Big data:

Velocity, Variety, Volume

Big data = too big to store in 1 machine!

1 machine can store = 2 numbers

1,2,3 = big data

Small data: SQL server, mysql, postgreSQL

M1 = { 1, 2}   M2 = {3}
M3 = {1,2}   M4= {3}

Big data-> sharding

M1 = {1,2}   M2={2,3}   M3={3,1}

Apache Spark/Hadoop/Beam/hive/hbase -> 3

Find the sum of all numbers

Small data; M1 = { 1, 2}   M2 = {3}
M1 = 3, M2= 3
M1+M2 = 3+3 = 6

Big data: M1 = {1,2}   M2={2,3}   M3={3,1}

M1=3, M2=5, M3=5
1+2+3 = 13

YOU CANNOT RUN SMALL DATA PROGRAMS
On BIG DATA machines directly!

Imperative v/s Declarative languages

Imperative-> Java, c#, c++, python
-> how to do
Declarative-> SQL, HTML, CSS
-> what to do
-> diff flavours of SQL, HTML/css renders
Separately for diff browser types!

imperative
declarative
a=1   a=?   1

True
b=2 b=? 2
True
c=a+b c=? 3 True
print(c) ?? “3” Execute()->”
3”
Eager
LAZY
Stack, Heap Directed
Acyclic Graph

LINQ, Spark, TensorFlow, Beam, DialogFlow,
Airflow, SQL…..

Big data-> Transformations, Actions

Actions-> execute the graph
Transformation-> add and optimise nodes in graph!

Big data pipelines-> wait for execution to see the results!!!

MapReduce -> problems are mapped to individual machines; machine solve the problem; but the result is AGGREGATED by a master (driver) machine
(Map-Combine is the same thing)

Apache Spark-> in memory analytics on big data
-> sharding, mapreduce…..
-> Databricks, HDInsight, spark on vm

Variety of Data::
SQL, NoSQL, images, videos, binary, text, csv/json/xml….

DUMPING YARD-> we do not care about the type of data -> data lake

Before cloud-> Apache Hadoop

After cloud-> Hadoop-compatible storages
Azure-> Azure Storage; AWS-> S3; Google Cloud Storage; Sharepoint; dropbox; OneDrive; google drive

Velocity::
Data at rest
Data in motion

2 separate architectures for the batch, stream (speed == stream)

Lambda -> 2 separate architectures for batch/stream
Kappa -> unified architecture for batch/stream
-> Delta architecture

Bias->hampering the results because of preconceived notions or discrimination
-> can exist in data collection
-> data labelling
-> feature selection
-> model development
-> model testing
-> usage
Bias-> not always bad
0 -> we are in trouble!
1 -> racism!
Bias is a very tiny number added to our ML to get familiar prediction

Fairness != Bias

Bias -> class imbalance or insufficiency
Fairness detection tools -> ways to counter bias

Fairness tools->
Data, model -> WhatIf analysis tools
example: https://pair-code.github.io/what-if-tool/
http://aequitas.dssg.io/upload.html

Model Explanations:

1. **LOCAL**
    1. **For each column-> what are acceptable values, outliers, min/max, std**
2. **GLOBAL**
    1. **Most important and least important columns**
    2. **100 columns:**
        1. **Column importance = 1**
        2. **Each columns' importance = 1/x**
        3. **Loan Approval/Rejection-> CIBIL, previous loan paid, breakfast, breed of cat**
            1. **Correlation**
            2. **Pair plots of distributions**
            3. **Feature selection (permutation, PCA....)**