert systems and ANN would give erratic predictions for inputs that were dependent on each other or outside of the range used to develop the system. Furthermore, one could not drill down to determine the major contributors to the strange result. These systems fell into disuse as soon as the developer left the scene. There are so many failures of expert systems, it's difficult to keep track of all of them.

Top 10 Failures of Expert Systems

10. Failure to say you should have bought control valves instead of those cheap on-off valves

9. Failure to say you should have bought Coriolis meters instead of those cheap rotameters
8. Failure to explain why expert systems failed
7. Failure to explain what engineers will do when all the manufacturing is offshore
6. Failure to predict the next layoff
5. Failure to predict the last and next economic crises
4. Failure to explain what is really said in congressional bills
3. Failure to predict your drug costs under the Medicare prescription plan
2. Failure to predict what the cost of medical care will be under the new healthcare plan
1. Failure to figure out where the governor of South Carolina was last June.

(Greg McMillan, 2010)

1.4 Discussion on the future of analytics

The reliance on data driven decision making will continue to grow. Just like the widespread usage of metrics and reports today, companies will start expecting to see some predictive analytics insights as part of regular dashboards.

As analytics becomes more and more prevalent in the corporate consciousness, a basic awareness and understanding of analytical techniques will become a required skill for career growth at the middle to senior management tiers, irrespective of industry and function. There will also be an increased demand for some super specialized roles. These will require intensive expertise with programming and technology to support the actual analytics implementation.

In the next decade we will witness technological advances that will play an increasingly important role in the ability of companies to mine data for real time insights and actions in the context of the rapid pace of data produced and the variety of data that is being captured.

The future of data analytics will see data discovery and preparation change, in a practice known as augmented data preparation and discovery. Machine learning automation augments and streamlines data profiling, modelling, enrichment, data cataloguing and metadata development, making the data preparation process more flexible. Traditional methods often involve rule-based approaches to transform data. However, augmented data preparation makes the process more flexible because it automatically adapts fresh data, especially outlier variables.

Machine learning augments data discovery because the algorithms allow data analysts to visualise and narrate relevant findings easily. Machine learning also paves the way for several functions like clusters, links, exceptions, correlations predictions and data exceptions without having to rely on end-users to generate all these results. Augmented data preparation and discovery will play a huge role in the future of data analytics because it streamlines data preparation and discovery, giving analysts large sets of clean data. (Michael Dixon, 2019)

1.5 Stance taken and justification

Successfully managing the "data deluge" will allow scientists to compare the genomes of similar types of cancers to identify how critical regulatory pathways go awry, to ferret out previously unknown and unsuspected drug interactions and side effects, to precisely track the genetic changes that have allowed evolving humans to populate the globe, and even to determine how our genes and environment interact to cause obesity, osteoporosis and other chronic diseases.

I do believe there are many factors that can make the data so overpowering that information retrieval becomes difficult or even impossible in some cases, but there are a few measures that can be taken to mitigate so,

1. Countering data deluge by using the right data

The primary challenge for any company is to select the right information that serves its customers. Data is growing rapidly and it's difficult for marketing analysts to collect and analyse data on the go. Additionally, it's important to make sure that the data collected by you really reflects the purchase decisions taken by your customer base. Data deluge is going to increase in the coming time, but it is manageable if companies learn to select the right amount of data as it will help them build better and reliable customer relationships in the long run.

2. Countering data deluge by controlling costs

Traditionally, companies have always been convinced by the theory of throwing money for technology solutions. This, however, can no longer be called a sensible strategy because of the exponential growth in data that calls for sensible analysis and handpicking only "useful" data. Companies can train its data scientists to reject duplicate data and cull useful information that can save a lot of money in the long run.

3. Countering data deluge by updating and auditing policies

According to the 2011 McKinsey Global Institute report, many large U.S companies have more data stored as compared to the U.S Library of Congress and that has become a cause of concern for data managers. The only way to counter this is by having effective data retention and data destruction policies. Companies can have such policies that allow auto-deletion of inessential data after a specific period of time. Also, there must be policies which allow retrieval of data such as important emails, files, and documents in times of litigation. Companies must also diligently follow both internal as well as government compliances in the process. Moreover, while data should be transparent and available to all employees, they should be given the right to access it only for a limited time. (Naveen Joshi, 2017)

2 Question 2

Solution to Question 1 Part B

2.1 Introduction

The Data analytic lifecycle is designed for Big Data problems and data science projects. The cycle is iterative to represent real project. To address the distinct requirements for performing analysis on Big Data, step – by – step methodology is needed to organize the activities and tasks involved with acquiring, processing, analyzing, and repurposing data.

2.2 Discuss data preparation phase tools

Phase 2: Data Preparation

- Steps to explore, preprocess, and condition data prior to modeling and analysis.
- It requires the presence of an analytic sandbox, the team execute, load, and transform, to get data into the sandbox.
- Data preparation tasks are likely to be performed multiple times and not in predefined order.
- Several tools commonly used for this phase are – Hadoop, Alpine Miner, Open Refine, Storm, Spark etc.

1. Apache Spark

Spark is a framework for parallel processing of Big Data. Spark is designed to use the basis of Hadoop MapReduce with some modifications that enables it to perform more efficiently than Hadoop MapReduce. Spark has its own streaming API and independent processes for continuous batch processing across varying short time intervals.

Spark runs up to 100 times faster than Hadoop in certain circumstances, however it still uses Hadoop distributed file system. This is the reason why most of the Big Data projects install Spark on Hadoop so that the advanced Big Data applications can be run on Spark by using the data stored in Hadoop distributed file system. So, we can consider Spark as an extension of Hadoop, which has some features for real-time analytics like being fast, simple, and supportive of applications such as machine learning, stream processing, and graph computation. Xu, Wu, Xu, Zhu, and Bass implement Spark into their idea for real-time data analytics as a service. It is able to support both stream and batch processing while Hadoop is made mostly for batch processing.

Spark provides many real-time processing and evaluation options that Hadoop alone cannot. Therefore, to manage the data for their architecture, they utilize Spark specifically. Though Bilal et al. are making use of a graph database, Neo4J, to store datasets, Spark is the graph processing system being used. Their use of Spark will allow them to process the waste data and analyze it efficiently. The research on distributed computing engines shows that Spark has consistent scalability for large datasets. Yan, Huang, and Yi show Spark is scalable to process seismic data with its in-memory computation and data locality features. (Yadranjiaghdam, 2016)

2. Apache Storm

Storm is another real-time computation system. It is a task parallel distributed computing system which can reliably process unbounded streams of importing data. Storm uses an independent workflow, Directed Acyclic Graphs, in its platform. Storm utilizes Zookeeper, a minion worker to manage its processes, instead of running on Hadoop clusters. Many of the explored resources make use of Storm with their new contributions to real-time data analytics. Storm, unlike Hadoop alone, can continue to analyze data as it arrives. As Storm is a complex event processing system that has the ability to detect important event occurrences, it is the processing system that Jones utilizes to detect crucial events through the processing of Twitter feeds. (Yadranjiaghdam, 2016).

Apache Storm is based on the 'fail fast, auto restart' approach that allows it to restart the process once a node fails without disturbing the entire operation. This feature makes Storm a fault-tolerant engine. It guarantees that each tuple will be processed 'at least once or exactly once', even if any of the nodes fail or a message is lost. The standard configuration of Storm makes it fit for production instantly. Once the Storm cluster is deployed, it can be easily operated. Besides, it is a robust and user-friendly technology, making it suitable for both small- and big-sized firms.

2.3 Discuss model building phase tools

Phase 4: Model Building –

- Team develops datasets for testing, training, and production purposes.
- Team also considers whether its existing tools will suffice for running the models or if they need more robust environment for executing models.
- Free or open-source tools – Rand PL/R, Octave, WEKA, Python
- Commercial tools – MATLAB, STASTICA.

1. Python – Scikit-learn

The Python programming language is establishing itself as one of the most popular languages for scientific computing. Thanks to its high-level interactive nature and its maturing ecosystem of scientific libraries, it is an appealing choice for algorithmic development and exploratory data analysis.

One of the popular library used for building models is Scikit-learn, Scikit-learn harnesses this rich environment to provide state-of-the-art implementations of many well-known machine learning algorithms, while maintaining an easy-to-use interface tightly integrated with the Python language. This answers the growing need for statistical data analysis by non-specialists in the software and web industries, as well as in fields outside of computer-science, such as biology or physics. Since it relies on the scientific Python ecosystem, it can easily be integrated into applications outside the traditional range of statistical data analysis. Importantly, the algorithms, implemented in a high-level language, can be used as building blocks for approaches specific to a use case, for example, in medical imaging (Dubois, 2007; Milmann and Avaizis, 2011)

Strong points

– General purpose, open-source, commercially usable, and popular Python ML tools.
– Funded by INRIA, Telecom Paristech, Google and others.
– Well-updated and comprehensive set of algorithms and implementations.
– It is a part of many ecosystems; it is closely coupled with statistic and scientific Python packages.

Weak points

– API-oriented only.
– The library does not support GPUs.
– Basic tools for NNs.

2. Weka3

Weka collects a general purpose and very popular wide set of ML algorithms implemented in Java and engineered specifically for DM (Weka3 2018; Waikato 2018) . It is a product of the University of Waikato, New Zealand and is released under GNU GPLv3-licensed for non-commercial purposes. Weka has a package system to extend its functionality, with both official and unofficial packages available, which increases the number of implemented DM methods. It offers four options for DM: command-line interface (CLI), Explorer, Experimenter, and Knowledge Flow.

Weka can be used with Hadoop thanks to a set of wrappers produced for the most recent versions of Weka3. At the moment, it supports MapReduce but not yet Apache Spark. Clojure (Hickey 2018) users can also leverage Weka, thanks to the Clj-ml library (Clj-ml 2018).

Related to Weka, Massive Online Analysis is also a popular open-source framework written in Java for data stream mining, while scaling to more demanding larger-scale problems.

Strong points

– General purpose, involving wide set of algorithms with learning schemes, models and algorithms.
– It comes with GUI and is API-oriented.
– Supports standard DM tasks, including feature selection, clustering, classification, regression and visualization.
– Very popular ML tool in the academic community.

Weak points

– Limited to Big Data, text mining, and semi-supervised learning.
– Weak for sequence modelling; e.g., time-series

(Giang, 2019)

2.4 Justify with suitable scenarios

Model Building Tools

1. Use case for Python-Scikit-learn

Machine learning for neuroimaging with Scikit-Learn – Alexandre et. Al (2014)

In this paper they have illustrated with simple examples how machine learning techniques can be applied to fMRI data using the scikit-learn Python toolkit in order to tackle neuroscientific problems. Encoding and decoding can rely on supervised learning to link brain images with stimuli. Unsupervised learning can extract structure such as functional networks or brain regions from resting-state data. The accompanying Python code for the machine learning tasks is straightforward. Difficulties lie in applying proper preprocessing to the data,

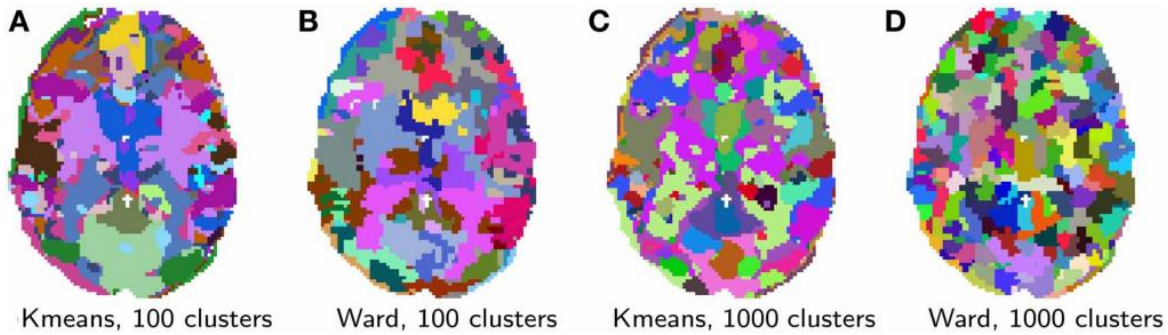


Figure 1 Brain parcellations extracted by clustering using scikit-learn

choosing the right model for the problem, and interpreting the results. Tackling these difficulties while providing the scientists with simple and readable code requires building a domain-specific library, dedicated to applying scikit-learn to neuroimaging data. This effort is underway in a nascent project, nilearn, that aims to facilitate the use of scikit-learn on neuroimaging data.

2. Use case for Weka3

Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification – Ignacio et al. (2017)

State-of-the-art light and electron microscopes are capable of acquiring large image datasets, but quantitatively evaluating the data often involves manually annotating structures of interest. This process is time-consuming and often a major bottleneck in the evaluation pipeline. To overcome this problem, we have introduced the Trainable Weka Segmentation (TWS), a machine learning tool that leverages a limited number of manual annotations in order to train a classifier and segment the remaining data automatically.

To segment the input image data (2D/3D grayscale or color), TWS transforms the segmentation problem into a pixel classification problem in which each pixel can be classified as belonging to a specific segment or class. A set of input pixels that has been labeled is represented in the feature space and then used as the training set for a selected classifier. Once the classifier is trained, it can be used to classify either the rest of the input pixels or completely new image data. All methods available in WEKA can be used.

Data Preparation Tools

1. Use case for Apache Spark

Bioinformatics applications on Apache Spark - Guo, R., Zhao, Y., Zou, Q., Fang, X. and Peng, S., 2018

Among the state-of–the-art parallel computing platforms, Apache Spark is a fast, general-purpose, in-memory, iterative computing framework for large-scale data processing that ensures high fault tolerance and high scalability by introducing the resilient distributed dataset abstraction. They surveyed Spark-based applications used in next-generation sequencing and other biological domains, such as epigenetics, phylogeny, and drug discovery.

Phylogeny reconstruction is important in molecular evolutionary studies but faces significant computational challenges. Before Spark-based tools were created, while several tools had been put forward for phylogeny reconstruction, they did not scale well, and there was a significant increase in the number of datasets. Therefore, in 2016, Xu et al. proposed CloudPhylo, a fast and scalable phylogeny reconstruction tool that made use of Spark. It evenly distributed the entire computational workload between working nodes.

An experiment was conducted using 5,220 bacteria whole-genome DNA sequences. The results showed that CloudPhylo took 24,508 seconds with one worker node, and it was able to scale well with increasing numbers of worker nodes. Moreover, CloudPhylo performed better than several existing tools when using more worker nodes. In addition, CloudPhylo achieved faster speeds on a larger dataset of about 100 Gb generated by simulation.

2. Use case for Apache Storm

Apache Storm Based on Topology for Real-Time Processing of Streaming Data from Social Networks – Batyuk, A. and Voityshyn, V. (2016)

In this paper we represented architectural concept of the Apache Storm based real-time data processing topology.

Experiments with the system allowed concluding the following:

1. The chosen toolset (mostly based on Apache Storm) was convenient in usage and shown its effectiveness on the implementation and testing stages.

2. The implemented topology demonstrated enough flexibility for the sample task. Therefore, it can be evolved in order to be applied for resolving more complex and valuable problems.

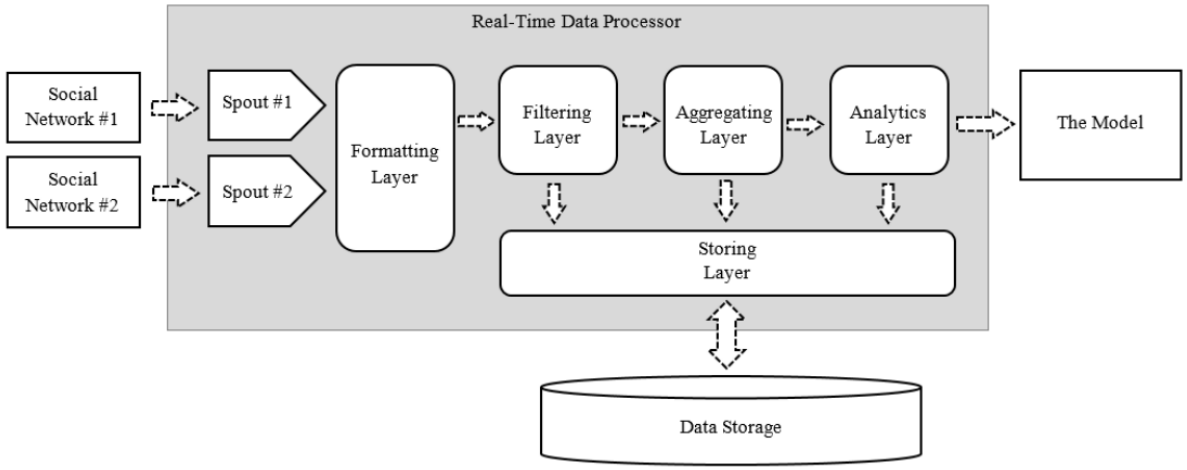


Figure 2 Real-Time Data Processing Topology with Apache Storm

3 Question 3

Solution to Question 2 Part B

3.1 Model different method(s) to address the above issues

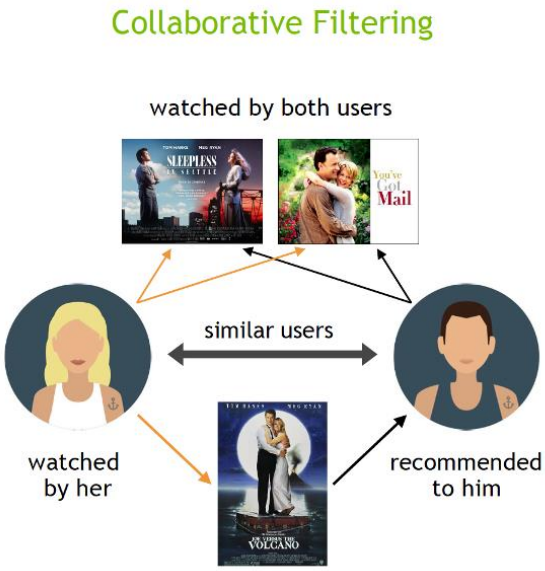
Recommender systems (RecSys) have become a key component in many online services, such as e-commerce, social media, news service, or online video streaming. However, with their growth in importance, the growth in scale of industry datasets, and more sophisticated models, the bar has been raised for computational resources required for recommendation systems.

Recommender systems are trained to understand the preferences, previous decisions, and characteristics of people and products, using data gathered about their interactions, which include impressions, clicks, likes, and purchases. Recommender systems help solve information overload by helping users find relevant products from a wide range of selections by providing personalized content. Because of their capability to predict consumer interests and desires on a highly personalized level, recommender systems are a favorite with content and product providers because they drive consumers to just about any product or service that interests them, from books to videos to health classes to clothing.

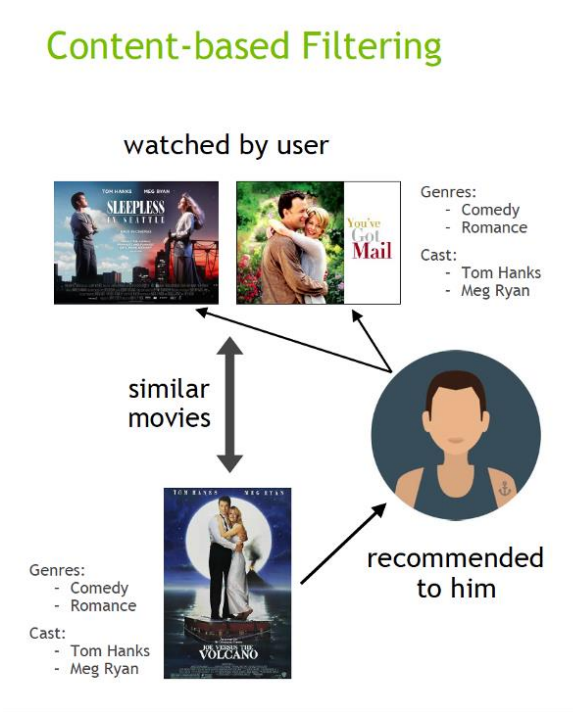
3.1.1 Types of Recommendation Systems

Traditionally, recommender systems approaches could be divided into these broad categories: collaborative filtering, content filtering, and hybrid recommender systems. More recently, some variations have been proposed to leverage explicitly the user context (context-aware recommendation), the sequence of user interactions (sequential recommendation) and the interactions of the current user session for next-click prediction (session-based recommendation).

Collaborative filtering algorithms recommend items (this is the filtering part) based on preference information from many users (this is the collaborative part). This approach uses similarity of user preference behavior, given previous interactions between users and items, recommender algorithms learn to predict future interaction. These recommender systems build a model from a user's past behavior, such as items purchased previously or ratings given to those items and similar decisions by other users. The idea is that if some people have made similar decisions and purchases in the past, like a movie choice, then there is a high probability they will agree on additional future selections. For example,



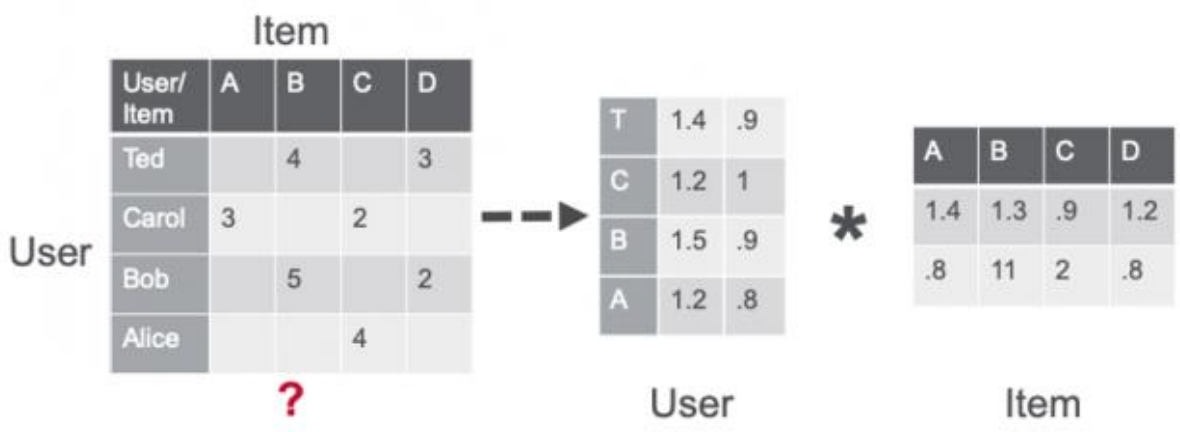
if a collaborative filtering recommender knows you and another user share similar tastes in movies, it might recommend a movie to you that it knows this other user already likes.



Content filtering, by contrast, uses the attributes or features of an item (this is the content part) to recommend other items similar to the user's preferences. This approach is based on similarity of items and user features, given information about a user and items they have interacted with, (e.g. a user's demographics, like age or gender, the category of a restaurant's cuisine, the average review for a movie), model the likelihood of a new interaction. For example, if a content filtering recommender sees you liked the movies "You've Got Mail" and "Sleepless in Seattle," it might recommend another movie to you with the same genres and/or cast, such as "Joe Versus the Volcano."

Hybrid recommender systems combine the advantages of the types above to create a more comprehensive recommending system. (Carol et al, Nvidia, 2021)

Matrix factorization (MF) techniques are the core of many popular algorithms, including word embedding and topic modeling, and have become a dominant methodology within the collaborative-filtering-based recommendations. MF can be used to calculate the similarity in user's ratings or interactions to provide recommendations.



3.2 Identify suitable attributes

To give a suitable example of how the Book Recommendation System will work we will be using the data set goodbooks-10k, this dataset contains ratings for ten thousand popular books. As to the source, let's say that these ratings were found on the internet. Generally, there are 100 reviews for each book, although some have less - fewer - ratings. Ratings go from one to five. Both book IDs and user IDs are contiguous. For books, they are 1-10000, for users, 1-53424. All users have made at least two ratings. Median number of ratings per user is 8. There are also books marked to read by the users, book metadata (author, year, etc.) and tags.

Books.csv `[ id, book_id, best_book_id, work_id, books_count, isbn, isbn13, authors, original_publication_year, original_title, title, language_code, average_rating, ratings_count, work_text_reviews_count, ratings_1, ratings_2, ratings_3, ratings_4, ratings_5 ]`

Book_tags.csv `[ goodreads_book_id, tag_id, count ]`

Tags.csv `[ tag_id, tag_name ]`

Ratings.csv `[ book_id, user_id, rating ]`

Throughout the solution of this part of Assignment we will be considering the goodbooks-10k dataset and try to build a recommender system in R.

3.2.1 Collaborative Filtering

Collaborative filtering is a standard method for product recommendations. To get the general idea consider this example:

Imagine you want to read a new book, but you don't know which one might be worth reading. You have a certain friend, with whom you have talked about some books and you typically have had quite a similar opinion on those books. It would then be a good idea to ask this friend whether he read and liked some books that you don't know yet. These would be good candidates for your next book.

What described above is exactly the main idea of the so-called user-based collaborative filtering. It works as follows:

1. You first identify other users similar to the current user in terms of their ratings on the same set of books.

   For example, if you liked all the "Lord of the rings" books, you identify users which also liked those books.

2. If you found those similar users you take their average rating of books the current user has not yet read …

   So, how did those "Lord of the rings" lovers rate other books? Maybe they rated "The Hobbit" very high and recommend those books with the highest average rating to him.

   Accordingly, "The Hobbit" has a high average rating and might be recommended to you.

These three steps can easily be translated into an alogrithm.

However, before we can do that, we have to restructure our data. For collaborative filtering data are usually structured that each row corresponds to a user and each column corresponds to a book. This could for example look like this, for 3 users and 5 books. Note that not every user rated every book. For example, user 1 only rated book 3, while user 2 rated book 1 and book 2.

```
##         book_id
## user_id  1  2  3  4  5
##       1 NA NA  4 NA NA
##       2  2  1 NA NA NA
##       3 NA NA  3 NA  3
```

Step 1: Find Similar Users

For this step we select users that have in common that they rated the same books. To make it easier let's select one example user `"David" (user_id: 17329)`. First, we select users that rated at least one book that David also rated. In total there are 440 users who have at least one book in common.

```
current_user <- "17329"
rated_items <- which(!is.na((as.data.frame(ratingmat[current_user, ]))))
selected_users <- names(which(apply(!is.na(ratingmat[ ,rated_items]), 1, sum) >= 2))
head(selected_users, 40)
```

```
##  [1] "35"   "153"  "158"  "202"  "343"  "368"  "958"  "1169" "1185" "1339"
## [11] "1449" "1456" "1464" "1518" "1571" "1634" "1677" "1759" "2166" "2218"
## [21] "2347" "2421" "2467" "2619" "3050" "3075" "3246" "3263" "3399" "3580"
## [31] "3641" "3662" "3757" "3796" "4005" "4204" "4242" "4276" "4289" "4489"
```

For these users, we can calculate the similarity of their ratings with "David" s ratings. There is a number of options to calculate similarity. Typically, cosine similarity or pearson's correlation coefficient are used. Here, we chose pearson's correlation. We would now go through all the selected users and calculate the similarity between their and David's ratings.

$$Pearson's\ Correlation = r = \frac{\sum(x_i - x)(y_i - y)}{\sqrt{\sum\ x_i - x\ \ y_i - y\ ^2}}$$

We can calculate the similarity of all others users with David and sort them according to the highest similarity.

```
similarities <- cor(t(rmat[rownames(rmat)!=current_user, ]), rmat[current_user, ], use =
'pairwise.complete.obs')
sim <- as.vector(similarities)
names(sim) <- rownames(similarities)
res <- sort(sim, decreasing = TRUE)
head(res, 40)
```

```
##      1339      1449      2347      6290      8682     12320     17404
## 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##     18199     18373     19481     20859     21766     24420     27920
## 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##     28448     29356     31754     32058     40579     44408     45602
## 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
##     47653     49299     51612     52297      5109     41018      6660
## 1.0000000 1.0000000 1.0000000 1.0000000 0.9622504 0.9045340 0.8944272
##      2218      3246      9787     43956     37746     25340       202
## 0.8660254 0.8660254 0.8660254 0.8660254 0.8528029 0.8473185 0.8164966
##     30761       158     48822     30601     31305
## 0.7717436 0.7385489 0.6827243 0.6741999 0.6688701
```

We can now select the 4 most similar users: 1339, 1449, 2347, 6290

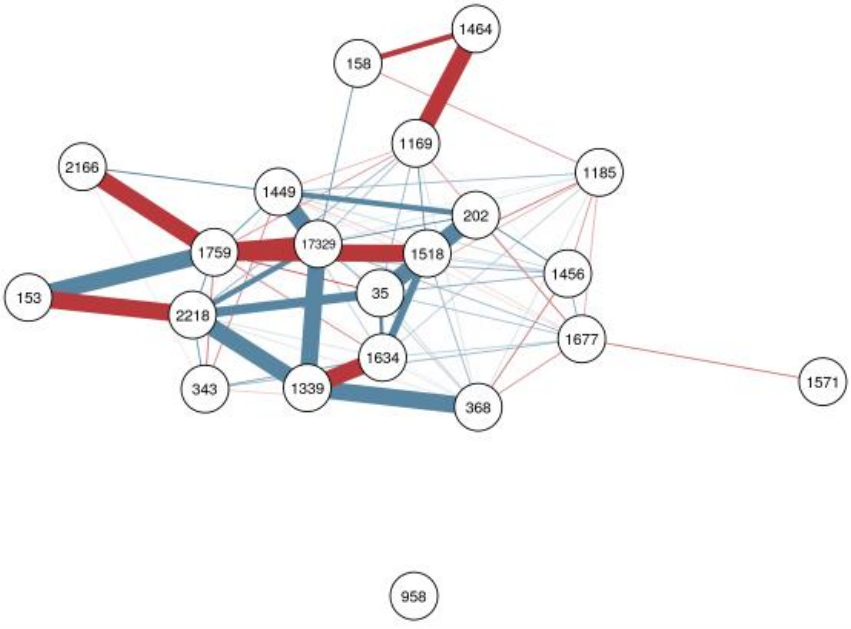


Figure 3 User Similarity Graph

Summary

In collaborative filtering the main idea is to use ratings from similar users to create recommendations. The basic algorithm is easy to implement by hand. *(Book Recommender, Philipp Spachtholz, 2017)*

More advanced ideas

The procedure shown above is a quite simple process. You can do other smart things to improve the algorithm:

- Instead of simply averaging the predictions of the similar users you can weight the ratings by similarity. This means that the more similar a user is to the current user the more weight his/her ratings receive in the calculation of the predictions.
- The similarity calculation can also be weighted, according to how many books users co-rated. The more books users co-rated the more reliable is their similarity score.

As the growth in the volume of data available to power recommender systems accelerates rapidly, data scientists are increasingly turning from more traditional machine learning methods to highly expressive deep learning models to improve the quality of their recommendations.

Broadly, the life-cycle of deep learning for recommendation can be split into two phases: training and inference. In the training phase, the model is trained to predict user-item interaction probabilities (calculate a preference score) by presenting it with examples of interactions (or non-interactions) between users and items from the past.

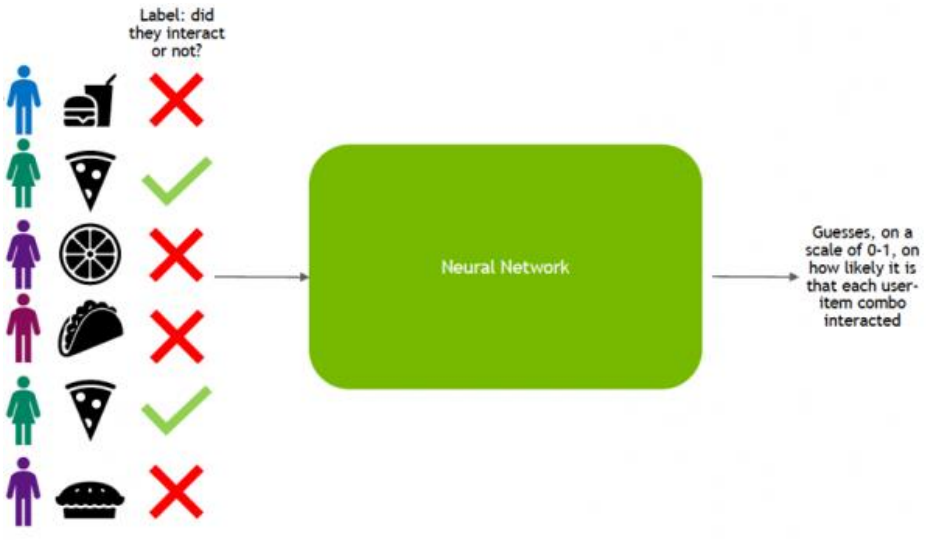
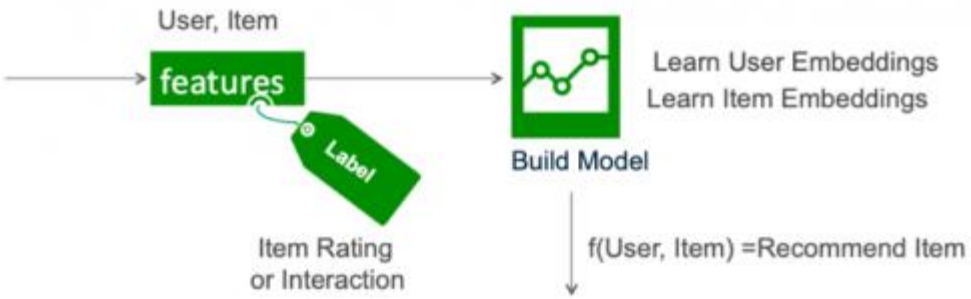


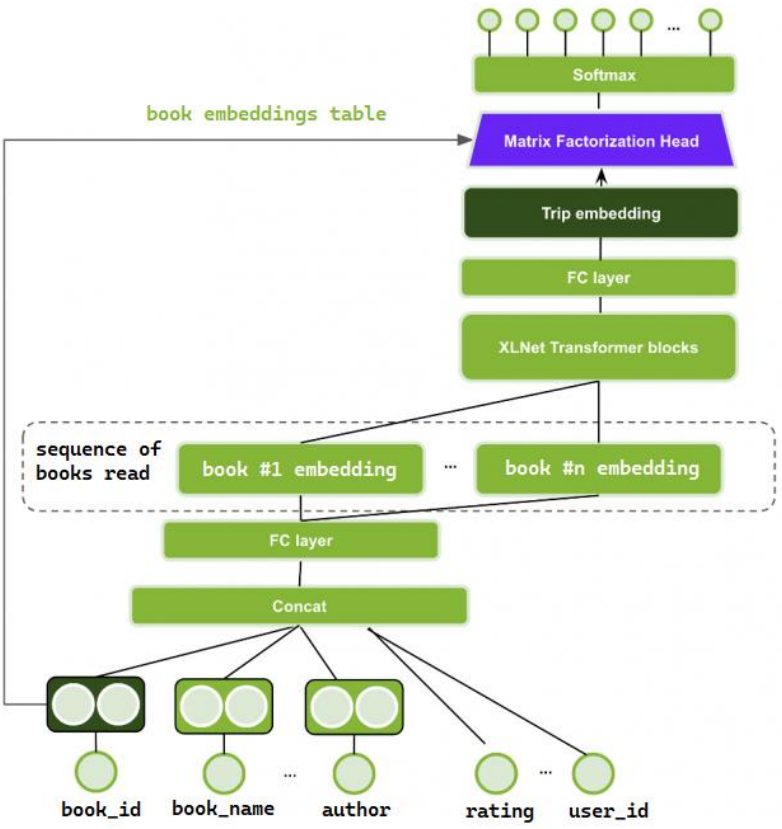
Figure 4 Deep Learning for Recommendation Training

Deep learning (DL) recommender models build upon existing techniques such as factorization to model the interactions between variables and embeddings to handle categorical variables. An embedding is a learned vector of numbers representing entity features so that similar entities (users or items) have similar distances in the vector space. For example, a deep learning approach to collaborative filtering learns the user and item embeddings (latent feature vectors) based on user and item interactions with a neural network.



3.3 Justify your solution by comparison

I would not say that the solution given is the best one out there, there are much better *newer* solutions that have come up, which I discovered after solving the book recommender system by collaborative filtering, now we will compare the collaborative filtering system with the ones which could have been much better and faster.



XLNet with Session-based Matrix Factorization head (XLNet-SMF)

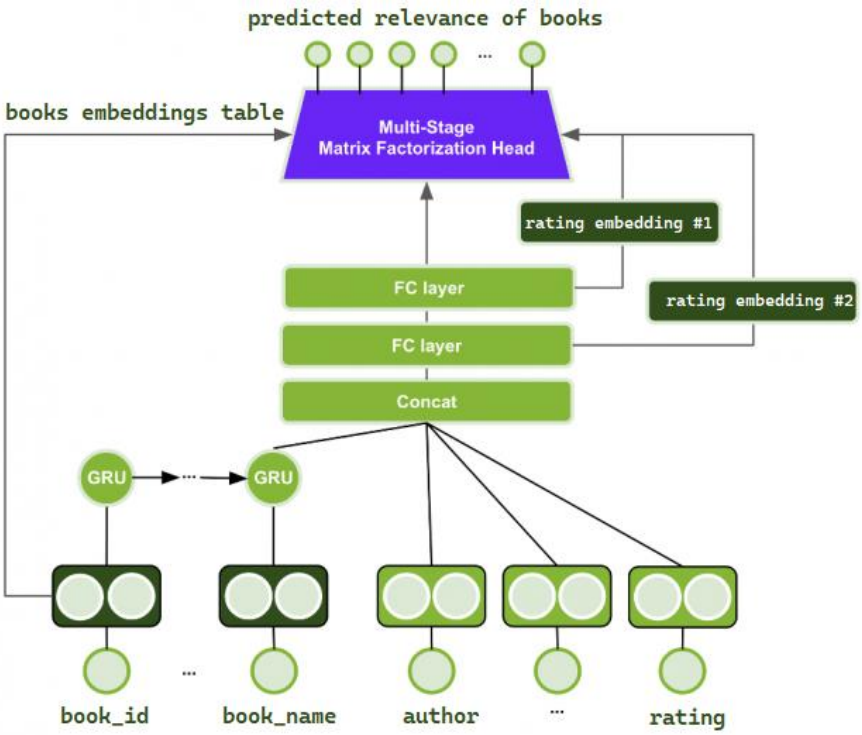
The XLNet with Session-based Matrix Factorization head (XLNet-SMF) uses a Transformer architecture named XLNet, originally proposed for the permutation-based language modeling task in Natural Language Processing (NLP). In this case, the sequence of items in the session (book) are modeled instead of the sequence of word tokens.

The XLNet training task was adapted for Masked Language Modeling (also known as Cloze task), like proposed by BERT4Rec for sequential recommendation. In that approach, for each training step, a proportion of all items are masked from the input sequence (i.e., replaced with a single

trainable embedding), and then the original ids of the masked items are predicted using other items of the sequence, from both left and right sides.

GRU with MultiStage Session-based Matrix Factorization head (GRU-MS-SMF)

The GRU with MultiStage Session-based Matrix Factorization head uses a GRU cell for embedding the historical user actions (previous 5 visited cities), similar to GRU4Rec. Missing values (sequences with less than 5 length) are padded with 0s. The last GRU hidden state is concatenated with the embeddings of the other categorical input features.



Ensembling

Ensembling is a proven approach to improve the accuracy of models, by combining their predictions and improving generalization. K-fold cross-validation

split data in k-folds
for *each architecture* **do**
  j = number of bags for this architecture
  for *each fold in folds* **do**
    for *i* in *[1, …, j]* **do**
      concatenate train and test out-of-fold data (OOF);
      train model on OOF;
      evaluate model based on book rating of in-fold train data;
      predict book rating of all folds of test data;
    end
  end
  ensemble by averaging over each fold and each bag;
end
ensemble by averaging over architecture;

In general, the higher the diversity of the model's predictions, the more the ensembling technique can potentially improve the final scores. In this case, the correlation of the predicted book scores between each two combinations of the three architectures was around 80%, which resulted in a representative improvement with ensembling in the final CV score. ***(Carol et al, Nvidia Winning Recommender System, 2021)***

Scores Comparison

	Single bag CV	Ensemble CV	Final LB
MLP-SMF	0.5667	0.5756	
GRU-MS-SMF	0.5664	0.5762	
XLNet-SMF	0.5681	0.5751	
Final Ensemble (Nvidia)		**0.5825**	**0.5939**
Collaborative Filtering (Ours)			**0.4252**

4 Question 4

Solution to Question 3 Part B

4.1 Recommend a solution

Mining and analyzing the valuable knowledge hidden behind the amount of data available in social media is becoming a fundamental prerequisite for any effective and successful strategic marketing campaign.

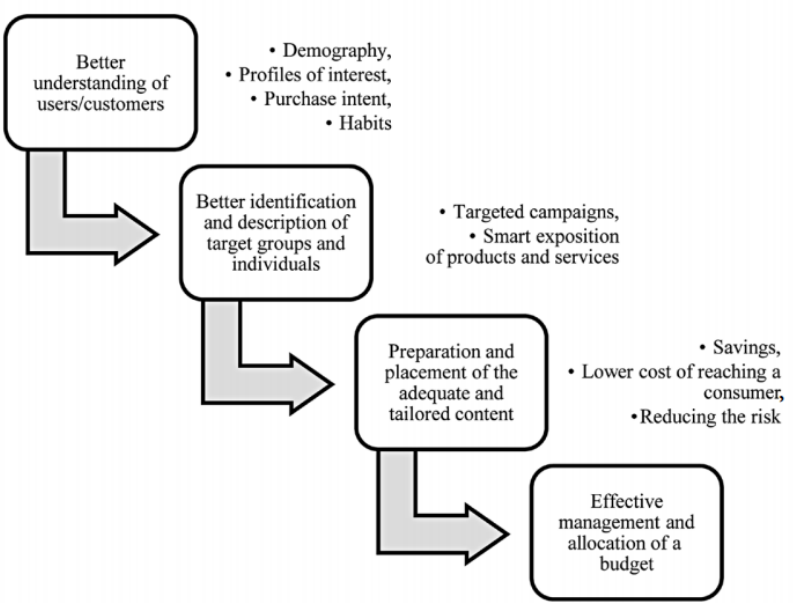


Figure 5 Data Driven Marketing

Big data has transformed the way businesses use to spend and generate the revenues. According to International Data Corporation (IDC) report, "total revenues from big data and business analytics will rise from $122 billion in 2015 to $187 billion in 2019. And enterprises who invest in big data and obtain the power to quickly analyze the large-scale data and extract actionable information can get an additional $430 billion in terms of productivity benefits over their competitors."

The enterprises are going to evidence lots of investment in big data in the coming year. Though the investments might vary from enterprise to enterprise, the investments on big data are going to discharge a higher return on investment (ROI).

McKinsey studied how different sectors will withdraw significant financial benefits from big data.

- US healthcare will generate $300 billion value per year.
- US retail will see 60% increase in their net margin.
- Manufacturing industries will notice up to 50% decrease in product development and assembly costs.

Big data has raised the opportunities for organizations to use data to track benefits, customers' and company's activities across the world. According to NewVantage Partners Big Data Executive Survey, 48.4% of corporate executives say that their firm has achieved "measurable results" from their Big Data investments. (McKinsley Research, 2017)

4.1.1 The Solution

Product recommendation

Analyze customer's buying behavior helps marketers to offer more relevant product recommendations for their customers. Amazon is an excellent example of a product recommendation. It gives a better suggestion based on your searches, interests, and previous purchase history.

When Amazon recommends a product it also uses market basket analysis for cross-selling. MBA is the most common technique among marketers to identify what products customers bought together. It helps to recommend a product based on the buying history and other people's buying history who bought the same item.

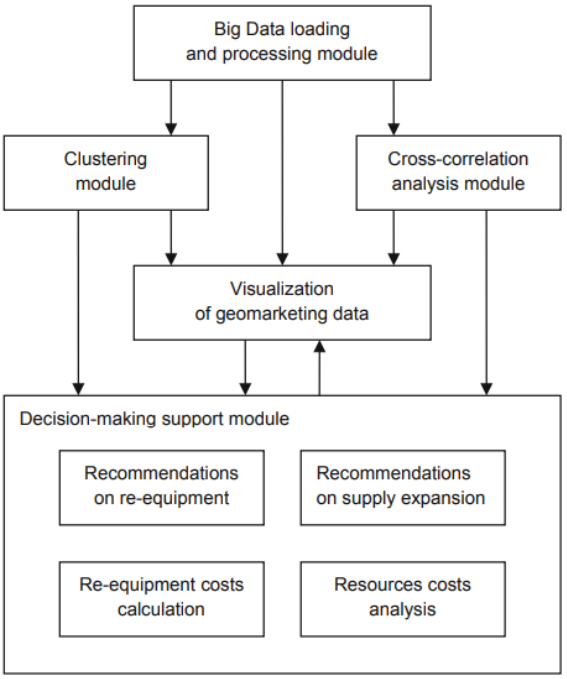


Figure 6 Solution Architecture

Targeted Emails

Personalized marketing allows marketers to reach a specific audience. If companies send the right email to the right person at the right time helps, companies to build a personal bond with their customers and that leads to increased sales. You can create an effective email campaign for your target audience based on their interest, demography, search history, and preferred content.

More targeted Ads

Information gathered from browsing history such as website visits and which deals or offers they consider are used to creating more targeted and effective Ad campaigns. Digital advertising means display company Ads on third-party websites to its site visitors that will help companies to gain more revenue. Google and Facebook is the best example of digital advertising. Personalized targeting help companies to improve customer experience, increase brand loyalty, and boost sales as well.

Increasing sales

With the help of Big data analysis, companies can know more about their customers such as what they buy? how frequently they buy a certain product? And which payment method they prefer? It will help to make the right offer at the right time that leads to increased sales.

Demand forecasting

Big data also helps to predict the demand for a product. With the help of Predictive and Prescriptive analytics company can improve their demand forecasting exactly. Does this allow companies can determine How much they produce? Which product? When it produces? and which location?

- Predictive analytics – what can happen? – It is about the future.
- Prescriptive analytics – what should we do? – It provides advice based on predictions.
- Demand forecasting reduces the risk of stockouts. It allows companies to control their production costs as well.

Price optimization

Big data analysis allows companies to identify the best price for products based on competitors' price, seasonal price, cost of goods, and other variables. Marketers can also identify the price according to the demand. Price optimization can maximize sales and revenue.

Predictive analytics with data visualization

Predictive analytics forecast the future based on current data and historical facts. With the help of Predictive analytics, marketers can determine which customer or customer segments to target and the right content for each customer. It also helps to discover the right channels and the right timings for the campaign. That will help to increase the response rates. Data visualization is the process of presentation of information or data in visual formats such as graphs, charts, tables, diagrams, and maps. It helps to understand huge data sets more easily and fast because humans are visual by nature.

Budget optimization

Budget optimization is one of the biggest challenges for digital marketers. Does a customer directly go to your website and make a purchase? Rarely. In this digital age, before making a purchase decision consumer use social media channels for peer reviews and they also consume a lot of digital content for research and comparison.

Marketers use many ways to reach out to their customers like blogs, Emails, Social media channels, affiliate networks, and Adwords. Marketers need to determine which channel or touchpoint is highly contributing to conversion, revenue, ROI, and what channels are creating more sales opportunities. They should ignore the channel which has a poor click-through rate. That will help marketers to manage their budget smartly and that is possible with the Attribution model. (Rajanarthagi, GecDesigns, 2021)

4.2 Discuss issues

Surveillance on Social Media

Social media marketing has developed as an extensive but too little scrutinized digital data collection apparatus. Companies such as Facebook suggest that somehow consumers of what they call the "social web" operate with a different set of expectations for privacy. As Facebook recently explained to the Obama administration's Internet Policy Task Force, "certain aspects of the social web... exist precisely because people want to share rather than limiting the sharing of their information to others. Imposing burdensome privacy restrictions could limit Facebook's ability to innovate, making it harder for Facebook to compete in a constantly evolving industry" (Facebook 2011a).

The Limits of Self-regulation and Voluntary Codes

The threat to privacy of consumers and citizens throughout the digitally connected world grows daily. In the US and the EU, digital marketers have banded together to offer various self-regulatory plans designed to blunt new regulatory safeguards (Dixon 2007). The IAB on both sides of the Atlantic have offered a new selfregulatory system using graphical "icons" to inform online users that data are being collected. The real goals of such a program is to offer a set of self-regulatory privacy principles and an "opt-out" scheme that will blunt the growing support for serious reform designed to protect Internet privacy (EU has Trouble Digesting New Law on Internet Cookies—IAB Europe Offers Solution 2010).

Cookies on Digital Steroids

One of the ironies of the debate about behaviorally targeted (BT) advertising and privacy is that marketing industry representatives primarily tell regulators that such data techniques aren't targeted to individuals. When pressed about the privacy concerns connected to BT, they generally retort that there is a misunderstanding. Such targeting is both "anonymous and innocuous," and is only aimed at providing consumers with ads they will find of greater interest. In the US, what is currently considered "personally identifiable" information (or PII) is undergoing review. But traditionally it has meant one's physical and email address, birth date, and Social Security number. Online marketers cling to a claim that most, if not all, of the information they collect on a user is non-personally identifiable (non-PII). But such arguments don't hold up to serious scrutiny (not to mention the claims marketers make to each other and to prospective clients).

(J Chester et. al, 2012)

4.3 Justification

The proposed approach can be used to solve the problems of the location of new emerging institutions, shopping and entertainment centers, social services. The approach supports the formation of development strategies on the mechanisms of machine learning with the use of Big Data analysis to take into account the dynamically changing environment.

The semantic-statistical analysis allows building dependencies between objects generating demand, the automated search helps finding a balance between supply and demand, and the visualization is used to present the solution options. Using the introduced spatial clustering model, they have developed an original concept of data organization based on multiple layers including digital map, semantic web (knowledge base), and overlay network.

New approaches to organization of user interfaces allow increasing the effectiveness of situational monitoring centers through the implementation of intelligent interaction management technologies, taking into account the characteristics of the perception of the situation by decision makers. Comparing the range of demand and supply actions on an interactive map, it becomes possible to accurately assess the efficiency of location, operating time, and characteristics of urban infrastructure, thereby improving the quality of customer service and improving business processes. (Ivaschenko et al, 2019)

4.3.1 Justification by Real Word Survey Results

Digital Marketing with BigData in India

Big data in E-business	Opinion on usefulness of big data in E-business			Chi- square value	p-value
	useful	Not useful	Total		
Using	30	06	36	36.2	P < 0.0001 Significant
Not Using	21	65	86		
Total	51	71	122		

From the above table, out of 122 companies 36 companies are using big data in ebusiness applications and out of that 30 i.e. 83.33% companies agreed on usefulness of Big data in E-business & 6 i.e. 16.66% companies are not agreed on the usefulness of big data in E- business. While 86 companies not using big data in e- business, from that 21 i.e. 24.41% companies where considered usefulness of big data in e-business and 65 i.e. 75.58% companies where not considered usefulness of big data in e-business. There is statistically significant association between the marketing strategies using Big Data and usefulness of big data in E-business domain (P = 0.0001) (Sayyad et. al, 2020)

Digital Marketing with BigData in Poland & USA

According to estimates by Forrester Research (Forrester 2014), the expenditure of companies in USA to cover these costs will rise from 24% in 2014 to 35% in 2019. The biggest chunk here goes to cover the costs of search marketing, display advertising, with less spending on social media and email marketing. The above situation results from the shift in the time spent consuming different types of media. The data can be successfully applied to the Polish market, where – according to different estimates – current expenditure on online marketing exceeds 20% of marketing budgets. It is expected that in the next two or three years, both in Poland and in developed countries, digital media will replace traditional television as the most consumed media. The implications to be expected are, on the one hand, marketing automation using the potential of Big Data; on the other hand, the need to seek new and better automated modes of analysing and measuring marketing operations, using technological solutions which, although designed by man, employ machine learning mechanisms, virtual and augmented reality (VR and AR), artificial intelligence (AI), and advanced semantic analysis allowing to translate machine language into natural language. This leads to the conclusion that the challenges faced by researchers and marketing practitioners centre around: measuring and the ability to reliably analyse the effectiveness of operations, the integration of operations carried out using different channels, the constant need to adjust technological and IT solutions to the requirements of the marketing sector, and finally the search for measurable effects of digital marketing, resulting from relations with stakeholders and non-customers. (Jacuński et al, 2018)

Justify Marketing Intelligence as Marketing 4.0

Following the development of industry 4.0 with the networking of man and machine, a marketing 4.0 debate is taking place on the performance and cost-effective availability of (mobile) internet.

- Marketing 1.0: The origin as a prototype puts the core competence of Marketing on a product and its distribution. Marketing activities are geared towards this, so that the market is at the center.
- Marketing 2.0: The focus shifts to the consumer. Companies further differentiate from each other as consumers become more self-confident (from the 1970s).
- Marketing 3.0: The focus is on people. They are determined by values that depend on their environment. Customer management instead of market-oriented corporate management is prevalent, as human centricity characterizes Marketing (from the 1980s onwards).
- Marketing 4.0: The focus here is on digitalization and thus the convergence of technologies, without losing sight of the previous stage. This means an online-offline integration (from approx. 2010 onwards).

- Marketing 5.0: Expectations regarding further developments include current popular discussions like block chain and platform marketing as a part of the blockchain economy. Analogue to the backbone of bitcoins block chain will provide marketing services which depend on the trust provided by closed IT systems, which are worthy e.g. for guaranteeing selected target media of programmatic advertising or to avoid fake accounts in social media. So, marketing 5.0 will probably be the epoch of digital trust marketing. (Lier J, 2019)

The solution described for this question conforms to Marketing 4.0, but we see that the solution is slowly becoming outdated, Marketing 5.0 is starting to show up, which will include cutting edge technology for marketing with blockchain.

5 Question 5

Solution to Question 4 Part B

5.1 Introduction to Big Data Platform

What is Big Data?

> *Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making. - Gartner*

Nowadays big data analytics is a very broad area for both academia and industry. Big data analytics has attracted intense interest from all academia and industry recently for its attempt to extract knowledge, information and wisdom form big data. Big data and cloud computing, two of the most important trends that are defining the new emerging analytical tools. Big data analytical capabilities using cloud delivery models could ease adoption for many industry, and most important thinking to cost saving, it could simplify useful insights that could providing them with different kinds of competitive advantage. Many companies to provide online Big Data analytical tools some of the top most companies like Amazon Big data Analytics Platform, HIVE web-based Interface, SAP Big data Analytics, IBM InfoSphere BigInsights, TERADATA Big Data Analytics, 1010data Big Data Platform, Cloudera Big Data Solution etc. Those companies analyze huge amount of data with help of different type of tools and also provide easy or simple user interface for analyzing data. (Rahul KC, et al, 2016)

5.1.1 Big Data Platform

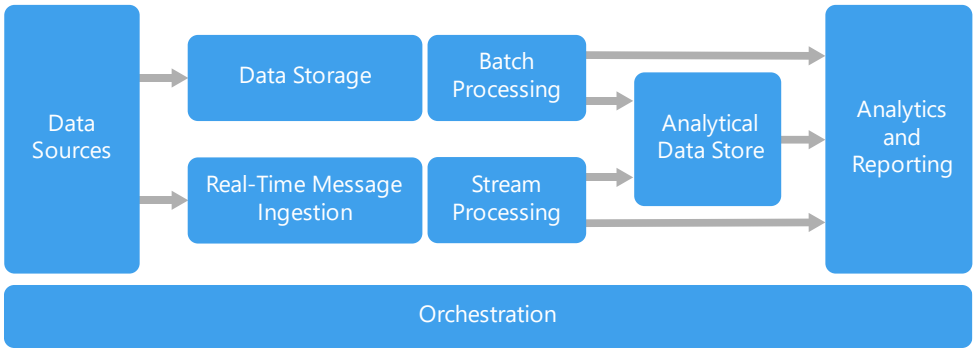


Figure 7 Big Data Platform

Big data solutions typically involve one or more of the following types of workload:

- Batch processing of big data sources at rest.

- Real-time processing of big data in motion.
- Interactive exploration of big data.
- Predictive analytics and machine learning.

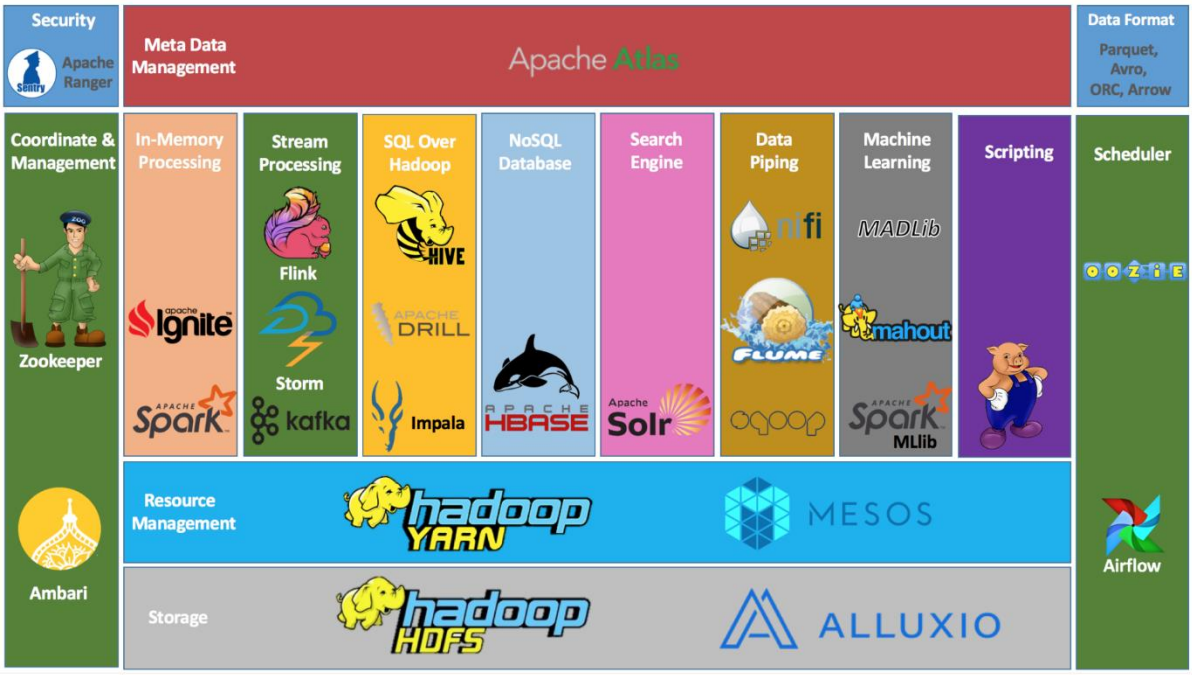


Figure 8 Hadoop Ecosystem

Hadoop Ecosystem is neither a programming language nor a service, it is a platform or framework which solves big data problems. You can consider it as a suite which encompasses a number of services (ingesting, storing, analyzing and maintaining) inside it.

Below are the Hadoop components, that together form a Hadoop ecosystem,

- HDFS: Hadoop Distributed File System
- YARN: Yet Another Resource Negotiator
- MapReduce: Data processing using programming
- Spark: In-memory Data Processing
- PIG, HIVE: Data Processing Services using Query (SQL-like)
- HBase: NoSQL Database
- Mahout, Spark MLlib: Machine Learning
- Apache Drill: SQL on Hadoop
- Zookeeper: Managing Cluster
- Oozie: Job Scheduling

- Flume, Sqoop: Data Ingesting Services
- Solr & Lucene: Searching & Indexing
- Ambari: Provision, Monitor and Maintain cluster

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically, both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically, the compute nodes and the storage nodes are the same, that is, the MapReduce framework and the Hadoop Distributed File System (see HDFS Architecture Guide) are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster. (Apache Software Foundation, 2021)

5.2 Problem solving approach

The problem is defined as an Inverted Index which is mapping of text in the document. The method used is Map method which can read the input file and output (word, filename) as the keyvalue pair and the other method is Reducer method that can use a hash map of (filename, count) to count the occurrences of each filename for a particular word key.

Input: A text file f (text over several lines)

Output: A sequence of key-value pairs (w, d), where w is the word and d is the number of occurrences of w in f.

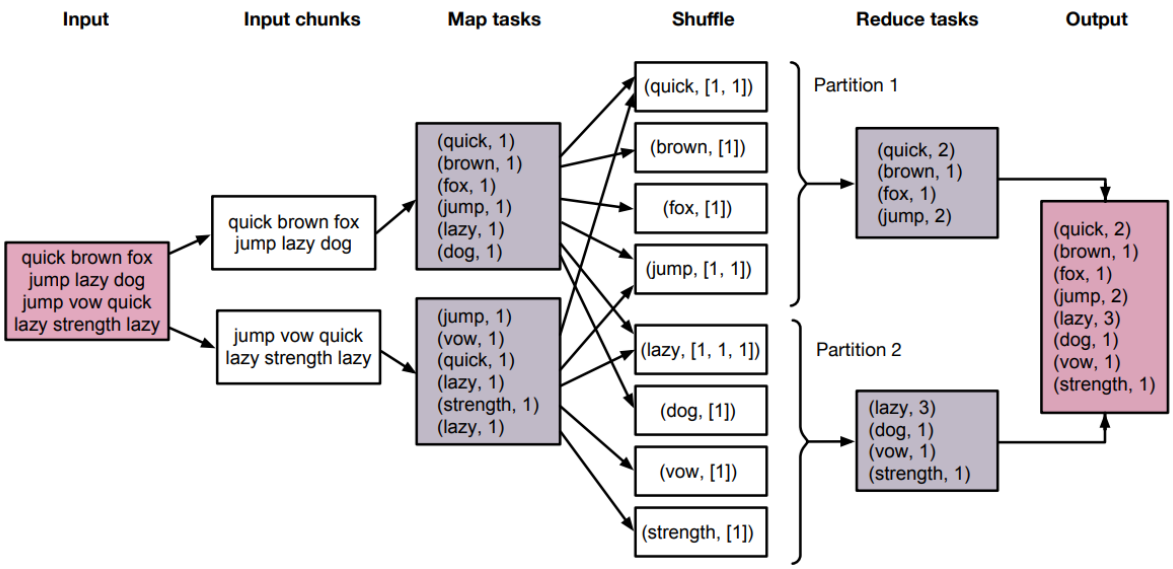


Figure 9 Problem Solving Approach

5.2.1 Algorithm

```
class MAPPER
   method INITIALIZE
      H ← new ASSOCIATIVEARRAY
   method MAP(docid a, doc d)
      for all term t ∈ doc d do
         H{t} ← H{t} + 1
   Method CLOSE
      for all term t ∈ H do
         EMIT(term t, count H{t})

class REDUCER
   method REDUCE(term t, counts [c1, c2, . . .])
      sum ← 0
      for all count c ∈ counts [c1, c2, . . .] do
         sum ← sum + c
      EMIT(term t, count sum)
```

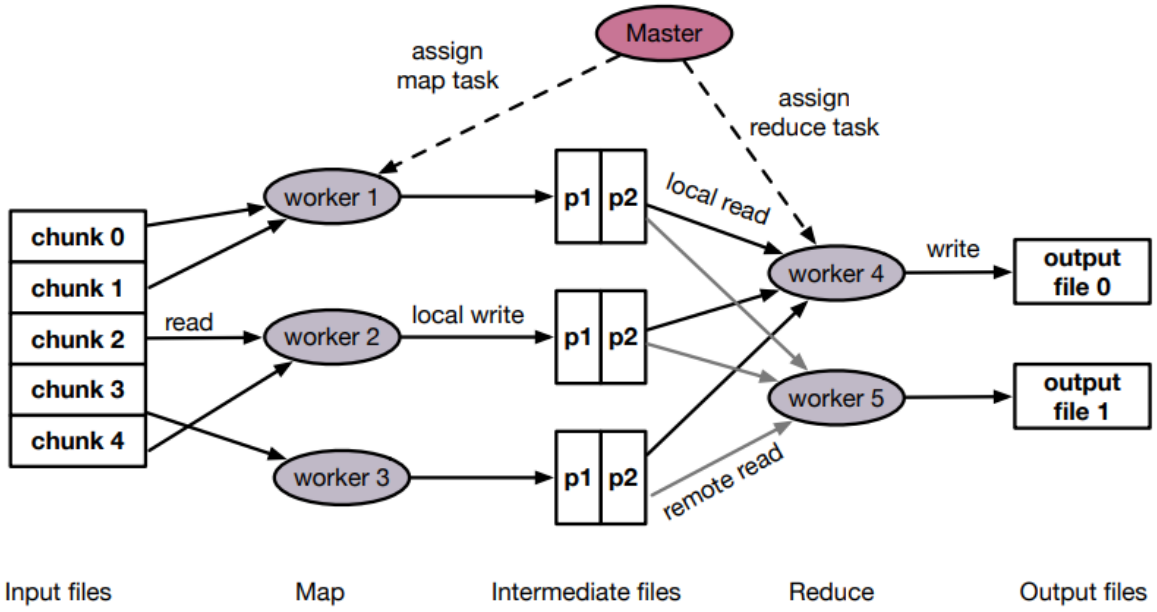


Figure 10 Map-Reduce Architecture

Initialization

- Split the input file into m chunks
- Start a Master process
- The Master assigns map tasks to the workers

Map

- Generates key-value pairs
- For each key-value pair (k, v): p = h(k) mod r
- Assign (k, v) to the partition p.

Shuffle

- The output of the map is grouped by key, sorted by key and is copied to the reducers

Sort

- Each reducer merges the key-value pairs obtained from different mappers

Reduce

- The reduce function is applied

5.3 Design and Implementation

The WordCount Application is Quite Straight-Forward

Let's say we have the following file `example.txt`

```
Hello World Bye World
Hello Hadoop Goodbye Hadoop
```

The Mapper implementation, via the map method, processes one line at a time, as provided by the specified TextInputFormat. It then splits the line into tokens separated by whitespaces, via the StringTokenizer, and emits a key-value pair of < <word>, 1>.

For the given sample input the map emits:

```
< Hello, 1>
< World, 1>
< Bye, 1>
< World, 1>
< Hello, 1>
< Hadoop, 1>
< Goodbye, 1>
< Hadoop, 1>
```

WordCount also specifies a combiner. Hence, the output of each map is passed through the local combiner (which is same as the Reducer as per the job configuration) for local aggregation, after being sorted on the keys. The Reducer implementation, via the reduce method just sums up the values, which are the occurrence counts for each key (i.e. words in this example).

The output of the job is:

```
< Hello, 2>
< World, 2>
< Bye, 1>
< Hadoop, 2>
< Goodbye, 1>
```

The main method specifies various facets of the job, such as the input/output paths (passed via the command line), key/value types, input/output formats etc., in the Job. It then calls the `job.waitForCompletion` to submit the job and monitor its progress.

`WordCount.java`

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.net.URI;
```

```
import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.util.GenericOptionsParser;
import org.apache.hadoop.util.StringUtils;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    static enum CountersEnum { INPUT_WORDS }

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    private boolean caseSensitive;
    private Set<String> patternsToSkip = new HashSet<String>();

    private Configuration conf;
    private BufferedReader fis;

    @Override
    public void setup(Context context) throws IOException,
        InterruptedException {
      conf = context.getConfiguration();
      caseSensitive = conf.getBoolean("wordcount.case.sensitive", true);
      if (conf.getBoolean("wordcount.skip.patterns", false)) {
        URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
        for (URI patternsURI : patternsURIs) {
          Path patternsPath = new Path(patternsURI.getPath());
          String patternsFileName = patternsPath.getName().toString();
          parseSkipFile(patternsFileName);
        }
      }
    }

    private void parseSkipFile(String fileName) {
      try {
        fis = new BufferedReader(new FileReader(fileName));
        String pattern = null;
```

```
      while ((pattern = fis.readLine()) != null) {
        patternsToSkip.add(pattern);
      }
    } catch (IOException ioe) {
      System.err.println("Caught exception while parsing the cached file '"
          + StringUtils.stringifyException(ioe));
    }
  }

  @Override
  public void map(Object key, Text value, Context context
                  ) throws IOException, InterruptedException {
    String line = (caseSensitive) ?
        value.toString() : value.toString().toLowerCase();
    for (String pattern : patternsToSkip) {
      line = line.replaceAll(pattern, "");
    }
    StringTokenizer itr = new StringTokenizer(line);
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      context.write(word, one);
      Counter counter = context.getCounter(CountersEnum.class.getName(),
          CountersEnum.INPUT_WORDS.toString());
      counter.increment(1);
    }
  }
}

public static class IntSumReducer
     extends Reducer<Text,IntWritable,Text,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Text key, Iterable<IntWritable> values,
                     Context context
                     ) throws IOException, InterruptedException {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}

public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  GenericOptionsParser optionParser = new GenericOptionsParser(conf, args);
  String[] remainingArgs = optionParser.getRemainingArgs();
  if ((remainingArgs.length != 2) && (remainingArgs.length != 4)) {
    System.err.println("Usage: wordcount <in> <out> [-skip skipPatternFile]");
    System.exit(2);
  }
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount.class);
```

```
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    List<String> otherArgs = new ArrayList<String>();
    for (int i=0; i < remainingArgs.length; ++i) {
      if ("-skip".equals(remainingArgs[i])) {
        job.addCacheFile(new Path(remainingArgs[++i]).toUri());
        job.getConfiguration().setBoolean("wordcount.skip.patterns", true);
      } else {
        otherArgs.add(remainingArgs[i]);
      }
    }
    FileInputFormat.addInputPath(job, new Path(otherArgs.get(0)));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs.get(1)));

    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

NOTE: The above solution is heavily inspired from the Official Apache Hadoop MapReduce Example Client Application,

https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

The Program was run on Hadoop Docker Container with Docker Compose for Worker Nodes, the Results and Outputs of which are attached in Appendix A of this Assignment.

5.3.1 Input

The Input file was taken from Apache Hadoop's `README.txt`, but any file can be taken as input.

`README.txt`

```
For the latest information about Hadoop, please visit our website at:

   http://hadoop.apache.org/

and our wiki, at:

   http://wiki.apache.org/hadoop/

This distribution includes cryptographic software.  The country in
which you currently reside may have restrictions on the import,
possession, use, and/or re-export to another country, of
encryption software.  BEFORE using any encryption software, please
```

```
check your country's laws, regulations and policies concerning the
import, possession, or use, and re-export of encryption software, to
see if this is permitted.  See <http://www.wassenaar.org/> for more
information.

The U.S. Government Department of Commerce, Bureau of Industry and
Security (BIS), has classified this software as Export Commodity
Control Number (ECCN) 5D002.C.1, which includes information security
software using or performing cryptographic functions with asymmetric
algorithms.  The form and manner of this Apache Software Foundation
distribution makes it eligible for export under the License Exception
ENC Technology Software Unrestricted (TSU) exception (see the BIS
Export Administration Regulations, Section 740.13) for both object
code and source code.

The following provides more details on the included cryptographic
software:
  Hadoop Core uses the SSL libraries from the Jetty project written
by mortbay.org.
```

5.3.2 Output

The output is the mapping `<key, value>` where key is the word and value is the word count

```
(BIS),  1
(ECCN)  1
(TSU)   1
(see    1
5D002.C.1,      1
740.13) 1
<http://www.wassenaar.org/>     1
Administration  1
Apache  1
BEFORE  1
BIS     1
Bureau  1
Commerce,       1
Commodity       1
Control 1
Core    1
Department      1
ENC     1
Exception       1
Export  2
For     1
Foundation      1
Government      1
Hadoop  1
Hadoop, 1
Industry        1
Jetty   1
License 1
Number  1
```

```
Regulations,    1
SSL     1
Section 1
Security        1
See     1
Software        2
Technology      1
The     4
This    1
U.S.    1
Unrestricted    1
about   1
algorithms.     1
and     6
and/or  1
another 1
any     1
as      1
asymmetric      1
at:     2
both    1
by      1
check   1
classified      1
code    1
code.   1
concerning      1
country 1
country's       1
country,        1
cryptographic   3
currently       1
details 1
distribution    2
eligible        1
encryption      3
exception       1
export  1
following       1
for     3
form    1
from    1
functions       1
has     1
have    1
http://hadoop.apache.org/       1
http://wiki.apache.org/hadoop/  1
if      1
import, 2
in      1
included        1
includes        2
information     2
information.    1
```

```
is      1
it      1
latest  1
laws,   1
libraries       1
makes   1
manner  1
may     1
more    2
mortbay.org.    1
object  1
of      5
on      2
or      2
our     2
performing      1
permitted.      1
please  2
policies        1
possession,     2
project 1
provides        1
re-export       2
regulations     1
reside  1
restrictions    1
security        1
see     1
software        2
software,       2
software.       2
software:       1
source  1
the     8
this    3
to      2
under   1
use,    2
uses    1
using   2
visit   1
website 1
which   2
wiki,   1
with    1
written 1
you     1
your    1
```

5.4 Performance analysis

We can look at the logs to figure out the performance numbers for the job

```
2021-05-28 09:32:33,817 INFO mapreduce.Job: Running job: job_1622193027243_0002
2021-05-28 09:32:38,907 INFO mapreduce.Job: Job job_1622193027243_0002 running in uber mode : false
2021-05-28 09:32:38,909 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-28 09:32:43,984 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-28 09:32:48,012 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-28 09:32:48,024 INFO mapreduce.Job: Job job_1622193027243_0002 completed successfully
2021-05-28 09:32:48,110 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=877
                FILE: Number of bytes written=459769
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1463
                HDFS: Number of bytes written=1301
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6720
                Total time spent by all reduces in occupied slots (ms)=14416
                Total time spent by all map tasks (ms)=1680
                Total time spent by all reduce tasks (ms)=1802
                Total vcore-milliseconds taken by all map tasks=1680
                Total vcore-milliseconds taken by all reduce tasks=1802
                Total megabyte-milliseconds taken by all map tasks=6881280
                Total megabyte-milliseconds taken by all reduce tasks=14761984
        Map-Reduce Framework
                Map input records=31
                Map output records=179
                Map output bytes=2050
                Map output materialized bytes=869
                Input split bytes=102
                Combine input records=179
                Combine output records=131
                Reduce input groups=131
                Reduce shuffle bytes=869
                Reduce input records=131
                Reduce output records=131
                Spilled Records=262
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=69
                CPU time spent (ms)=870
                Physical memory (bytes) snapshot=644034560
                Virtual memory (bytes) snapshot=13581504512
```

```
        Total committed heap usage (bytes)=2254962688
        Peak Map Physical memory (bytes)=331821056
        Peak Map Virtual memory (bytes)=5114970112
        Peak Reduce Physical memory (bytes)=312213504
        Peak Reduce Virtual memory (bytes)=8466534400
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=1361
File Output Format Counters
        Bytes Written=1301
```

5.4.1 Interpretation

The program used 870ms of the CPU, Read 1361 bytes of data and Wrote 1301 bytes of data. Which is about 1564.367 bytes/s Read and 1495.402 bytes/s of Write. A thing to notice in these readings is that they are not a good representation of a real world scenario. Hadoop is capable of much better performance, but for the sheer fact that our input file is so small that most of the time is spent by Hadoop to setup the Job, but when using a large enough file this time will get negligible and that is where the parallel job performance of Hadoop will shine.

The Input file was split into 102 bytes, so that means 13 processes were spawned by Hadoop to do the Map Operation, and to do all of this operation it took 614MB of physical memory, which might seem more for this task, but it becomes negligible when the input file size goes into Tera Bytes and Peta Bytes.

From the Paper Hadoop Performance Analysis Model with Deep Data Locality by Sungchul Lee et al., (2019) the paper introduced the concept of data locality on HDFS and the Hadoop performance analysis model. The deep data locality on the model was applied to improve the performance of the Hadoop system. The authors made two DDL methods, such as block-based DDL and key-based DDL. The two DDL methods were combined on HDFS and increased over 34.4% more performance than the default MR. The DDL methods on the Hadoop system were tested on a cloud, Hadoop simulation and physical implement Hadoop system. According to the test, the block-based DDL increased the Hadoop performance by 9.8% more than the default MR, and key-based DDL improved it by 21.9% more than the default one. Also, the combined methods increased the Hadoop performance upto 34.4% more than the default method.

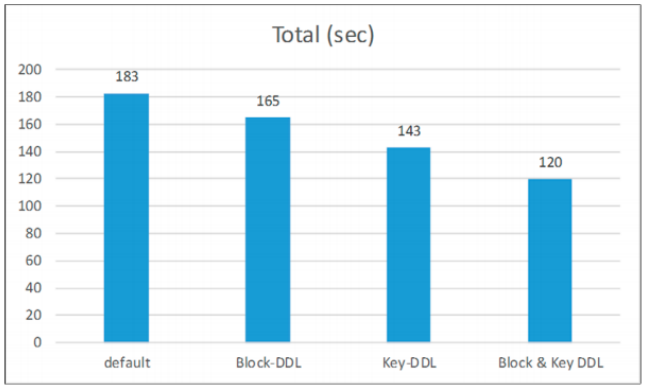


Figure 19. Performance comparison of total time.

Table 4. Improvement of Hadoop performance by DDL.

	Block-DDL	**Key-DDL**	**Block& Key-DDL**
Default	9.8%	21.9%	34.4%
Block-DDL	N/A	13.3%	27.3%
Key-DDL	N/A	N/A	16.1%

Figure 11 Hadoop Performance

Bibliography

1. Moreira, J., de Leon Ferreira, A.C.P. and Horváth, T., 2019. A general introduction to data analytics. Wiley.
2. Orman, L.V., 2015. Information paradox: Drowning in information, starving for knowledge. IEEE Technology and Society.
3. Anderson, C., 2008. The end of theory: The data deluge makes the scientific method obsolete. Wired magazine, 16(7), pp.16-07.
4. Controlglobal.com. 2021. [online]

   Available at: <https://www.controlglobal.com/articles/2010/drowninginfo1002/> [Accessed 26 May 2021].
5. Drowning in Data but Thirsty for Insight? Uncommonlogic.com [online]

   Available at: < https://www.uncommonlogic.com/wp-content/uploads/2018/08/Drowning-in-Data-But-Thirsty-for-Insight-ebook-unCommon-Logic-Bulldog-Solutions.pdf> [Accessed 26 May 2021].
6. Data Deluge and How to avoid it, Naveen Joshi, 2017,

   https://www.allerin.com/blog/data-deluge-and-how-to-avoid-it
7. Preparing for the future of data analytics, Michael Dixon, December 6, 2019

   https://seleritysas.com/blog/2019/12/06/preparing-for-the-future-of-data-analytics/
8. https://www.geeksforgeeks.org/life-cycle-phases-of-data-analytics/
9. Yadranjiaghdam, B., Pool, N. and Tabrizi, N., 2016, December. A survey on real-time big data analytics: applications and tools. In 2016 international conference on computational science and computational intelligence (CSCI) (pp. 404-409). IEEE.
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.
11. Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á.L., Heredia, I., Malík, P. and Hluchý, L., 2019. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review, 52(1), pp.77-124.
12. Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K.W., Schindelin, J., Cardona, A. and Sebastian Seung, H., 2017. Trainable Weka Segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics, 33(15), pp.2424-2426.
13. https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.0

14. Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B. and Varoquaux, G., 2014. Machine learning for neuroimaging with scikit-learn. Frontiers in neuroinformatics, 8, p.14.

15. Batyuk, A. and Voityshyn, V., 2016, August. Apache storm based on topology for real-time processing of streaming data from social networks. In 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP) (pp. 345-349). IEEE.

16. Guo, Runxin, Yi Zhao, Quan Zou, Xiaodong Fang, and Shaoliang Peng. "Bioinformatics applications on apache spark." GigaScience 7, no. 8 (2018): giy098.

17. Lee, S., Jo, J.Y. and Kim, Y., 2019. Hadoop performance analysis model with deep data locality. Information, 10(7), p.222.

18. https://developer.nvidia.com/blog/how-to-build-a-winning-recommendation-system-part-2-deep-learning-for-recommender-systems/

19. https://developer.nvidia.com/blog/how-to-build-a-winning-recommendation-system-part-1/

20. https://developer.nvidia.com/blog/how-to-build-a-winning-deep-learning-powered-recommender-system-part-3/

21. https://gecdesigns.com/blog/role-of-big-data-in-digital-marketing

22. Chester, J., 2012. Cookie wars: How new data profiling and targeting techniques threaten citizens and consumers in the “big data” era. In European Data Protection: In Good Health? (pp. 53-77). Springer, Dordrecht.

23. Ivaschenko, A., Stolbova, A. and Golovnin, O., 2019, October. Spatial clustering based on analysis of Big Data in digital marketing. In Russian Conference on Artificial Intelligence (pp. 335-347). Springer, Cham.

24. Sayyad, S., Mohammed, A., Shaga, V., Kumar, A. and Vengatesan, K., 2018, December. Digital Marketing Framework Strategies Through Big Data. In International conference on Computer Networks, Big data and IoT (pp. 1065-1073). Springer, Cham.

25. Jacuński, M., 2018. Measuring and analysis of digital marketing. RESEARCH PRIVACY, p.11.

26. Lies, J., 2019. Marketing Intelligence and Big Data: Digital Marketing Techniques on their Way to Becoming Social Engineering Techniques in Marketing. International Journal of Interactive Multimedia & Artificial Intelligence, 5(5).

Appendix A

Complete OUTPUT for the Hadoop Program

```
inkadmin@beast2:~/satyajit/docker-hadoop$ make wordcount
docker build -t hadoop-wordcount ./submit
Sending build context to Docker daemon  7.168kB
Step 1/9 : FROM bde2020/hadoop-base:2.0.0-hadoop3.2.1-java8
 ---> a89a06d383e8
Step 2/9 : MAINTAINER Ivan Ermilov <ivan.s.ermilov@gmail.com>
 ---> Using cache
 ---> 37cf173866e7
Step 3/9 : COPY WordCount.jar /opt/hadoop/applications/WordCount.jar
 ---> Using cache
 ---> 865a1008ad28
Step 4/9 : ENV JAR_FILEPATH="/opt/hadoop/applications/WordCount.jar"
 ---> Using cache
 ---> b9f038c2ae3f
Step 5/9 : ENV CLASS_TO_RUN="WordCount"
 ---> Using cache
 ---> 158ccbd33bbd
Step 6/9 : ENV PARAMS="/input /output"
 ---> Using cache
 ---> e0917bd13f4f
Step 7/9 : ADD run.sh /run.sh
 ---> Using cache
 ---> 45e50c892ffe
Step 8/9 : RUN chmod a+x /run.sh
 ---> Using cache
 ---> 0a3f31438c71
Step 9/9 : CMD ["/run.sh"]
 ---> Using cache
 ---> 25c45210ca3b
Successfully built 25c45210ca3b
Successfully tagged hadoop-wordcount:latest
docker run --network docker-hadoop_default --env-file hadoop.env bde2020/hadoop-base:latest hdfs dfs -mkdir
-p /input/
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
 - Setting dfs.permissions.enabled=false
Configuring yarn
```

```
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
docker run --network docker-hadoop_default --env-file hadoop.env bde2020/hadoop-base:latest hdfs dfs -
copyFromLocal -f /opt/hadoop-3.2.1/README.txt /input/
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
```

```
 - Setting dfs.permissions.enabled=false
Configuring yarn
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
2021-05-28 09:32:25,738 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
docker run --network docker-hadoop_default --env-file hadoop.env hadoop-wordcount
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
```

```
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
 - Setting dfs.permissions.enabled=false
Configuring yarn
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
2021-05-28 09:32:32,068 INFO client.RMProxy: Connecting to ResourceManager at
resourcemanager/172.18.0.2:8032
2021-05-28 09:32:32,222 INFO client.AHSProxy: Connecting to Application History server at
historyserver/172.18.0.3:10200
2021-05-28 09:32:32,363 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not
performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
```

```
2021-05-28 09:32:32,411 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-
yarn/staging/root/.staging/job_1622193027243_0002
2021-05-28 09:32:32,523 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
2021-05-28 09:32:32,667 INFO input.FileInputFormat: Total input files to process : 1
2021-05-28 09:32:32,726 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
2021-05-28 09:32:32,776 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
2021-05-28 09:32:32,792 INFO mapreduce.JobSubmitter: number of splits:1
2021-05-28 09:32:32,935 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
2021-05-28 09:32:33,383 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1622193027243_0002
2021-05-28 09:32:33,384 INFO mapreduce.JobSubmitter: Executing with tokens: []
2021-05-28 09:32:33,533 INFO conf.Configuration: resource-types.xml not found
2021-05-28 09:32:33,533 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2021-05-28 09:32:33,784 INFO impl.YarnClientImpl: Submitted application application_1622193027243_0002
2021-05-28 09:32:33,817 INFO mapreduce.Job: The url to track the job:
http://resourcemanager:8088/proxy/application_1622193027243_0002/
2021-05-28 09:32:33,817 INFO mapreduce.Job: Running job: job_1622193027243_0002
2021-05-28 09:32:38,907 INFO mapreduce.Job: Job job_1622193027243_0002 running in uber mode : false
2021-05-28 09:32:38,909 INFO mapreduce.Job:  map 0% reduce 0%
2021-05-28 09:32:43,984 INFO mapreduce.Job:  map 100% reduce 0%
2021-05-28 09:32:48,012 INFO mapreduce.Job:  map 100% reduce 100%
2021-05-28 09:32:48,024 INFO mapreduce.Job: Job job_1622193027243_0002 completed successfully
2021-05-28 09:32:48,110 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=877
                FILE: Number of bytes written=459769
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1463
                HDFS: Number of bytes written=1301
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Rack-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6720
                Total time spent by all reduces in occupied slots (ms)=14416
                Total time spent by all map tasks (ms)=1680
                Total time spent by all reduce tasks (ms)=1802
                Total vcore-milliseconds taken by all map tasks=1680
                Total vcore-milliseconds taken by all reduce tasks=1802
                Total megabyte-milliseconds taken by all map tasks=6881280
                Total megabyte-milliseconds taken by all reduce tasks=14761984
        Map-Reduce Framework
```

```
                Map input records=31
                Map output records=179
                Map output bytes=2050
                Map output materialized bytes=869
                Input split bytes=102
                Combine input records=179
                Combine output records=131
                Reduce input groups=131
                Reduce shuffle bytes=869
                Reduce input records=131
                Reduce output records=131
                Spilled Records=262
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=69
                CPU time spent (ms)=870
                Physical memory (bytes) snapshot=644034560
                Virtual memory (bytes) snapshot=13581504512
                Total committed heap usage (bytes)=2254962688
                Peak Map Physical memory (bytes)=331821056
                Peak Map Virtual memory (bytes)=5114970112
                Peak Reduce Physical memory (bytes)=312213504
                Peak Reduce Virtual memory (bytes)=8466534400
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1361
        File Output Format Counters
                Bytes Written=1301
docker run --network docker-hadoop_default --env-file hadoop.env bde2020/hadoop-base:latest hdfs dfs -cat
/output/*
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
 - Setting dfs.permissions.enabled=false
Configuring yarn
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
```

```
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
2021-05-28 09:32:54,462 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted =
false, remoteHostTrusted = false
(BIS),  1
(ECCN)  1
(TSU)   1
(see    1
5D002.C.1,      1
740.13) 1
<http://www.wassenaar.org/>     1
Administration  1
Apache  1
BEFORE  1
BIS     1
Bureau  1
```

```
Commerce,       1
Commodity       1
Control 1
Core    1
Department      1
ENC     1
Exception       1
Export  2
For     1
Foundation      1
Government      1
Hadoop  1
Hadoop, 1
Industry        1
Jetty   1
License 1
Number  1
Regulations,    1
SSL     1
Section 1
Security        1
See     1
Software        2
Technology      1
The     4
This    1
U.S.    1
Unrestricted    1
about   1
algorithms.     1
and     6
and/or  1
another 1
any     1
as      1
asymmetric      1
at:     2
both    1
by      1
check   1
classified      1
code    1
code.   1
concerning      1
country 1
country's       1
country,        1
cryptographic   3
currently       1
details 1
distribution    2
```

```
eligible        1
encryption      3
exception       1
export  1
following       1
for     3
form    1
from    1
functions       1
has     1
have    1
http://hadoop.apache.org/       1
http://wiki.apache.org/hadoop/  1
if      1
import, 2
in      1
included        1
includes        2
information     2
information.    1
is      1
it      1
latest  1
laws,   1
libraries       1
makes   1
manner  1
may     1
more    2
mortbay.org.    1
object  1
of      5
on      2
or      2
our     2
performing      1
permitted.      1
please  2
policies        1
possession,     2
project 1
provides        1
re-export       2
regulations     1
reside  1
restrictions    1
security        1
see     1
software        2
software,       2
software.       2
```

```
software:       1
source  1
the     8
this    3
to      2
under   1
use,    2
uses    1
using   2
visit   1
website 1
which   2
wiki,   1
with    1
written 1
you     1
your    1
docker run --network docker-hadoop_default --env-file hadoop.env bde2020/hadoop-base:latest hdfs dfs -rm -r
/output
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
 - Setting dfs.permissions.enabled=false
Configuring yarn
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
```

```
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
Deleted /output
docker run --network docker-hadoop_default --env-file hadoop.env bde2020/hadoop-base:latest hdfs dfs -rm -r
/input
Configuring core
 - Setting hadoop.proxyuser.hue.hosts=*
 - Setting fs.defaultFS=hdfs://namenode:9000
 - Setting hadoop.http.staticuser.user=root
 - Setting io.compression.codecs=org.apache.hadoop.io.compress.SnappyCodec
 - Setting hadoop.proxyuser.hue.groups=*
Configuring hdfs
 - Setting dfs.namenode.datanode.registration.ip-hostname-check=false
 - Setting dfs.webhdfs.enabled=true
 - Setting dfs.permissions.enabled=false
Configuring yarn
 - Setting yarn.timeline-service.enabled=true
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-vcores=4
 - Setting yarn.resourcemanager.system-metrics-publisher.enabled=true
 - Setting
yarn.resourcemanager.store.class=org.apache.hadoop.yarn.server.resourcemanager.recovery.FileSystemRMStateSto
re
 - Setting yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage=98.5
 - Setting yarn.log.server.url=http://historyserver:8188/applicationhistory/logs/
 - Setting yarn.resourcemanager.fs.state-store.uri=/rmstate
 - Setting yarn.timeline-service.generic-application-history.enabled=true
 - Setting yarn.log-aggregation-enable=true
 - Setting yarn.resourcemanager.hostname=resourcemanager
 - Setting yarn.scheduler.capacity.root.default.maximum-allocation-mb=8192
 - Setting yarn.nodemanager.aux-services=mapreduce_shuffle
 - Setting yarn.resourcemanager.resource_tracker.address=resourcemanager:8031
 - Setting yarn.timeline-service.hostname=historyserver
 - Setting yarn.resourcemanager.scheduler.address=resourcemanager:8030
```

```
 - Setting yarn.resourcemanager.address=resourcemanager:8032
 - Setting mapred.map.output.compress.codec=org.apache.hadoop.io.compress.SnappyCodec
 - Setting yarn.nodemanager.remote-app-log-dir=/app-logs
 - Setting
yarn.resourcemanager.scheduler.class=org.apache.hadoop.yarn.server.resourcemanager.scheduler.capacity.Capaci
tyScheduler
 - Setting mapreduce.map.output.compress=true
 - Setting yarn.nodemanager.resource.memory-mb=16384
 - Setting yarn.resourcemanager.recovery.enabled=true
 - Setting yarn.nodemanager.resource.cpu-vcores=8
Configuring httpfs
Configuring kms
Configuring mapred
 - Setting mapreduce.map.java.opts=-Xmx3072m
 - Setting mapreduce.reduce.java.opts=-Xmx6144m
 - Setting mapreduce.reduce.memory.mb=8192
 - Setting yarn.app.mapreduce.am.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.map.memory.mb=4096
 - Setting mapred.child.java.opts=-Xmx4096m
 - Setting mapreduce.reduce.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
 - Setting mapreduce.framework.name=yarn
 - Setting mapreduce.map.env=HADOOP_MAPRED_HOME=/opt/hadoop-3.2.1/
Configuring for multihomed network
Deleted /input
```