

# Cloud and Big Data

## Big Data Overview

Gaurav Parashar

January 6, 2020

## 1 Objectives

## 2 Introduction

- What does "Big Data" Mean?
- Applications of Big Data
- Generation of Big Data

## 3 6 Vs of Big Data

- Limitations of Existing Systems

## 4 Application of Big Data in Industry Verticals

- Telecom
  - Call Detail Record

## 5 Hadoop HDFS and its features

- Hadoop Design
- Interacting with HDFS

## 6 Interacting with HDFS

- List Directory Contents
- Creating a Directory
- Copy Data onto HDFS
- Display Content of a file from HDFS

## 7 MapReduce

## 8 Python

## 9 References

# Objectives

- Understand Big Data?

# Objectives

- Understand Big Data?
- Application areas of Big Data

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features
- How to configure Virtual Machine

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features
- How to configure Virtual Machine
- Perform read and write in Hadoop

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features
- How to configure Virtual Machine
- Perform read and write in Hadoop
- Understand what is Cloud?

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features
- How to configure Virtual Machine
- Perform read and write in Hadoop
- Understand what is Cloud?
- How to configure Cloud Environment?

# Objectives

- Understand Big Data?
- Application areas of Big Data
- Analyse limitations of existing systems
- Big Data Analytics in Industry Verticals
- Understand Hadoop and its features
- How to configure Virtual Machine
- Perform read and write in Hadoop
- Understand what is Cloud?
- How to configure Cloud Environment?
- How to perform operations on Big Data in Cloud Environment

# What does "Big Data" Mean?

# What does "Big Data" Mean?

- Collecting large amounts of data

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

- Early Fraud Detection, Credit Card Frauds, Tick [1] Analytics

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

- Early Fraud Detection, Credit Card Frauds, Tick [1] Analytics
- Content personalisation, Recommendation System

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

- Early Fraud Detection, Credit Card Frauds, Tick [1] Analytics
- Content personalisation, Recommendation System
- Insurance: Personalised Pricing

# What does "Big Data" Mean?

- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

- Early Fraud Detection, Credit Card Frauds, Tick [1] Analytics
- Content personalisation, Recommendation System
- Insurance: Personalised Pricing
- Customer Loyalty Data

# What does "Big Data" Mean?

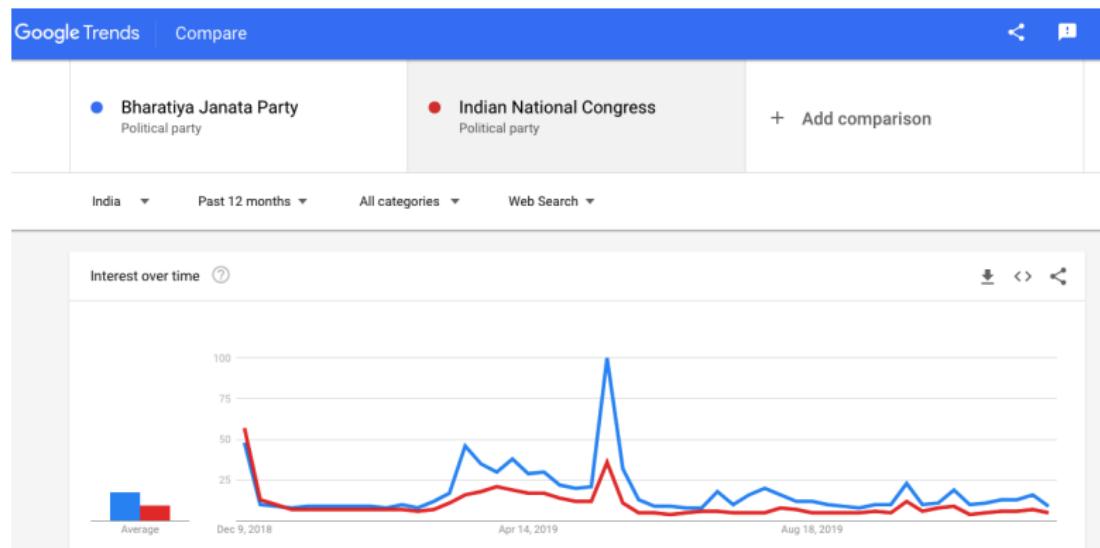
- Collecting large amounts of data
  - Computers, Databases, Video, Voice , tweets, comments, blogs, web pages, logs, calls, messages (Text + Whatsapp + ...) ,...

What do we do with this data?

- Early Fraud Detection, Credit Card Frauds, Tick [1] Analytics
- Content personalisation, Recommendation System
- Insurance: Personalised Pricing
- Customer Loyalty Data
- Predicting Future

# Applications of Big Data

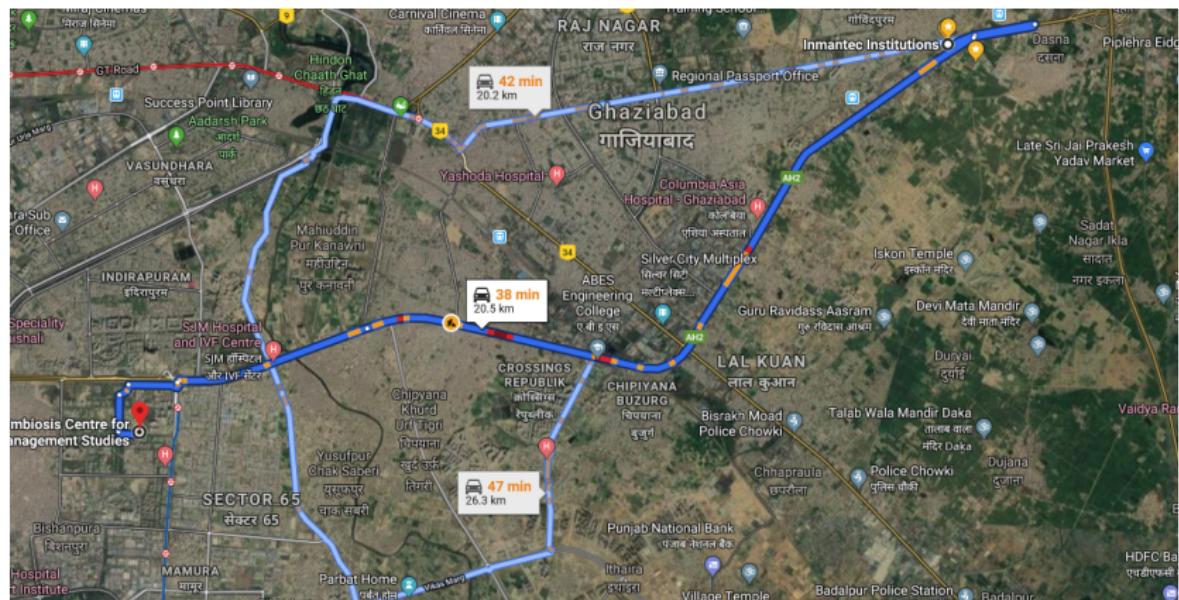
Examples:  
Google Trends



<sup>1</sup>Google Trends for BJP and INC

# Applications of Big Data

## Maps



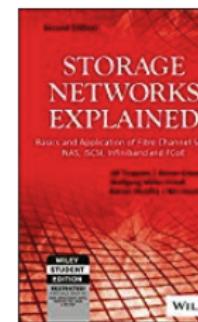
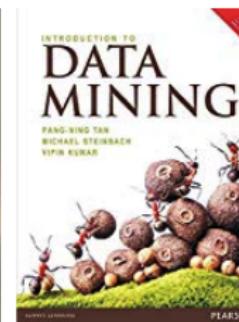
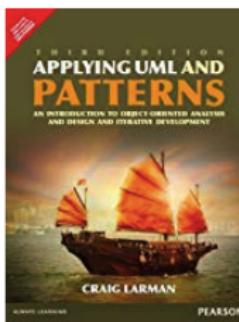
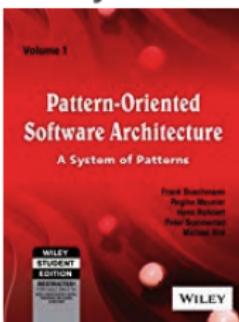
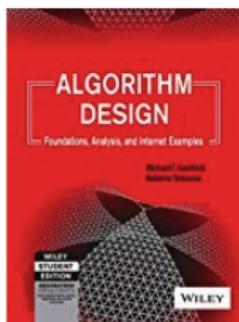
<sup>2</sup>Display path, traffic, terrain, etc. time

# Applications of Big Data

## Recommendation Systems

[Explore all](#)[See more](#)[See more](#)

### Recommendations for you in Books



### <sup>3</sup>Recommendation Engine

# Applications of Big Data

## Twitter Trends

What's happening?

Sumit Kumar @sumitmeetg2007 · Dec 5  
15K Continuous Hill Run followed by drills and strength exercises ✓  
#MarathonTraining #running 💪

Check out my activity on Strava: strava.app.link/EVnBA3x091

STRAVA

Bengaluru Running

Sumit Kumar  
Dec 5 @ 5:16 AM • Running

Trends for you

#ChangeWhatYouCan MG ZS EV - India's First Pure Electric Internet SUV - Coming Soon! Promoted by Morris Garages India

Trending in India #hyderabadpolice 130K Tweets

NDTV @ndtv Opinion divided on killing of accused in Telangana vet's ...

Trending in India #HumanRights 25K Tweets

Trending in India #Encounter 205K Tweets

GHAZIABAD POLICE is Tweeting about this

Trending in India #RIPDisha 6,200 Tweets

4

<sup>4</sup>Trending in Twitter

# Applications of Big Data

## Big Data in Sports [2]

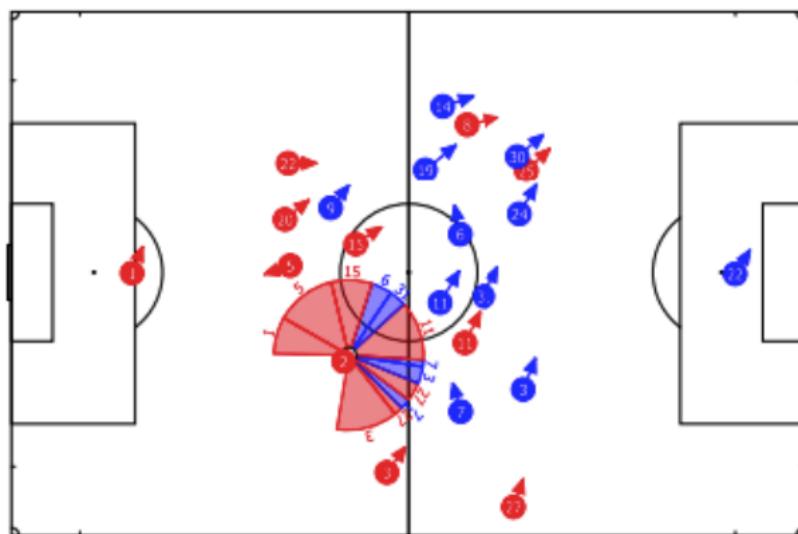


Figure: Available receivers of pass by Red2

# Applications of Big Data

- Weather Prediction
- Medical Diagnosis
- Smart Cities and Buildings

# Generation of Big Data

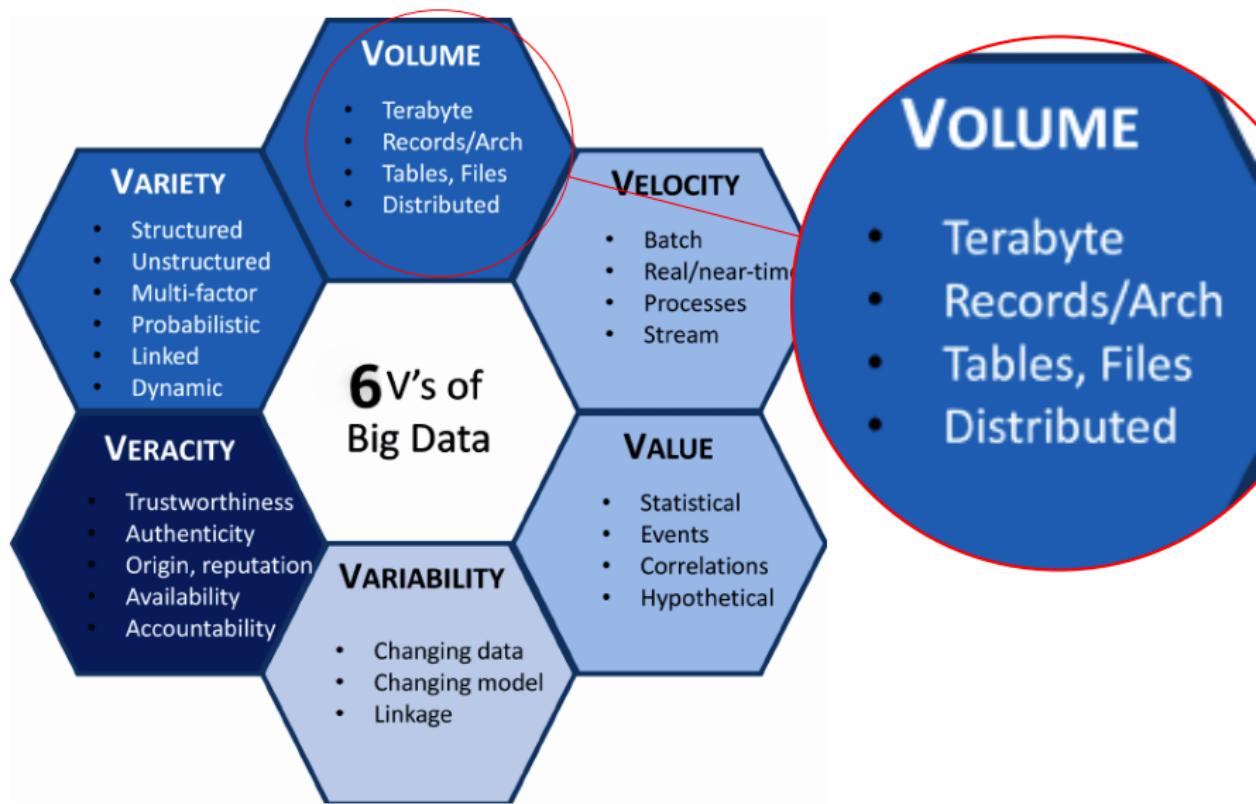
## What Happens in an Internet Minute?



And Future Growth is Staggering



# 6 Vs of Big Data[3]: Volume 20



## Scenario: Volume

What is the starting limit of Big Data?

A > 1 GB - < 1TB

## Scenario: Volume

What is the starting limit of Big Data?

- A > 1 GB - < 1TB
- B > 1 TB - < 1PB

## Scenario: Volume

What is the starting limit of Big Data?

- A > 1 GB - < 1TB
- B > 1 TB - < 1PB
- C > 1 PB - < 1EB

## Scenario: Volume

What is the starting limit of Big Data?

- A > 1 GB - < 1TB
- B > 1 TB - < 1PB
- C > 1 PB - < 1EB
- D > 1 EB - < 1ZB

## Scenario: Volume

What is the starting limit of Big Data?

- A > 1 GB - < 1TB
- B > 1 TB - < 1PB
- C > 1 PB - < 1EB
- D > 1 EB - < 1ZB

Answer:

Depends upon the organisation definition of Big Data. Relative term.

# Scenario: Volume 20

**40 ZETTABYTES**

[ 43 TRILLION GIGABYTES ]

of data will be created by 2020, an increase of 300 times from 2005

2020

2005



**6 BILLION PEOPLE**  
have cell phones



WORLD POPULATION: 7 BILLION

# Volume SCALE OF DATA

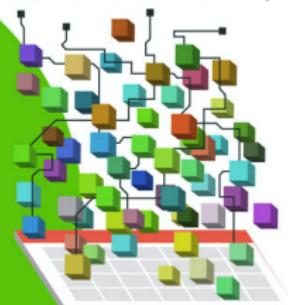


It's estimated that

**2.5 QUINTILLION BYTES**

[ 2.3 TRILLION GIGABYTES ]

of data are created each day



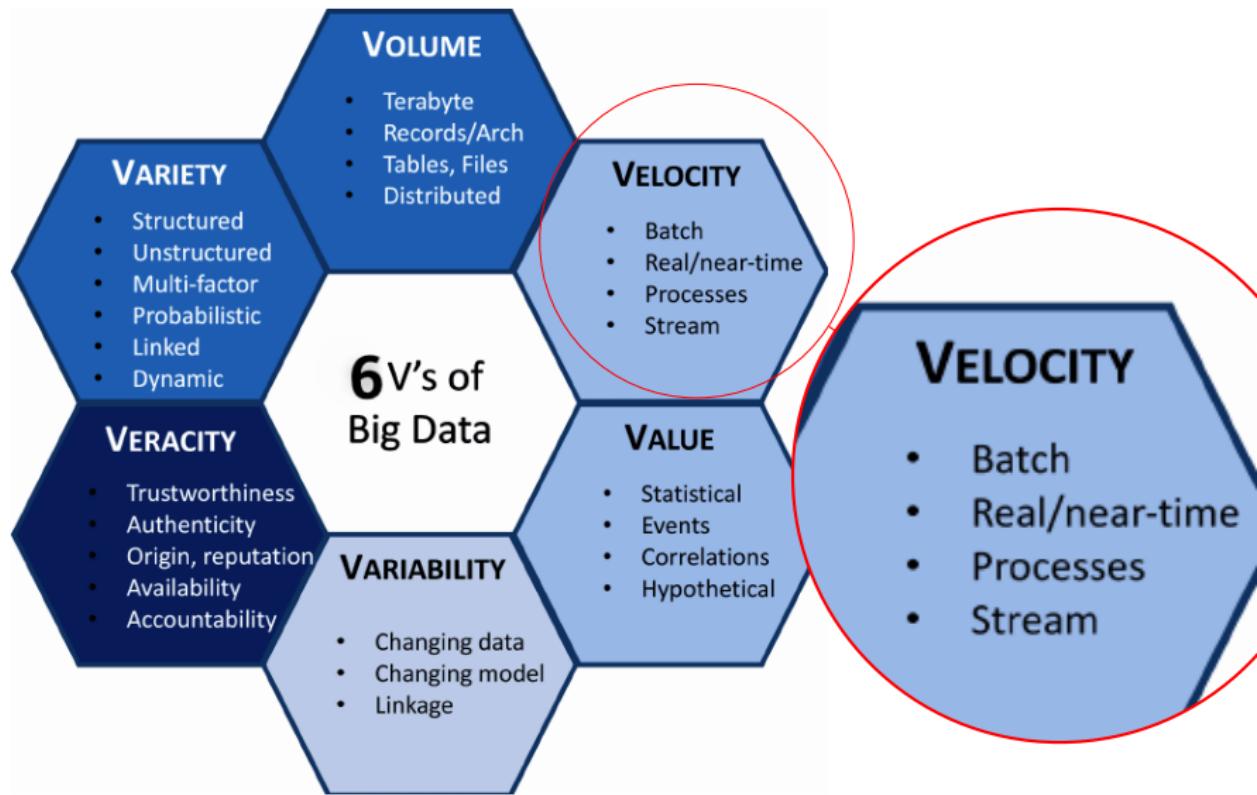
Most companies in the U.S. have at least

**100 TERABYTES**

[ 100,000 GIGABYTES ]

of data stored

# 6 Vs of Big Data: Velocity 20



# 6 Vs of Big Data: Velocity

The New York Stock Exchange captures

**1 TB OF TRADE INFORMATION**

during each trading session



By 2016, it is projected there will be

**18.9 BILLION NETWORK CONNECTIONS**

– almost 2.5 connections per person on earth

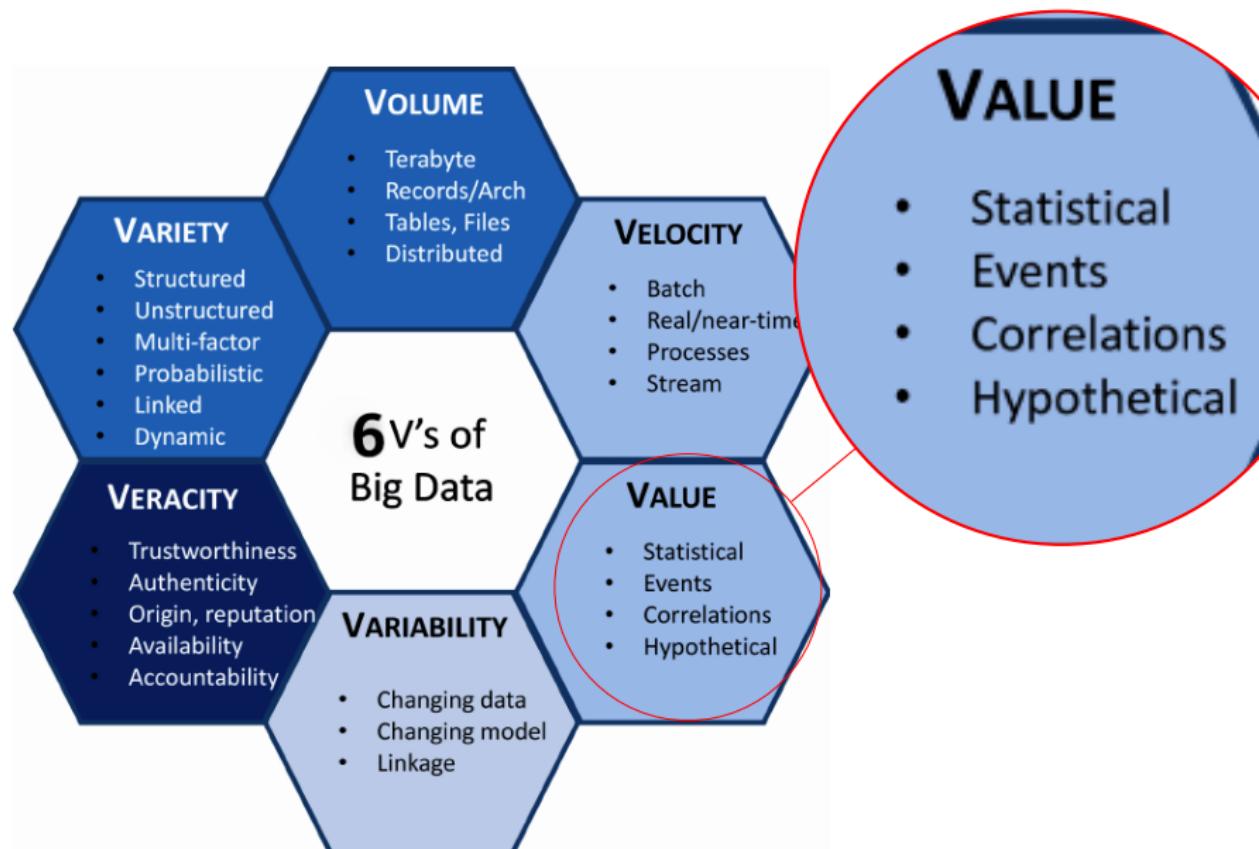


Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

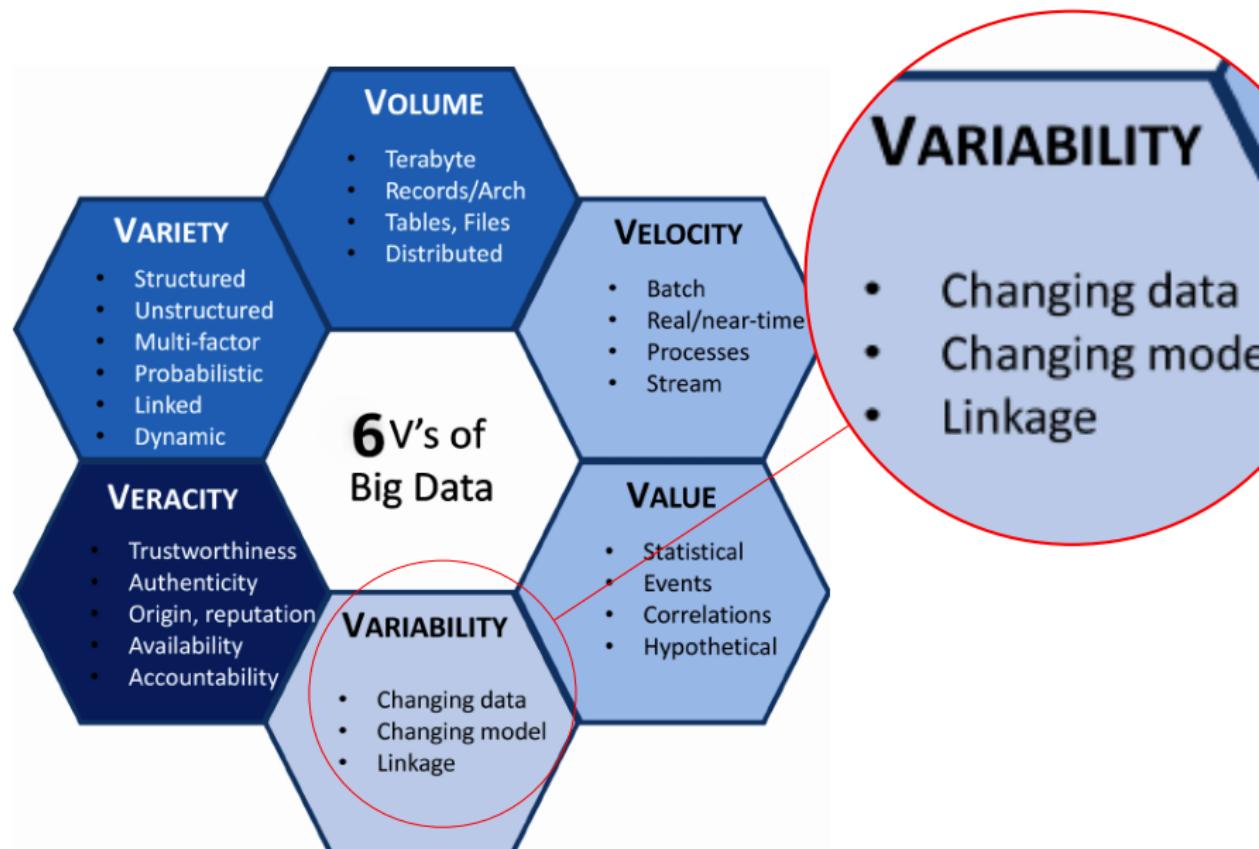
**Velocity**  
ANALYSIS OF STREAMING DATA



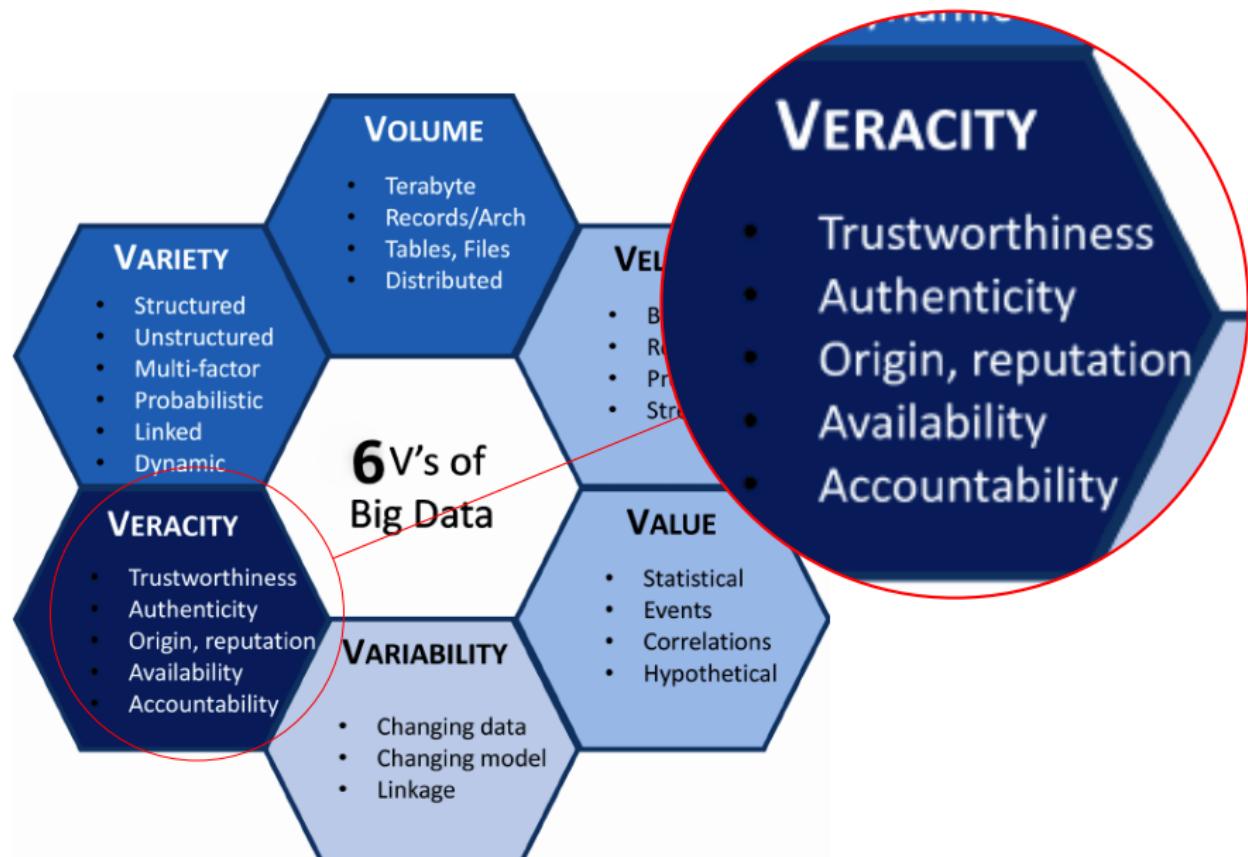
# 6 Vs of Big Data: Value 20



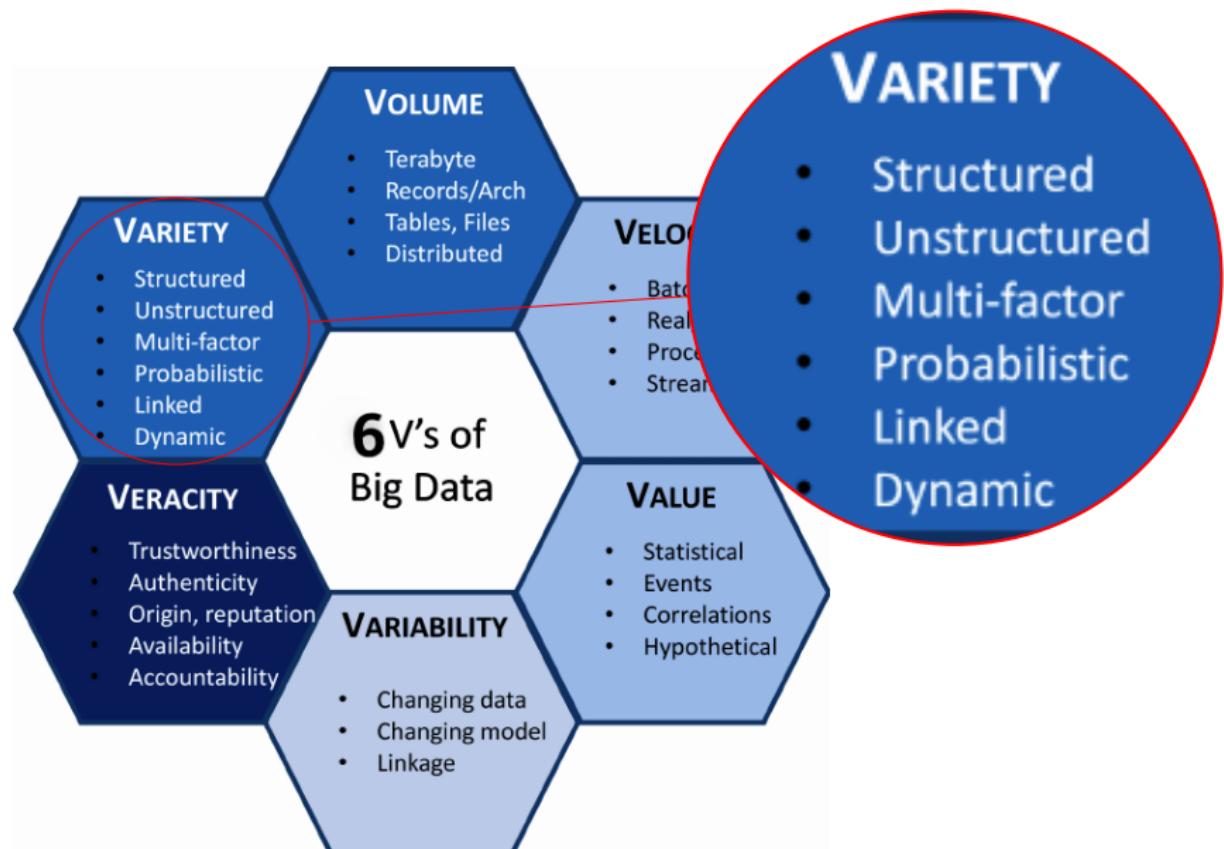
# 6 Vs of Big Data: Variability 20



# 6 Vs of Big Data: Veracity 20



# 6 Vs of Big Data: Variety 20



# Limitations of Existing Systems

- ① Cost of scaling

# Limitations of Existing Systems

- ① Cost of scaling
- ② Vertical Scaling

# Limitations of Existing Systems

- ① Cost of scaling
- ② Vertical Scaling
- ③ Integration with legacy systems

# Limitations of Existing Systems

- ① Cost of scaling
- ② Vertical Scaling
- ③ Integration with legacy systems
- ④ Rapid Change in type of data

# Limitations of Existing Systems

- ① Cost of scaling
- ② Vertical Scaling
- ③ Integration with legacy systems
- ④ Rapid Change in type of data
- ⑤ Lack of skills

# Home Work Question 1: Telecom

A telco[4] serving 8 million prepaid mobile subscribers

- ① Volume 10: \_\_\_\_\_
- ② Velocity 13: \_\_\_\_\_
- ③ Value 15: \_\_\_\_\_
- ④ Variability 16: \_\_\_\_\_
- ⑤ Veracity 17: \_\_\_\_\_
- ⑥ Variety 18: \_\_\_\_\_

Link: <https://github.com/gauravparashar/symbiosis>

---

UID, DATE (yyyy-MM-dd), TRIP\_SEQUENCE\_ID, MOBILITY\_TYPE, TRANSPORT\_MODE,  
TOTAL\_DISTANCE, TOTAL\_TIME, START\_TIME, END\_TIME, TOTAL\_POINTS, POINT\_LIST  
422a837717,2015-06-06,1,STAY,STAY,0.000,57749.000,00:00:00,16:02:29,1,1|2015-06-01  
00:00:00|6.373743|-10.772951

422a837717,2015-06-06,2,MOVE,WALK,3153.708,2323.000,16:02:29,16:41:12,39,1|2015-  
06-01 16:02:29|6.373497|-10.773267;2|2015-06-01 16:03:30|6.374243|-10.773447;3|2015-06-  
01 16:04:31|6.374983|-10.773652;4|2015-06-01 16:05:32|6.375711|-10.773898;5|2015-06-01  
16:06:33|6.376103|-10.774265;6|2015-06-01 16:07:34|6.375691|-10.774913;7|2015-06-01  
16:08:35|6.375280|-10.775561;8|2015-06-01 16:09:36|6.374868|-10.776209;9|2015-06-01  
16:10:38|6.374457|-10.776858;10|2015-06-01 16:11:39|6.374046|-10.777506;11|2015-06-01  
16:12:40|6.373634|-10.778154;12|2015-06-01 16:13:41|6.373223|-10.778802;13|2015-06-01  
16:14:42|6.372842|-10.779469;14|2015-06-01 16:15:43|6.372488|-10.780150;15|2015-06-01  
16:16:44|6.372098|-10.780811;16|2015-06-01 16:17:45|6.371690|-10.781461;17|2015-06-01  
16:18:47|6.371286|-10.782115;18|2015-06-01 16:19:48|6.370883|-10.782768;19|2015-06-01  
16:20:49|6.370479|-10.783421;20|2015-06-01 16:21:50|6.370076|-10.784074;21|2015-06-01  
16:22:51|6.369672|-10.784727;22|2015-06-01 16:23:52|6.369238|-10.785360;23|2015-06-01  
16:24:53|6.368795|-10.785988;24|2015-06-01 16:25:55|6.368358|-10.786618;25|2015-06-01  
16:26:56|6.368053|-10.787320;26|2015-06-01 16:27:57|6.368022|-10.787936;27|2015-06-01  
16:28:58|6.368780|-10.788057;28|2015-06-01 16:29:59|6.369542|-10.788139;29|2015-06-01  
16:31:00|6.370310|-10.788158;30|2015-06-01 16:32:01|6.371075|-10.788096;31|2015-06-01  
16:33:02|6.371611|-10.788376;32|2015-06-01 16:34:04|6.371883|-10.789094;33|2015-06-01  
16:35:05|6.372137|-10.789817;34|2015-06-01 16:36:06|6.372195|-10.790581;35|2015-06-01  
16:37:07|6.372183|-10.791348;36|2015-06-01 16:38:08|6.372054|-10.791999;37|2015-06-01  
16:39:09|6.371293|-10.791979;38|2015-06-01 16:40:10|6.371313|-10.792746;39|2015-06-01  
16:41:12|6.371295|-10.793513

---

422a837717,2015-06-06,3,STAY,STAY,0.000,34295.000,16:41:12,23:59:59,1,1|2015-06-06  
16:41:12|6.371295|-10.793513

Hadoop is a distributed system which is based on HDFS (Hadoop Distributed File System). It is scalable, distributed, and portable file system for large commodity systems.

- stores large amount of data
- reliable way to store data
- scalable way to manage resources
- HDFS is composed of two main components
  - Name Node
  - Data Node

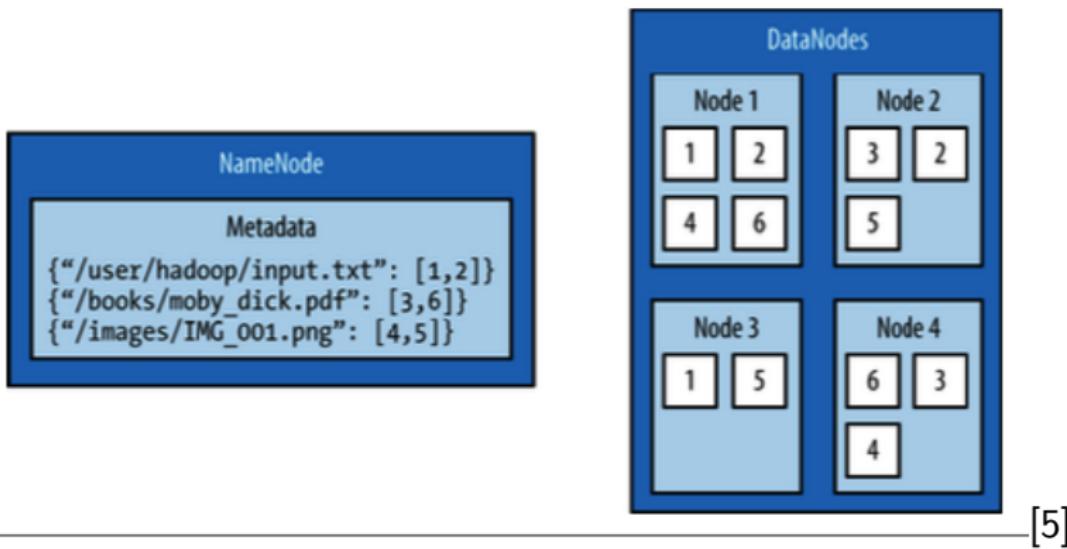


Figure: HDFS Architecture <sup>6</sup>

<sup>6</sup>HDFS cluster with replication factor of 2

Interaction with HDFS is primarily performed from the command line using command **hdfs**.

```
$ hdfs COMMAND [-option <arg>]
```

The command argument instructs which functionality of HDFS to be used.

# Common File Operations

To perform basic operations on HDFS, we use **dfs** command with **hdfs**. **dfs** supports many file operations like copy, move, remove , etc.

List directory contents

```
$ hdfs dfs -ls /
```

It lists out the content of the root filesystem of HDFS.

# List Directory Contents

Output:

```
$ hdfs dfs -ls /
Found 3 items
drwxr-xr-x - hadoop supergroup 0 2019-12-28 23:20 /
    ↪ input
drwxr-xr-x - hadoop supergroup 0 2019-12-29 00:18 /
    ↪ output
drwx----- - hadoop supergroup 0 2019-12-18 22:33 /
    ↪ tmp
```

To create a directory within HDFS.

```
$ hdfs dfs -mkdir /user
```

Output:

```
$ hdfs dfs -mkdir /user
Found 4 items
drwxr-xr-x - hadoop supergroup 0 2019-12-28 23:20 /
    ↪ input
drwxr-xr-x - hadoop supergroup 0 2019-12-29 00:18 /
    ↪ output
drwx----- - hadoop supergroup 0 2019-12-18 22:33 /
    ↪ tmp
drwxr-xr-x - hadoop supergroup 0 2020-01-03 22:22 /
    ↪ user
```

Copy data file(s) onto HDFS.

```
$ hdfs dfs -put source destination
```

Output:

```
$ hdfs dfs -put hw.csv /user
```

-get command can be used to retrieve data from HDFS to local file system

To visualise the content in the file copied onto HDFS

```
$ hdfs dfs -cat filename
```

Output:

```
$ hdfs dfs -cat /user/hw.csv
1,65.78331,112.9925
2,71.51521,136.4873
3,69.39874,153.0269
4,68.2166,142.3354
....
24999,67.52918,132.2682
25000,68.87761,124.8742
```

-head command can be used to display 50 lines from top.

-tail command can be used to display last 50 lines from bottom.

MapReduce is a programming model that enables large volumes of data to be processed and generated by dividing work into independent tasks and executing the tasks in parallel across a cluster of machines.

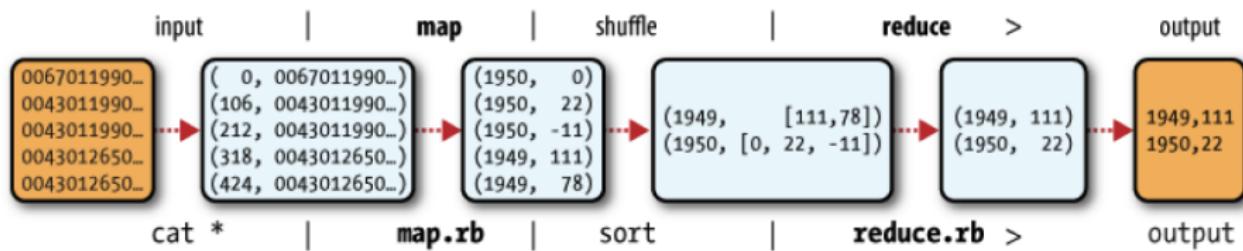


Figure: MapReduce logical data flow[6]

## Word Count Example - Prerequisite

- sys module : A file containing a set of functions you want to include in your application.

```
import sys
```

- for loop : A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

```
students = ["Aman", "Anita", "Cherry"]  
for student in students:  
    print(student)
```

- lists : List is a collection which is ordered and changeable.  
Allows duplicate members

```
students = ["Aman", "Anita", "Cherry"]
```

## Word Count Example - Prerequisite

- print : The print() function prints the specified message to the screen, or other standard output device.

```
print "%s\t%s" %("Hello", "how are you?")
print '{0}\t{1}'.format("Hello", "how are
    ↪ you?")
```

- split : split(separator) function returns a list of strings after breaking the given string by the specified separator.

```
line = "Mary had a little lamb"
print (line.split(" "))
```

# Word Count Example - Prerequisite

- **dictionary** : Dictionary is an unordered collection of data values, used to store data values like a map, which unlike other Data Types that hold only single value as an element, Dictionary holds key:value pair.

```
# Creating an empty Dictionary
```

```
Dict = {}  
print("Empty Dictionary: ")  
print(Dict)
```

```
# Creating a Dictionary with Integer Keys
```

```
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}  
print("\nDictionary with the use of Integer Keys:  
      ")  
print(Dict)
```

# Python Hadoop: Final Mapper code

```
for line in sys.stdin:  
    line = line.strip() # Remove the leading and  
    ↪ trailing spaces  
    words = line.split() # Split the line on space  
    for word in words:  
        print "%s\t%s" %(word,1) # you can use this  
        ↪ method or  
        print '{0}\t{1}'.format(word, 1) #this method
```

# Python Hadoop: Final Mapper output

```
$ echo Hello world. I am an Indian. and I love my  
↪ country | python mapper.py
```

Output:

```
Hello 1  
world. 1  
I 1  
am 1  
an 1  
Indian. 1  
and 1  
I 1  
love 1  
my 1  
country 1
```

# Python Hadoop: Final Reducer code

```
import sys
w = {}
# Process each key-value pair from the mapper
for line in sys.stdin:
    # Get the key and value from the current line
    word, count = line.split('\t')
    # Convert the count to an int
    count = int(count)
    if word in w:
        w[word] = w[word] + count
    else:
        w[word] = count

for word in w.keys():
    print "%s\t%s" %(word,w[word])
```

# Python Hadoop: Final Reducer output

```
$ echo Hello world. I am an Indian. and I love my  
→ country | python mapper.py | python reducer.py
```

Output:

```
and 1  
love 1  
I 2  
my 1  
am 1  
an 1  
Indian. 1  
country 1  
world. 1  
Hello 1
```

-  [Investopedia, "Financial advisory website."](#)  
Accessed on 2019-12-06.
-  [J. Gudmundsson and M. Horton, "Spatio-temporal analysis of team sports - A survey," \*CoRR\*, vol. abs/1602.06994, 2016.](#)
-  [IBM, "Ibm big data platform - bringing big data to the enterprise."](#)  
Accessed on 2019-12-06.
-  [T. T. B. Services, "Big data and the telecom industry."](#)  
Accessed on 2019-12-09.
-  [D. M. Zachary Radtka, \*Hadoop with Python\*.](#)  
O' Reilly Media, Inc., 2016.
-  [T. White, \*Hadoop: The Definitive Guide\*.](#)  
O' Reilly Media, Inc., 2012.