Big Data Analytics

* Introduction to Big Data Analytics

* Introduction to Hadoop

* Hadoop HDFS & Map Reduce

* NoSQL

* Link Analysis

* Mining Data Streams

* Books that will Change your life (My recommendations)

Introduction to Big Data

Q1 Write a short note on Big Data.

=> Data means the quantities, characters or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.

- So big data is also a data but with a huge size. 'Big Data' is a term used to describe collection of data that is huge in size and yet growing exponentially with time. In short, such a data is so large and complex that ~~one~~ none of the traditional data management tools are able to store it or process it efficiently

- It takes terrabytes to 10s of petabytes of data to be called as big data.

- Big data can come from various quarters like social media sites, Sensors, Digital photos, Business transactions, Location based data. ~~For~~ facebook, E-commerce Netflix are the Major Big Data Users.

Q. Write a short Note on Types of Big Data.

=> There are Three major types of Big data.

1. Structured Data: Any data that can be stored, accessed and processed in the form of fixed format is termed as a Structured data.

- Over the period of time, talent in computer Sience have achieved greater success in developing techniques for working with Such Kind of data (where the format is well known in advance) and also deriving value out of it.

- However nowadays we are forseeing issues when size of such data grows to a huge extent, typically sizes are being in the rage of multiple Zettabyte.

- Example of Structured data is an Employee table in a database :

Employee ID	Employee Name	Gender	Description	Salary in k
3021	Naresh Ahuja	M	Admin	35
1924	Walter White	M	H.R.	39

2. Unstructured Data

- Any data with unknown form or a structure is called as unstructured data. In addition to the size being huge, unstructured data poses multiple challenges in terms of its processing for deriving value out of it.

- Typical example of unstructured data is, a heterogeneous data source containing a combination of simple text files, images, videos etc.

- Now a days organizations have wealth of data available with them but unfortunately they don't know how to derive value out of it since this data is in its raw form or unstructured form

- Example of Unstructured Data is output returned by Google Search.

3. Semi-Structured Data.

- Semi Structured data can contain both the form of data.

- We can see semi structured data as a structured in form but it is actually not defined with eg: a table definition in relational DBMS.

- Example of Semi Structured data is a data represented in XML file.

Q. Explain 5 V's of the Big Data.

1. Volume:
   - The name 'Big Data' itself is related to a size which is enormous.
   - Size of data plays very crucial role in determining value out of data. Also whether a particular data can actually be considered as a Big Data or not, is dependent on Volume of data.
   - Hence 'Volume' is one characteristics which needs to be considered while dealing with Big Data.

2. Velocity: The term velocity refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.
   - It deals with the speed at which data flows in from sources like business processes, application logs, networks and social media site, sensors, mobile devices, etc. The flow of data is massive and continuous.

3. Variety ⁰₀ Variety refers to heterogeneous sources and the nature of data, both structured and unstructured.

- During early days, spreadsheets and databases were the only sources of data considered by most of the application

- Now a days, data in the form of emails, photos, videos, PDF audio etc. is also being considered in the analysis application. This variety of unstructured data poses certain issues for storage, mining and analysing data.

4. Veracity: Trustworthiness of data.

- Data involves some uncertainty and ambiguities
- Mistakes can be introduced by humans and machines
  - People sharing accounts.
  - Like it today, dislike it tomorrow
  - Wrong system timestamps.
- Data qualities is vital.
- Analytics and conclusion rely on good data qualities.

  Garbage data + perfect model => garbage result
  perfect data + garbage or model => garbage results

5. Value:
Raw data of Big Data is of low value for example, Single observations

- Analytics and theory about the data increases the value

- Analytics transform big data into Smart data.

Q Differentiate Between Traditional Data and Big data.

parameters	Traditional Data	Big Data.
1. Volume	GB	TB or PB (even more)
2. Generate rate	per hour, Per Day	Every second or milisecond.
3. Structure	Structured	Semistructured or unstructured
4. Data Source	Centralized	fully Distributed
5. Data integration	easy	Difficult
6. Data Store	RDBMS	HDFS, NoSQL
7. Access	Interactive	Batch or near real time
8. Update Scenario	Repeated Read & write	write once Repeated Read
9. Data Structure	Static Schema	Dynamic Schema
10. Scaling Potential	Non Linear	Somewhat close to Linear.

Introduction to Hadoop

Q1. What is Hadoop

- It is an open source framework of tools designed for storage and processing of large scale data.
- Hadoop is maintained by Apache
- ~~Created by~~ It is created by Doug Cutting and Mike Carafella in 2005
- Cutting named the program after his son's toy elephant
- In 2005 Doug Cutting and Michall J. Carafella developed Hadoop to support Distruction for the Nutch Search Engine project.
- This project was funded by Jahoo and in 2006 yahoo gave the projet to Apache Software foundation.

Q. What are features of Hadoop

1. Low cost : As Hadoop is an open source framework, it is free. It uses commodity hardware to store and process huge data. Hence it is not much costly

2. High computing power:
Hadoop uses distributed computing model. Due to this, task can be distributed amongst different nodes and can be processed quickly. Cluster have thousands of nodes which gives high computing capability to hadoop.

3. Scalability :
Nodes can be easily added and removed. Failed nodes can be easily detected. For all these activities very little administration is required

4. Huge and Flexible Storage :
Massive data storage is available due to thousands of nodes in the cluster. It supports both structured and unstructured data. No preprocessing is required on data before storing it.

5. Fault tolerance and data protection :
If any node fails the tasks in hand are automatically redirected to other node. Multiple copies of all data are automatically stored. Due to this even if any node fails that data is available on some other nodes also.

Page: 4
Date:

Q. Difference between Hadoop and Traditional RDBMS.

	Hadoop	RDBMS
1.	Hadoop stores both Structured and unstructured data	RDBMS stores data in a structural way
2.	SQL can be implemented on top of Hadoop as the execution engine	SQL (Structured Query Language) is used.
3.	Basic data unit is key/value pairs	Basic data unit is relational tables.
4.	With MapReduce we can use scripts and codes to tell actual steps in processing the data.	With SQL we can state Expected result and database engine derives it.
5.	Hadoop is designed for offline processing and analysis of large-scale data.	RDBMS is designed for online transactions

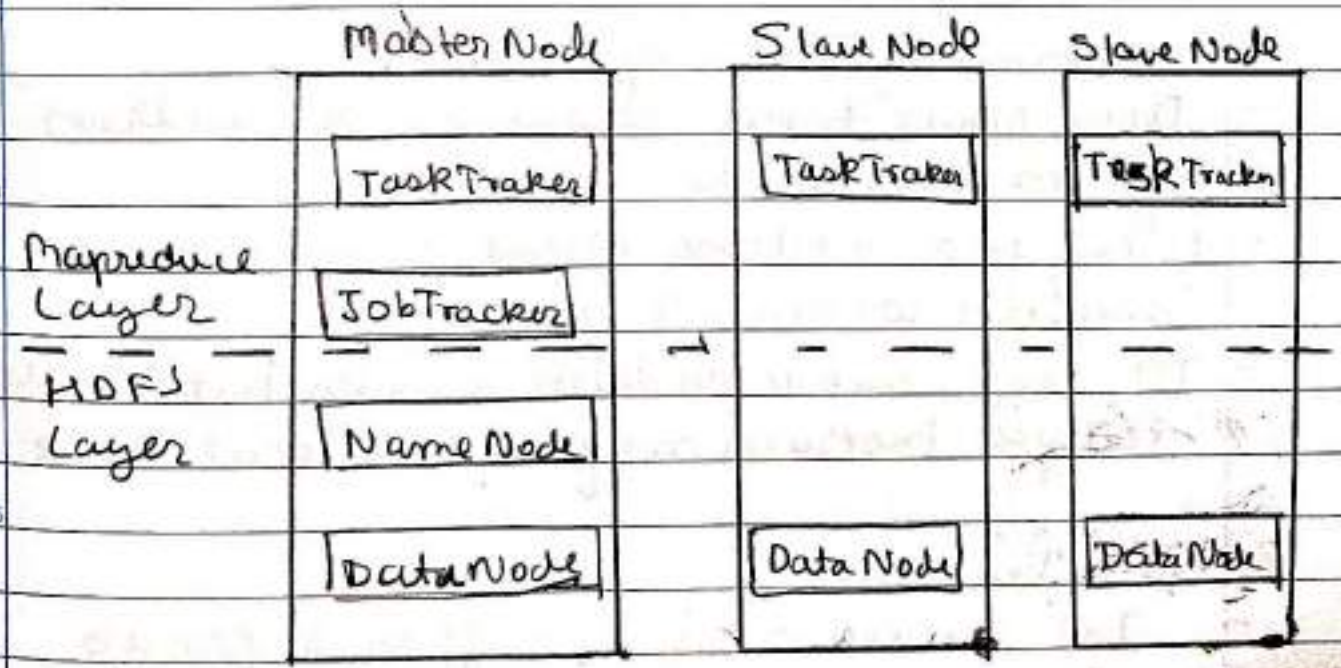
Q. Hadoop Architecture

=> Physical Architecture:

- Name Node represented every files and ~~dirc~~ directory which is used in the namespace (just like a index)

- Datanode helps you to manage the State of an HDFS node and allows you to interact with the blocks where data is actually stored

- The Master node allows you to conduct parallel processing of data using ~~H~~ Hadoop Mapreduce.



→ The Slave nodes are the additional machines in the Hadoop cluster which allows you to Store data to conduct complex calculations. Moreover, all the Slave node comes with Task Tracker and a Data Node.

Now we will see every component in detail

1. Name Node:
   - Known as master of HDFS
   - Data node is known as Slave of HDFS
   - It has job tracker which keeps track of files distributed to Data Nodes.
   - Name node is only Single point of failure.

2. Data Node:
   - Known as Slave of HDFS
   - Data Node takes client back address from Name Node.
   - Using this address client communicates directly with the Data Node.
   - TO create, move or delete blocks Data Node receives instructions from the local disk.

3. Job Tracker:
   - Job tracker determines files to process, node assignments for different tasks, tasks monitoring etc
   - only one Job tracker daemon per hadoop cluster is allowed
   - Job Tracker runs on a server as a master node of the cluster.

Task Tracker: Individual Tasks assigned by Job Tracker are executed by Task Tracker

- There is a single Task Tracker per slave node.
- Task Tracker may handle multiple tasks parallelly by using multiple JVMs.

Q. Write a short note on Hadoop Distributed File System

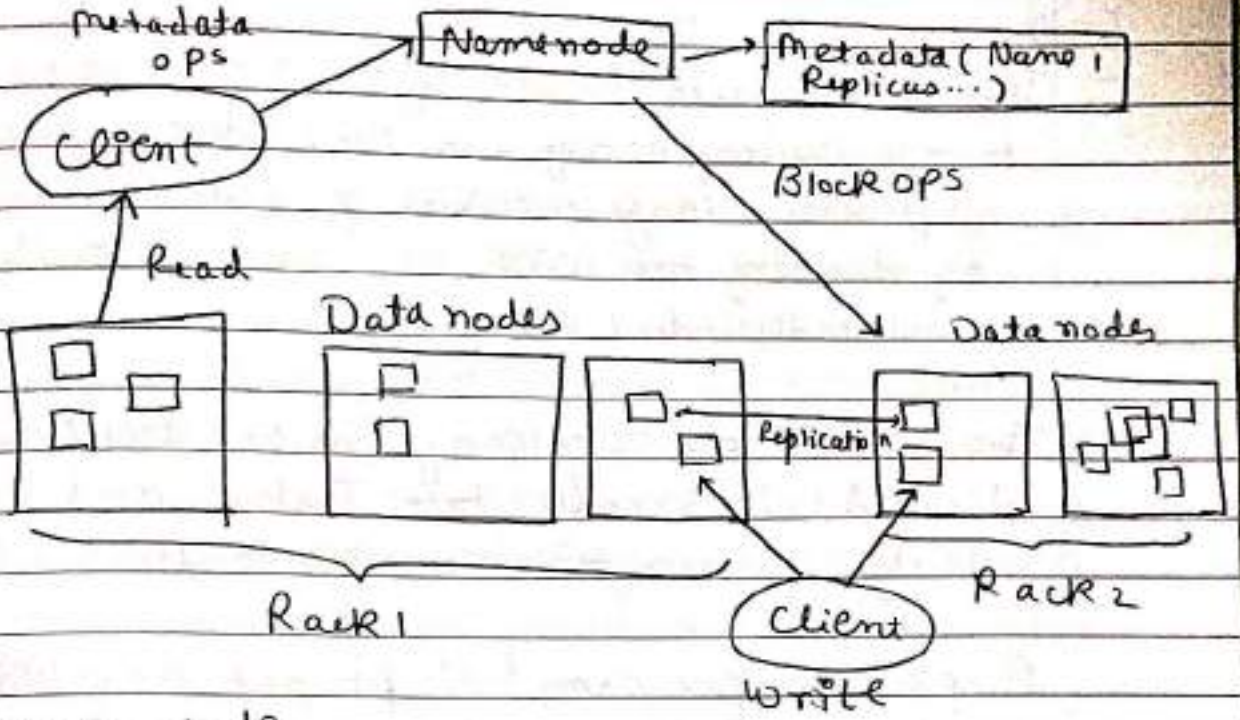
=> For Distributed storage and distributed computation Hadoop uses a master/slave architecture. The distributed storage system in Hadoop is called as the Hadoop Distributed file System or HDFS. In HDFS a file is chopped into 64 MB chunks and then stored / known as blocks.

- ~~As previously discussed HDFS cluster has~~

- HDFS Cluster has master and slave architecture. Name node manages the namespace and of the filesystem.

- In this namespace the information regarding file system tree, ~~met~~ metadata for all the files and directories in that tree etc is stored. For this it creates two files the namespace image and one edit log and stories information in it on consistent basis.

- A client interacts with HDFS by communicating with the Name Node and Data node. The user does not know about the assignment of name node and Data Node for functioning ie. which Namenode and Data node are assigned or will be assigned.



1 Data Node

- Data node is known as the slave in HDFS
- The Data Node takes client block addresses from Name Node
- Using this address client communicates directly with one Data Node.
- For replication of Data a Data Node may communicate with other Data Nodes
- Data Node continuesly informs local change updates to Name Node.
- To create move or delete blocks Data Node receives instructions from the local disk.

2. Name Node.

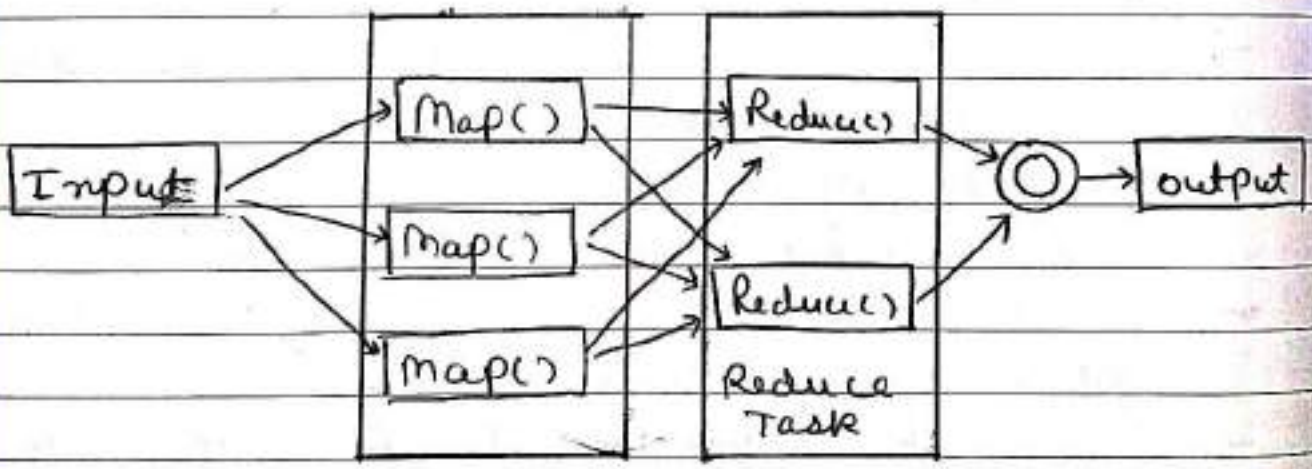
- The Namenode is known as the master of HDFS
- Data Node is known as the slave of HDFS
- The Name node has Job tracker which keeps track of files distributed to Data node.
- Namenode directs DataNode regarding the low level I/O tasks and it is only the Single point of failure.

Q. MapReduce – write a short Note

- The Mapreduce is one of the main components of the Hadoop ecosystem. Mapreduce is designed to process a large amount of data in parallel by dividing the work into some smaller and independent tasks.

- The whole job is taken from the user and divided into smaller ~~tasks~~ tasks, and assign them do ~~to~~ the worker nodes.

- MapReduce program take input as a list and convert to the output as a list also



- The Map Task:
  - The map/ mapper takes a set of keys and values. We can say it as a key value pair as input. The data may be in a structured or unstructured form. The framework can make it into keys and values.

- The Keys are the reference of input files, and values are the dataset.

- The user can create a custom business logic based on their needs from data processing.

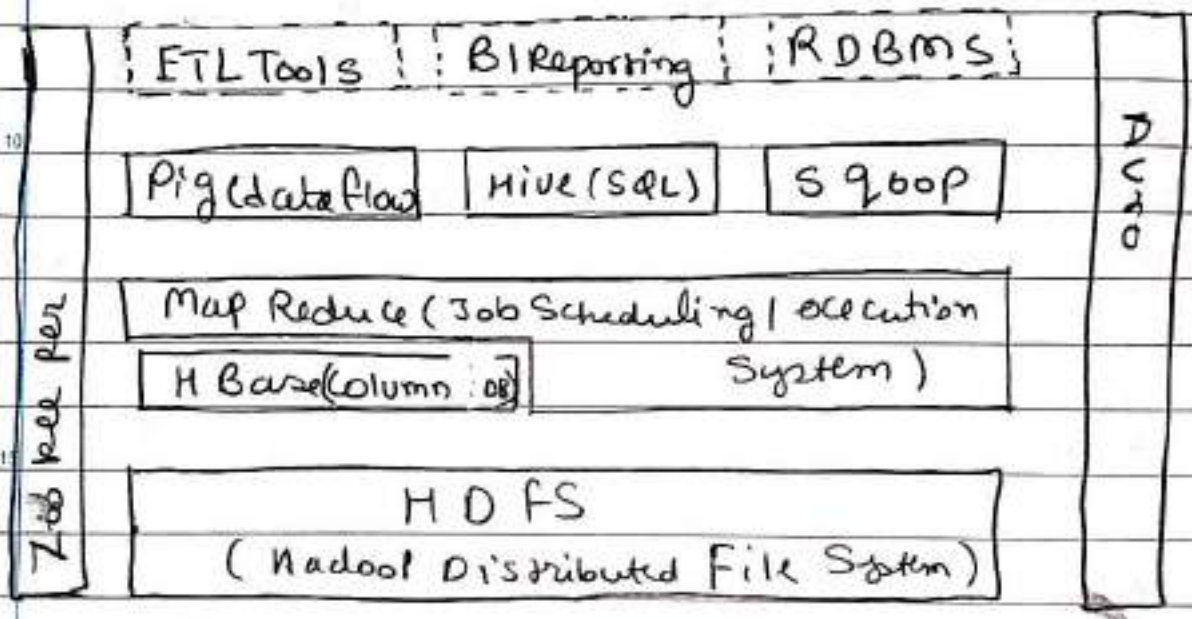
- The task is applied on every input value.

- The Reduce Task:
  - The reducer takes the key value pair, which is created by the mapper as input. The key value pairs are sorted by the key elements.
  - In the reducer we perform the sorting, aggregation or summation type jobs.

- How mapReduce task works?
  - The given inputs are processed by the user-defined methods. All different business logics are working on one mapper section. Mapper generate intermediate data and reducer ~~tasks~~ takes them as input. The data are processed by user defined function in the reducer section. The final output is stored in HDFS.

Q Write a short note on Hadoop Ecosystem

=> Hadoop Ecosystem is a platform or a suite which provides various services to solve the big data problems.



1 Flume: A distributed service for collecting aggrigating, and moving large amounts of log data.

- Has a simple and flixible architecture based on streaming data flow
- Robust and fault Tolerant
- Uses a simple extensible data model that allows for online analytic application.

2. Hive: A data warehouse infrastructure built on top of Hadoop for providing data summarization, queries and analysis.

3. HBase: It is a distributed column oriented database

- It is a hadoop application built on top of HDFS
- It is not a relational database. Hence it does not support SQL

4. HDFS: It is a distributed file system suitable for storing large file.

- HdFS does not support fast individual ~~record~~ lookups.
- It provides high latency batch processing.

5. Mahout: It is a distributed and scalable machine learning algorithm on the hadoop platform.

Eg: provides recommendations on users taste.

6. Pig: It is a high level platform for creating mapreduce program with the use of language called Pig Latin.

- The Pig is used for analyzing and querying on large data set which is stored in HDFS and you can use the pig by using Pigstall

7. Sqoop: It is a tool which is designed for efficiently transferring bulk data between Apache Hadoop and structured data stores such as relational databases.

8. Zookeeper: Zookeeper is used for the coordination. It is a hadoop component used to maintain all the configuration information, naming etc. And this also provides distributed synchronization and manages large clusters of machine doing a proper communication between them.

Hadoop HDFS and Map Reduce.

Distributed file System:

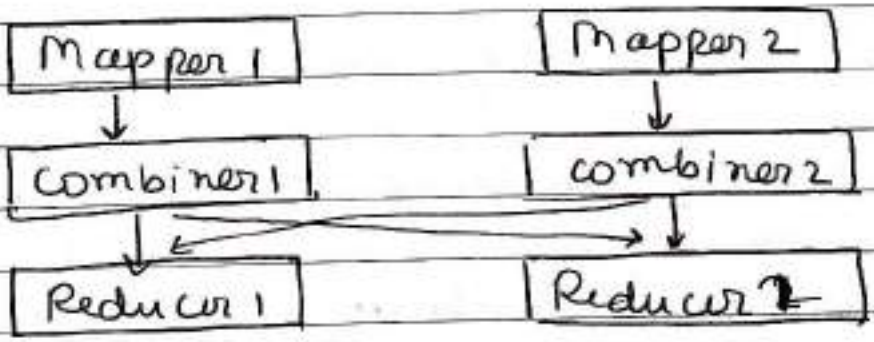
- Distributed File System is any file System that allows acces to file from multiple hosts sharing via a computer network.
- It may include facilities for transparent replication and fault tolerance.
- Different types of Distributed File Systems are Google File System and Hadoop Distributed File Systems.

Q. Write a short note on combiner.

- A combiner is a type of mediator between the mapper phase and the reducer phase. The use of combiners is totally optional.

- As a combiner sits between the mapper and reducer, it accepts the output of map phase as an input and pases the key value pairs to the reduce operation.

- In other words we can say the combiner is also known as mini reducer. It processes the intermediate data from the mapper. Use of combiner is optional.

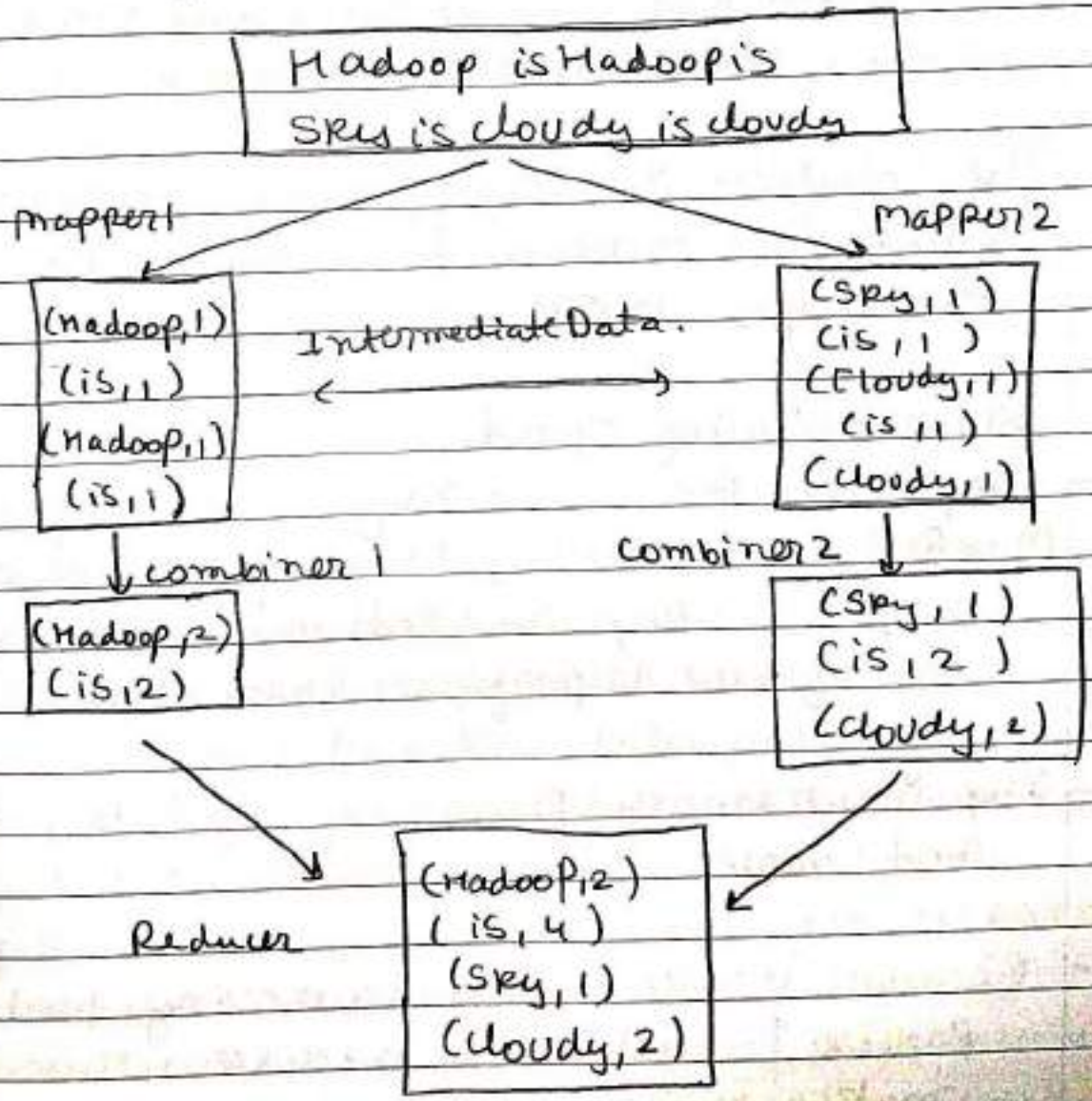


- From the Hadoop mapper, it creates a huge amount of intermediate data. In the process of sending these data to the reducer creates a massive network congestion. The combiner are used to overcome this problem.

- In the below example we can see there is two mappers. The main data is divided into two parts - The mapper finds the intermediate key value pairs.

- The output of the mapper is partially reduced by the combiner

- In the last stage, all output of the combiners is again reduced by the reducer and finds the final output.



Q. Write a short note on Limitations of Hadoop.

- Small File concern
  - Not suitable for small data.
  - HDFS lacks the ability to support the random reading of small file due to its high capacity design.
  - If there are lot of many small files, then the Name Node will be overloaded since it stores the namespace of HDFS
  - The solution is simply merging all the small files to create bigger files and then copy to HDFS
- Slow Processing Speed
  - MapReduce Processes a huge amount of data.
  - MapReduce works by breaking the processing into phases: Map and Reduce. Which requires a lot of time to perform these tasks.
  - The In memory processing of data, Apache and Spark overcome this issue.
- Because it takes no time in moving the data / Process in and Out of the disk, thus this makes it faster.
  - Apache Spark is 100 times faster as compared to MapReduce because it processes everything in memory

o Batch processing / Real time processing

- Hadoop only supports batch processing, it is not suitable for streaming data. Hence overall performance is slower. Mapreduce framework doesn't leverage the memory of the hadoop cluster to the maximum.

- Apache Spark solves this problem as it supports the stream processing.

o Iterative Processing.

- Apache Hadoop is not much efficient for iterative processing.
- As Hadoop is not supported cyclic data flow.
- Spark overcomes this issue.
- As it accesses data from RAM instead of DISK.

o Latency.

- Slower because it supports different format, structured and huge amount of data.
- In Mapreduce, Mapper takes a set of data and broke down into a key value pair.
- Reducer takes the output from the map as input and process further.
- MapReduce requires a lot of time to perform these tasks thereby increasing latency.
- Apache Spark can Reduce this problem.
- Apache Flink data streaming achieves low latency and high throughput.

NoSQL

NoSQL

- NoSQL database Stands for 'Not only SQL" or 'Not SQL'

- Tradidional RPBMS uses SQL Syntax and queries to analyze and get the data for furtner insights.

- No SQL database Management System that provides mechanism for storage and retrival of massive amount of unstructured data in distributed environment.

- The concept of NoSQL database become popular with internet giants like google, FaceBOOK, Amazon, etc. Who deal with huge volume of data. The system response time becomes slow when you use RDBMS for massive volumes of data.

- TO solve this problem, we could 'Scale up' our Systems by upgrading our existing hardware. This process is expensive

- The alternative for this issue is to disribute database load on multiple hosts whenever the load increases. This method is known as "Scaling out".

dt: Noor

8. Features of NoSQL

1. Non-Relational
- NoSQL databases never follow the relational model
- Never provide tables with flat fixed-column records.
- work with self contained aggregated or BLOBs
- Dosent requir Object ~~Orient~~ relational mapping and data normalization
- No complex features like query languages, query planners, referential integrity joins, ACID.

2. ~~Features~~ Schema-free
- No SQL databases are either schema free or have relaxed schemas.
- Do not requir any sort of definition of the schema of the data.
- Offers heterogeneous structures of data in the same domain

3. Simple API
- Offers easy to use interfaces for storage and querying data provided.
- APIs allow low level data manipulation and selection methods.
- Text based protocols mostly used with HTTP REST with JSON.

4. Distributed.
- Multiple NoSQL databases can be executed in a distributed fashion

- Offers autoscaling and fail over capabilities

5. Scalability:
- This can be scaled up and scale out (horizontal scaling).

Q. Differentiate SQL and NoSQL

	parameters	SQL	NoSQL
1	Types of Database	Relational Database	Non-Relational Database.
2.	Schema	predifined Schema	Dynamic Schema
3.	Database catagory	Table-based Database	4 types of database
4.	Scalability	Vertically	Horizontally
5.	Language	SQL	~~NoSQL~~ UQL (unstructured Query Lang)
6.	online processing	online tran-saction processing	Online Analytical processing.
7.	Base property	ACID property	CAP Theorem.

Q. Write ~~two~~ NoSQL Database Types. ~~Give~~ Short note

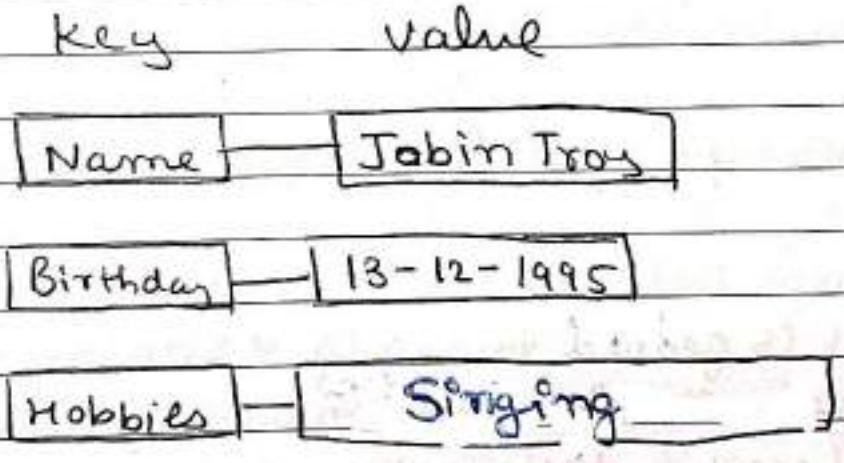
=> There are 4 types of NoSQL Database

1. Key-value ~~Database~~ Store Database

- Key value Stores are the least complex of the NoSQL databases. They are, as the name suggests, a collection of key value pairs.

- This simplicity makes them the most scalable of the NoSQL database type, capable of storing huge amounts of data.

Example



- The value in a key-value store can be anything: a string, a number, but also an entire new set of key value pairs encapsulated in an object.

- Example: Redis, Voldemort, Riak, Amazon's Dynamo

2- Document Store Database

- Document Stores are one step up in complexity from Key Value Stores.

- Document Stores appear the most natural among the NoSQL database types because they're designed to store everyday documents as is, and they allow for complex querying and calculations on this often already aggregated for form of data.

- The way things are stored in a relational database makes sense from a normalization point of view: everything should be stored only once and connected via foreign keys. Document Stores care little about normalization as long as the data is in a structure that makes sense.

- Examples are MongoDB and CouchDB

3. Graph Database

- It is geared towards storing relations between entities in an efficient manner.
- When the data is highly interconnected, such as for social networks, scientific paper citations, or capital asset clusters, graph database is the answer
- Graph or network data has two main concepts
  Node: The entities themselves. In a social network this could be people
  Edge: The relationship between two entities. This relationship is represented by a line and has

its own properties. An edge can have a direction, for example, if the arrow indicates who is whose boss.

- One of the popular example is Neo4j

4. Column Store Database.
- Instead of storing data in relational tuples (table rows), it is stored in cells grouped in columns.
- It offers very high performance and a highly scalable architecture.

- Some common examples of column family database include event logging and blogs like document databases, but the data would be stored in a different fashion

- In logging, every application can write its own set of columns and have each row key formatted in such a way to promote easy look up based on application and timestamp.

~~Cassandra can be a column store~~.

- Examples are: HBase, BigTable, Hyper Table.

Q. Benefits of NoSQL

1 Big Data Analytics

- Big Data is one of main feature, promotes growth and popularity of NoSQL
- NoSQL has good provision to handle such big data.

2. Better data availability

- NoSQL database works with distributed environments.
- NoSQL database environment should provide good availability across multiple data servers
- NoSQL data base can ~~read and~~ supply high performance

3. Location independence

- NoSQL database can read and write database regardless of location of database operation.

Q. Write a short note on Cassandra

=> Apache Cassandra is a highly scalable, high performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL Database.

Features of cassandra

i] Scability:
Cassandra is highly scalable system; it also allow to add more hardware as per data requirment.

iii] 24x7 Availability

- In cassandra there are very few chances of failure
- It is always available for business application.

iii] Good performance:
cassandra system can be scaled up linearly, which means, it increases your outputs as you increase the number of nodes in the cluster.

iv) Good Data Storage:
- Cassandra can store almost all possible data formats including: Structured, semi and Unstructured

- It can manage all change to your data structures as per your need.

v] Data Distribution

Cassandra system provides flexible data distribution, as per need of data replication across multiple data centers.

Q. What is MongoDB?

=> MongoDB is a cross platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

- It supports basic and advanced concepts of the SQL
- It is opensource Database Management System
- It is designed to work with commodity servers. So it is acceptable across all types of industries.

o Advantages of MongoDB over RDBMS
1. Schemaless
2. No complex join
3. Ease to scale out etc

o Features of MongoDB

1. Supports ad hoc queries:
In MongoDB, you can search by field, range queries and it also supports regular expression searches.

2. Indexing:
You can index any field in a document.

3. Replication:

MongoDB supports Master Slave replication. A master can perform Reads and writes and a Slave copies data from the master and can only be used for reads or backups.

4. Duplication of data.

MongoDB can run over multiple servers. The data is duplicated to keep the system up and also keep its running condition in case of hardware failure.

5. Load Balancing:

It has an automatic load balancing configuration because of data placed in Shards.

Q. What is DynamoDB?

=> Dynamo DB allows users to create databases capable of storing and retrieving any amount of data and comes in handy while serving any amount of traffic.

Advantages of dynamo DB

1. It has fast and predictable performance.
2. It is highly scalable.
3. It offloads the administrative burden operating and scaling.
4. It's scalability is highly flexible
5. It provides with on demand backups.

Limitations of DynamoDB

1. All tables and global secondary indexes must have a minimum of one read and one write capacity unit.
2. Only 5 local and five global Secondary index per table are permitted.
3. It does not prevent the use of reserved words as names.

Finding Similar Items

Q. Write a short note on Distance Measure.

=> A set of points is called a space and it is necessary to define any distance measure. Let x and y be two points in the space, then a distance measure is defined as a function which takes the two points x and y as inputs, and produces the distance between the two points x and y as outputs. The distance function is denoted as: $d(x, y)$

- The Output produced by the function d is a real number which satisfies the following axioms:

1. Negativity of distances: The distance between two points x and y cannot be negative
$$d(x, y) \geq 0$$

2. Positivity of distances: The distance between any two points is zero if they have same coordinates.
$$d(x, y) = 0 \text{ iff } x = y$$

3. Symmetry of Distances: The distance between x and y is same as distance between y and x
$$d(x, y) = d(y, x)$$

4. Triangular Inequality of distances: The property says that,
$$d(x, y) \leq d(x, y) + d(z, y)$$

Q. Write a short note on Euclidean Distance.

=> Euclidean Distance is the most popular out of all the different distance measures.

- It is measured on the Euclidean space. If we consider an n-dimensional Euclidean space then each point in that space is a vector of n real numbers. For example, if we coneider the two D Euclidean Space then each point in space is represented by (x_1, x_2) where x_1 and x_2 are real numbers.

* In Euclidean
- There are three majorly used norms here

1. L_2-norm = $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

2. L_1-norm = $|x_1-x_2| + |y_1-y_2|$

3. L-infinity = $\max(|x_1-x_2|, |y_1-y_2|)$

Manhattan distance

=> Consider two points (6,4) and (2,7) in 2D Euclidean space. Find Euclidean distance between them

Ans L2 norm = $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

$= \sqrt{(6-2)^2+(4-7)^2}$

$= 5$

L1 norm = $|x_1-x_2| + |y_1-y_2|$

$= |6-2| + |4-7|$

$= 4+3$

$= 7$

L infinity = $\max(|6-2|, |4-7|)$

$= \max(4, 3)$

$= 4$

=> Consider two points (1,2,2) and (2,5,3) in 3D Euclidean space. Find Euclidean distance between them.

L2 norm = $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}$

$= \sqrt{(1-2)^2+(2-5)^2+(2-3)^2}$

$= \sqrt{11}$

L1 norm = $|1-2|+|2-5|+|2-3|$

$= 1+3+1 = 5$

L∞ norm = $\max(|1-2|, |2-5|, |2-3|)$

$= \max(1, 3, 1) = 3$

Q Write a short note on Jaccard Distance

=> Jaccard Distance is measured in the space of sets. Jaccard distance between two sets is defined as:
$d(x,y) = 1 - SIM(x,y)$

$SIM(x,y)$ is the Jaccard similarity which measures the closeness of two sets. Jaccard similarity is given by the ratio of the size of the intersection and the size of the union sets x and y.

Q Consider two sets $A = \{1,2,3\}$ and $B = \{1,2,4,5\}$. Evaluate how similar are A and B. also find Jacard distance

→ $$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$J(A,B) = |A \cap B| / |A| + |B| - |A \cap B|$$

$$= 2 / 3 + 4 - 2$$
$$= 2/5 = 0.4$$

Jacard Distance = 1 − Jaccard Similarity
$= 1 - 0.4$
$= 0.6$

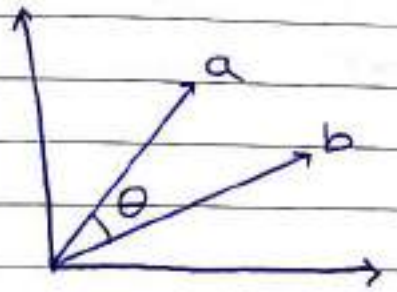
Q. Write short note on Cosine distance

=> Cosine similarity is used in Space

- Space is where points are represented as vectors

- Cosine distance is nothing but angle between 2 vectors.

Note: The angle range from 0° to 180°



less the angle between two vectors more is the similarity.

=> Consider following are the two vectors in Euclidean Space:
$X = (1, 2, -1)$ and $y = (2, 1, 1)$
Calculate the cosine distance between X & y

Ans Cosine Distance is the dot product between two vectors divided by L2 norm of X and Y

$$\text{Cosine Angle} = \frac{\text{Dot Product of Vectors}}{\text{L2 norm of both vectors}}$$

Dot product $= 1 \times 2 + 2 \times 1 + (-1) \times 1 = 3$

L_2 norm of $x = \sqrt{(1)^2 + (2)^2 + (-1)^2}$

$= \sqrt{6}$

L_2 norm of $y = \sqrt{(2)^2 + (1)^2 + (1)^2} = \sqrt{6}$

$$\cos\theta = \frac{\text{Dot product}}{L \text{ norm of } x \cdot L \text{ norm of } y} = \frac{3}{\sqrt{6}\sqrt{6}} = \frac{3}{6} = \frac{1}{2}$$

$$\theta = \cos^{-1}\frac{1}{2} = 60^\circ$$

The angle between two vectors x & y is 60°

Q. Write a short note on edit distance

=> Edit distance is used for calculating the distances between two points where points are represented as strings.

- For Example:
  1. The edit distance between "good" and "goodbye" is 3
  2. The edit distance between "Hello" and "Jello" is 1

- The longest common subsequence (LCS) of x and y can also be used to calculate edit distance.

- An LCS of x and y is a string that is constructed by deleting positions from x and y, and that is as long as any string that can be constructed that way.

- Here we perform deletion operations on characters on respective strings.
  Suppose 'x' and 'y' are two points represented as strings

$$d(x,y) = \text{length of string } 'x' + \text{length of string } 'y' - 2\,LCS(\text{common character})$$

Example 1

1. X = ABCDE Y = ACFDEG

=>

Length of X = 5

length of y = 6

LCS = 4 (ACDE)

d(x, y) = 5+6-(2×4) = 3

Example 2

=> X = abcde y = bcdve

length of 'x' = 5

length of 'y' = 6

LCS = 4 (bcde)

d(x,y) = 5+6-(2×4) = 3

=> Example 3

X = A B C D E

y = A C F D E G

Find Edit Distance

Step 1: we delete 'B' from position 2 of String X

X = A C D E
Y = A C F D E G

Step 2: Insert F at position 3 of String X

X = A C F D E
Y = A C F D E G

Step 3: Insert G at position 6 of string X

X = A C F D E G
Y = A C F D E G

Step 4:

Edit distance = No. of insertion + No. of deletion
= 2 + 1
= 3.

Q. Write a short note on Hamming distance.

=> Hamming Distance is used for the boolean vectors, that is which contain only 0 and 1

- The no. of items in which the two items differ is the Hamming Distance between them

Eg:
consider vectors

$p1 = 10101$
$p2 = 11110$

$d(p1, p2) = 3$

because these vectors differ in the second, fourth and fifth components, while they agree in the first and third components.

Q. Solve using K means Clustering

$K = \{2, 3, 4, 10, 11, 12, 20, 25, 30\}$

Consider: 2 clusters and $m_1 = 4$ and $m_2 = 12$

=> Step 1: $K_1 = \{2, 3, 4\}$ $K_2 = \{10, 11, 12, 20, 25, 30\}$

Average: $m_1 = 3$ $m_2 = \frac{108}{6} = 18$

Step 2: $K_1 = \{2, 3, 4, 10\}$ $K_2 = \{11, 12, 20, 25, 30\}$

Take average $m_1 = 4.75$ $m_2 = 19.6$

Step 3: $K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

Average $m_1 = 7$ $m_2 = 25$

Step 4: $K_1 = \{2, 3, 4, 10, 11, 12\}$ $K_2 = \{20, 25, 30\}$

Average $m_1 = 7$ $m_2 = 25$

We are getting same means!

So the new clusters are

$K_1 = \{2, 3, 4, 10, 11, 12\}$

$K_2 = \{20, 25, 30\}$

Clustering

What is Clustering?

When you are trying to learn ~~about~~ about something, say music, one approach might be to look for meaningful groups or collections. you might organize music by genre, while your friend might orginize music by decade. How

How you choose to group help you understand more about them as individual piece of music. ~~For~~ you might find that you have a deep affinity for lofi music and further break down the genre into different approaches of music from different locations. On the other hand your friend might look at music from the 1980s and be able to understand how the music across genres at that time was influenced by the Sociopolitical climate. In both cases you and your friend have learned something interesting about music, even though you took different approaches.

In machine learning too we often group examples as a first step to understand a subject in a machine learning system. Grouping unlabeled examples is called clustering.

As the examples are unlabeled, clustering relies on unsupervised machine learning. If the examples are labeled, then clustering becomes classification.

Q. Write a short note on CURE algorithm.

=> Whenever we have given a large dataset it is important to cluster it and to cluster this data we use CURE algorithm which stands for Clustering Using Representatives.

- We use Representative points to cluster
- Clustering is useful for discovering groups and identifying interesting distributions in the underlying data.
- Traditional clustering algorithms either favour clusters with spherical shapes and similar size or are very fragile in the presence of outliers.
-

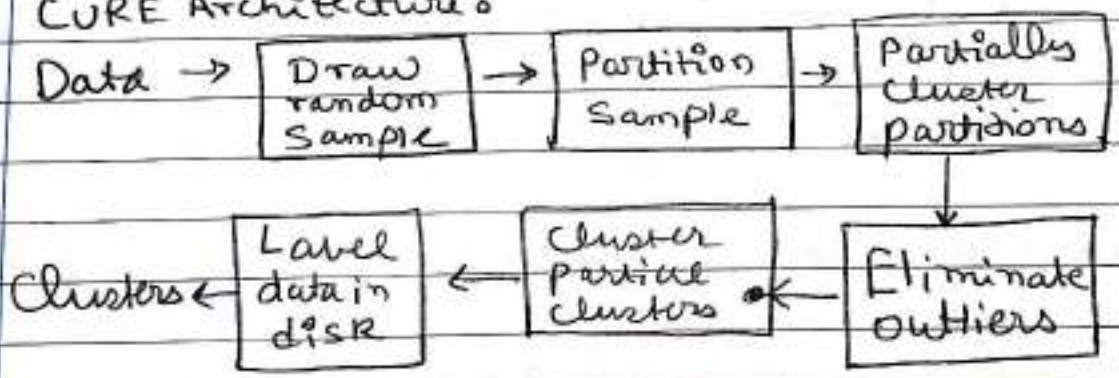
- Thus CURE algorithm came into picture which is more robust to outliers and identifies clusters having non spherical shapes and wide variances in size

- CURE algorithm works better in spherical as well as non-spherical clusters.

- It prefers a set of points which are scattered as representative cluster than all points or centroid approach.

- CURE uses random sampling and partitioning to speed up clustering.

CURE Architecture :



⇒ Advantages of CURE Algorithm

* CURE can detect clusters of non-spherical shape, with variation of size with the representative points for each cluster.

* Good execution time with large database and sets using random sampling and partition methods.

* Works well with outliers, which are detected and merged or eliminated.

Extra points about CURE Algorithm

- For each cluster, B well scattered points within the cluster are chosen, and then shrinking them towards the mean of the cluster by a fraction α
- The distance between two clusters is then the distance between the closest pair of representative points from each cluster
- The B representative points attempt to capture the physical shape and geometry of the cluster
- Shrinking the scattered points towards the mean gets rid of surface abnormalities and decrease the effects of outliers.

Recommendation System

Q. Write a Short Note on Recommendation System.

=> There are two ways that user can interact with large data item set:

① Search: Users know what they are looking for. They know the precise item

② Recommendation: If the items in the products are large and similar then user often doesn't know what he/she is looking for. So in such case recommendation systems come into picture.

- The Reason behind ~~we need~~ why do we need recommendation system is the system recommends to the user certain items that they think user may be interested in, based on the study of user profile

- The key that made recommendation System so important is that we moved from era of Scarcity to the era of abundance.

- Self space is a scarce commodity for traditional retailers
  Also: TV Networks, movies theaters

- The web enables near-zero-cost dissemination of information about the products. The web has many more products ever before. ie there is limitation to self space. Product with Amazon are much much more than products which individual retailer can hold.

- Recommendation system is a facility that involves predicting user responses to option in web applications. We have seen the following recommendations:
  ① "you may also like these", "People who liked this also liked.."

  ② If you download presentations from Slideshare, it says "Similar content you can save and browse later"

- These suggestions are from recommendation systems. The paradigm used are as follows:
  ① collaborative filtering systems
  ② Content Based systems.

collaborative filtering

=> The Recommendations are done based on the users behaviour. History of users plays an important role.

- For Example: If user X likes 'ACDC', Nucleya and Green day while the user y likes 'ACDC, Nucleya and lil Nas X then they have similar interests.

- So there is a huge similarity that the user X would like lil Nas X and user y would like green day

- This is the way collaborative Filtering is done

- There are two types of collaborative filtering techniques
  1. User user collaborative Filtering
  2. Item - Item collaborative Filtering.

1. User - User collaborative Filtering

- In this the user vector includes all the items purchased by the user and rating given for each particular product

- The similarity is calculated between user using an $n * n$ matrix in which n is the no. of users present.

- The similarity is calculated using the cosine similarity formula

- Now, the recommending matrix is calculated. In this the rating is multiplied by the similarity between the users who have bought this item and the user to which item has to be recommended

- The value is calculated for all items that are new for that user and are sorted in descending order.

- Then the top items are recommended to that user

- If a new user comes or old user changes his or her rating or provides new ratings then recommendations may change

2. Item-Item collaborative filtering

- In this rather than considering similar users similar items are considered.

- ~~Avengers hey may~~
~~Batman~~

- Here the recommendation matrix is m * m matrix where m is the number of items present.

5. Collaborative Filtering

Pros

① No Knowledge engineering efforts needed

② Serendipity in results

③ Continuous learning for market process

Cons

① Rating feedback is required

② New items and users face to cold start.

Content based Filtering

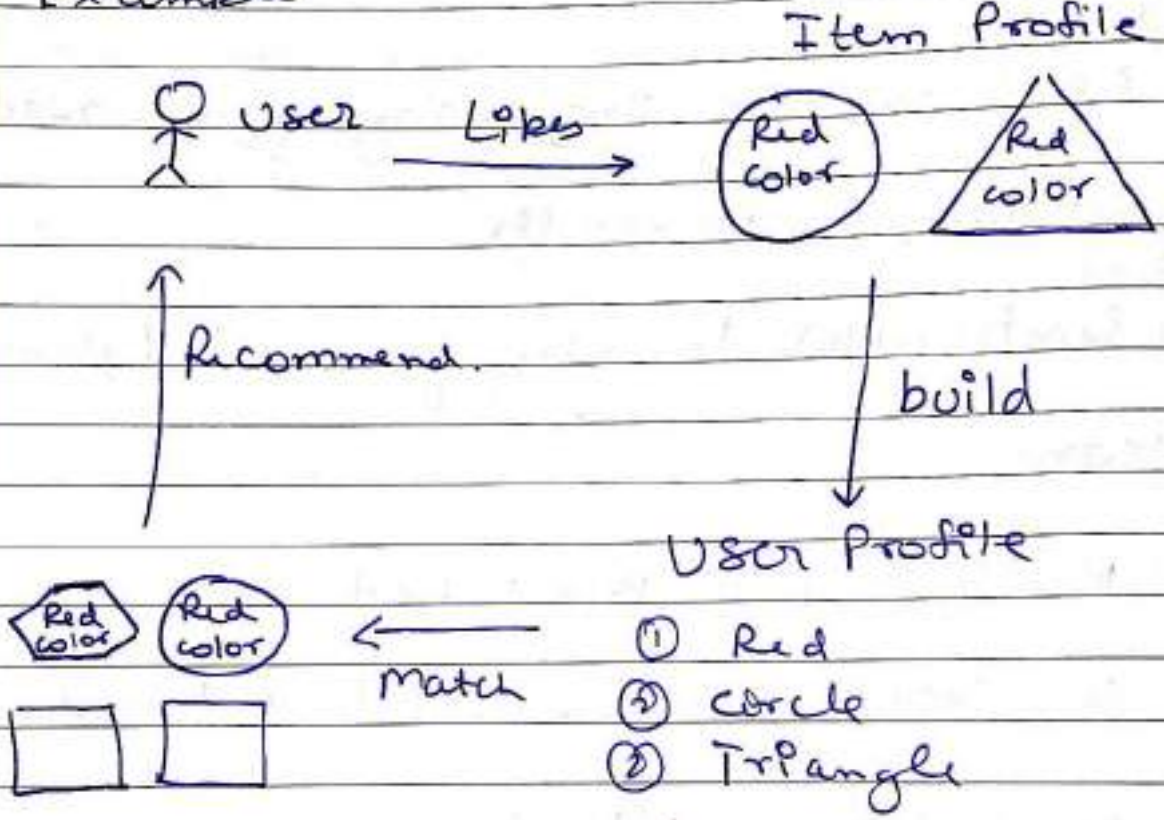
- This technique is based on the information of & or some description provided for the product.

- The system finds the similarity between products based on its context or description

- The person or user's previous history is taken into account to find similar products that user may like.

for example if a user likes movies such as '3 idiots' then we can recommend him the movies of 'Amir Khan' or movies with the genre 'Comedy'.

Example



- In this filtering two types of datacare used first, the likes of users, the users interests, users personal information such as age or ~~something~~ sometimes the user history too. This data is represented by the user vector

- Second information related to the products known as an item vector. The item vector contains the top features of all items based on which similarity between them can be calculated.

Pros :

1. No. community Requirment
2. Items can be compared among themselves

Cons

1. Need of content description
2. New users face cold start.

Mining Social Media Network Graph

Introduction

- A Social Network is a Social Structure between actors, mostly individuals or organizations.

- It indicates the ways in which they are connected through various social familiarities, ranging from casual acquaintance to close familial bonds

- A Social Network, we think of Facebook, Twitter, Google+, or another website that is called a "Social network", and indeed this kind of network is representative of the border class of networks called "Social".

- The essential characteristics of a social network are:

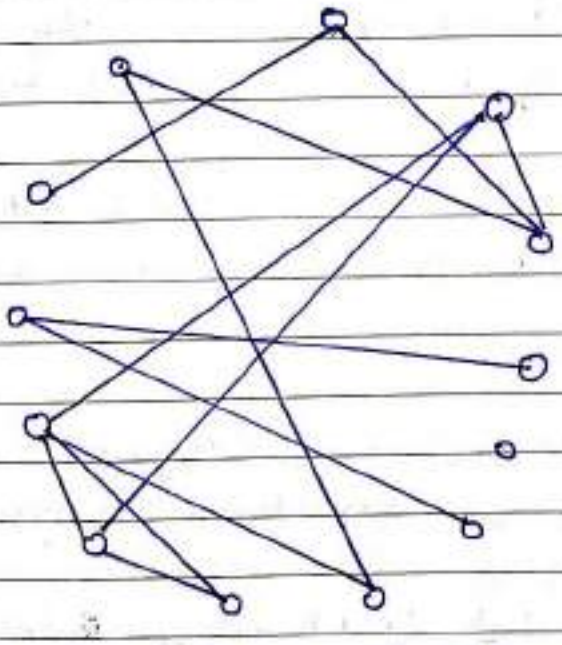
1. There is a collection of entities that participate in the network. These entities are ~~and~~ people, but they could be something else entirely.

2. There is at least one relationship between entities of the network. On facebook or its like, this relationship is called friends. Sometimes the relationship is all or nothing two people are either friends or they are not

- However in other examples of social networks, the relationship has a degree. This degree could be discrete eg. friends, family, acquaintances, or none as in Google+. It could be a real number; an example would be the fraction of the average day that two people spend talking to each other.

Social Networks as Graphs.

- People are represented as nodes
- Relationships are represented as edges: relationships may be acquaintanceship, friendship, co-authorship etc.



- Two or more people, who interact with one another, share similar characteristics and attributes and collectively have a sense of unity

- There are multiple reasons to find Social groups and communities

1. Behavior analysis
2. link prediction
3. Media use
4. Security
5. Social Studies.

8. Write short Note on Girvan Newman Algorithm

=> The Girvan Newman technique for the detection and analysis of community structure depends upon the iterative elimination of edges with the highest number of the shortest paths that pass through them

- By getting rid of the edges the network breaks down into smaller networks or communities.

- The algorithm as the name suggest is introduced by Girvan and Newman

- The idea was to find with edge is in a network occur most frequently between other pairs of nodes by finding edge betweenness. betweenness

- The edges joining communities are then expected to have high edge betweenness

- We can express Girvan Newman algorithm in the following procedure:

(1) Calculate edge betweenness for every edge in the graph.

(2) Remove the edge with highest edge betweenness.

(3) Calculate edge betweenness for remaining edges.

(4) Repeat Step 2-4 until all edges are removed

- In order to calculate edge betweenness it is necessary to find all shortest paths in graph

- The algorithm starts with one vertex, calculate edge weights for paths going through that vertex and then repeat it for every vertex in the graph & sums the weights for every edge.

limitations:

- The algorithm is not very time efficient with networks containing large number of nodes and data.
- Communities in huge and complex networks are difficult to detect and therefore Girvan Newman is not favourable for very large number of data set.

Link Analysis

Page Rank.

- Page Rank is an algorithm used by Google Search to rank websites in their search engine results.

- Page Rank was named after Larry Page, one of the founders of Google. Page Rank is a way of measuring the importance of website pages.

- According to Google:

① Page Rank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

② It is not the only algorithm used by Google to order search engine results, but it is the first algorithm that was used by the company, and it is one best known.

- There are two popular algorithms to rank web pages by popularity.
  ① HITS – Hypertext Induced Topic Search Algorithm.

① Pagerank Algorithm

About Page Rank Algorithm

- The Page Rank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page

- PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process

- The Page Rank computations require several passes called 'iterations', through the collection to adjust approximate

- Pagerank values do more closely reflect the theoretical true value.

Links in PageRank.

- If we consider that, there are 150 million web pages exists in the some part of world wide web (www) then all pages may have approximately 1.7 billion links to different web pages.

- The number of links exists between two or more web pages can be categorized as follows

(1) Back links: Backlink indicates given web page is referred by how many number of other web pages.

(2) Forward link

- Forward link represents the fact that, how many web pages will be referred by a given web pages.

- Clearly, out of these two types of links back links are very important from Ranking of documents perspective

- A web page which contains number of backlinks is said to be important web page and will get upper position in Ranking

About Spider Traps and Dead Ends

- Spider traps: There are sets of web Pages with the property that if you enter that set of pages, you can never leave because there are no links from any page in the set to any page outside the set.

Dead Ends: Some Web Pages have no Out links. If the random walker arrives at such a page there is no place to go next, and the walk ends

- Any dead end is, by itself, a Spider trap. Any page that links only to itself is a Spider trap

- If a Spider trap can be reached from outside then the random walker may wind up there eventually and never leave.

Hiding Spider Traps and Dead Ends

- Limiting random walker is allowed to wander at random. We let the walker follow a random Out link, if there is one, with probability β (β normally, $0.8 \leq \beta \leq 0.9$). With probability $1-\beta$ (called the taxation rate), we remove that walker and deposite a new walker at a randomly chosen Web page.

- If the walker gets stuck in a spider trap it doesn't matter because after a few time steps, that walker will disappear and be replaced by a new walker.

- If the walker reaches a dead end and disappears, a new walker takes over shortly.

Q. Write short note on Hubs and Authorities

- The hubs and authorities is an extension to the concept of Page Ranking. Hubs and authorities will add more preciseness to the existing page rank mechanism

- The ordinary, traditional page Rank algorithm will calculate the page rank for all the web pages available in a given web structure. But user doesn't want to examine or view all of these web pages. He/She just want first 20 to 50 pages in an average case.

- Hence the idea of hubs and authorities will come into existance to have efficiency and reduce work load calculating page rank.

- On hubs and authorities page rank will be calculated for only those web pages who will fetch in resultant set of web pages for a given search query.

- It is also known as hyperlink induced topic Search abbreviated as HITS.

- The traditional pagerank calculations have a single view for a given web page. But hubs and authorities algorithm will have two different shades of views for a given web page.

(1) Some web page has importance as they will present significant information of given topic so these web pages are known as the authorities.

(2) Some web pages has importance because they gives us the information of any randomly selected topic as well as they will direct us to other web pages to collect more information about the same. Such web pays known as hubs.

Mining Data Streams

Introduction

- Suppose that for our boat rental operation, we want ~~to~~ to use a network of sensors to track the current condition of the lake. One idea would be to store this data in a relational database

- We might also want to track this information in real time. For example, we might have a dashboard that updates with the latest average temperature.

- When dealing with such a system, the first issue is to do with storage.

- Not only will there be a massive number of readings to be stored; most of these readings are ~~not~~ only relevant for a limited period of time. So we'd have to deal with frequent insertion as well as removal

- The second issue is to do with performance.

- If we're looking to ~~compute~~ compute the latest average temperature, we might consider taking the average of all temperatures within a five minute window.

```
SELECT AVG (temp)
FROM weather
WHERE timestamp >= Sysdate - 5/(24*60)
```

- If we repeat this query every minute or so we'll be redoing a significant amount of computation each time. What about real time?

Data Streams

- What we're dealing with is a data stream (Now Data Stream comes into the picture)

- Data Stream are defined as infinite, time ~~Oriented~~ ordered sequence of tuples.

- Each of these tuples follow a schema - in our case, the Weather Schema.

- Traditional DBMSs are not built for data streams. When we process data streams, we have a different set of goals:

* Continuous results while tuples are arriving

* Not memorizing all past tuples for future use

- Few applications are Sensor networks, manufacturing process, web logs and click-streams, Telecom records etc.

- Few challenges are:

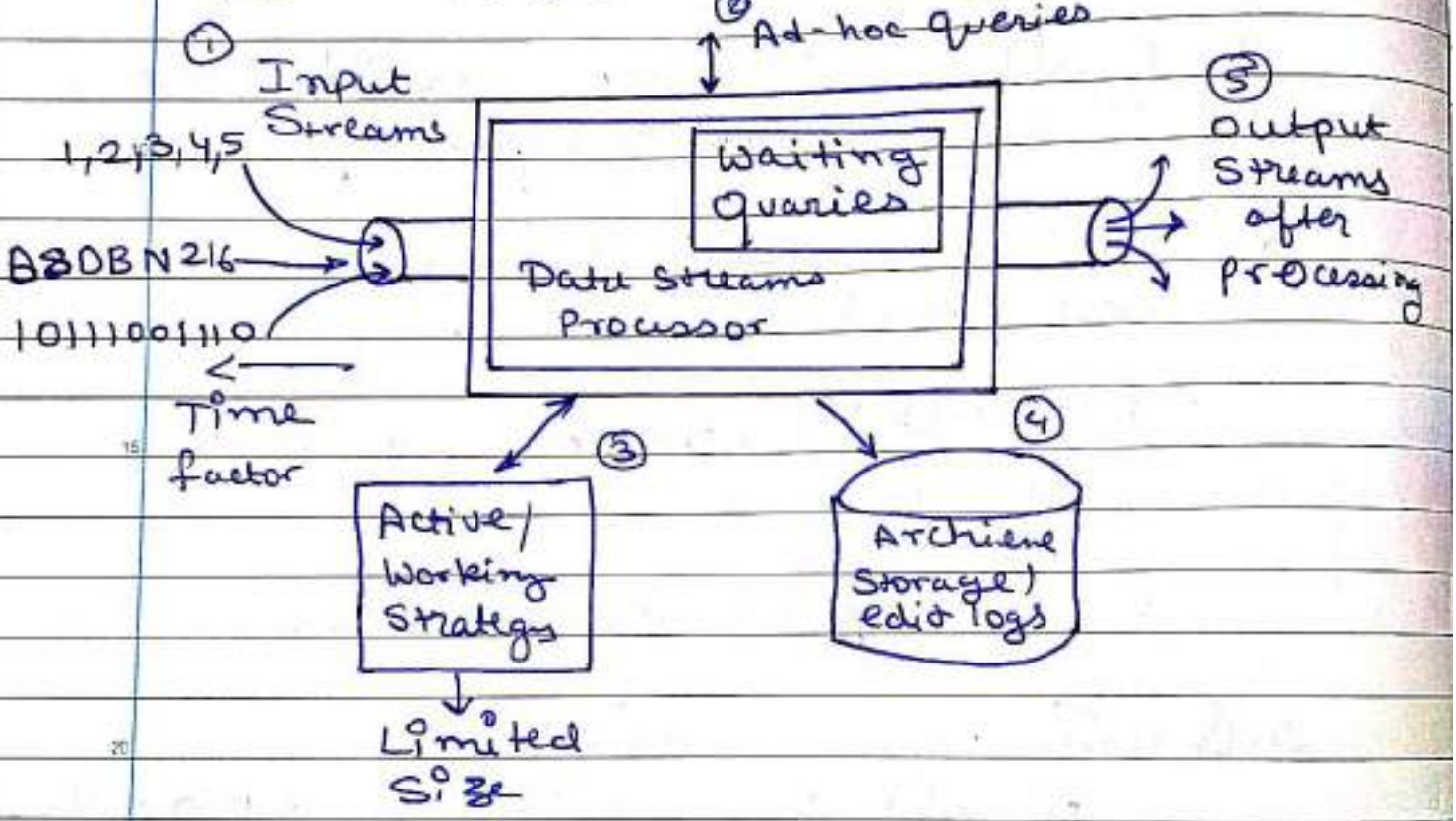
1. Multiple, continuous, rapid, time-varying streams of data.

2. Queries may be continuous (not just one-time)
   - Evaluated continuously as stream data arrives
   - Answer updated over time.

3. Queries may be complex.
   - Beyond element-at-a-time processing
   - Beyond stream-at-a-time processing

Q Write a short Note on Data Stream Management System (DSMS)

=> A DSMS is very similar to a conventional relational Database Management System.



1. Input Streams: The characteristics which it has are =>
   - There can be one or more number of input Streams entering the System
   - The Streams can have different data types.
   - The rate of data flow of each Stream may be different
   - Within a Stream the time interval between the arrival of data items

may differ. For examples, suppose the second data item arrives after 2 ms from the arrival of the first data item, then it is not necessary that the third data item will also arrives after 2 ms from the arrival of the second data item. It may arrive earlier or even later.

2. Stream Process: All types of processing such as sampling, cleaning, filtering and querying on the input stream data are done here. Two types of queries are supported which are standing queries and ad-hoc queries. We shall discuss both the ~~queries~~ query types in details in the upcoming section.

3. Working Storage: A limited memory such as a disk or main memory is used as the working storage for storing parts or summaries of streams so that queries can be executed. If faster processing is needed, main memory is used otherwise secondary storage disk is used. As the working storage is limited in size, it is not possible to store all the data received from all the streams.

4. Archival Storage: The archival store is a large storage area in which the streams may be archived but execution of queries directly on the archival store is not supported. Also, the fetching of data from this store takes a lot of time as compared to the fetching of

data from the working store.

5. Output Streams:

The output consists of the fully processed Streams and the results of the execution of queries on the Streams.

- The difference between a conventional database-management System and a data Stream management System is that in case of the database management System all of the data is available on the disk and the System can control the rate of data reads. On the other hand, in case of the data Stream management System the rate of arrival of data is not in the control of the System and the System has to take care of the possibilities of data getting lost and take the necessary precautionary measures.

Q. Explain different categories of stream queries

⇒ ① Standing queries

- This query is stored in a designated place inside the stream processor. The standing queries are executed whenever the conditions for that particular query becomes true.

- For example, if we take the case of a temperature sensor then we might have the following standing queries in the stream processor:

  - Whenever the temperature exceeds 50°C, output an alert
  - On arrival of a new temperature reading, produce the average of all the readings arrived so far starting from the beginning.
  - Output the maximum temperature ever recorded by the sensor, after every new reading arrival

② Ad-hoc queries

- An ad-hoc query is not predefined and is issued on the go at the current state of the streams. The nature of the ad-hoc queries cannot be determined in advance

- To allow a wide range of arbitrary ad-hoc queries it is necessary to store a sliding window of all the streams in the working storage. A sliding window is nothing but the most recent elements in the stream. The number of elements to be accomodated in the sliding window has to be determined beforehand. As and when new elements arrive, the oldest ones will be removed from the window and hence the name sliding window

- Instead of determining the size of the sliding window in advance we may also take another approach based on the unit of time. In this approach the sliding window may be designed to accomodate say all the stream data for an hour or a day or a month, etc.

- for example, a social networking website like facebook may want to know the number of unique active users over the past one month.

Q Write Short Note on Bloom Filter

- A Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set

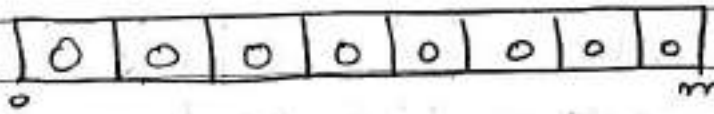
- For eg: Checking availability of username is a set membership problem, where the set is a list of all registered username

- The problem with the bloom filters is that they are probabilistic in nature that means there might be some false positive result.

False positive: It might tell that given username is already taken but actually it's not.

Working of Bloom filtering.

- An empty bloom filter is a bit array of m bits, all set to zero like this



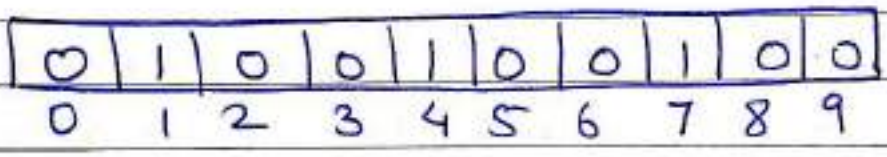
- We need K number of hash functions to calculate the hashes for a given input.

- When we want to add an item in the filter, the bits at K indices $h_1(x), h_2(x), \ldots h_k(x)$ are set where indices are calculated using hash functions.

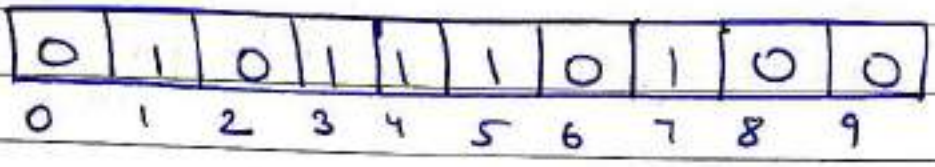
Example: Suppose we want to enter 'throw' in the filter, we are using 3 hash functions and a bit array of length 10, all set to 0 initially.
First we'll calculate the hashes as following:

h_1("throw") % 10 = 1
h_2("throw") % 10 = 4
h_3("throw") % 10 = 7

Note: These outputs are random for explanation only
Now we will set the bits at indices 1, 4, 7 to 1



again we want to enter "catch", similarly we'll calculate hashes
h_1("catch") % 10 = 3
h_2("catch") % 10 = 5
h_3("catch") % 10 = 4
Set the bits at indices 3, 5 and 4 to 1



- Now if we want to check 'throw' is present in filter or not. We'll do the same process

but in reverse order. Calculating respective hashes using h_1, h_2 and h_3 and check if all indices are set to 1 in the bit array.

- If all bits are set then we can say that "throw" is "Probably Present"

- If any of the bit at these indices are 0 then "throw" is "definitely not present".

* False Positive in Bloom Filters

- The Question is why we said "Probably Present" Why this uncertainty. Lets take an example:

Example: Suppose we want to check wether "cat" is present or not. We will calculate hashes using h_1, h_2, h_3

h_1("cat")%10 = 1
h_2("cat")%10 = 3
h_3("cat")%10 = 7

If we check the bit array, bits at these indices are set to 1 but we know that "cat" was never added to the filter. Bit at index 1 and 7 was set when we added "throw" and bit 3 was set when we added "catch".

- We can control probability of getting false positive by controlling the size of Bloom filter

- If we want to decrease the probability of false positivity then increase Number of hash function.

Probability of False Positivity

$$P = \left(1 - \left[1 - \frac{1}{m}\right]^{kn}\right)^k$$

where, m = size of bit array
n = No. of expected elements to be inserted in filter
k = No. of hash functions

Size of Bit Array: $m = \frac{-n \log P}{(\log 2)^2}$

Optimum number of hash functions: $K = \frac{m}{n} \log 2$

Q. Write Short note on Flajolet Martin Algorithm

=> It approximates the number of unique objects in a stream or a database in one pass.

- If the stream contains n elements with m of them unique, this algorithm runs in O(n) time and needs O(log(m)) memory.

- The Flajolet Martin algorithm is an algorithm for approximating the number of distinct elements in a stream with a single pass.

* The major components of this algorithm are:
  1. A collection of hash functions
  2. A bit-string of length L such that $2^L > n$. A 64-bit string is sufficient for most cases.

- Each incoming element will be ~~hashed~~ hashed using all the hash functions. Higher the number of distinct elements in the stream, higher will be the number of different hash values.

- On applying a hash function h on any element of the stream e, the hash value h(e) is produced. We convert h(e) into its equivalent binary bit-string.

This bid string will end in some number of zeroes. For instance, the 5 bid string 11010 ends with 1 zero and 10001 ends with no zeroes.

- This count of zeroes is known as the tail length. If R denotes the maximum tail length of any element e encountered thus far in the stream, then the estimate for the number of unique elements in the stream is 2^R.

- Now to see that this estimate makes sense we have to use the following arguments using probability theory.

  - The probability of h(e) having a tail length of atleast r is 2^{-r}

  - The probability that none of the m distinct elements have tail length of at least r is $(1-2^{-r})^m$

  - The above expression can also be written as $((1-2^{-r})^{2r})^{m2^{-r}}$

  - and we can reduce it to $e^{-m2^{-r}}$ as $(1-2^{-r})^{2^r} = e^{-1}$

- If $m \gg 2^r$ the probability of finding a tail of length at least r approaches 1
- If $m \ll 2^r$, the probability of finding a tail of length at least r approaches 0.
- So we can conclude that our estimate of 2^R is neither going to be too low nor too high.

Q. Write a Short Note on DGIM Algorithm

=> DGIM algorithm (Datar-Gionis-Indyk-Motwani), is designed to find number of 1's in a data set

- This algorithm uses $O(\log^2 N)$ bits to represent a window of N bit.

- It allows to estimate the number of 1's in the window with an error of no more than 50%.

- DGIM has two main components which are Timestamp and Buckets

- Each bit that arrives has a timestamp, for the position at which it arrives

- If the first bit has a timestamp 1, the second bit has a timestamp 2 and so on ... the positions are recognized with the window size N (which are usually taken as multiple of 2)

- The windows are divided into buckets consisting of 1's and 0's

- The following six conditions must be satisfied by the buckets:

1. The right end of every bucket must be occupied by a 1

2. Every 1 should be inside some bucket

3. A bit cannot be inside more than one bucket. In other words the buckets cannot overlap.

4. The number of buckets of a particular size can be either one or two. There will also be a limit of the maximum size of the bucket for a particular stream.

5. The size of a bucket is always a power of 2.

6. On moving from right to left the size of buckets will increase.

KING WARRIOR MAGICIAN LOVER

ROBERT MOORE
DOUGLAS GILLETTE

Tiny Changes, Remarkable Results

Atomic Habits

An Easy & Proven Way to Build Good Habits & Break Bad Ones

James Clear

INTERNATIONAL BESTSELLER

THE WAY OF THE SUPERIOR MAN

20th Anniversary Edition

A Spiritual Guide to Mastering the Challenges of Women, Work, and Sexual Desire

DAVID DEIDA

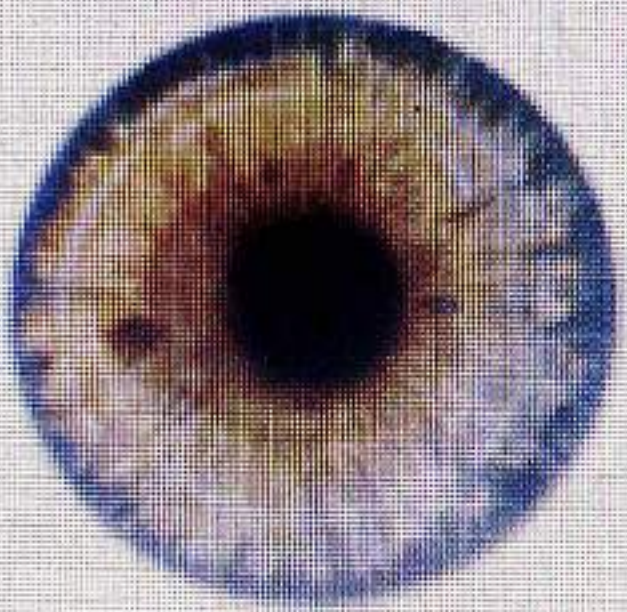
MEN ARE FROM MARS, *Women Are from Venus*

JOHN GRAY

A PRACTICAL GUIDE FOR IMPROVING COMMUNICATION AND GETTING WHAT YOU WANT IN RELATIONSHIPS

FROM THE AUTHOR OF *SAPIENS*

Yuval Noah Harari



21 Lessons for the 21st Century

DEEP

RULES FOR FOCUSED SUCCESS
IN A DISTRACTED WORLD

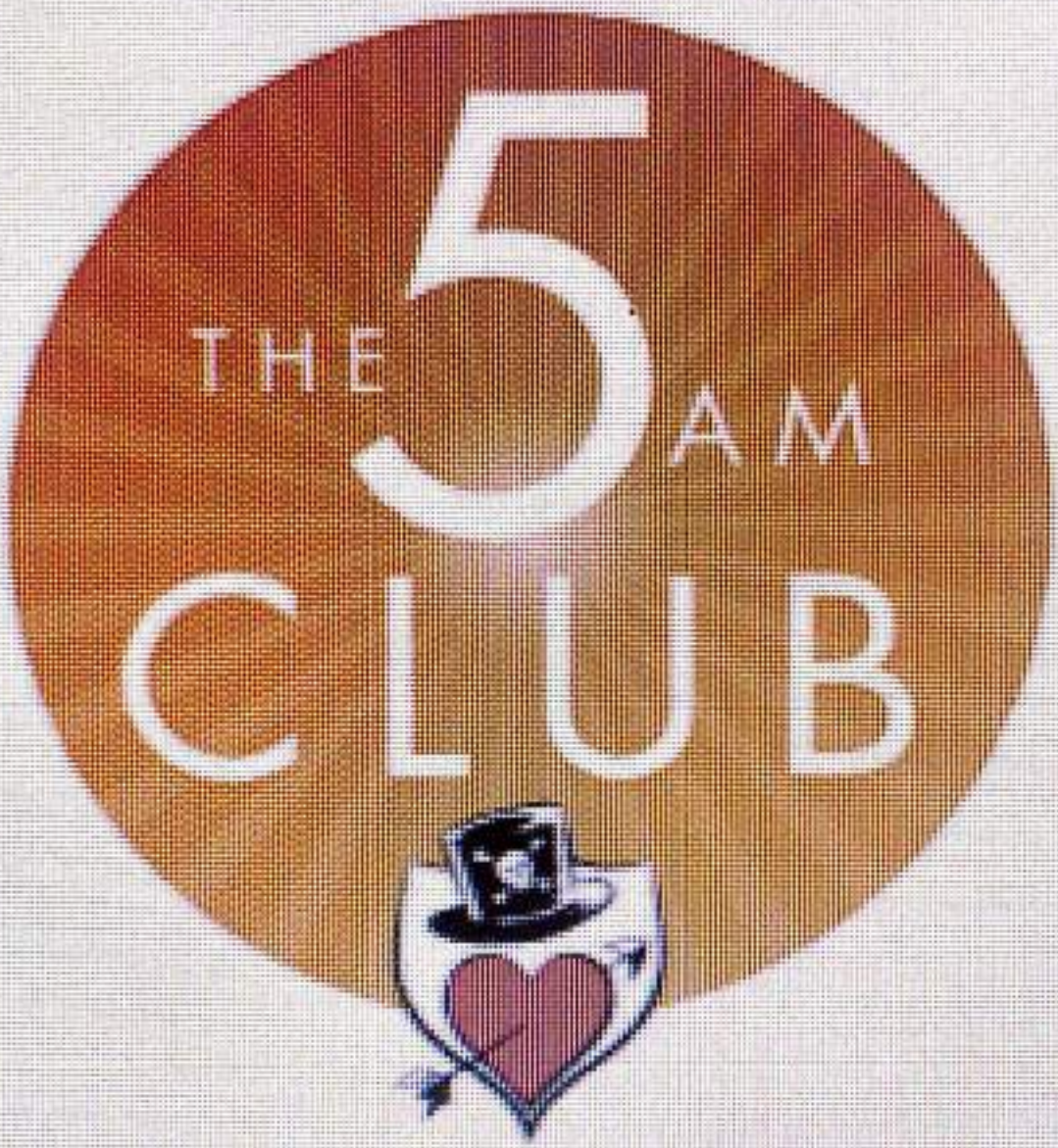
WORK

CAL NEWPORT

THE #1 BESTSELLING AUTHOR OF *THE MONK WHO SOLD HIS FERRARI*

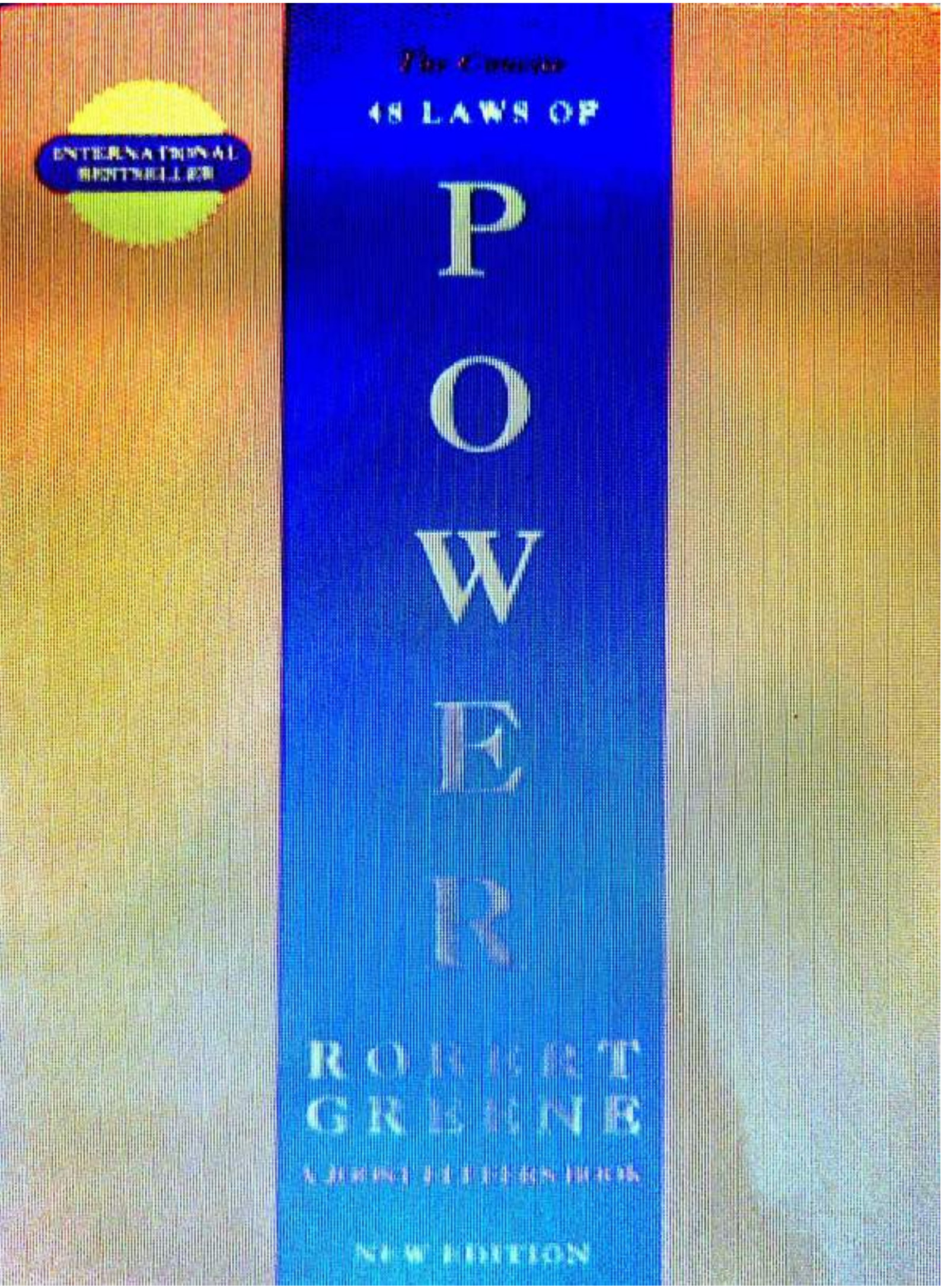
ROBIN SHARMA

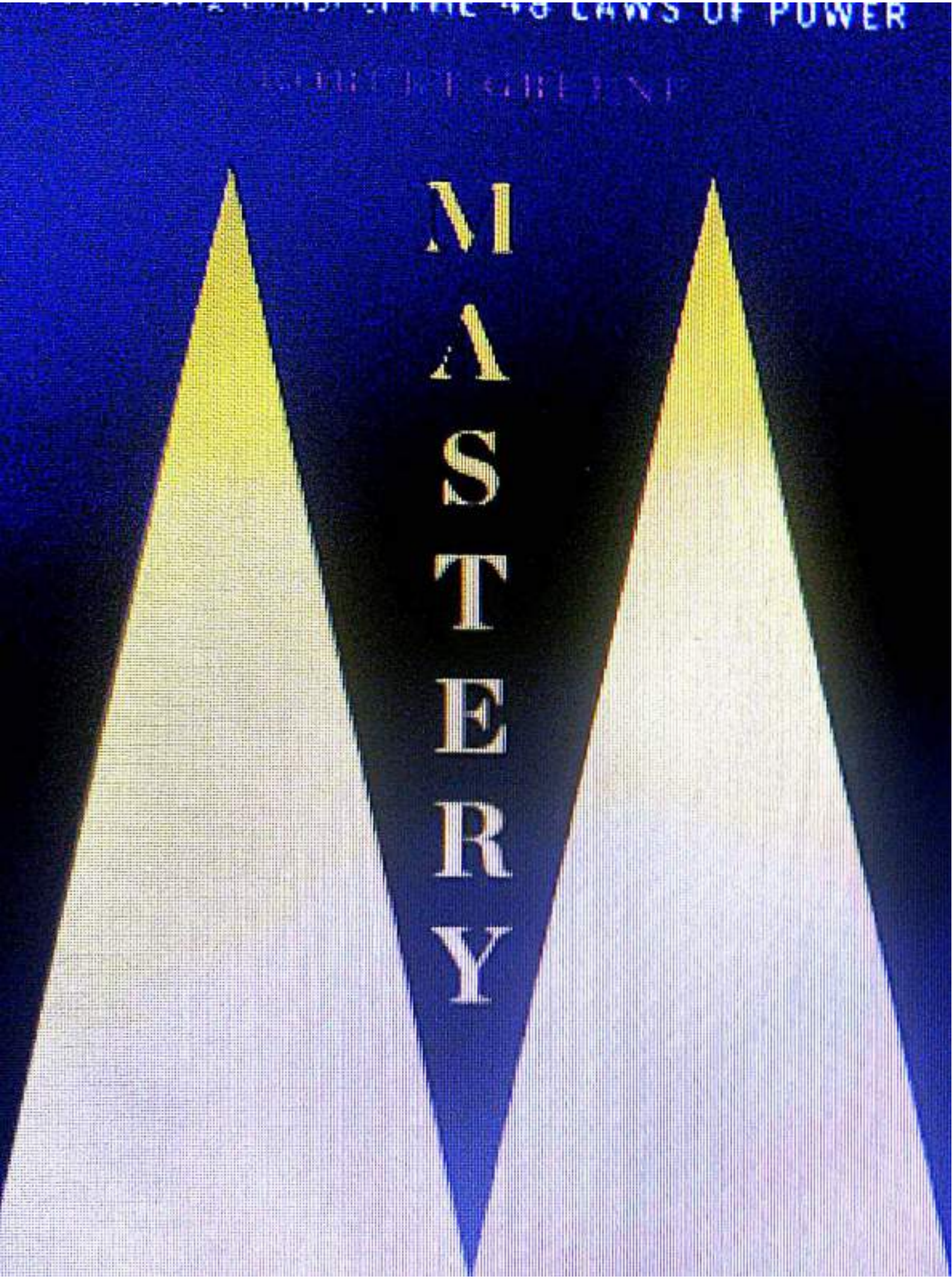
15 MILLION BOOKS SOLD WORLDWIDE



OWN YOUR MORNING

ELEVATE YOUR LIFE


THE CONCISE
48 LAWS OF
POWER
ROBERT GREENE
NEW EDITION


THE 48 LAWS OF POWER
MASTERY

JORDAN B. PETERSON

12 RULES FOR LIFE

AN ANTIDOTE TO CHAOS

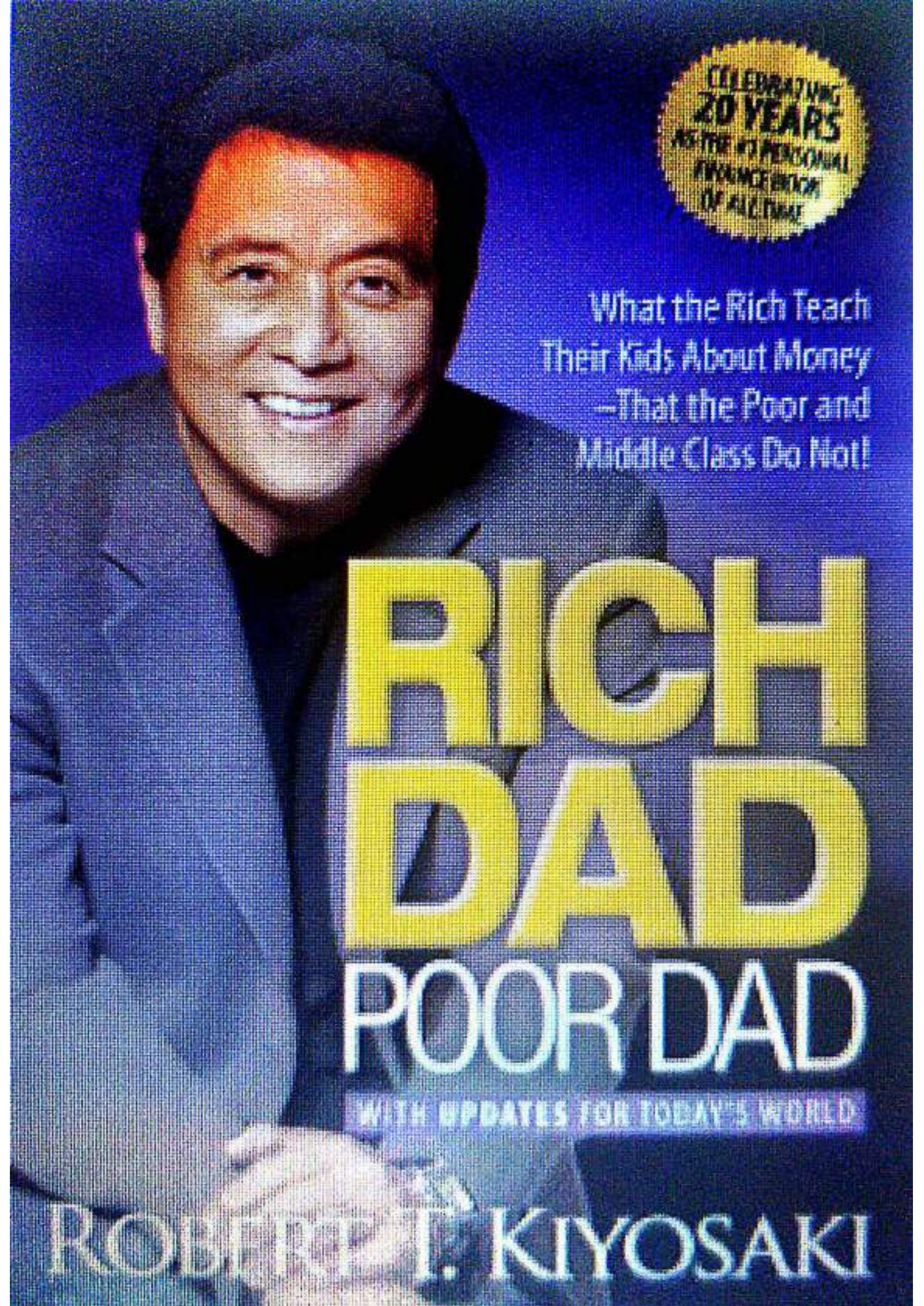
Allen Lane

JORDAN B. PETERSON

BEYOND ORDER

12 MORE RULES FOR LIFE

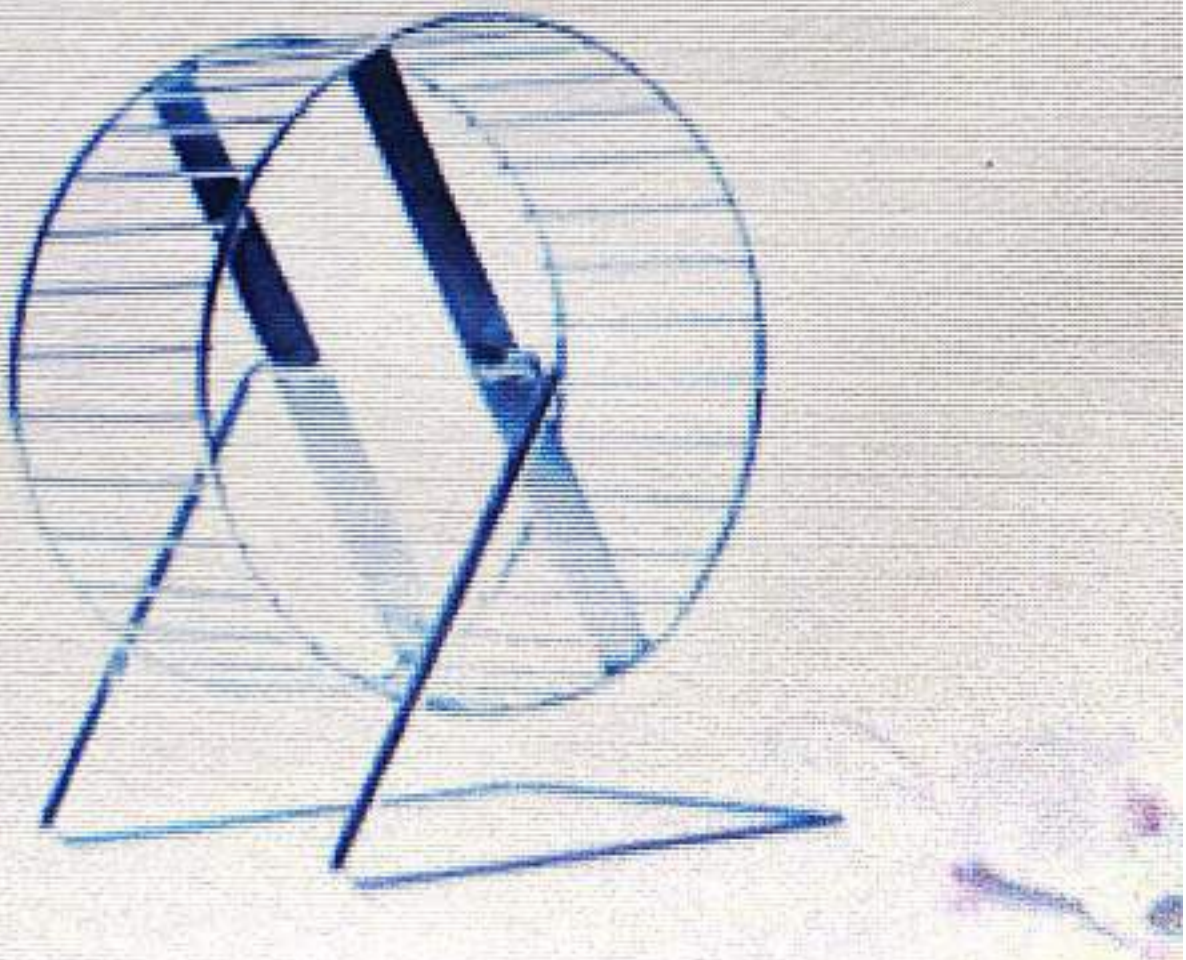
"The most influential public intellectual in the Western world right now"
THE NEW YORK TIMES


CELEBRATING
20 YEARS
AS THE #1 PERSONAL
FINANCE BOOK
OF ALL TIME
What the Rich Teach
Their Kids About Money
–That the Poor and
Middle Class Do Not!
RICH
DAD
POOR DAD
WITH UPDATES FOR TODAY'S WORLD
ROBERT T. KIYOSAKI

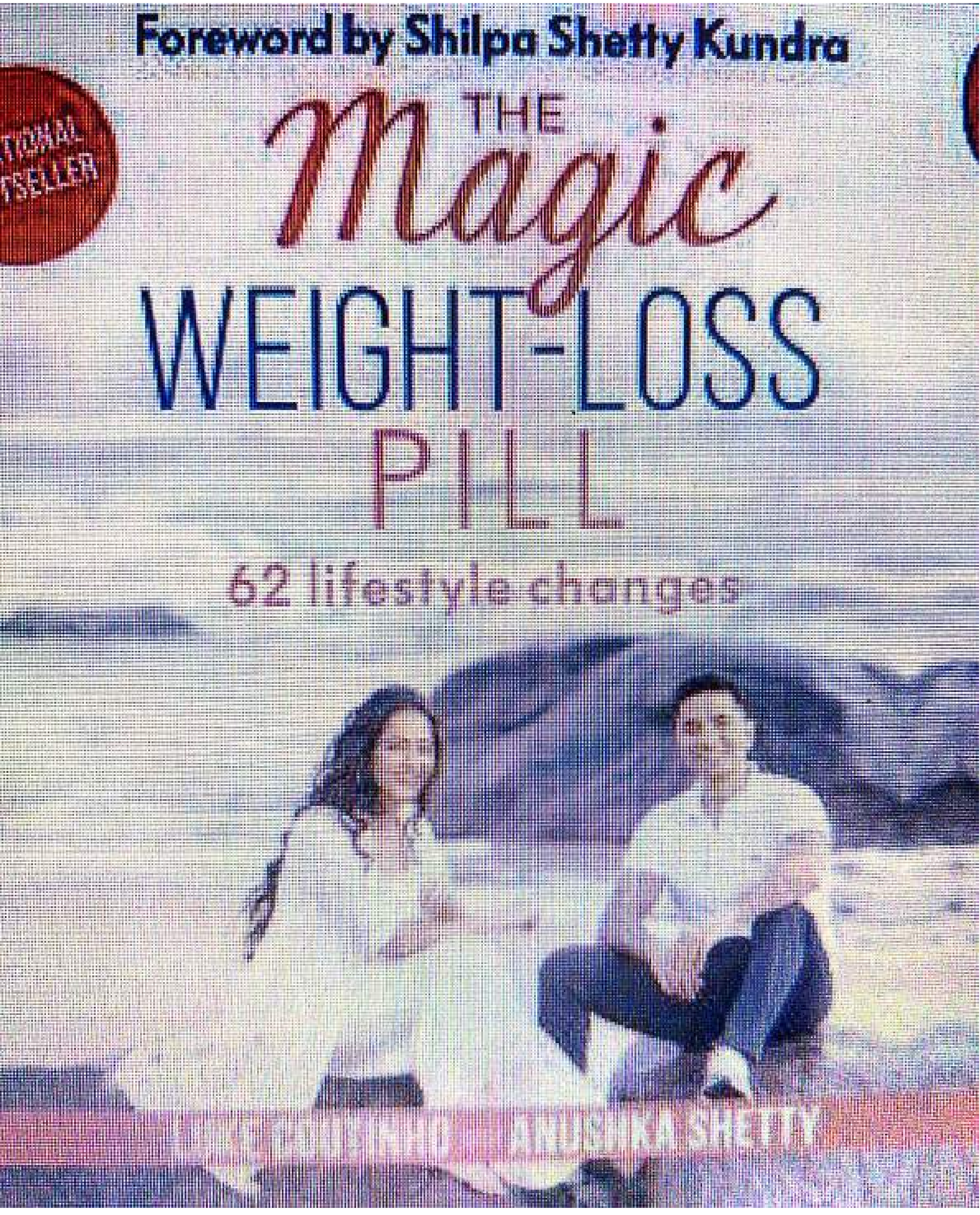
'Absolutely fascinating.' **Wired**

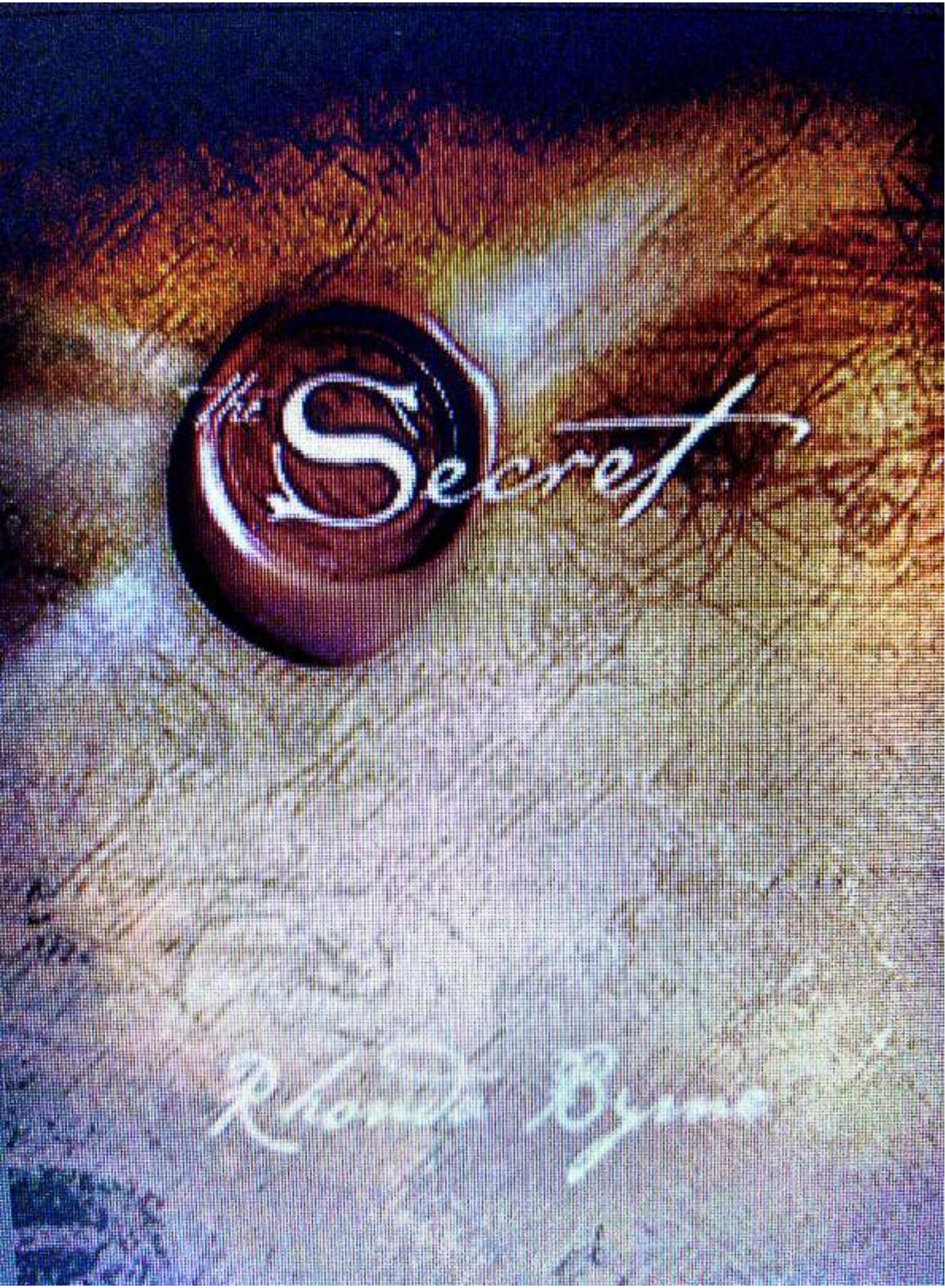
THE POWER OF HABIT

Why we do what we do and how to *change*



CHARLES DUHIGG


Foreword by Shilpa Shetty Kundra
THE
Magic
WEIGHT-LOSS
PILL
62 lifestyle changes
ANUSHKA SHETTY


The Secret
Rhonda Byrne

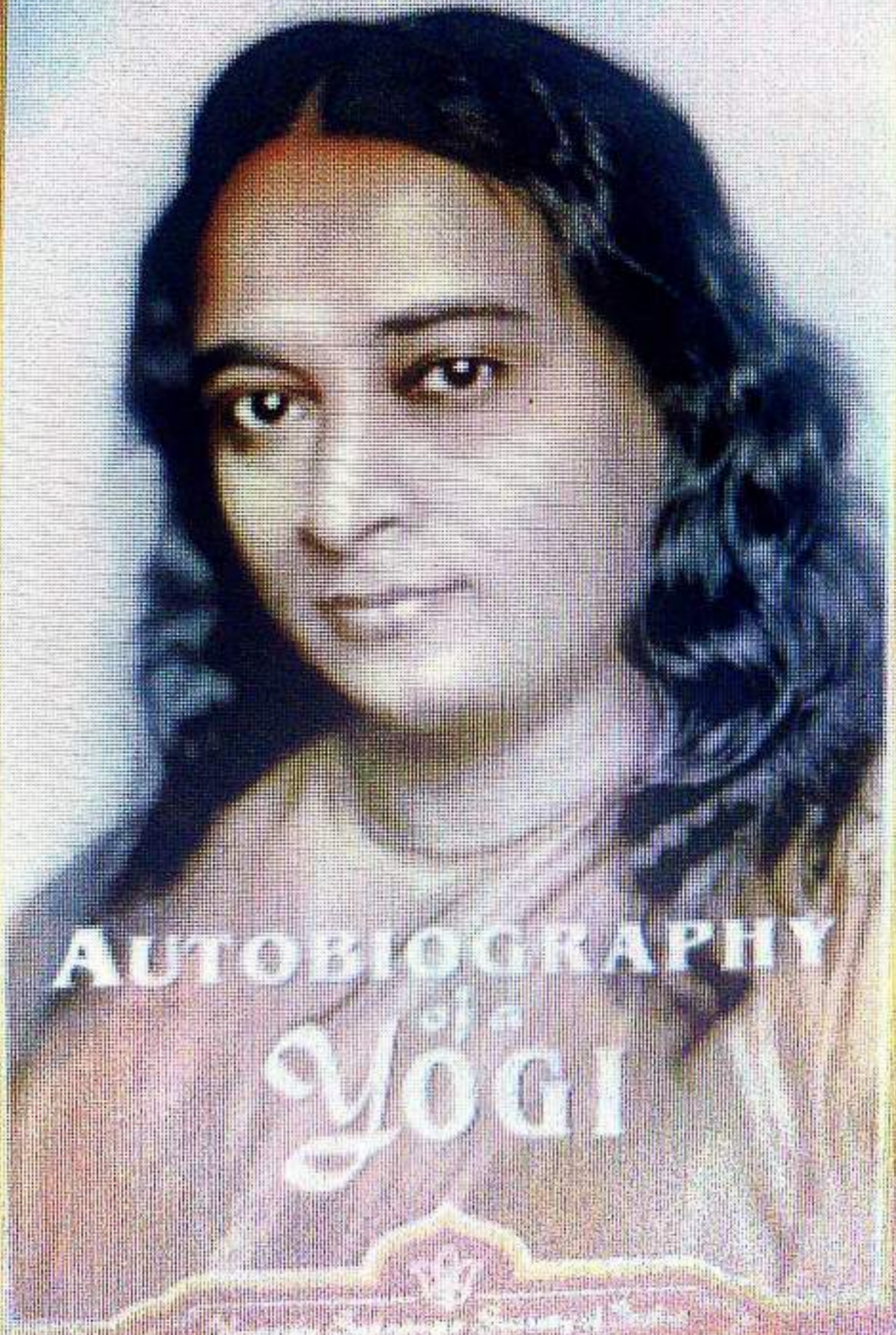
INTERNATIONAL BESTSELLER

EAT THAT FROG!

NEW REVISED EDITION

GET MORE OF THE IMPORTANT THINGS DONE TODAY

Paramahansa Yogananda



AUTOBIOGRAPHY of a YOGI

FROM THE AUTHOR OF THE 48 LAWS OF POWER

THE LAWS OF HUMAN NATURE

ROBERT GREENE

Everyday ayurveda

Daily Habits That Can Change Your Life in a Day

BHASWATI BHATTACHARYA

NEW YORK TIMES BESTSELLER

"Sapiens tackles the biggest questions of history and of the modern world, and it is written in unforgettably vivid language."
—JARED DIAMOND, Pulitzer Prize–winning author of *Guns, Germs, and Steel*

Yuval Noah Harari

Sapiens

A Brief History of Humankind

THE NUMBER ONE BESTSELLER

Yuval Noah Harari

Homo Deus

A Brief History of Tomorrow

"A MIND-BENDING BOOK" GUARDIAN