Introduction to Bigdata/AI Development with Cloud Services

Libraries installation and environment setup

Dataset operation (load/save)

Model preparation and training

- Saving/loading

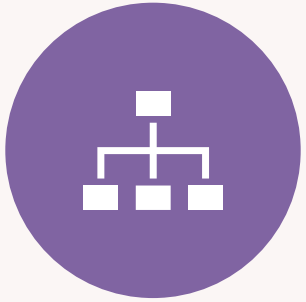
Data /results visualization

Dealing with large data sets

AI Development with Cloud Services

PROJECT 1:
REGRESSING ANALYSIS

PROJECT 2:
CLUSTERING



PROJECT 3:
CLASSIFICATIONS

- Regression analysis example
- https://colab.research.google.com/github/towardsai/tutorials/blob/master/machine_learning_algorithms_for_beginners/machine_learning_algorithms_for_beginners.ipynb#scrollTo=vPU_9qhzD2Z4
- Clustering example
- https://colab.research.google.com/github/csmastersUH/data_analysis_with_python_spring_2020/blob/master/clustering.ipynb#scrollTo=xI4cb-jzbyha
- Classifications
- https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/classification.ipynb#scrollTo=R32zteKHCaXT

Data Analysis with Cloud Services

Libraries installation and environment setup

Dataset operation (load/save)

AI Model preparation and training

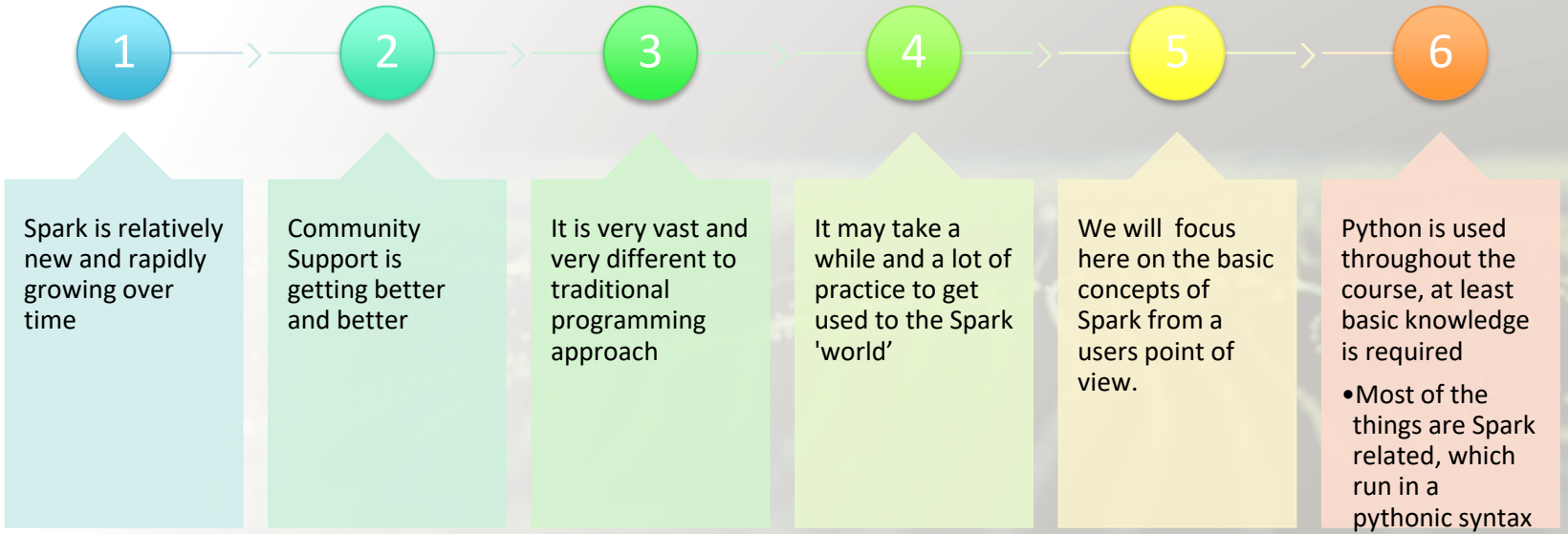
Saving/loading

Data /results visualization

Dealing with large data sets

Data Analytics with Spark

- Note: these slides are a modified version of the slides produced by: Apurva Nandan, Anni Pyysing

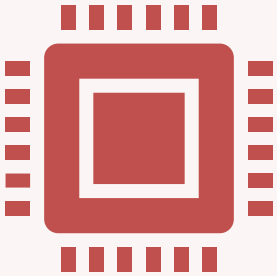

1
Spark is relatively new and rapidly growing over time
2
Community Support is getting better and better
3
It is very vast and very different to traditional programming approach
4
It may take a while and a lot of practice to get used to the Spark 'world’
5
We will focus here on the basic concepts of Spark from a users point of view.
6
Python is used throughout the course, at least basic knowledge is required
•Most of the things are Spark related, which run in a pythonic syntax

Apache Spark is a fast, Open Source, big data based engine for large scale data analysis and distributed processing

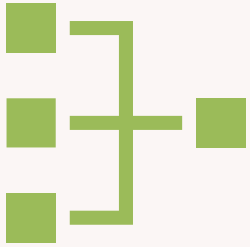
Developed in **Scala** and runs on **Java Virtual Machine**

Can be Used with Scala, Java, Python or R

Works on the Map-Reduce concept
to do the *distributed processing*



Stores the data into memory when needed, for rapid processing



Follows distributed processing concepts for distributing the data across the workers/nodes of the cluster

Apache Spark: Key Concepts

- **Spark Job:** Parallel computation consisting of multiple *tasks*
- **Driver** Program: Used to submit your code as a Spark Job to the cluster
- **Executor**:
  - è Processes running in Workers
  - è Launches individual *tasks* of a Spark Job
  - è Each executor uses certain 'Cores' and 'Memory'
- **Task:** Does the actual computation by computing an RDD partition
- **Stage**: Set of parallel tasks – one task per partition. Multiple stages = a spark job
- **RDD:** ?
- **RDD Partition:** ?

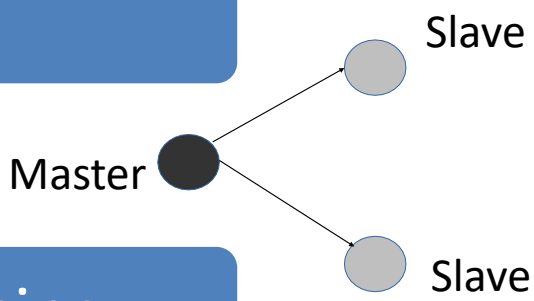
Cluster : Set of machines grouped together

Node: Individual machine in a cluster

Master:

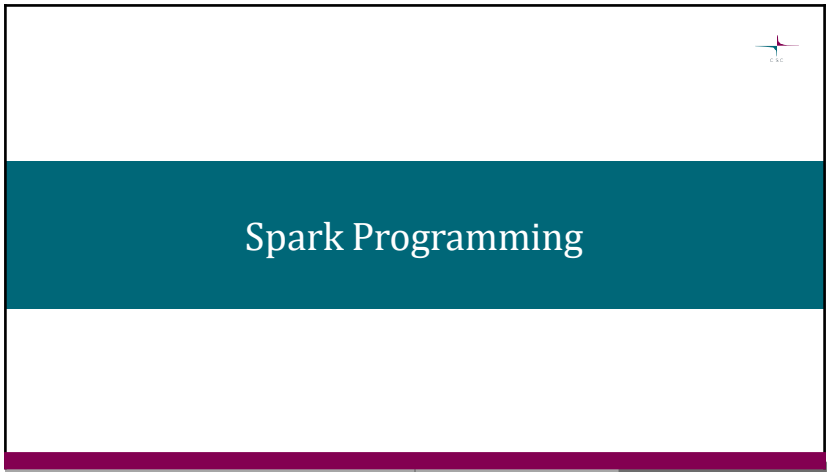
- Entry point of a cluster
- Delegates requests to workers

Worker/Slave: Carries out the processing



Why Run Spark on Cloud?

- You can request as much resource as you want
- You can pay per usage so adding more resources or deleting it would be flexible
- You can access it from anywhere
- You can use the same cluster for a group



Python: Basic Data Structures

A = [1, 2, 3, 4, 5]

[[1, 2], [3, 4, 5]]

B = (1, 2, 3, 4, 5)

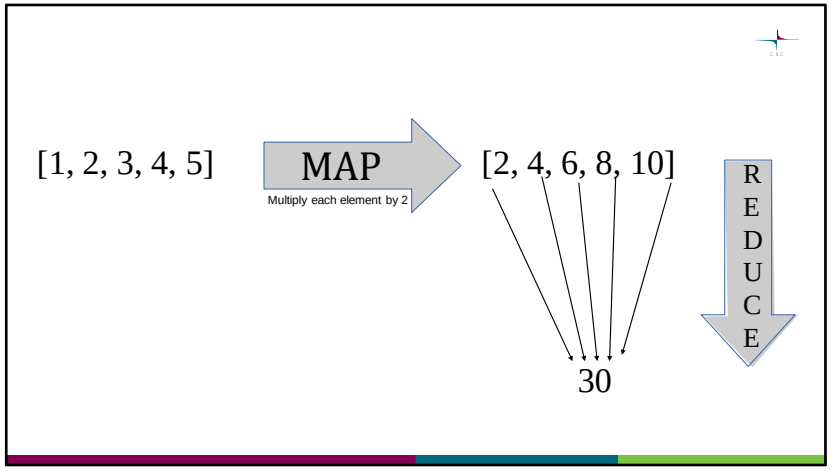
C = {'key1':'value1', 'key2': 'value2'}

A[0] = 1

B[4] = 5

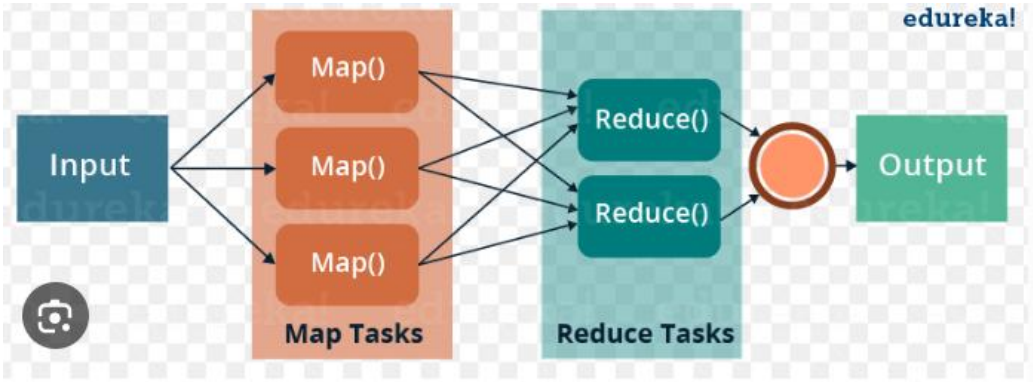
C['key1'] = value1

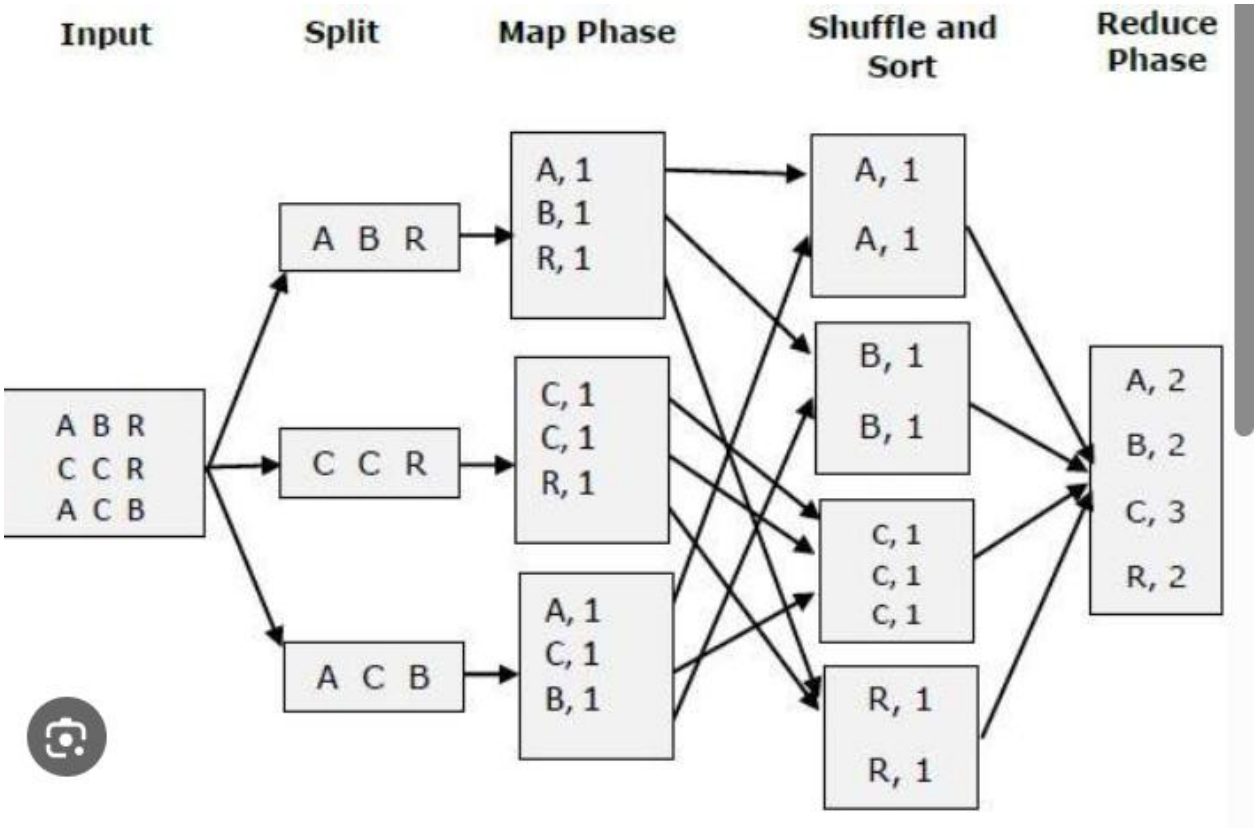
Map-Reduce Paradigm

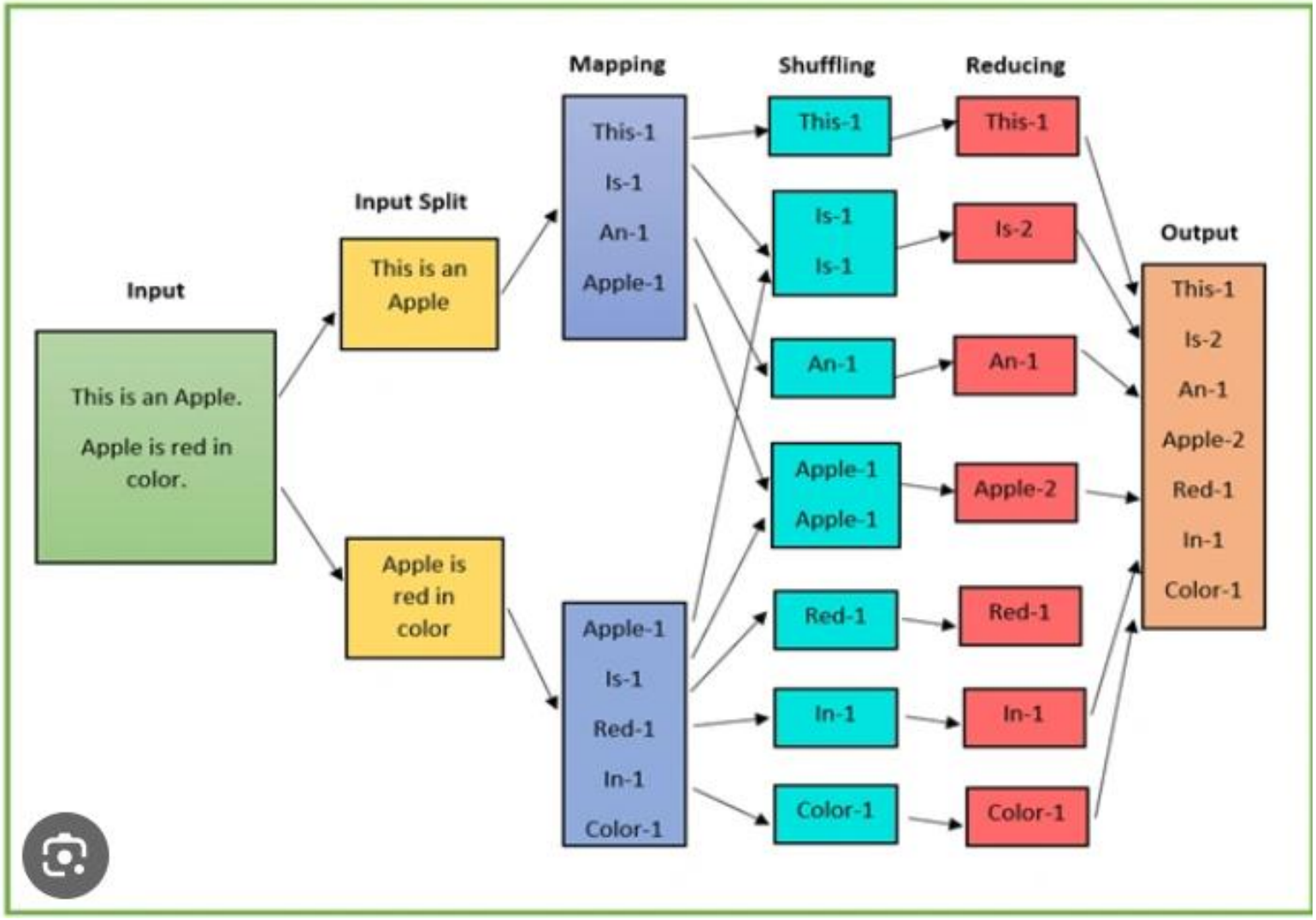


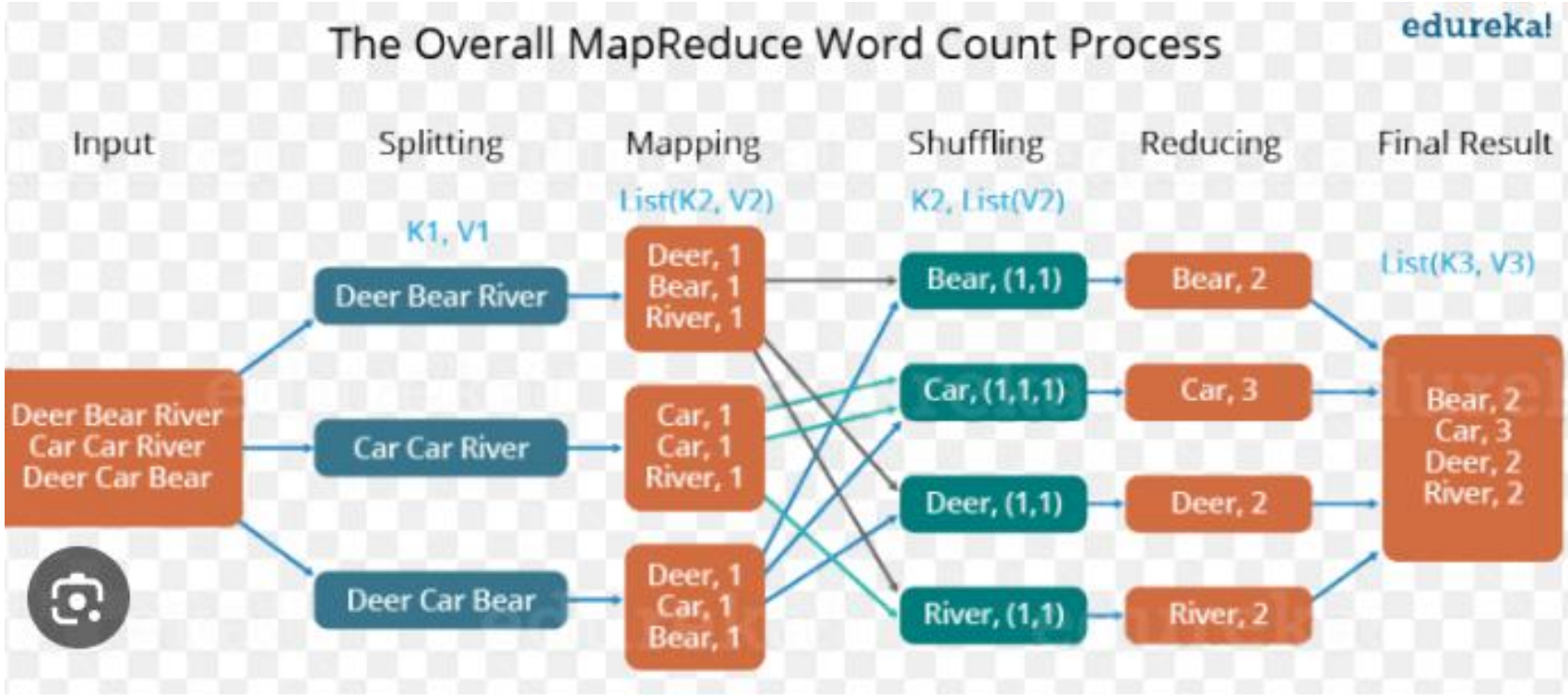
Spark: RDD

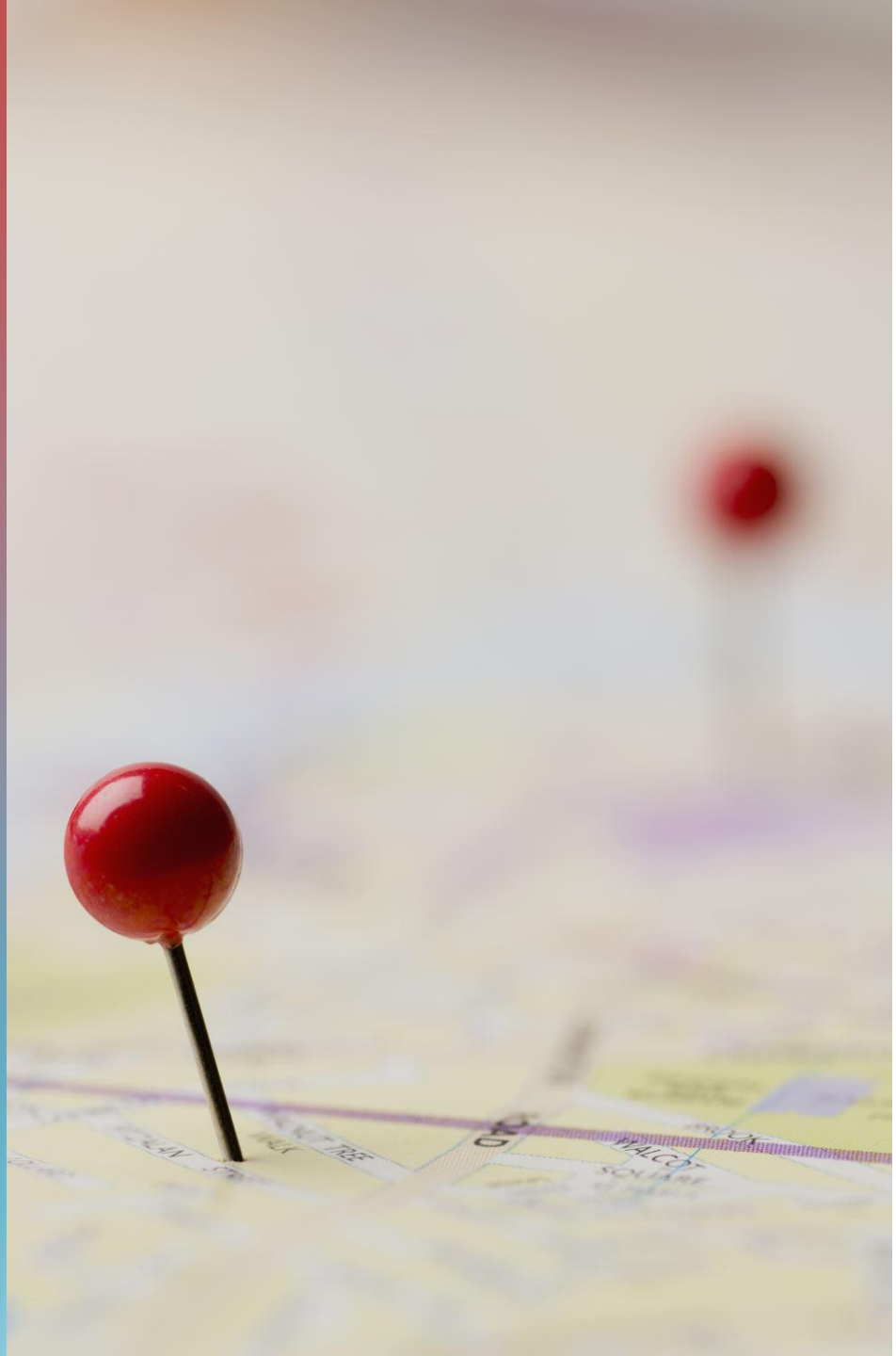
- RDD stands for Resilient Distributed Datasets
- Spark's fault tolerant dataset which can be operated in a distributed manner
  - ➔ Running the processing across all the nodes of a cluster
- Can be used to store any type of element in it
- *Lazy Evaluation* with all the transformations
- Some Special types of RDDs:
  - Double RDD: for storing numerical data
  - Pair RDD: For storing key pair values


edureka!
Input
Map()
Map()
Map()
Map Tasks
Reduce()
Reduce()
Reduce Tasks
Output


Input
Split
Map Phase
Shuffle and Sort
Reduce Phase
A B R
C C R
A C B
A B R
C C R
A C B
A, 1
B, 1
R, 1
C, 1
C, 1
R, 1
A, 1
C, 1
B, 1
A, 1
A, 1
B, 1
B, 1
C, 1
C, 1
C, 1
R, 1
R, 1
A, 2
B, 2
C, 3
R, 2


Input
This is an Apple.
Apple is red in color.
Input Split
This is an Apple
Apple is red in color
Mapping
This-1
Is-1
An-1
Apple-1
Apple-1
Is-1
Red-1
In-1
Color-1
Shuffling
This-1
Is-1
Is-1
An-1
Apple-1
Apple-1
Red-1
In-1
Color-1
Reducing
This-1
Is-2
An-1
Apple-2
Red-1
In-1
Color-1
Output
This-1
Is-2
An-1
Apple-2
Red-1
In-1
Color-1


The Overall MapReduce Word Count Process
edureka!
Input
Splitting
Mapping
Shuffling
Reducing
Final Result
K1, V1
List(K2, V2)
K2, List(V2)
List(K3, V3)
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, (1,1)
Car, (1,1,1)
Deer, (1,1)
River, (1,1)
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2



- **Practical examples**
- **Example 1:**
- **https://colab.research.google.com/github/RPI-DATA/course-intro-ml-app/blob/master/content/notebooks/18-big-data/02-intro-spark.ipynb#scrollTo=BYrclDoIEOMX**
- **Example 2:**
- **https://colab.research.google.com/github/arminnorouzi/sparkml/blob/main/Notebooks/sparkml_tutorial.ipynb#scrollTo=7aiotEBtRXyP**

Learn more?

- Check:
- Spark official home page
- https://spark.apache.org/examples.html
- GitHub tutorials/examples
- https://github.com/vara-co/Home_Sales
- Ebooks/tutorials
  https://runawayhorse001.github.io/LearningApacheSpark/pyspark.pdf