* Data migration to the cloud

⇒ solu around the digital workspace information management archiving & migration space

* Spark Core & SQL

⇒ Hadoop, MR, Ecosystem component

← Spark

↳ philosophy, features, components.

Core programming → python → pyspark

↳ (Major focus)

* Use Cases :-

* Hadoop ⇒ Open Source framework.

↳ processing big data.

↳ Additional layer on top of any existing system.

* Core Components

⇒ HDFS → Storage

⇒ MapReduce → process data.

↳ processing framework

⇒ Dist App' info of Cluster.

*] Components of Hadoop Ecosystem

Hive	→ Run SQL like queries on data in HDFS → Struct data
Pig	→ Scripts for Data cleansing (Data pre-processed)
Sqoop	→ Database Migrator
Mahout	⇒ Machine Learning Lib

*] Spark

⇒ * Hadoop (2006) ⇒ HDFS + MapReduce

(2006-2018) ⇒ Hadoop → Best Batch processing in the market.

* Limitations of Hadoop MapReduce

HDFS → Data Storage.

① High Latency → Batch processing Slowing down (WIP) (Not developed)

② Architecture → Different operating → Different tools.

③ Processing → Disk based ⇒ Involves DISK I/O ops

* MR → Disk based Processing
  → High Latency.

* Spark ⇒ General purpose, Lightning fast cluster computing framework.
  → Open Source, Wide range Data processing Engine.

* Spark ⊇ Batch processing & Streaming processing

* Types of Processing
  → Batch
  → Real time (Streaming)
  → Interactive
  → Iterative.

* General purpose ⇒

* Features ⇒ → Single library for SQL
  → Graph Analytics, Streaming & ML Algo.

* Spark Competent to Hadoop
  → Memory based processing
  → Open Source computing framework

Spark ⇒ Core API
↳ Programming in Spark.

Spark ⇒ Speed
100× faster than traditional frameworks (Hadoop MR)

⇒ Poweful Caching mechanism

[*] Deployment

⇒ Mesos, Hadoop, via YARN,
Spark's own Cluster Manager.

*Real Time ⇒ Low Latency by using in-memory processing (computations)

[*] Ployglot ⇒ Scala, Java, Python, R prog.
(Spark)

Python + Spark ⇒ Pyspark

* Why Spark (2010) = Fame → 2014

* Components of Spark ⇒
① Spark Core
② Spark SQL
③ Spark Streaming
④ Spark ML Lib
⑤ Spark Graphx
⑥ Spark R

+ R programming → Data visualization.

* Spark Core & Spark SQL

* Module 1 ⇒ ① Spark Architecture
② Spark Core API → RDD Prog.
③ Operate on RDDs → Transf / Actions
④ RDD Persistence
⑤ Advanced Spark Concepts → Shared variables

* Module 2 + ① Pyspark SQL – 20th Aug
②

* Spark Architecture
Hadoop → Dist. Env. → Multiple nod
↳ Clusters
→ 3 Components
① Data Storage
② API
③ Resource Management → Spark can be deployed as Stand alone serve
↳ Dist. Computed framework.
Cluster Mgr → service → Master node.

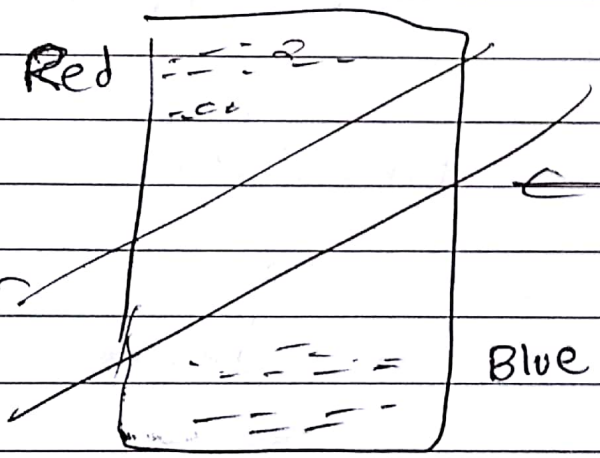
* Spark Context ⇒ Entry pt of Spark functionalitie

* App^n ⇒ Driver program → (SparkContext object)
Spark → Standalone, YARN, Mesos. ↳ Cluster Manager

* Spark RDD (Resilient Distributed Dataset) → fault tolerance
→ Fundamental unit of Spark programming
→ Fundamental Data structure.
→ Immutable collection of objects
↳ value of the dataset can be changed.

Dist → Multi Node Existence
Dataset → Represents records of the data.

SVM → predict diabetes, BMI etc.



← Marginal Line

Paration
ire.

'lamda fun' → 1 line fun
→ Executed in 1 line
→ (lambda x: "ERROR" if x)

* Bagging Technique

* RDD → Data in Stable storage, Other RDD
→ Parallelizing already.

In memory proc → 100x times faster
Disk based proc → 10x times faster.

* Softmax

Pytorch

map() ⇒ Iterates over Each line in RDD, split into new RDD

Filter() ⇒

Agile — set of values & principles → Standup meeting

Belief, decisions →

Virtual machine & Ubuntu

Desktop

Customer collaboration

Contract negotiation

2 of 4.3.1

⇒ Anaconda download, Left side of the list item more

└ Spyder (phyton 2.7)

Kivy

Tableau
Ubuntu
Pytorch & Kivy
Anaconda
Virtual machine

RDD ⇒ → In memory Computation Capability

RDD1 ⇒ Operation (Funct) → RDD2

MapReduce ⇒ funⁿ, Appⁿ, Job

* Dataset 1 → MapReduce Job — O/P adable

per year 2 opl

⇒ Collaboration & productivity → Automating
1. Infra
2. Workflow
3. Metrics
4. Testing

ind & interaction

process & tools

work S/W, Comp Docⁿ

Machine Learning

→ News Categorization - Google
Recommendations - Amazon

→ Deep Learning Algo, Computer Deep Fusion
Clustering

ML
- Computer to work without being explicitly programmed.

* Prediction Accuracy

Mean abs % Error

100 - mean abs % error = prediction

GLM → Generalized Linear model
↳ Family of Algorithms.

Bernoulli

→ Versicolor

SVM → Setosa Non-setosa
→ separating line → Greater segregation
Greater gap → Higher prediction activity