Machine Reasoning
Day 4

Course Manager: GU Zhan (Sam)

zhan.gu@nus.edu.sg

Machine Reasoning

Day 4

Day 4 Agenda

4.1 Knowledge Discovery by Machine Learning (Big Data)

- 4.1.1 Big Data Overview
- 4.1.2 NoSQL (Not Only SQL)
- 4.1.3 Exercise

4.2 Contemporary Reasoning Systems (Big Data)

- 4.2.1 Big Data Applications
- 4.2.2 Development Eco-systems (Hadoop & Spark)
- 4.2.3 Exercise

4.3 Building Machine Reasoning System [Workshop]

- 4.3.1 Big Data Machine Learning & Reasoning System
- 4.3.2 Workshop Submission

4.4 In-class Assessment

- 4.4.1 In-class Assessment

4.1 Knowledge Discovery by Machine Learning (Big Data)

4.1.1 Big Data Overview

4.1.2 NoSQL (Not Only SQL)

4.1.3 Exercise

4.1 Knowledge Discovery by Machine Learning (Big Data)

4.1.1 Big Data Overview

4.1.2 NoSQL (Not Only SQL)

4.1.3 Exercise

Big Data

- Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software.

- It is also a term used to describe large volumes of data which is hard to process using traditional database management systems (DBMS).

Big Data 5 V's

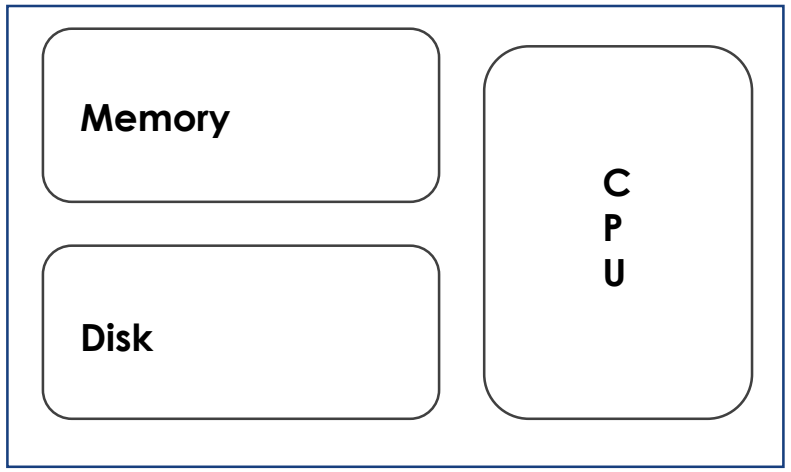
1. **Volume** : refers to the huge amount of data produced;
2. **Velocity** : refers to the speed at which data is generated;
3. **Variety** : The variety of data available/generated, which includes (structured/semi-structured/unstructured data);
4. **Veracity** : It refers to the uncertainties in data produced, i.e., real world data is messy and can be hard to analyze;
5. **Value**: Insights generated must be based on accurate data so that the knowledge discovered is of value to the company;

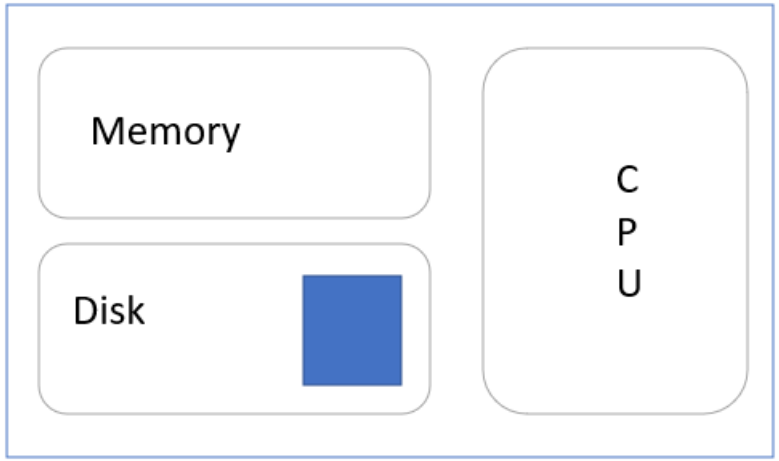
Traditional DBMS Systems

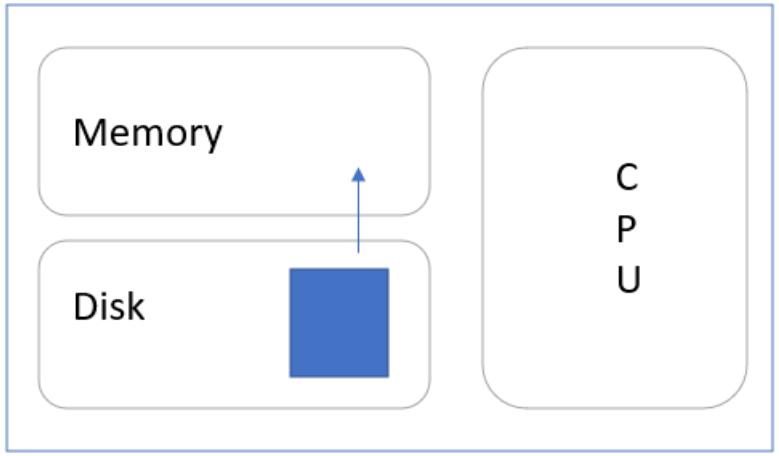
In a traditional database management system (DBMS):

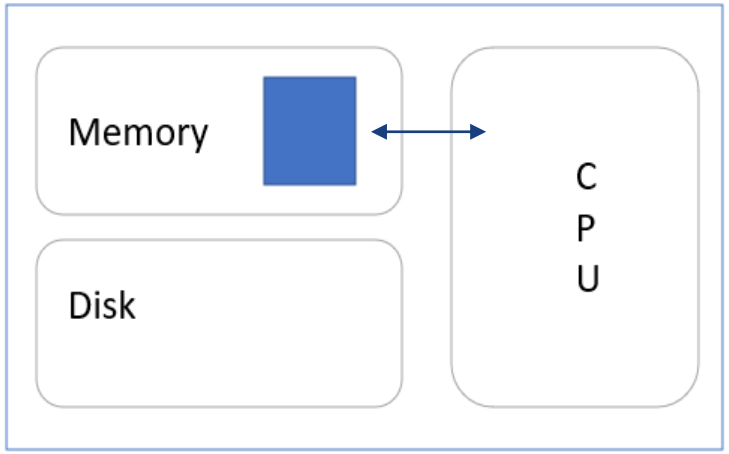
- Data resides on disk and loaded into memory when needed, we do processing by spending CPU cycles on the server and then the data is presented to the user for analysis, which in turn spends/uses User CPU and memory.
- Resources (**memory size**) are limited and its **slow (I/O: inputs/outputs)** to load data from disk. Big data or extremely large data sets that could be analyzed computationally to reveal insights like patterns, trends and associations is unlikely to fit in to memory **all at once**.

Traditional DBMS Systems









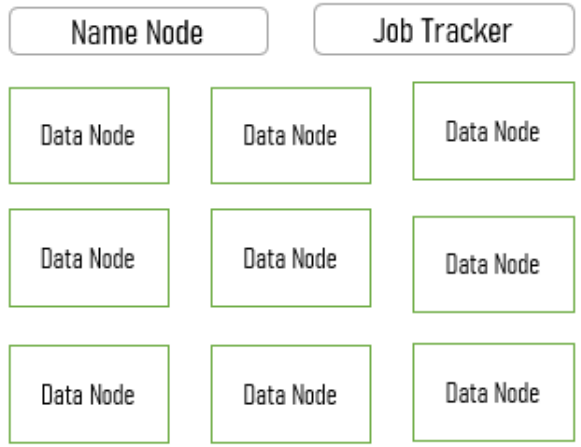
How to Handle Big Data?

- The solution is to scale out and use ***distributed computing*** by connecting many servers to perform jobs.

- A distributed system is a system whose components are located on ***different networked computers***, which communicate and coordinate their actions by passing messages to one another, via Remote Procedure Calls.

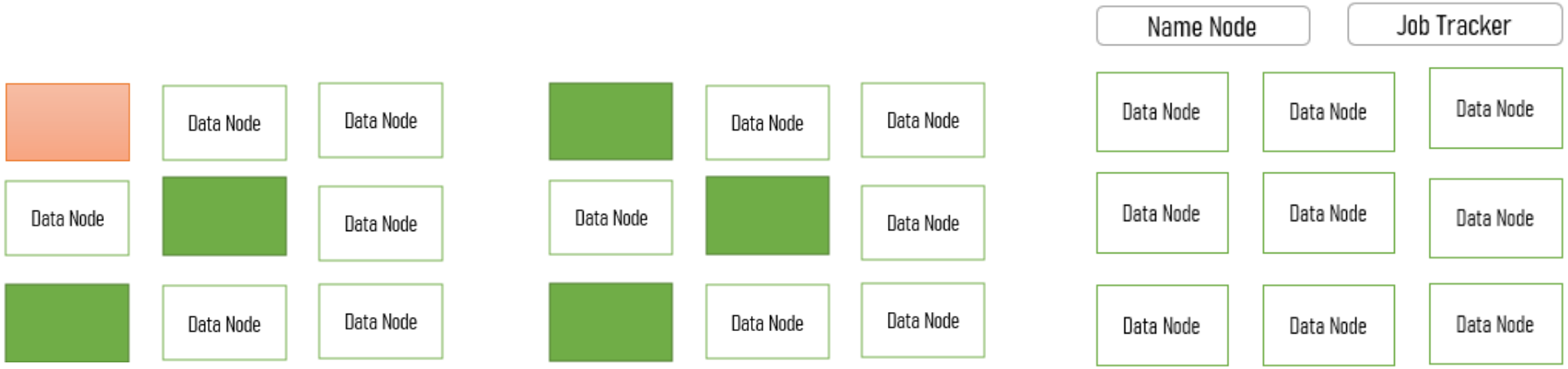
Distributed Computing

- Recover from failures
- Shared nothing architecture
- Hadoop distributed file system (HDFS)
- Map-Reduce (Divide-Conquer method)



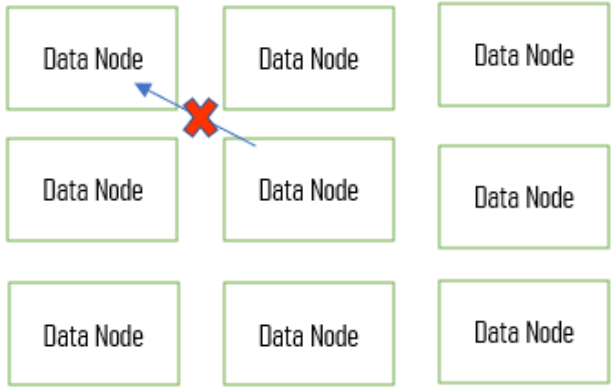
Distributed Computing

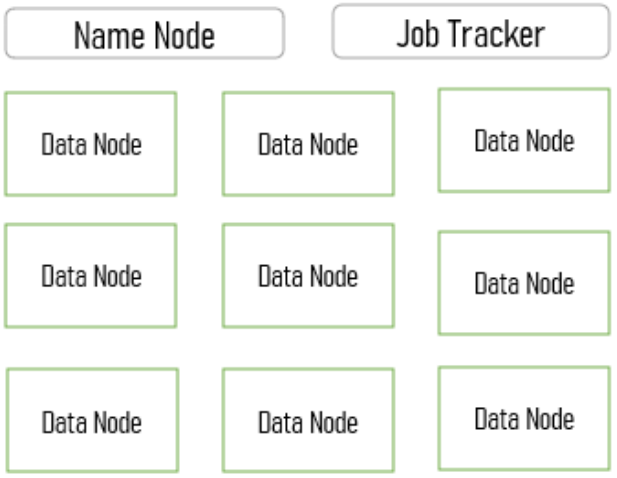
- Recover from failures: With more resources things are bound to fail more frequently, because of that they must be able to recover from failure in a scalable manner;



Distributed Computing

- Shared Nothing Architecture: In order to scale out efficiently, we do not burden the master servers / name nodes, like coordination and communication with other data nodes: reduced networking communication overheads;
- Each ***data node*** is unaware of the other nodes in a cluster;
- The Coordination task is left to the master servers: Name Node and Job Tracker;





Distributed Computing

- Each Data node is unaware of the other nodes in a cluster;
- Data Nodes be like …

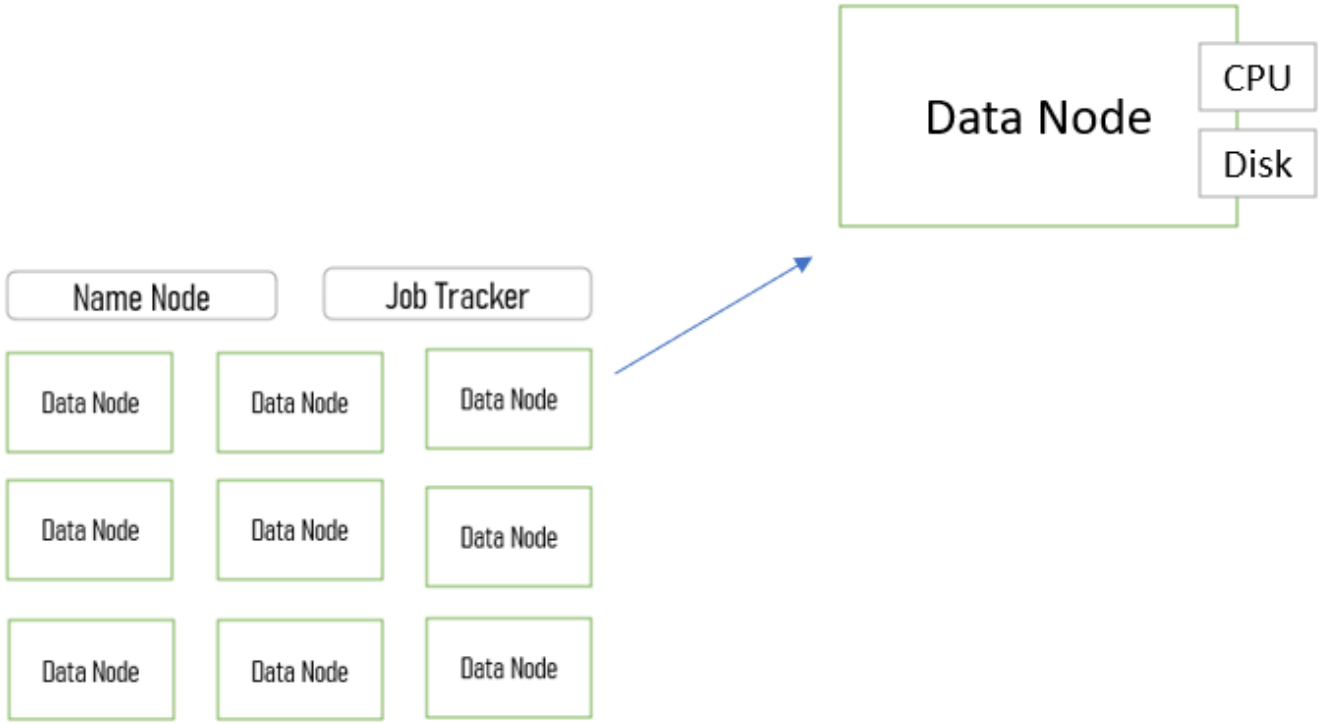
Distributed Computing

HDFS (Hadoop Distributed File System)

- Hadoop is an opensource implementation of the Hadoop distributed file System. It masks the complexities when with working with distributed systems;
- Provides fault tolerant distributed file system on top of these nodes; (servers/storages) carrying forward all the characteristics as well as an API that makes it easier to work on Hadoop;

Distributed Computing

- Map-Reduce: Takes compute against ***data stored on disk*** instead of ***data loaded in memory***;



4.1 Knowledge Discovery by Machine Learning (Big Data)

4.1.1 Big Data Overview

4.1.2 NoSQL (Not Only SQL)

4.1.3 Exercise

What is NoSQL?

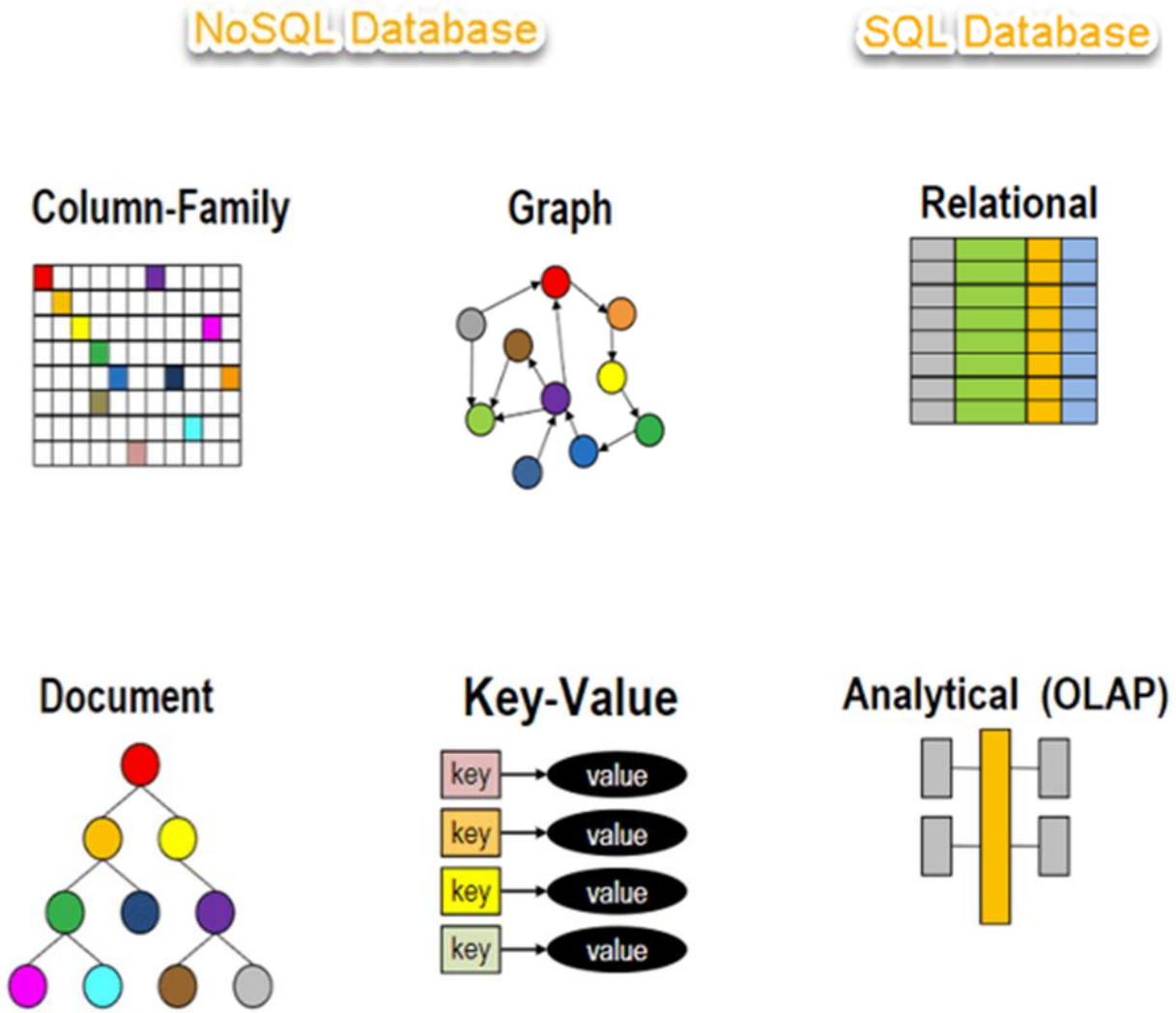
- **NoSQL** databases (aka. "not only SQL") are non tabular, and store data differently than relational tables;

- NoSQL databases come in a variety of types based on their data model / db schema;

- The main types are: column-family/wide-column, graph, document, key-value;

Types of NoSQL Databases

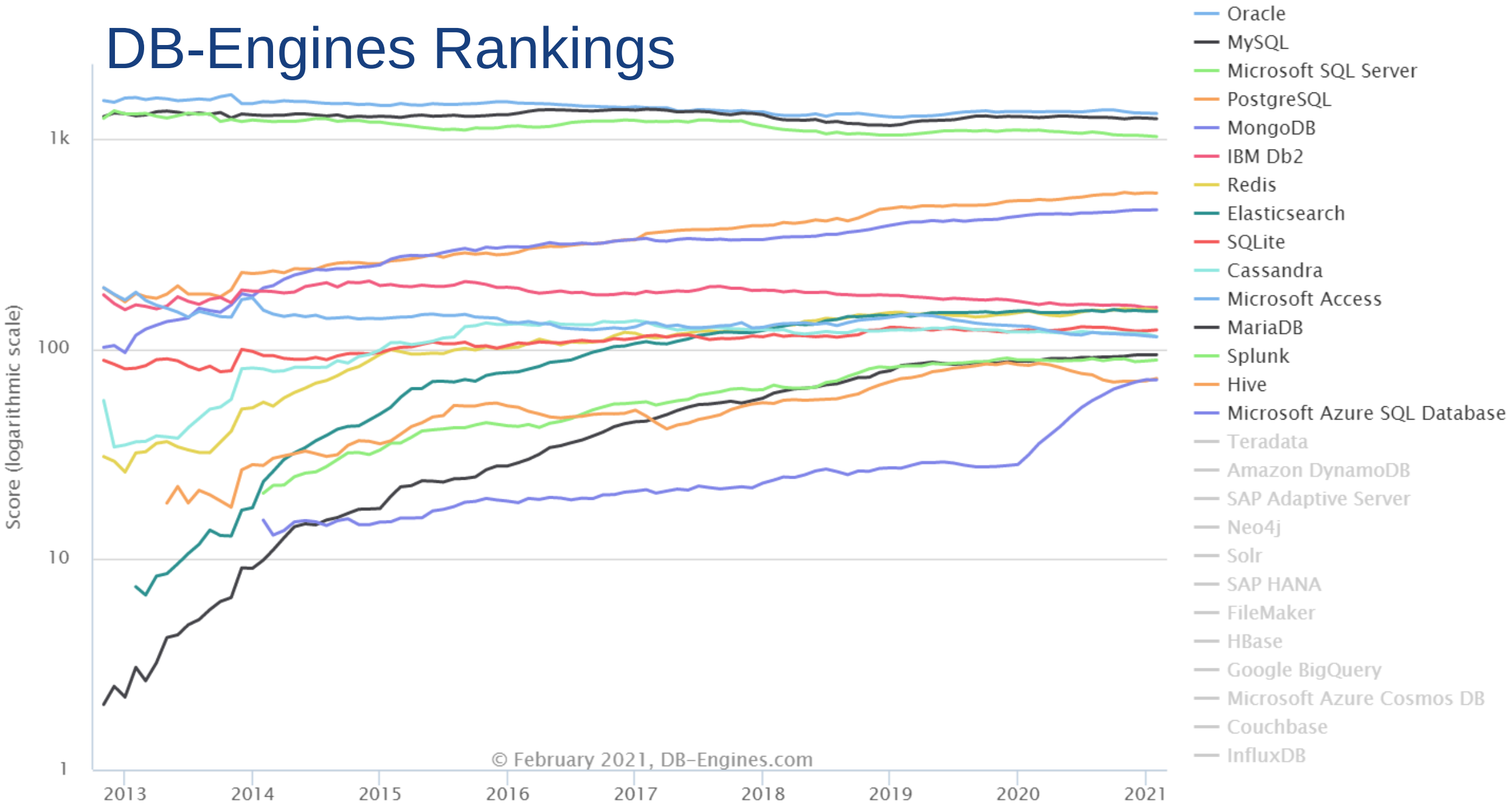
- **Column-family / Wide-column stores** store data in tables, rows, and dynamic columns;
- **Graph databases** store data in nodes and edges;
- **Document databases** store data in documents similar to JSON (**J**ava**S**cript **O**bject **N**otation) objects. Each document contains pairs of (embedded) fields and values;
- **Key-value databases** are a simpler type of database where each item contains keys and values;



Source https://www.mongodb.com/nosql-explained

	SQL Databases	NoSQL Databases
Data Storage Model	Tables with fixed rows and columns	Document: JSON documents, Key-value: key-value pairs, Wide-column: tables with rows and dynamic columns, Graph: nodes and edges
Development History	Developed in the 1970s with a focus on reducing data duplication	Developed in the late 2000s with a focus on scaling and allowing for rapid application change driven by agile and DevOps practices.
Examples	Oracle, MySQL, Microsoft SQL Server, and PostgreSQL	Document: MongoDB and CouchDB, Key-value: Redis and DynamoDB, Wide-column: Cassandra and HBase, Graph: Neo4j and Amazon Neptune

DB-Engines Rankings



Source https://db-engines.com/en/ranking

Data Structure in SQL DB

- Example: Employee DB
- Each row in each table is a single value(employee), and each column has fixed attributes:

Emp_ID	Emp_firstname	Emp_lastname	Emp_phone
1	Geet	Jethwani	81815656

Data Structure in NoSQL DB

- Example: document database – JSON;
- Each of the SQL column attributes would be fields;
- The details of an employee's record would be the data values associated with each field;
- For Example: Emp_firstname: "Geet", Emp_lastname: "Jethwani"

Data Structure: JSON

- Experience data represented in semi-structured format;
- We have looked at the structured (tabular) formats;
- However, we must understand JSON representation as in real world big data application, one may come across this very frequently;
- When setting up a data store, your first task is to answer the question: “What data would I like to store and how do the fields relate to each other”?

name	quantity	size	status	tags	rating
journal	25	14x21,cm	A	brown, lined	9
notebook	50	8.5x11,in	A	college-ruled,perforated	8

Data Structure: JSON

- JSON is formatted as name-value pairs;
- In JSON documents, fieldnames and values are separated by a colon, fieldname and value pairs are separated by commas, and sets of fields are encapsulated in “curly brackets” ({});
- Example:

```
{
 "name"        : "notebook",
 "quantity"    : 50,
 "size"        : { "length": 11, "width": 8.5, "unit": "in" },
 "status"      : "A",
 "tags"        : [ "college-ruled", "perforated" ],
 "rating"      : 8
}
```

Key

Value

Key

Value

4.1 Knowledge Discovery by Machine Learning (Big Data)

4.1.1 Big Data Overview

4.1.2 NoSQL (Not Only SQL)

4.1.3 Exercise

[Exercise] Knowledge Discovery by Machine Learning (Big Data)

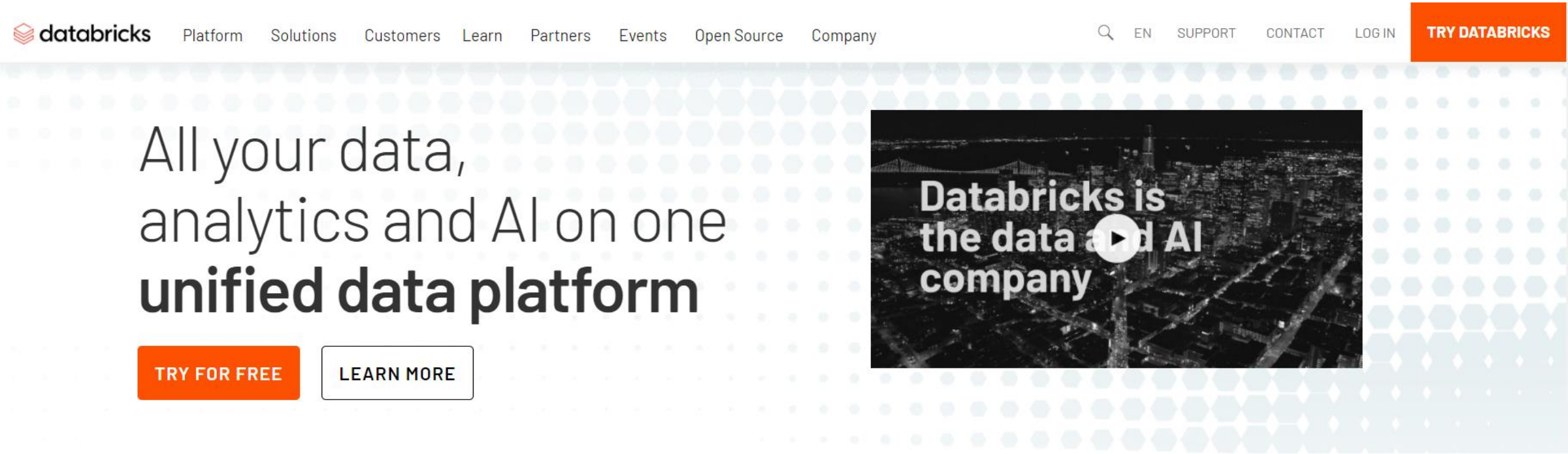
- **Setup Databricks account use for free: Community Edition.**
- **Structure data in JSON and flatten it into a table.**

What is DataBricks?

- Unified big data analytics platform;
- Run all knowledge discovery & analytics in one place;
- Powering dashboards, running reports;
- Machine learning and streaming jobs;
- Databricks is more optimized than open-source apache spark;
- Easier installation and no hassles of managing clusters;

DataBricks

- https://databricks.com/



databricks

Try Databricks for free

An open and unified data analytics platform for data engineering, data science, machine learning, and analytics.

From the original creators of Apache Spark™, Delta lake, MLflow, and Koalas.

aws Microsoft Azure

Databricks trial:

- Collaborative environment for data teams to build solutions together.
- Interactive notebooks to use Apache Spark™, SQL, Python, Scala, Delta Lake, MLflow, TensorFlow, Keras, Scikit-learn and more.
- Available as a 14-day full trial in your own cloud, or as a lightweight trial hosted by Databricks.

Used by:

Virgin hyperloop edmunds nielsen

RIOT GAMES REGENERON Grab

Please tell us about yourself

First Name: *

Last Name: *

Company *

Company Email *

Title *

Phone Number

Keep me informed with occasional updates about Databricks and related open source products

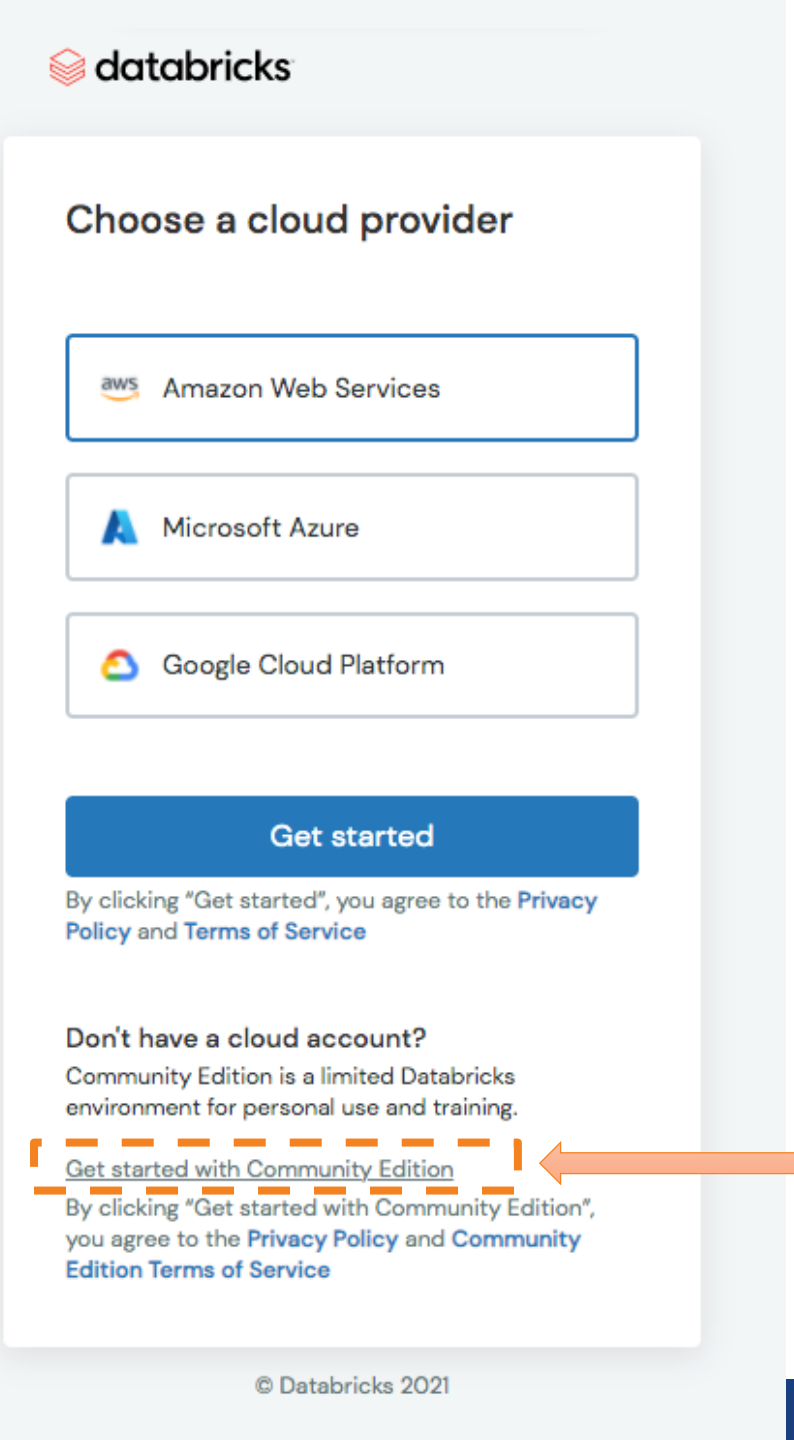
By Clicking "Get Started For Free", you agree to the Privacy Policy.

GET STARTED FOR FREE

I'm not a robot

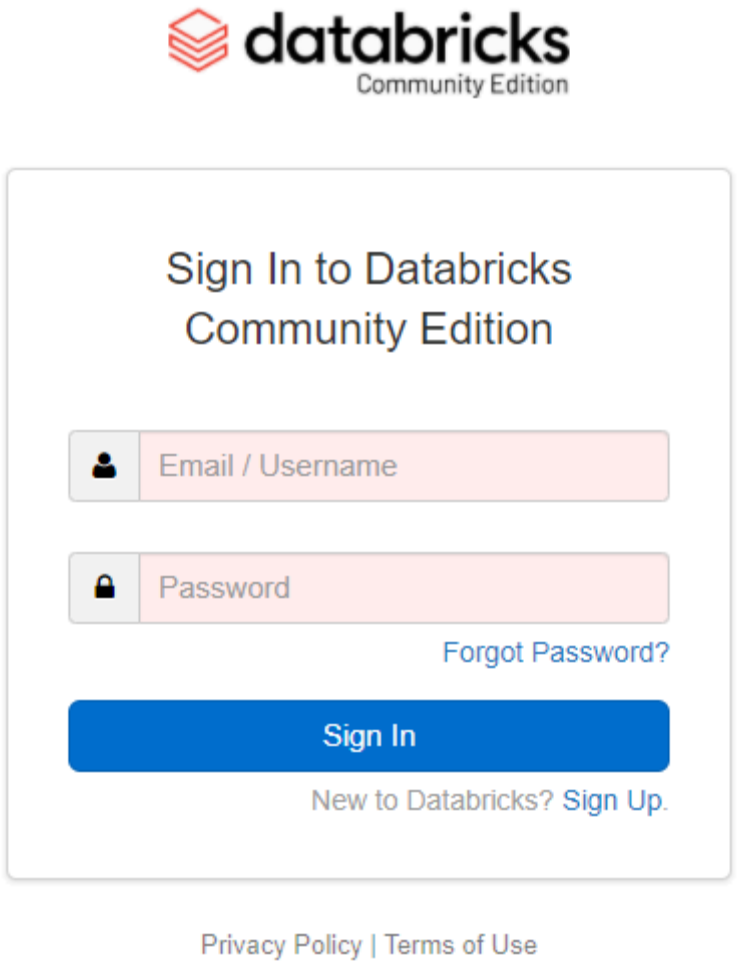
reCAPTCHA

Privacy - Terms



Select

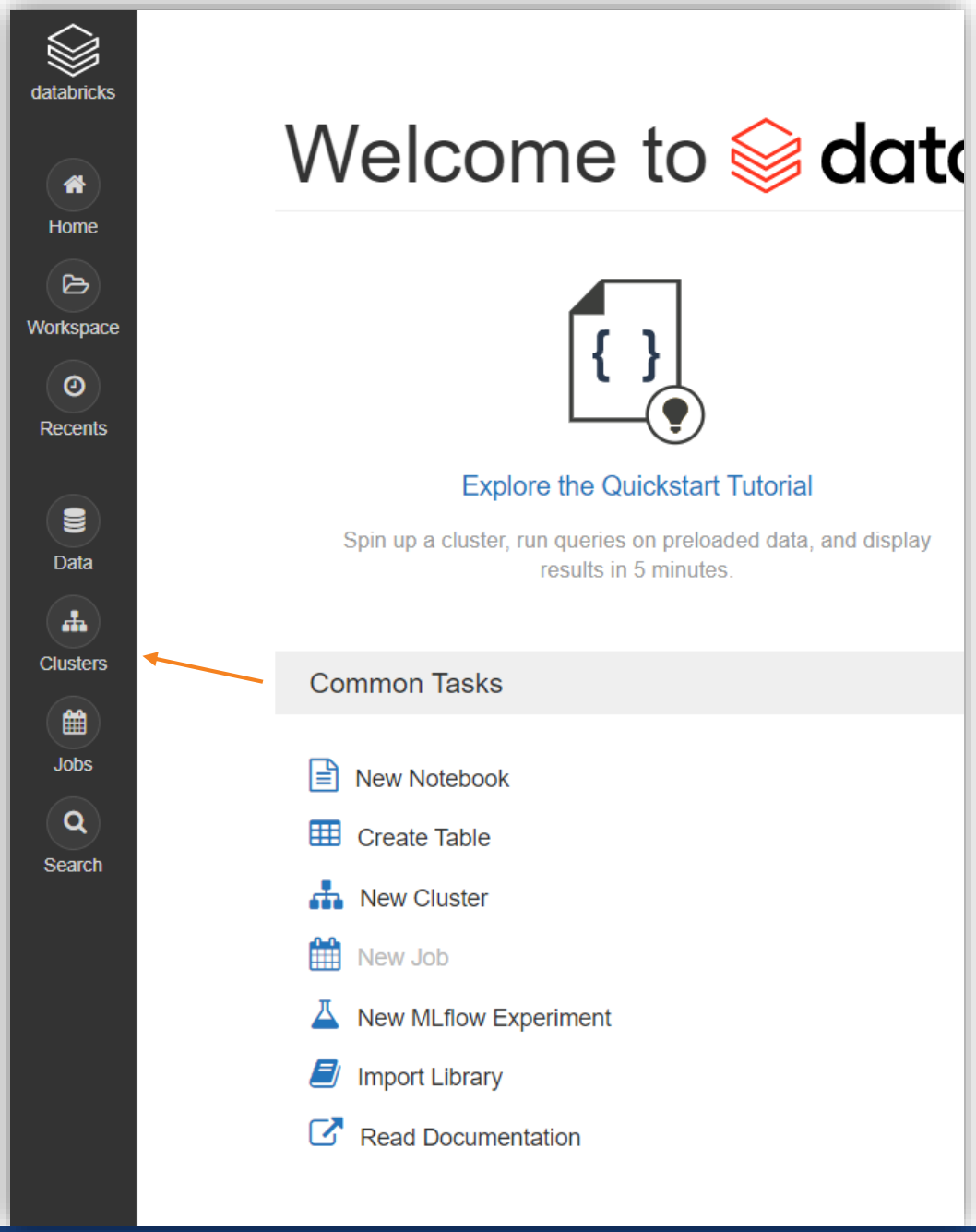
Community Edition



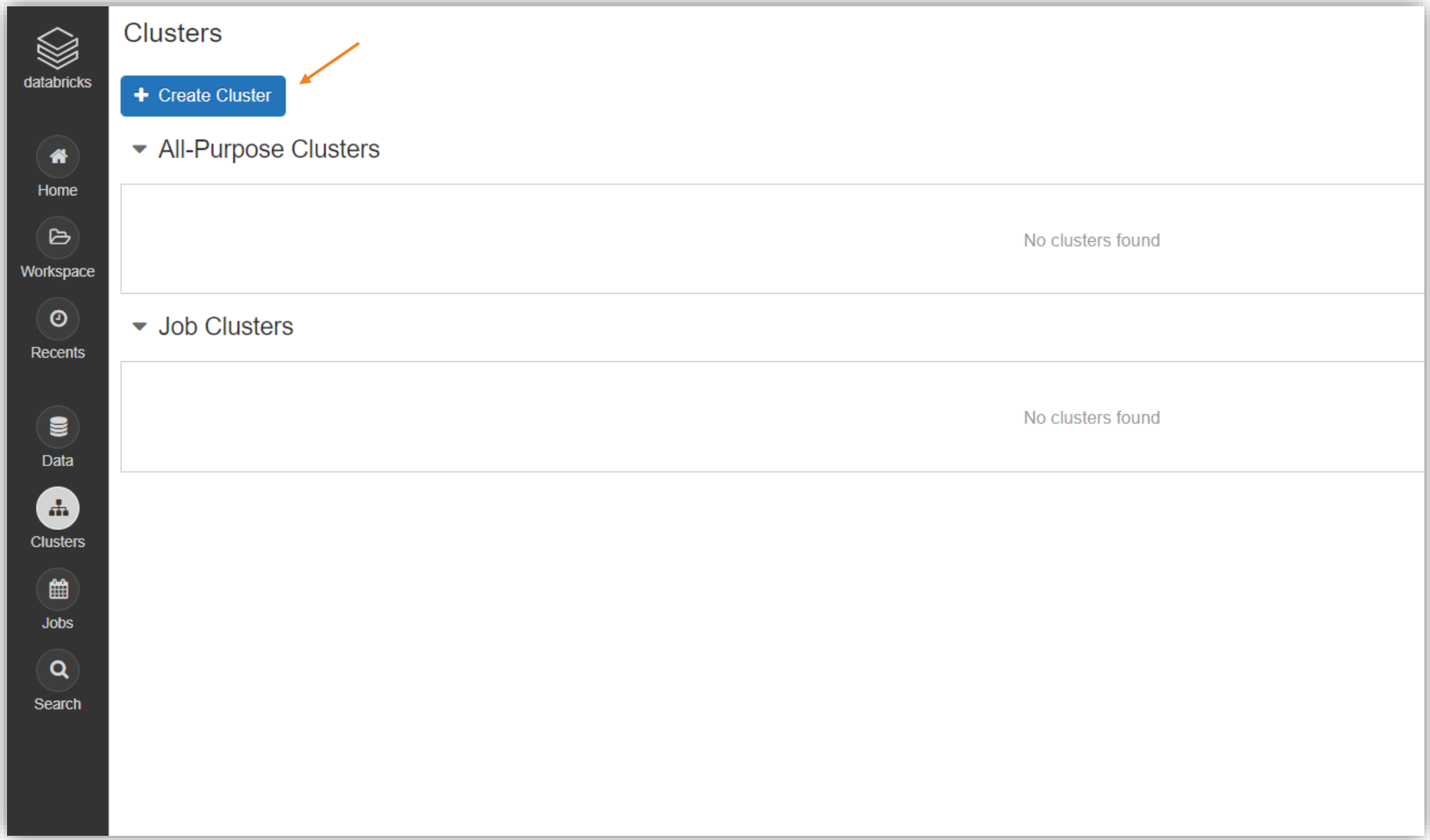
Login to your account;

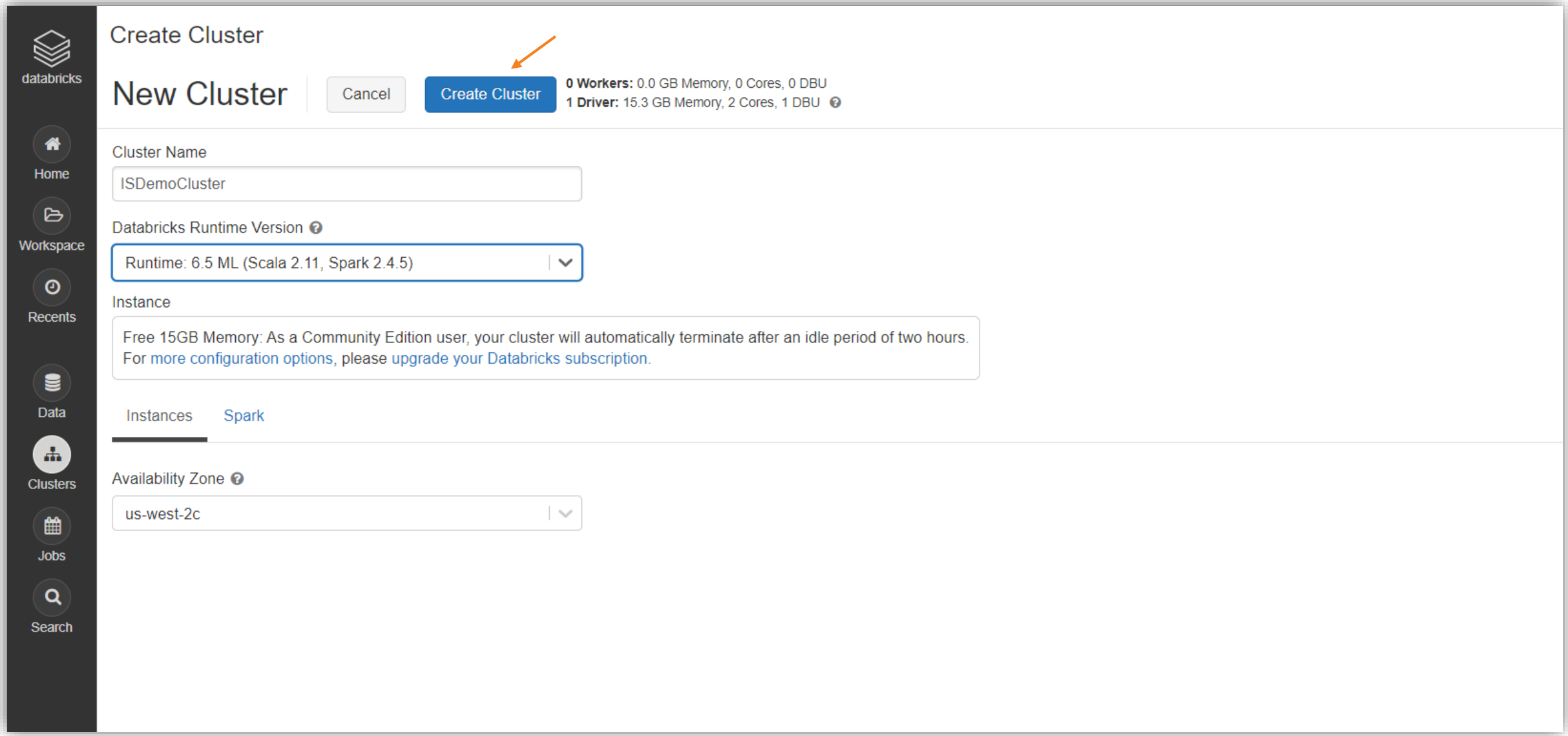
Community Edition log in portal:

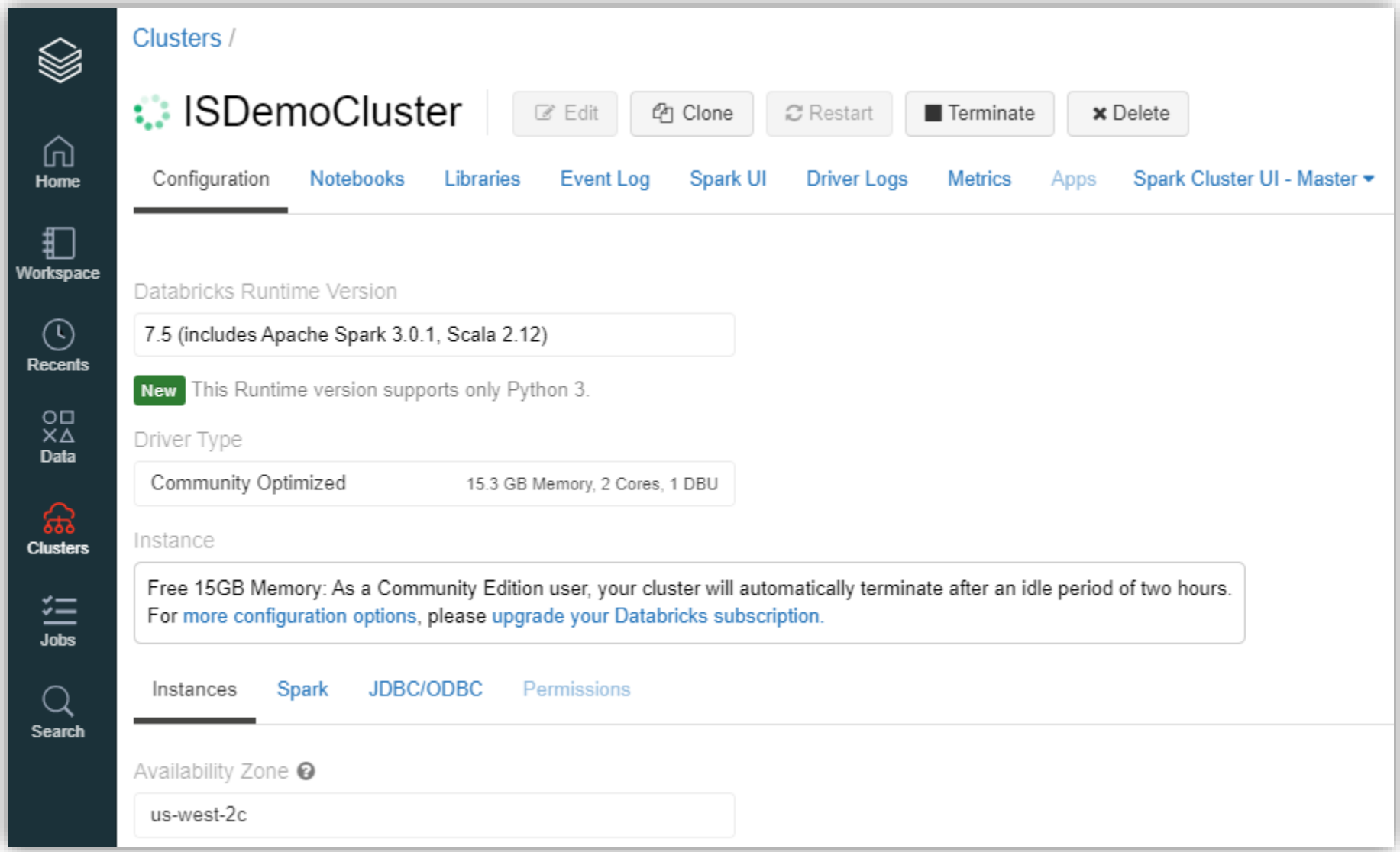
https://community.cloud.databricks.com/login.html



- Now create clusters;
- To get your cluster up and running;
- Click the clusters button as shown;


databricks
Clusters
+ Create Cluster
All-Purpose Clusters
No clusters found
Job Clusters
No clusters found
Home
Workspace
Recents
Data
Clusters
Jobs
Search




Clusters /
ISDemoCluster
Edit
Clone
Restart
Terminate
Delete
Home
Workspace
Recents
Data
Clusters
Jobs
Search
Configuration
Notebooks
Libraries
Event Log
Spark UI
Driver Logs
Metrics
Apps
Spark Cluster UI - Master
Databricks Runtime Version
7.5 (includes Apache Spark 3.0.1, Scala 2.12)
New
This Runtime version supports only Python 3.
Driver Type
Community Optimized
15.3 GB Memory, 2 Cores, 1 DBU
Instance
Free 15GB Memory: As a Community Edition user, your cluster will automatically terminate after an idle period of two hours.
For more configuration options, please upgrade your Databricks subscription.
Instances
Spark
JDBC/ODBC
Permissions
Availability Zone
us-west-2c

Data Representation in JSON

- In our Machine Memory module we used SQLite DB to represent and store data a structured tabular format;

- However, in other scenarios we may obtain data in different formats which may not be as simple to interpret as the relational data format (tables);

- Let us now structure our data in JSON format: https://en.wikipedia.org/wiki/JSON.

Data Representation in JSON Example

- Lets take an example of Iris flower dataset.

Iris Versicolor **Iris Setosa** **Iris Virginica**

- When represented in the tabular format:

SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa

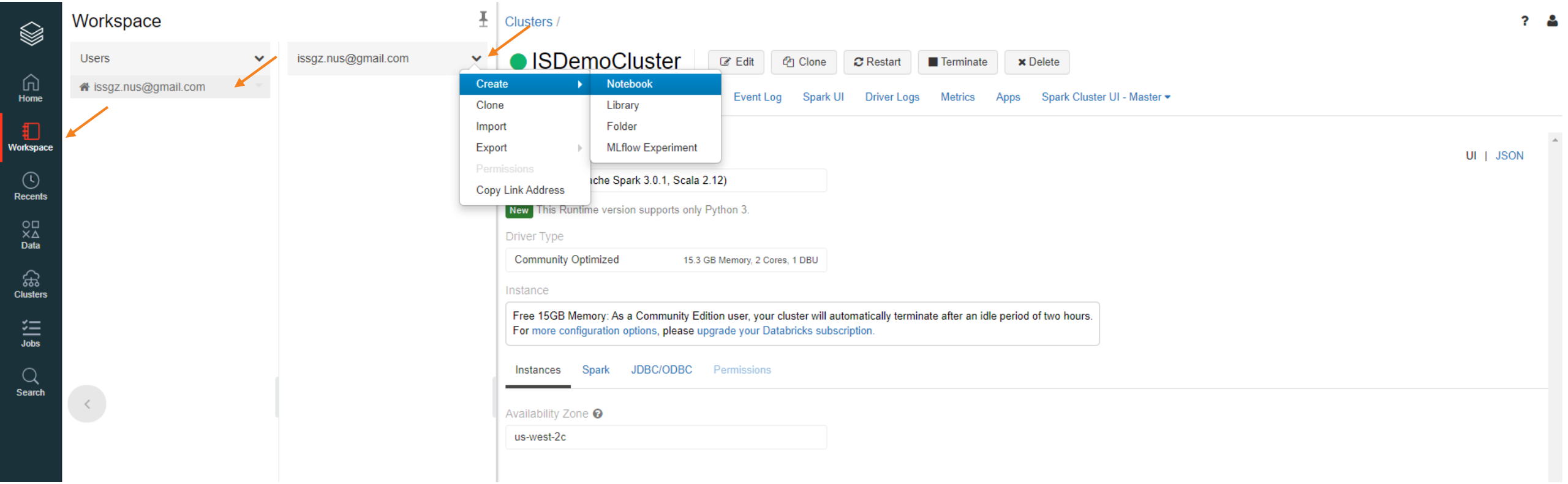
Source https://prasanth-ntu.github.io/html/ML-Course/Intro-to-ML-with-Python/images/iris-machinelearning.png

Data Representation in JSON Example

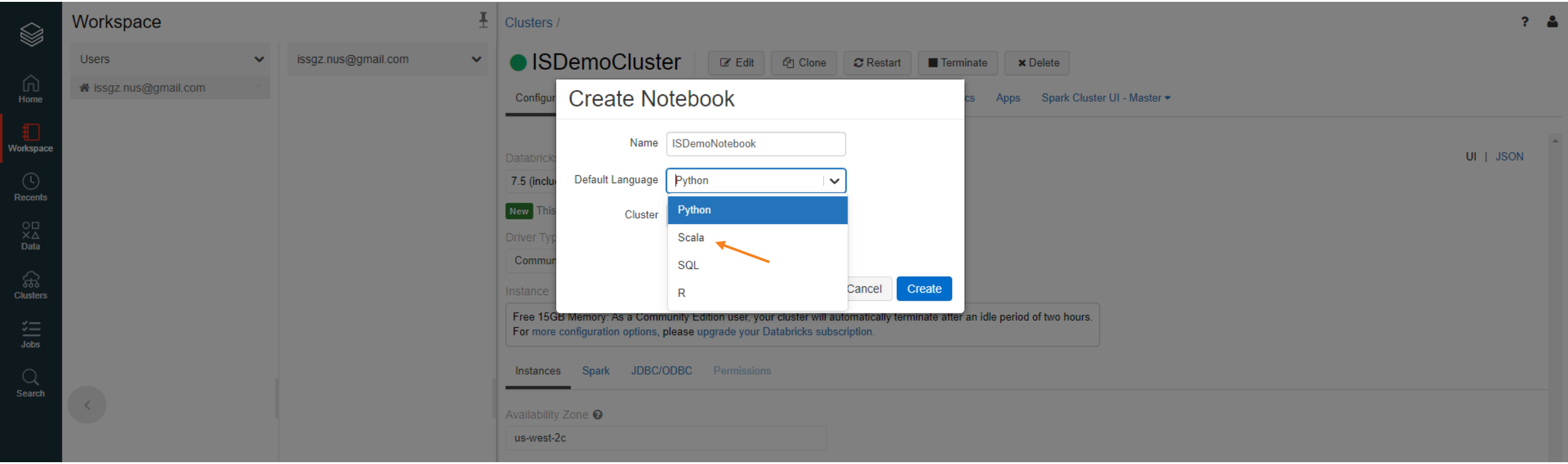
- The same content can be structured in JSON as:

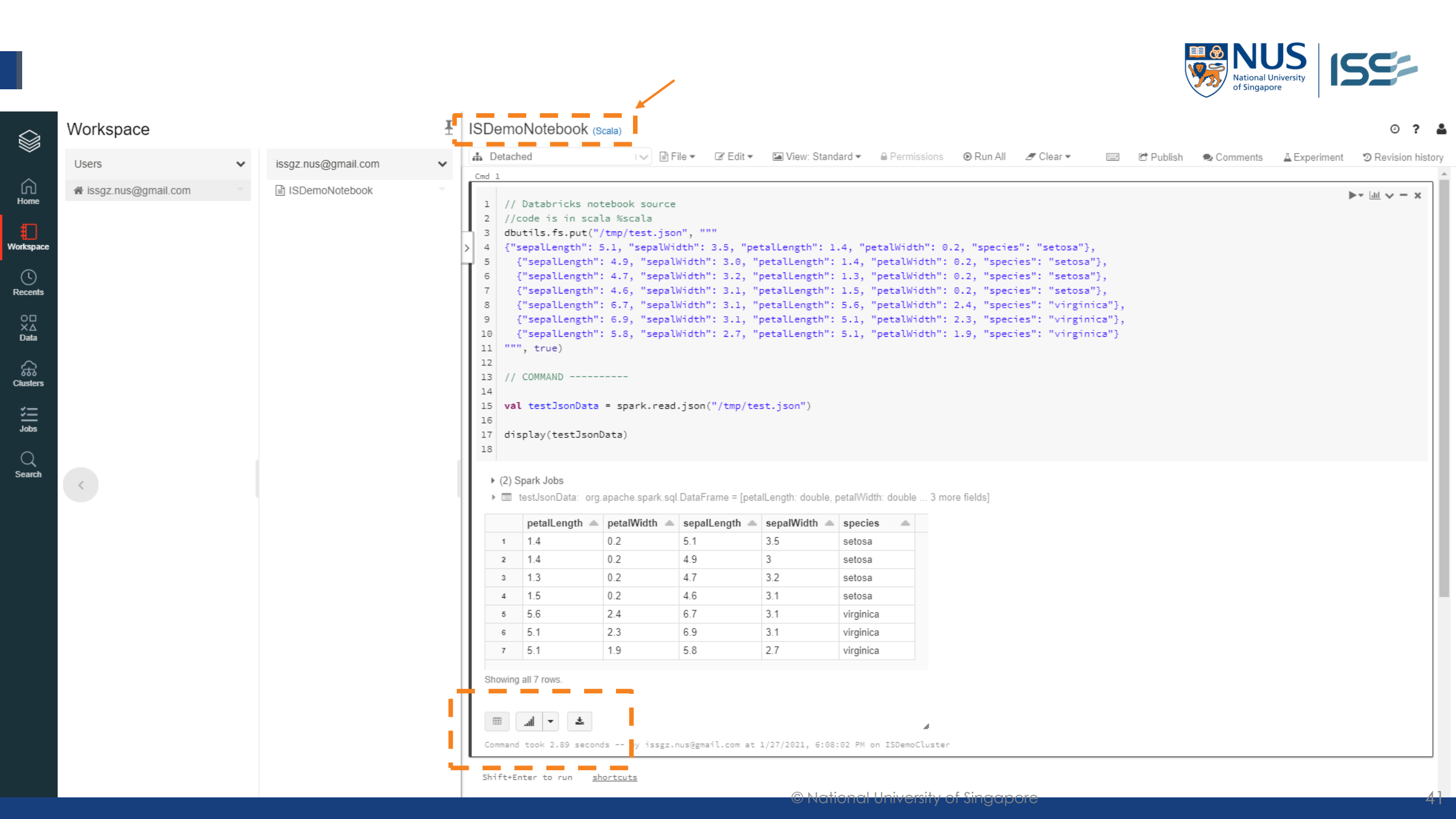
```
//code is in scala %scala
dbutils.fs.put("/tmp/test.json", """
{"SepalLengthCm": 5.1, "SepalWidthCm": 3.5, "PetalLengthCm": 1.4, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 4.9, "SepalWidthCm": 3.0, "PetalLengthCm": 1.4, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 4.7, "SepalWidthCm": 3.2, "PetalLengthCm": 1.3, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 4.6, "SepalWidthCm": 3.1, "PetalLengthCm": 1.5, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 5.0, "SepalWidthCm": 3.6, "PetalLengthCm": 1.4, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 5.4, "SepalWidthCm": 3.9, "PetalLengthCm": 1.7, "PetalWidthCm": 0.4, "Species": "setosa"},
{"SepalLengthCm": 4.6, "SepalWidthCm": 3.4, "PetalLengthCm": 1.4, "PetalWidthCm": 0.3, "Species": "setosa"},
{"SepalLengthCm": 5.0, "SepalWidthCm": 3.4, "PetalLengthCm": 1.5, "PetalWidthCm": 0.2, "Species": "setosa"},
{"SepalLengthCm": 4.4, "SepalWidthCm": 2.9, "PetalLengthCm": 1.4, "PetalWidthCm": 0.2, "Species": "setosa"},
""", true)
// COMMAND ----------
val testJsonData = spark.read.json("/tmp/test.json")
display(testJsonData)
```

Your email id should appear in the workspace, create notebooks here



Choose: Scala





How to enable DBFS file browser?

Help Center | Documentation | Knowledge Base

Search here

aws

- Getting started with Databricks
- Databricks SQL Analytics guide
- Databricks Workspace guide
  - Get started with Databricks Workspace
  - Best practices
  - Language roadmaps
  - User guide
  - Data guide
  - Delta Lake and Delta Engine guide
  - Machine learning and deep learning guide

Manage the DBFS file browser

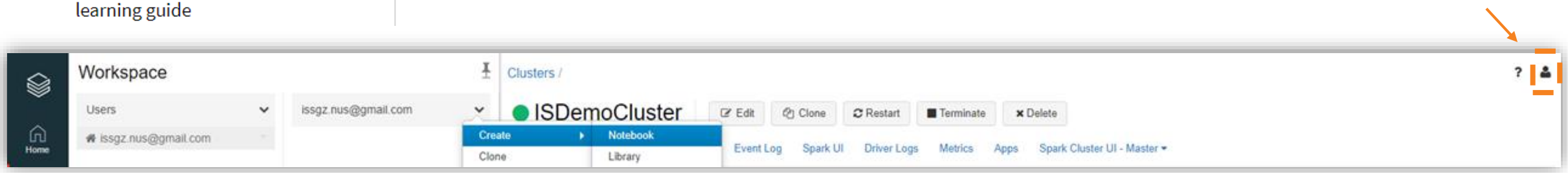
Manage the DBFS file browser

December 02, 2020

As an admin user, you can manage your users' ability to to browse data in the Databricks File System (DBFS) using the visual browser interface.

1. Go the Admin Console.
2. Click the **Advanced** tab.
3. Click the **Disable** or **Enable** button to the right of **DBFS File Browser**. This option is disabled by default.

This setting does not control programmatic access to the Databricks File System, for example through the DBFS command-line interface.



[Exercise]

- Refer to Iris flower example, structure all (or first 10) records of the toddler Autism data (csv) into JSON format;

- Use the created JSON format to import it into databrick cluster using notebook;

- **[Optional]** Visualize using the different graphs. Choose appropriate graphs to visualize your data;

4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

Big Data Applications (Hadoop)

- "One of the factors driving transformation for Western Union right now is the massive amount of transactional information that we accumulate as we serve our customers," noted Sanjay Saraf, the company's chief technology officer and senior vice president. "We've built an enterprise data hub on Cloudera to drive actionable insights that help the company create products and services that are relevant to our customers and help differentiate Western Union in a competitive marketplace."
- Solution: Western Union's enterprise data hub (EDH) feeds in structured data from multiple data warehouses as well as unstructured data including click streams, behavioral data, logs, and sentiment data collected by tools such as transactional, marketing, and other outreach systems. Western Union uses a combination of Apache Flume, Apache Sqoop, and Informatica Big Data Edition (BDE) to collect data from the various sources.

Source https://www.cloudera.com/about/customers/western-union.html

Big Data Applications (Hadoop)

- Tesla is using a Hadoop cluster to collect the increasing amount of data being generated by its connected cars.
- CIO Jay Vijayan said: “We are working on a big data platform... The car is connected, but it does not really talk to the network every minute because we want to keep it as smart and efficient as possible. It alerts us if the car is not functioning properly so service teams can take action.”

Source https://www.computerworld.com/article/3412201/12-hadoop-case-studies-in-the-enterprise.html

Big Data Applications (Spark)

NETFLIX

- Netflix has been determined to be able to predict what exactly its customers will enjoy watching with Big Data. As such, Big Data analytics is the fuel that fires the ‘recommendation engine’ designed to serve this purpose.
- More recently, Netflix started positioning itself as a content creator, not just a distribution method. Unsurprisingly, this strategy has been firmly driven by data. Netflix’s recommendation engines and new content decisions are fed by data points such as what titles customers watch, how often playback stopped, ratings are given, etc. The company’s data structure includes Hadoop, Hive and Pig with much other traditional business intelligence.

Source https://neilpatel.com/blog/how-netflix-uses-analytics/

The Secret Ingredients to Netflix's Success

1 Targeted Use Case

What are the business use cases upon which your big data and data science initiative should focus?

Why are these use cases important to the business?

How: Netflix's success rests on their ability to increase customer engagement through their **recommendation engine.**

2 Create Analytic Profiles

Viewer preferences are integrated with external data sources (like social media) in an **Analytic Profile**. These profiles capture analytic assets in a way that can be utilized across multiple use cases.

3 Capture Show Characteristics and Viewing Patterns

Build **Analytic Profiles** for each individual product. Netflix uses 100+ data points to tag each title.

4 Mastering Machine Learning

Data is fed into **Machine Learning** algorithms to create critical scores. The key to success here is to have a data science exploration and learning process to discover better predictors of customer behavior.

5 Management Fortitude to Become "Netflix Intelligent"

The management team must have a willingness to learn how to properly value and use the organization's data and analytics.

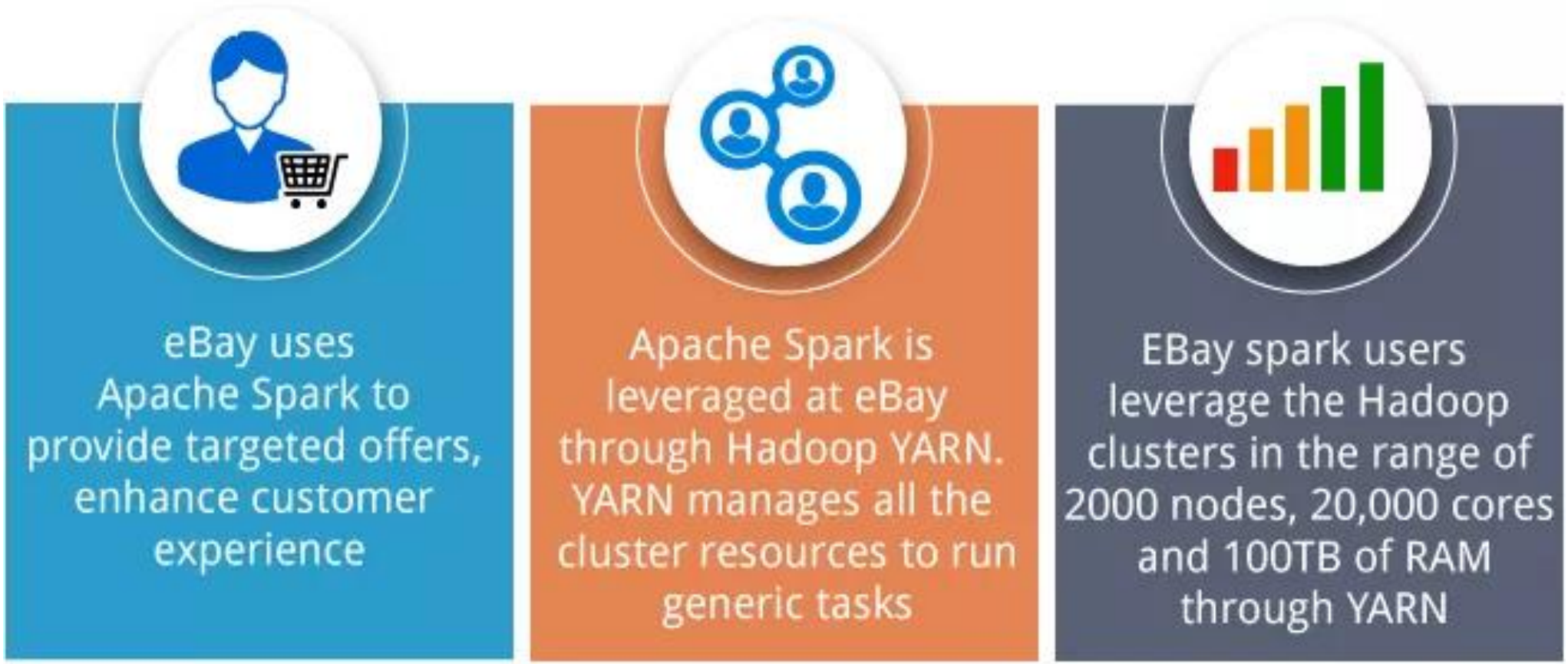
Source https://pbs.twimg.com/media/Dkvx0ySWwAYgH5o.jpg

Big Data Applications (Spark)

- A big technical challenge for eBay as a data-intensive business to exploit a system that can rapidly analyze and act on data as it arrives (streaming data).
- There are many rapidly evolving methods to support streaming data analysis. eBay is working with several tools including Apache Spark, Storm, Kafka. It allows the company's data analysts to search for information tags that have been associated with the data (metadata) and make it consumable to as many people as possible with the right level of security and permissions (data governance).
- The company has been at the forefront of using big data solutions and actively contributes its knowledge back to the open-source community.

Source https://www.forbes.com/sites/maribellopez/2016/10/04/how-ebay-uses-big-data-and-machine-learning-to-drive-business-value/?sh=5f1686531f35

Apache Spark at ebay



Source https://miro.medium.com/max/1304/1*aqux5d4MFNykUsw1vyuJLg.png

4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

Development Eco-systems (Hadoop & Spark)

is a distributed hard-disk computation platform.

- Hadoop distributed file systems (HDFS)
- Divide and conquer approach: Map-Reduce approach

is a distributed in-memory computation platform.

- Resilient Distributed Dataset (RDD)
- Spark RDD & ML library (decision tree)

Spark vs Hadoop MapReduce

Factors	Spark	Hadoop MapReduce
Speed	100x times than MapReduce	Faster than traditional system
Written In	Scala	Java
Data Processing	Batch / real-time / iterative / interactive /graph	Batch processing
Ease of Use	Compact & easier than Hadoop	Complex & lengthy
Caching	Caches the data in-memory & enhances the system performance	Doesn't support caching of data

Source https://data-flair.training/blogs/wp-content/uploads/sites/2/2016/09/Hadoop-MapReduce-vs-Apache-Spark.jpg

4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

Introduction to Hadoop Eco-systems

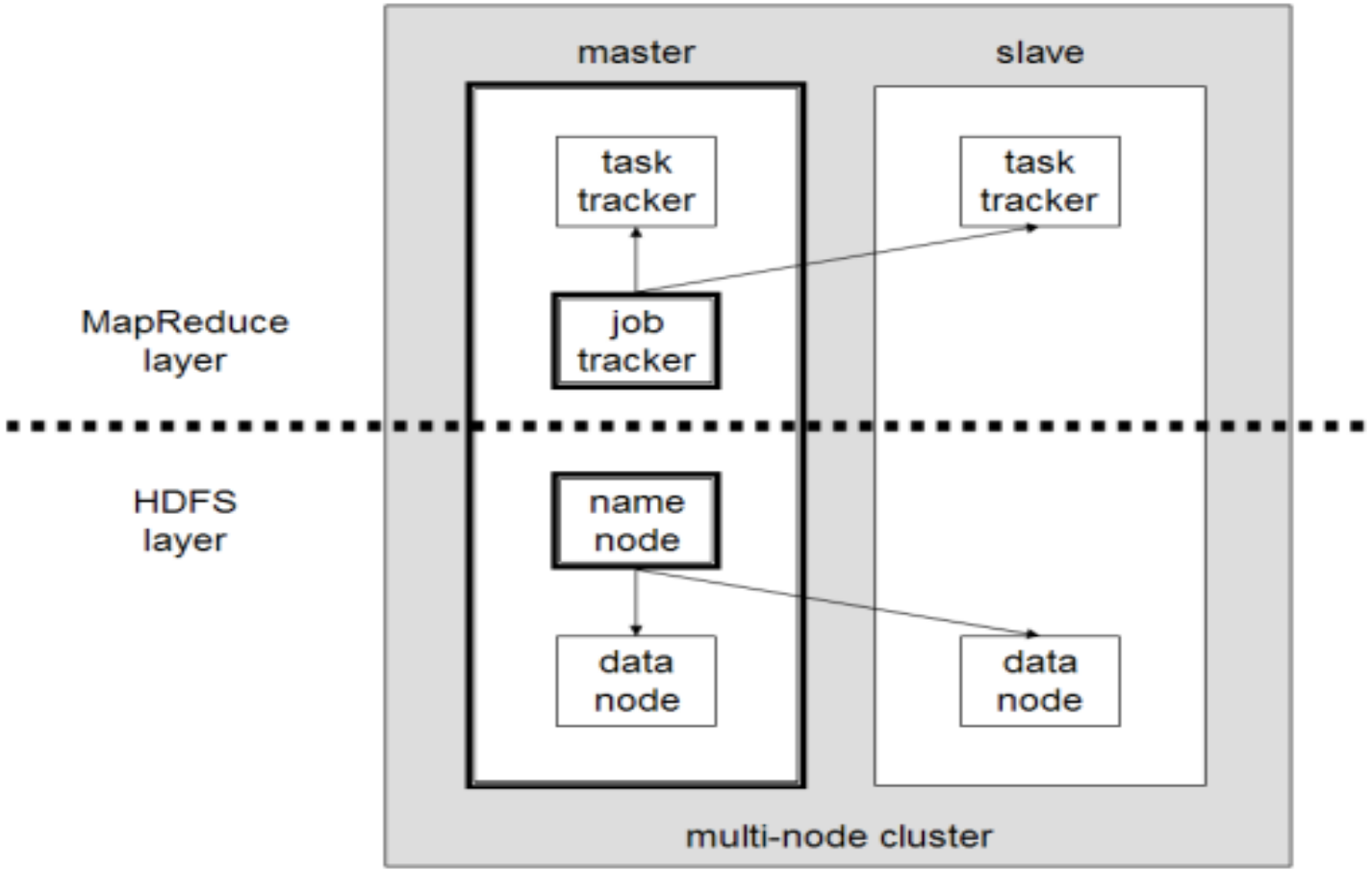
- Hadoop is the ***open source*** implementation of the google file system and Map-Reduce ***distributed computation***;
- Hadoop masks the complexities that are involved in working with distributed systems and provides a ***fault tolerant*** distributed file system (HDFS) as well as an API that makes programming in Hadoop a lot easier.
- HDFS: Hadoop Distributed File System is based upon Google File System.
- Hadoop is divided into ***HDFS*** and ***Map-Reduce***. HDFS is used for storing the data and Map-Reduce is used for processing data.
- Hadoop deals with ***files***; It is ***not a database***.

Hadoop HDFS Architecture

- The **Hadoop distributed file system (HDFS)** is a distributed, scalable, and portable file system written in Java for the Hadoop framework.
- HDFS has five services as follows:

1. Name Node
2. Data node
3. Job tracker
4. Secondary Name Node
5. Task Tracker

Hadoop HDFS Architecture: Services

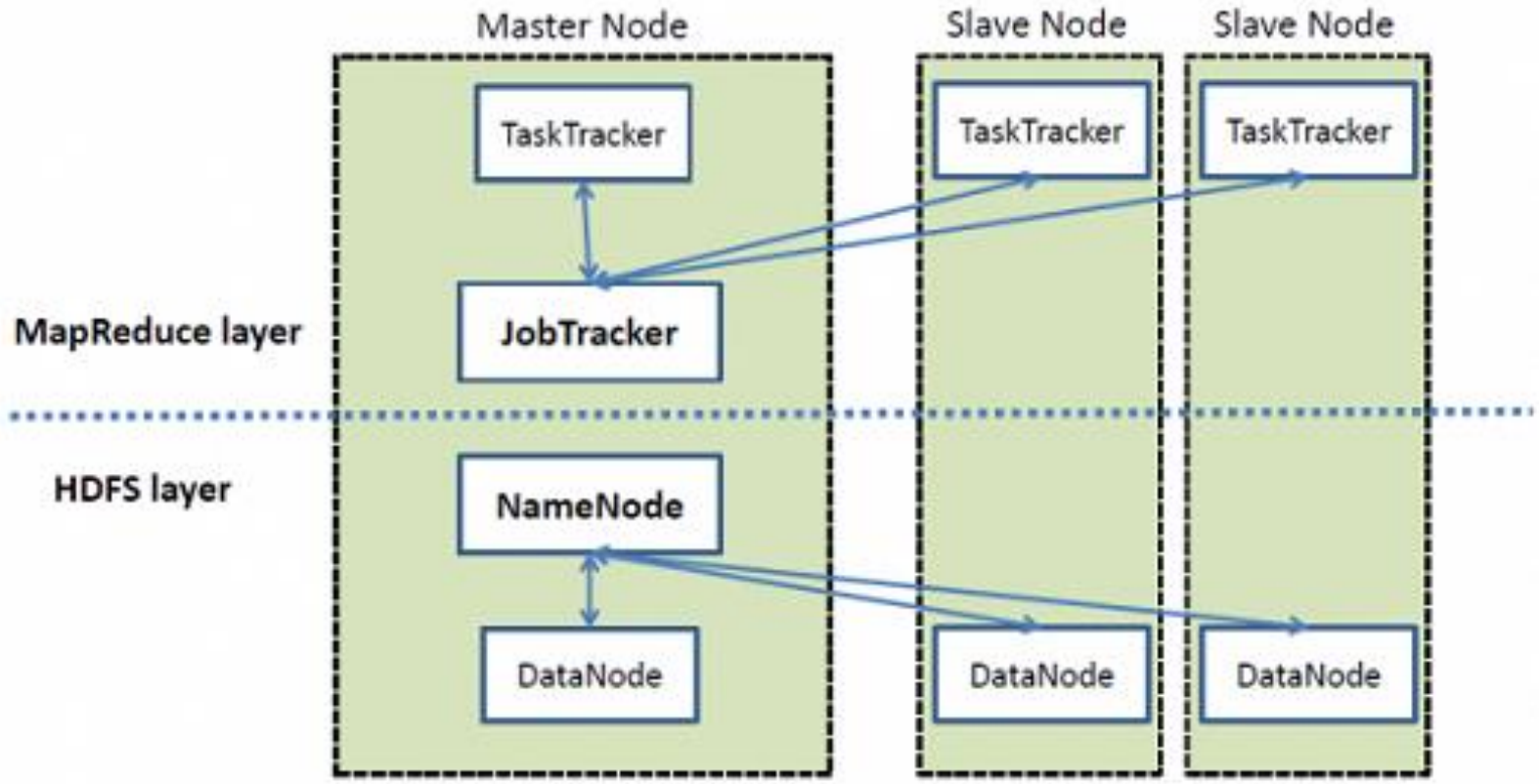


1. **Name Node**: HDFS consists of only one Name Node that is called the Master Node. The master node can track files, manage the file system and has the metadata of all of the stored data within it.
2. **Data Node**: This is also known as the slave node and it stores the actual data into HDFS.
3. **Job tracker**: Job tracker talks to the Name Node to know about the location of the data that will be used in processing.
4. **Secondary Name Node**: This is also known as the checkpoint Node. It is the helper Node for the Name Node.
5. **Task Tracker**: It is the Slave Node for the Job Tracker and it takes the task from the Job Tracker.

Source
https://s3.amazonaws.com/files.dezyre.com/images/blog/Hadoop+Architecture+Explained-What+it+is+and+why+it+matters/Hadoop+Architecture_OpenSource.png

Hadoop HDFS Architecture: Server Cluster

- Name Node is a master node and Data node is its corresponding Slave node and can talk with each other
- Enables ***multiple*** machines to perform computation of data;



Source
https://s3.amazonaws.com/files.dezyre.com/images/blog/Hadoop+Architecture+Explained-What+it+is+and+why+it+matters/Hadoop+Architecture_OpenSource.png

Hadoop HDFS Example

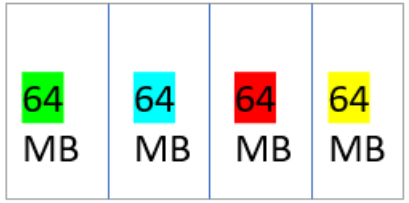
- Hadoop does two main things when a file is to be saved on HDFS:
  - It splits the file into chunks or blocks typically 64 to 128 mb data sizes;
  - Replicate each block of data (x3) and place them on a different data node;

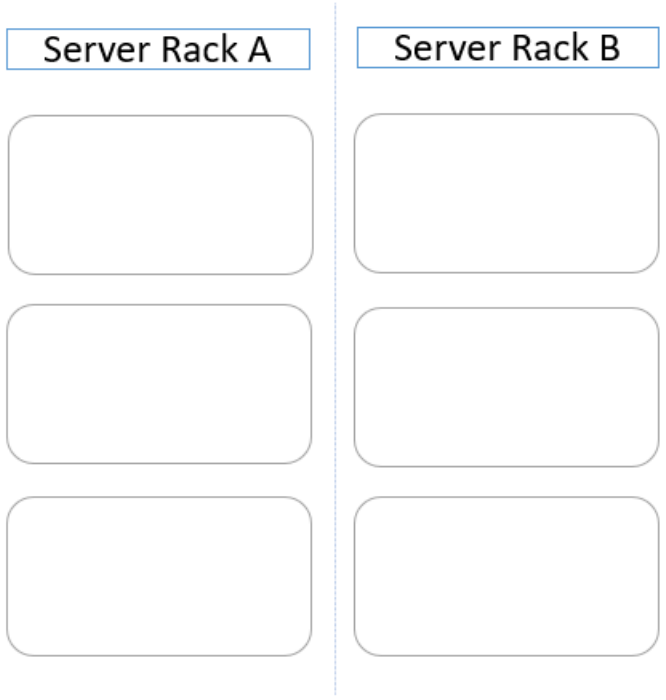
- With the replication factor of 3, the cluster has the power to choose among 3 servers, which one to use for computation.

- In case any node goes down, there is another copy of the data that is available, achieving ***fault tolerance***;

Hadoop HDFS Example

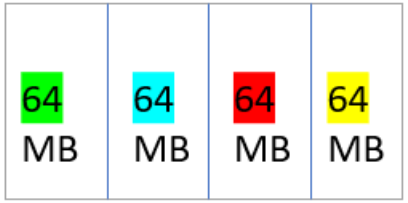
- It splits the file into chunks or blocks typically 64 to 128 MB in size

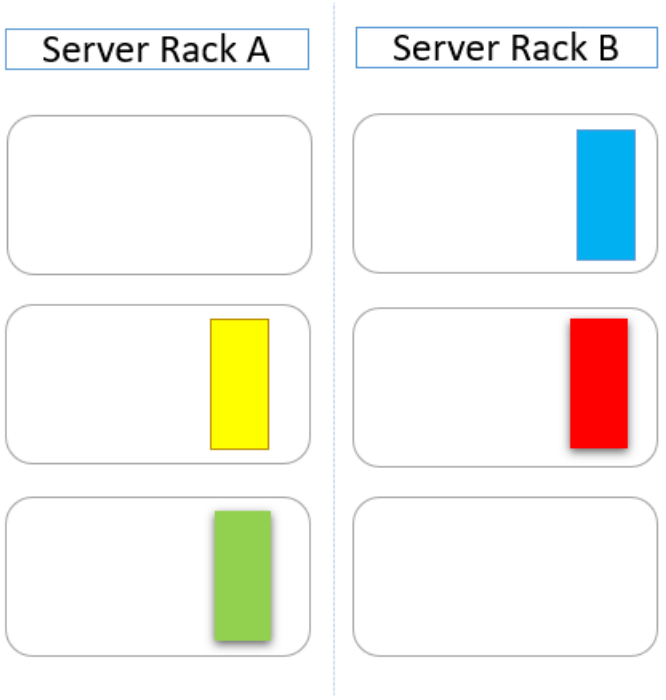




Hadoop HDFS Example

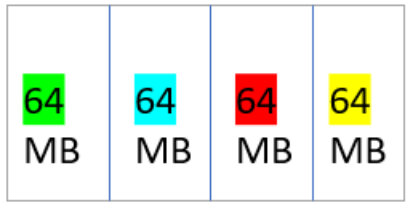
- Placing each block of data on a different data node

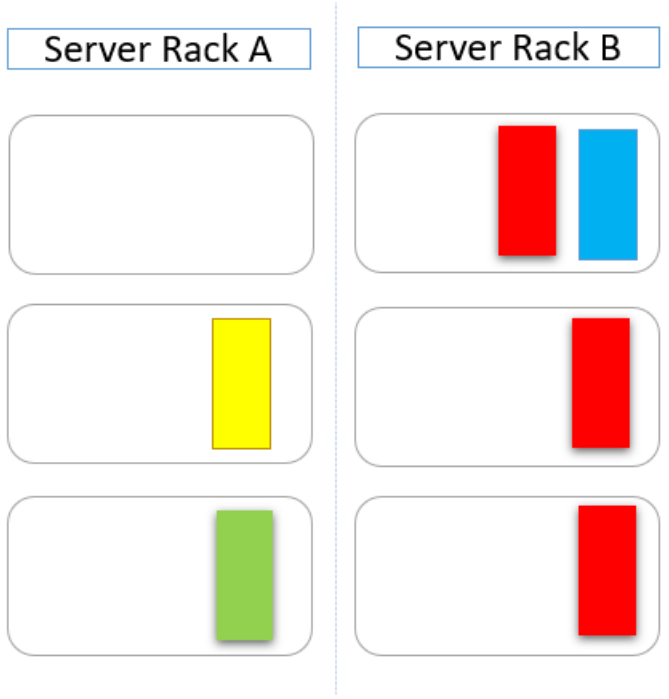




Hadoop HDFS Example

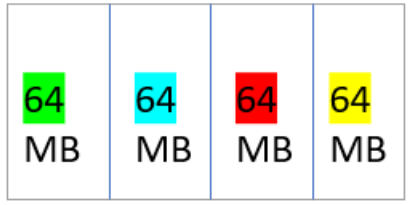
- Replicates each block to 3 nodes (data servers) by default

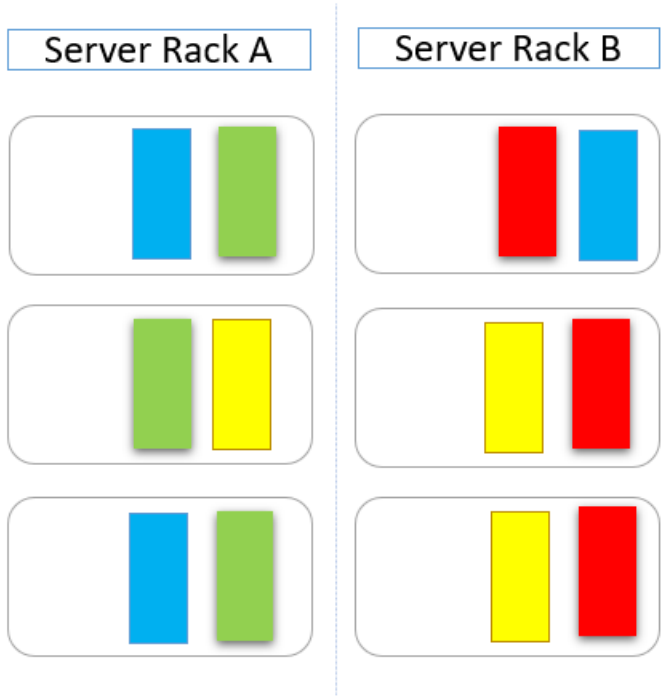




Hadoop HDFS Example

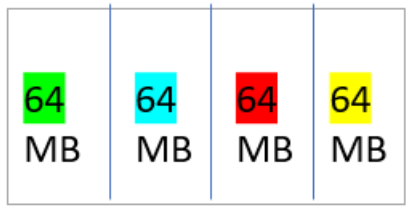
- Enables multiple machines to perform computation of data

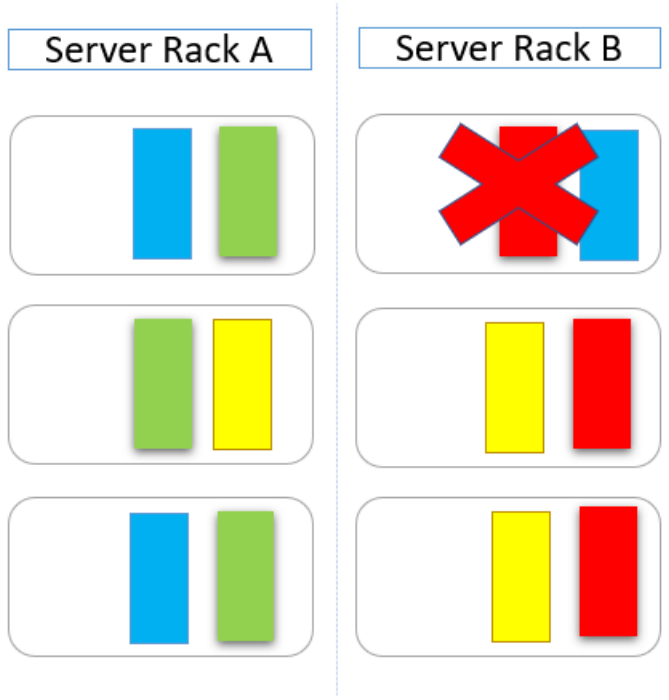




Hadoop HDFS Example

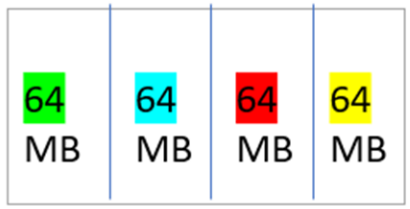
- Fault recovery: in case any node goes down, there's another copy of the data that is available, achieving fault tolerance

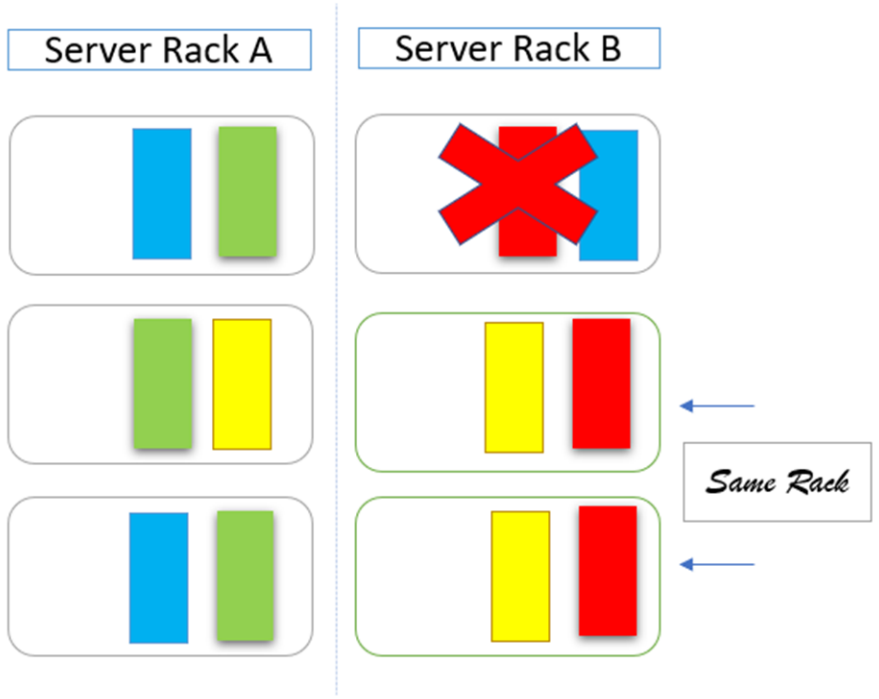




Hadoop HDFS Example

- It tries to serve data nodes on the same rack, to avoid network traffic between nodes, thus called: rack aware





Map-Reduce Computation

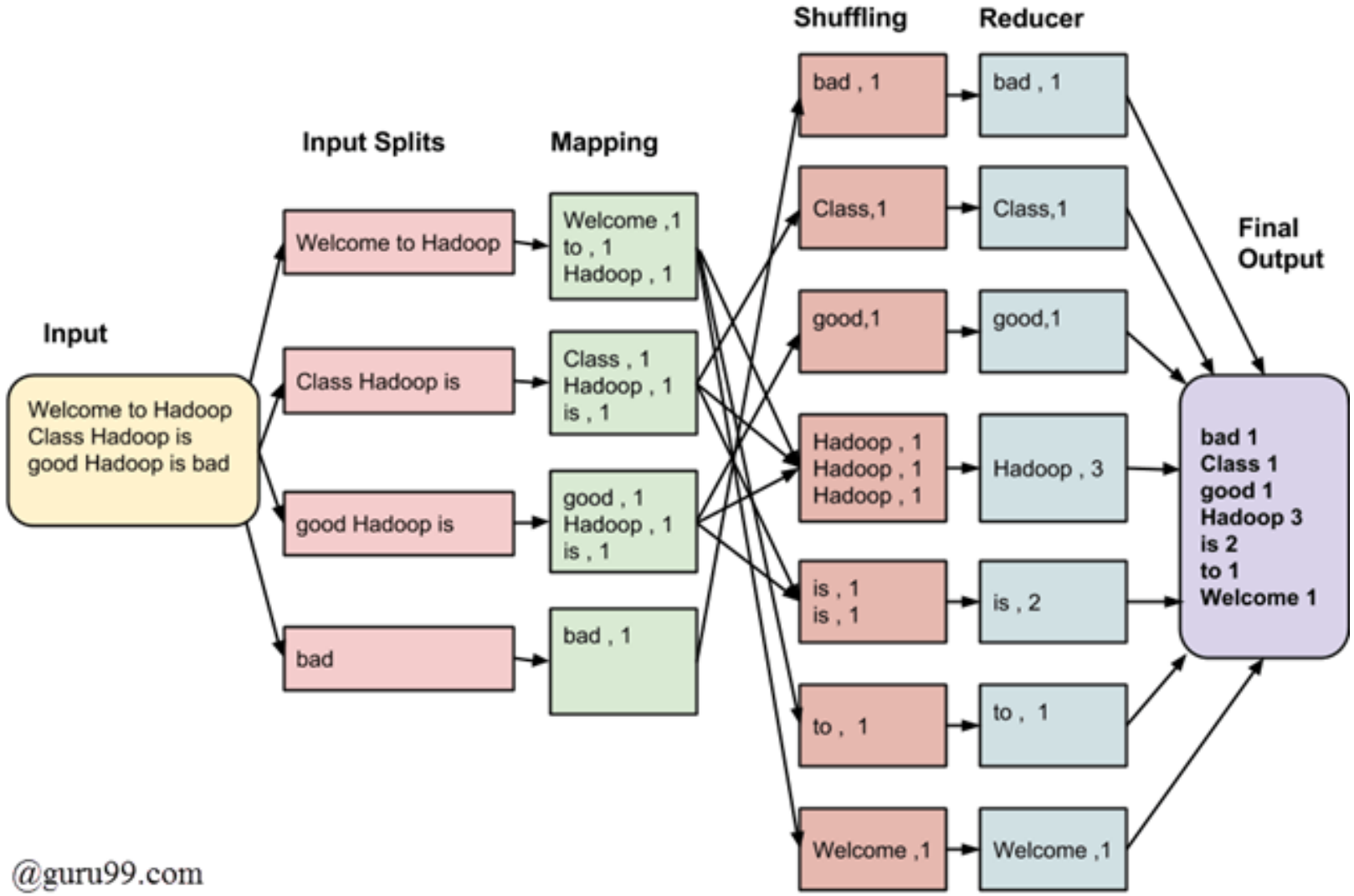
- Map-Reduce is a software framework used for ***parallelly*** computing or processing data that we store in HDFS;

- Framework where (Hive Sql) queries are interpreted;

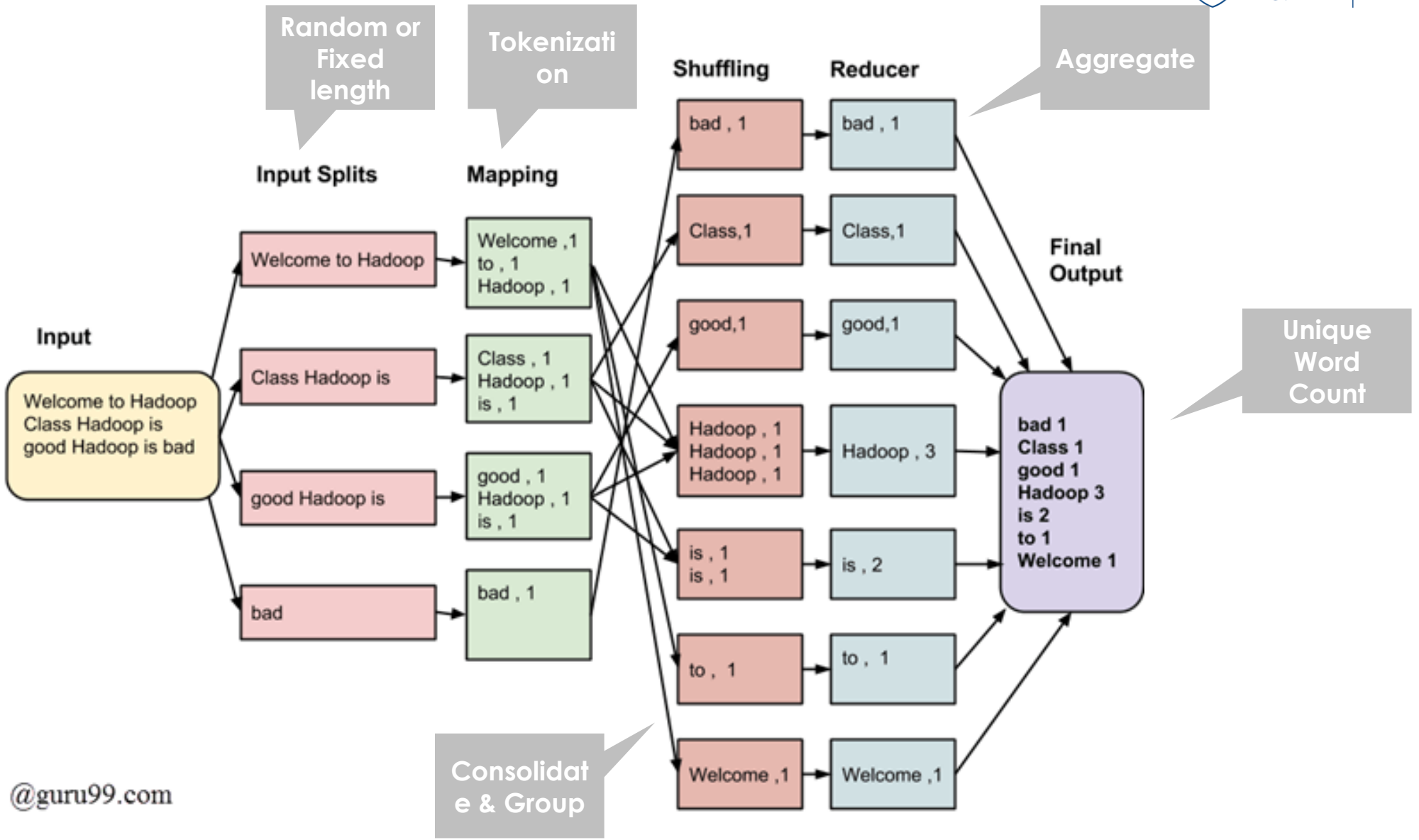
- A Map-Reduce job has three phases: (***divide & conquer method***)

    1. Map
    2. Shuffle & Sort
    3. Reduce

Map-Reduce Computation: Word Count

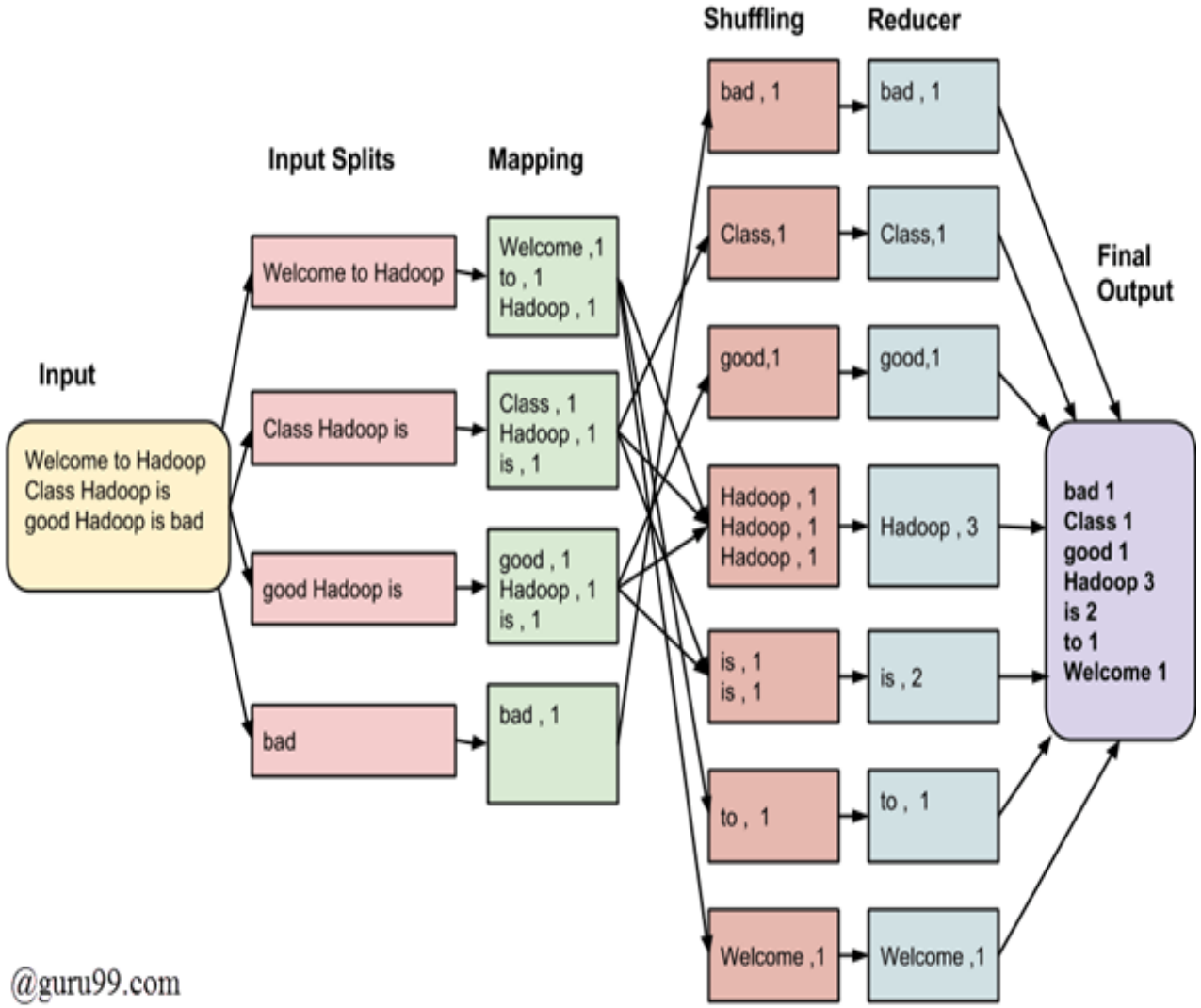


Map-Reduce Computation: Word Count



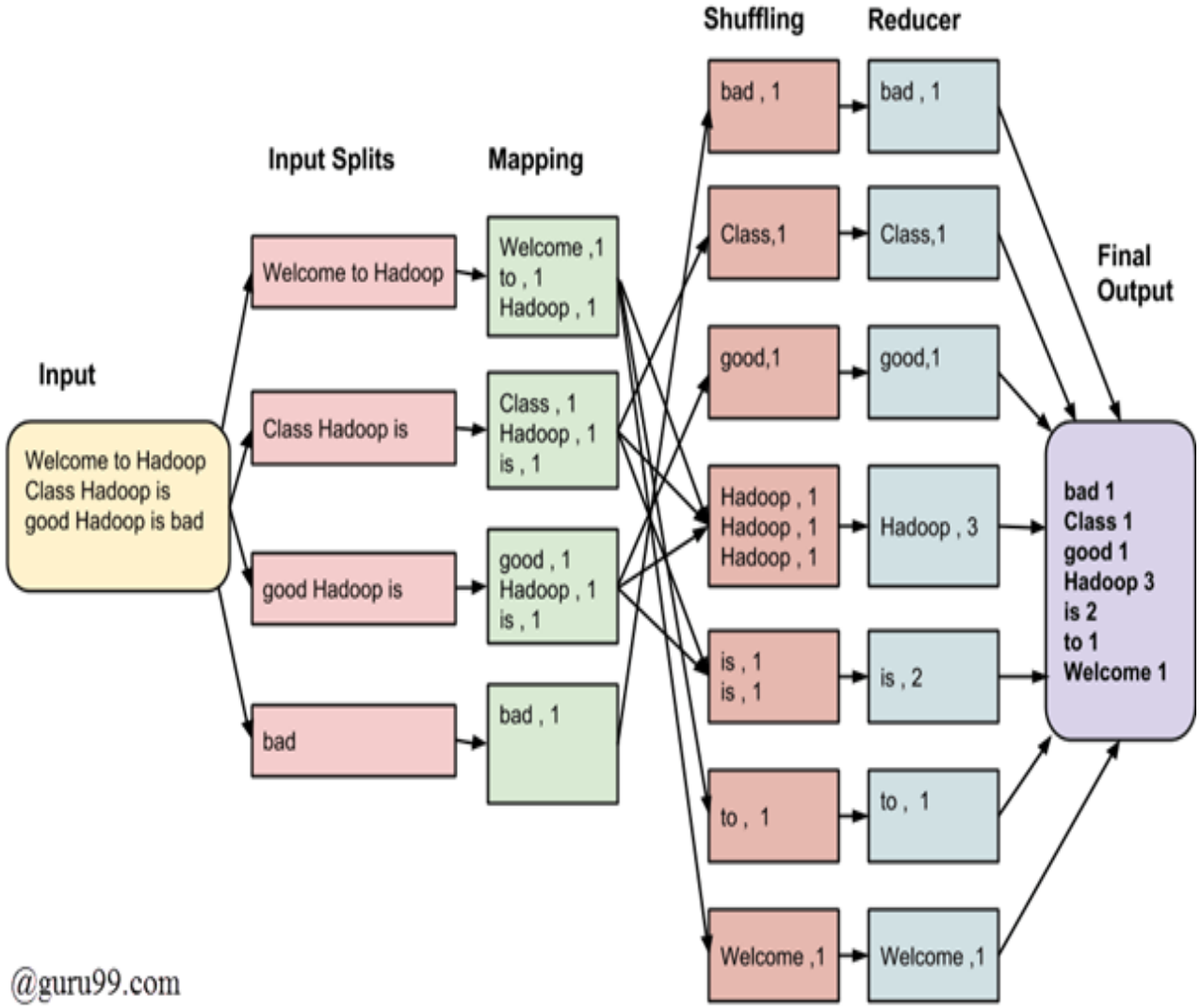
Map-Reduce Computation: Mappers

- A ***map-reduce job*** can use one or more mapper depending on the number of blocks the file spreads across.
- A ***mapper*** is assigned to each block. Here parallel distributed processing takes place given a file is split into blocks across multiple servers.
- Mappers take elements as a ***key*** and ***value*** and process them one at a time.



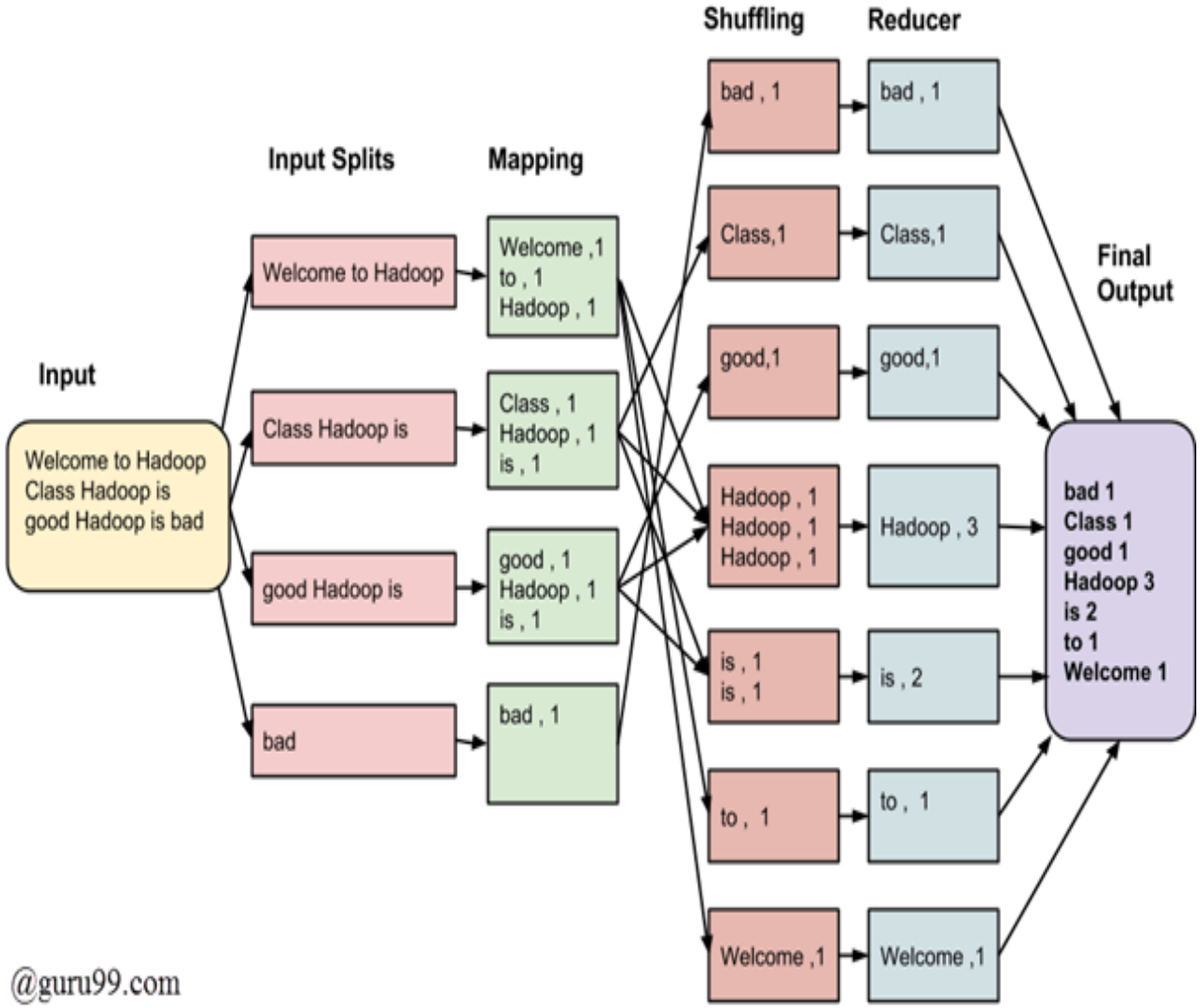
Map-Reduce Computation: Shuffle and Sort

- Maps are associated with the number of blocks of data required to read the input.
- ***Shuffle and sort*** has the task of sorting the key-value pairs.
- It uses the hash value of the key and splits keys into buckets according to the number of reducers.



Map-Reduce Computation: Reducer

- Mapper makes sense of data (calculation) while ***reducers*** get the key value pair as an output from mappers and ***aggregate*** the results together.
- No reducers are not dependent on the number of mappers. It can be configurated in the map-reduce job.
- Output of reducer is sent to a file directory in HDFS.



4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

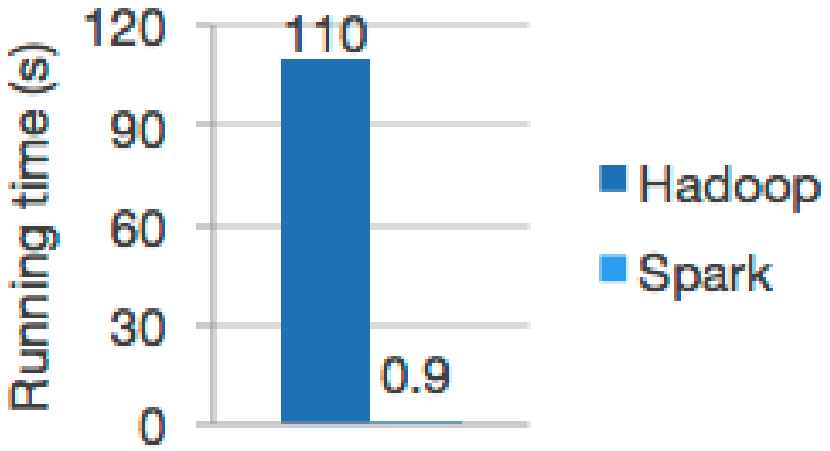
Spark Introduction

- **Apache Spark** is a unified analytics engine for large-scale data processing.
- It was the first unified analytics engine
- Spark simplifies working with data by supporting different languages (e.g. SQL, Java, Python, etc.)
- Spark brings data processing and analytics to one platform.

Why Spark?

- Speed: Spark runs workloads 100x faster than Hadoop.

- Apache Spark achieves high performance for both batch and streaming data, using ***in-memory computation*** compared with Hadoop's ***compute-on-disk***.



Spark Ease of Use

- Write applications quickly in Java, Scala, Python, R, and SQL;

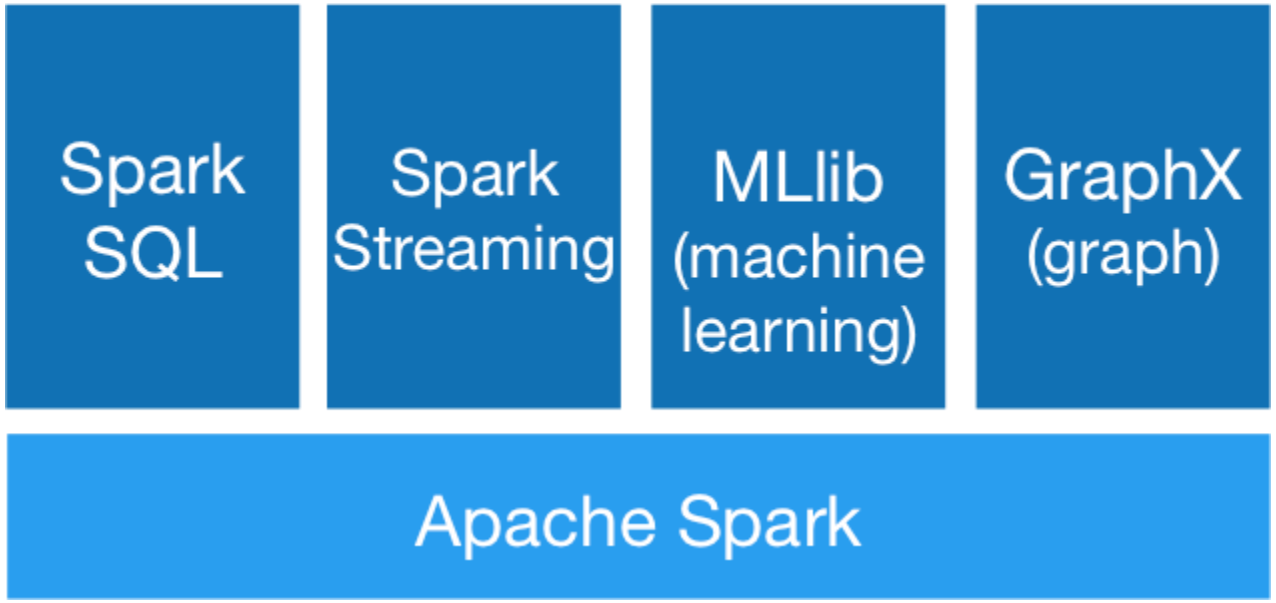
- Spark's Python DataFrame API Read JSON files with automatic schema inference;

https://www.codegrepper.com/code-examples/python/spark+read+json+schema

```
# Read JSON files with automatic schema inference

df = spark.read.json("logs.json")
df.where("age > 21").select("name.first").show()
```

Spark Generality

- Spark powers a stack of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, Spark Streaming, interactive dashboarding, and advanced analytics.



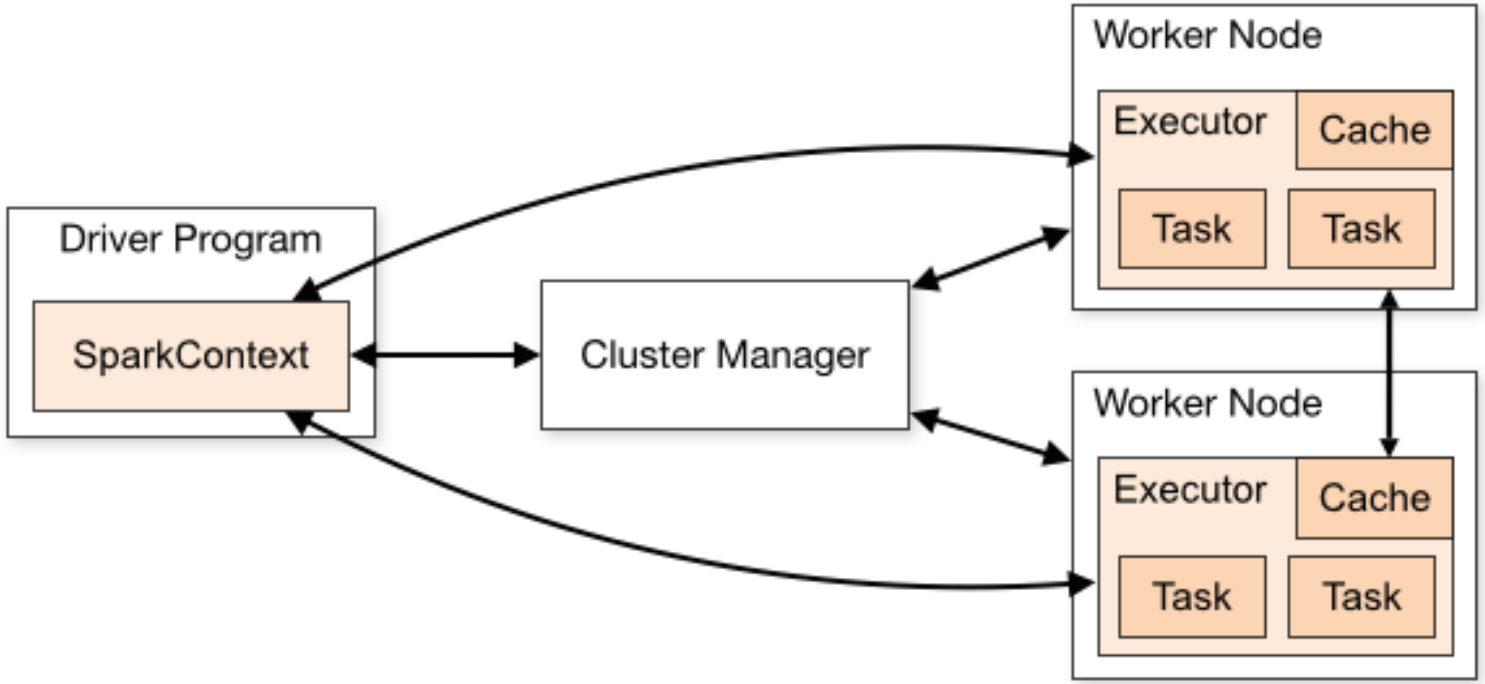
Spark Runs Everywhere

- Spark can run on Hadoop HDFS, Apache Mesos, Kubernetes, standalone, or in the cloud.

- It can access diverse data sources.

Spark Architecture

- Spark applications run as independent sets of processes on a cluster, coordinated by the ***SparkContext*** object in your main program (called the driver program).
- Specifically, to run on a cluster, the SparkContext can connect to several types of cluster managers (either Spark's own standalone cluster manager, Mesos or YARN), which allocate resources across applications.



1. **Once** connected, Spark acquires ***executors*** on worker nodes/servers in the cluster, which are processes that run computations and store data for your application.
2. **Next**, it sends your application code (Java JAR, Scala or Python passing via SparkContext) to the executors.
3. **Finally**, SparkContext sends tasks to the executors to run.

Spark Computation: Color Ball Count

- Imagine if you have to count the number of balls of each color in this bag;
- You are given the responsibility of performing this task, but you can call your friends (***executors***) over for help;
- How many friends will you call?
- Why?

Spark RDD

Resilient **D**istributed **D**atasets:

- **Definition**: a collection of elements partitioned across the nodes of the cluster that can be operated on in parallel.
- Example: consider the previous example , if a friend leaves before the completion of your task?
- How would you compute the count of balls: Recompute the whole set or only the set that was taken up by that left individual?
- RDD makes this re-work easier, when fault happens.

Spark Summary

- RDD is ***distributed*** and stored in various clusters on your system;
- Each ***executor (data node server)*** works on their part of the data, the results are then aggregated and send back to the ***driver (name node server)***;
- Dataframe in Spark inherits RDD properties (***resilient and distributed***) and metadata;
- ***SparkSQL*** commands execute against dataframes (table alike);
- Spark is not a dataframe, it's a ***in-memory compute engine*** that can read from databases;
- Data is ***ephemeral***: you never lose your data even when spark is down;
- Dataframe aggregation takes ***shorter*** time;

4.2 Contemporary Reasoning Systems (Big Data)

4.2.1 Big Data Applications

4.2.2 Development Eco-systems (Hadoop & Spark)

4.2.3 Exercise

[Exercise] Contemporary Reasoning Systems (Big Data)

Analyze the following case study: Google Flu Trends (GFT)

In 2008, researchers from Google explored this potential, claiming that they could “nowcast” the flu based on people’s searches. The essential idea, published in a paper in Nature, was that when people are sick with the flu, many search for flu-related information on Google, providing almost instant signals of overall flu prevalence. The paper demonstrated that search data, if properly tuned to the flu tracking information from the Centers for Disease Control and Prevention, could produce accurate estimates of flu prevalence two weeks earlier than the CDC’s data — turning the digital refuse of people’s searches into potentially life-saving insights.

[Exercise]

Google Flu Trends

- And then, GFT failed — and failed spectacularly — missing at the peak of the 2013 flu season by 140 percent. When Google quietly euthanized the GFT service, it turned the poster child of big data into the poster child of the foibles of big data.
- Google Flu Trends (GFT): https://youtu.be/lEDt89eQ64o

Answer questions:

- Why did GFT fail?
- What could Google have done differently?
- What does this teach us about analyzing Big data?

4.3 Building Machine Reasoning System [Workshop]

4.3.1 Big Data Machine Learning & Reasoning System

Special thanks to Geet Jethwani (A0215395B) for his contribution.

4.3.2 Workshop Submission

4.3 Building Machine Reasoning System [Workshop]

4.3.1 Big Data Machine Learning & Reasoning System

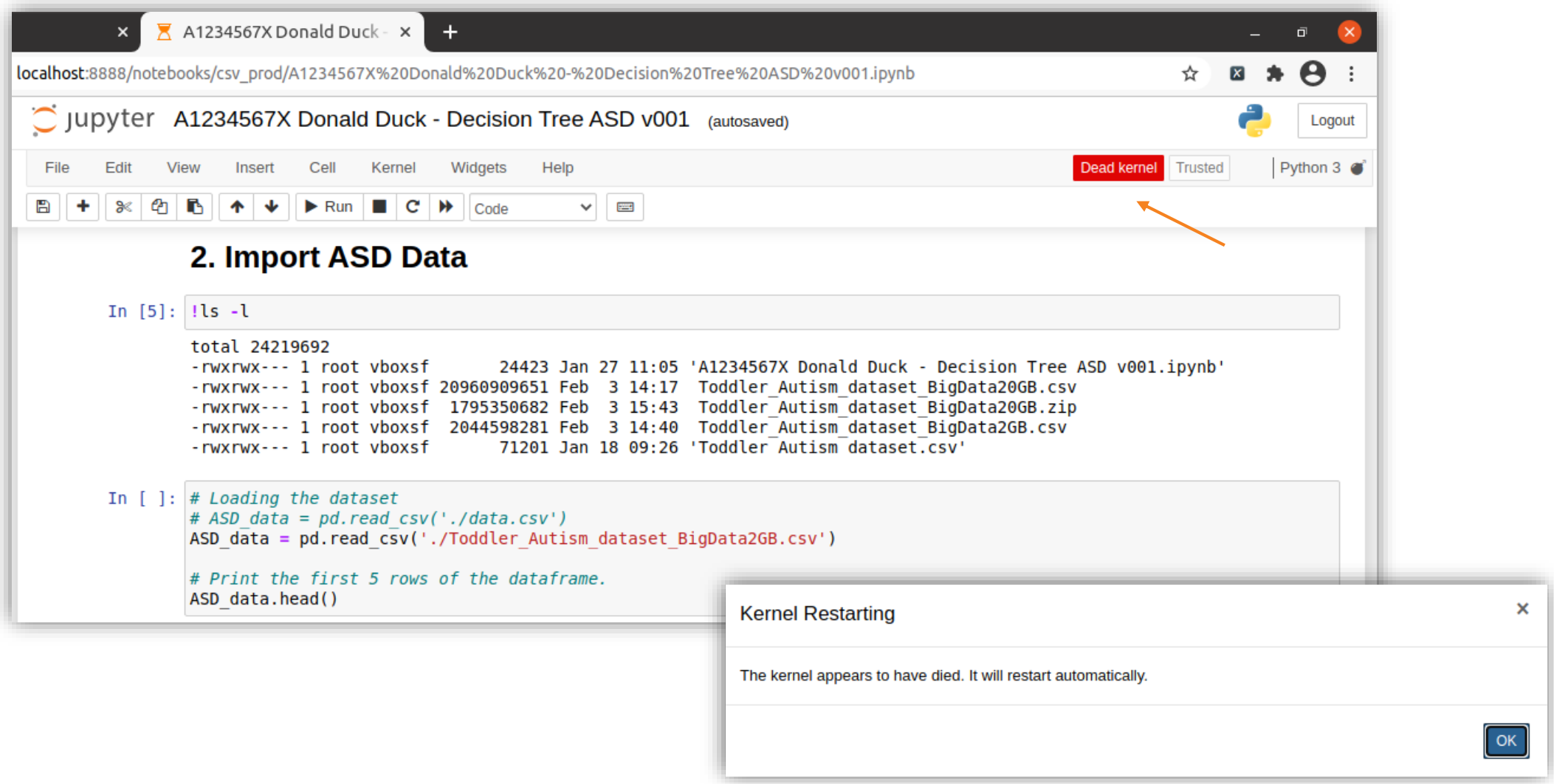
Special thanks to Geet Jethwani (A0215395B) for his contribution.

4.3.2 Workshop Submission

Learning Outcome

- Import data into big data platform.
- Apply SQL knowledge and RDBMS concepts over Spark.
- Apply Data Analysis techniques and Spark SQL queries on autism dataset .
- Finally, we are to build an ASD diagnosis solution using Spark ML big data algorithm library.

Issues with Local Machine (read 2GB csv in Jupyter)



Tools

- Databricks (import data/files);
- Spark SQL on Databricks (toddler autism case);
- SparkML (machine learning) library to build decision tree to detect autism;
- Anaconda with pyspark installed (if needed);

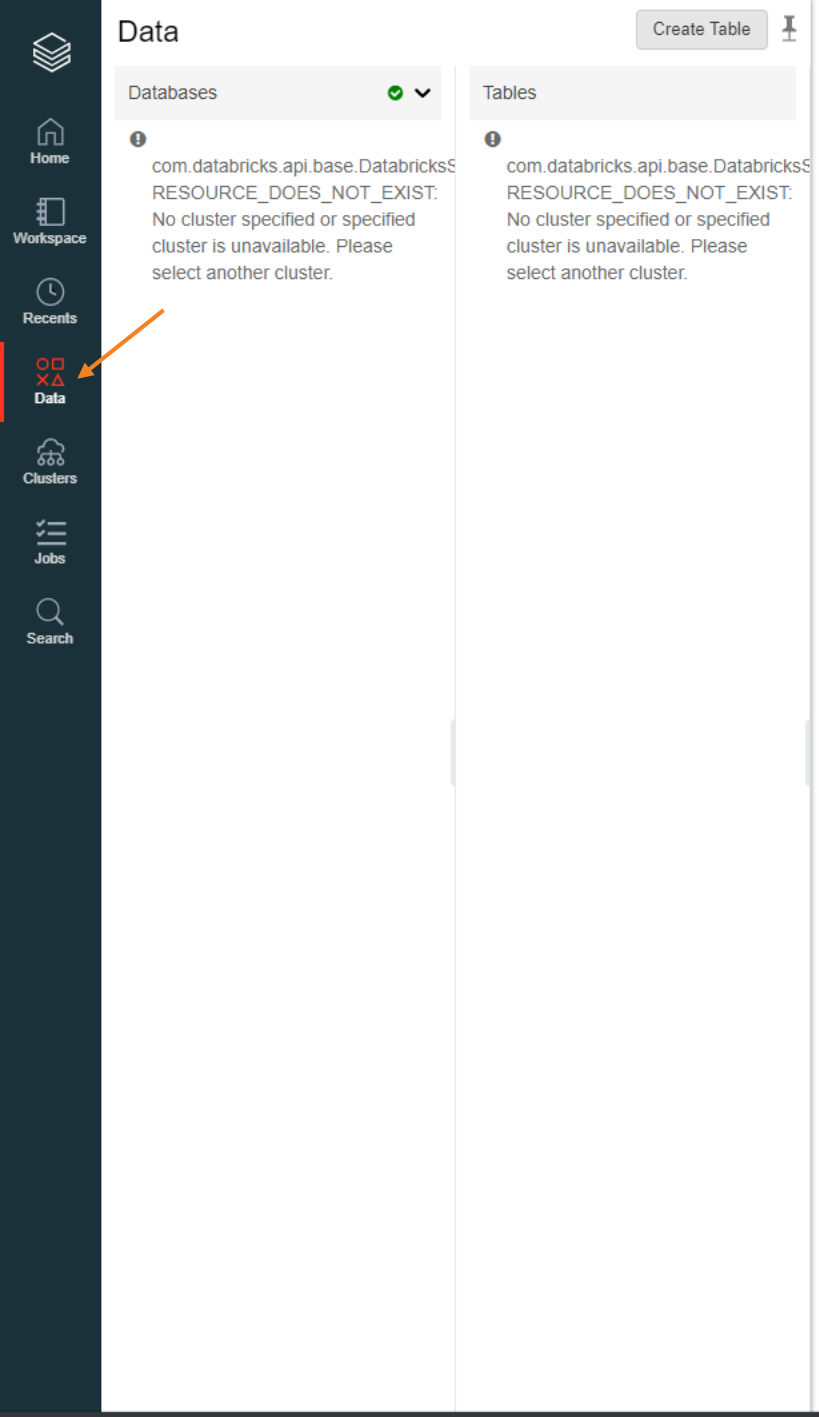
Additional Setup (if needed):

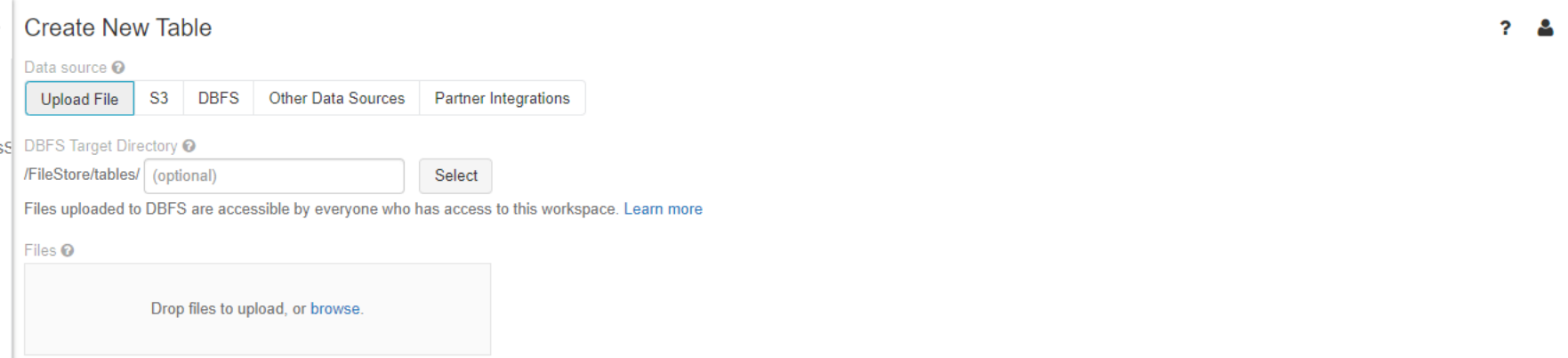
- Java 8 required on system;
- Set up JAVA_HOME and HADOOP_HOME System Variable;

[Workshop] Big Data Query

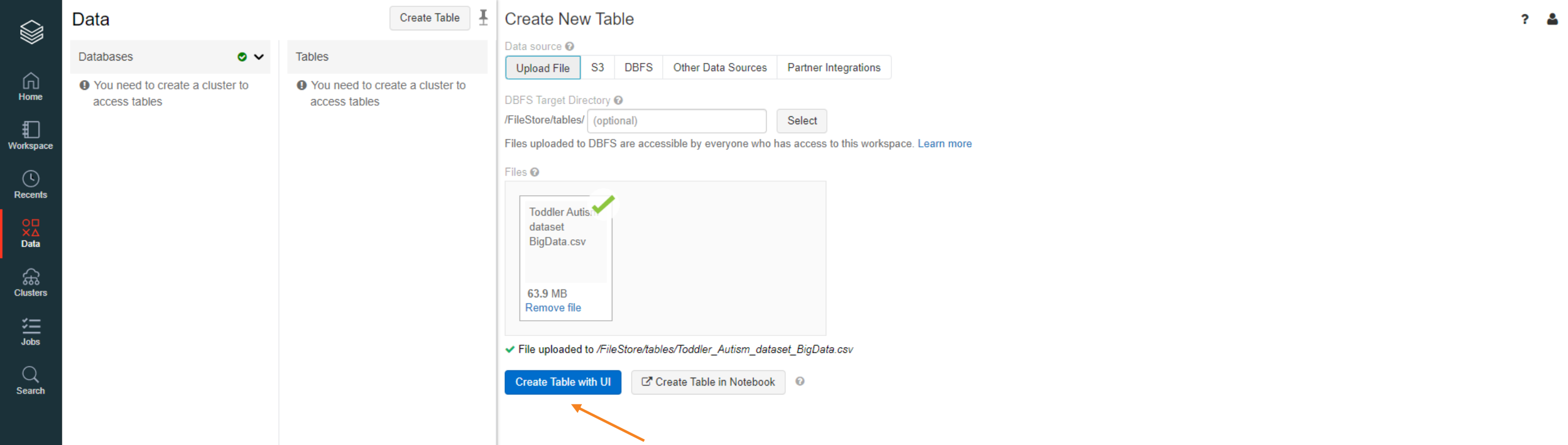
- Log in Community Edition log in portal: https://community.cloud.databricks.com/login.html

- Import & Query toddler ASD data;





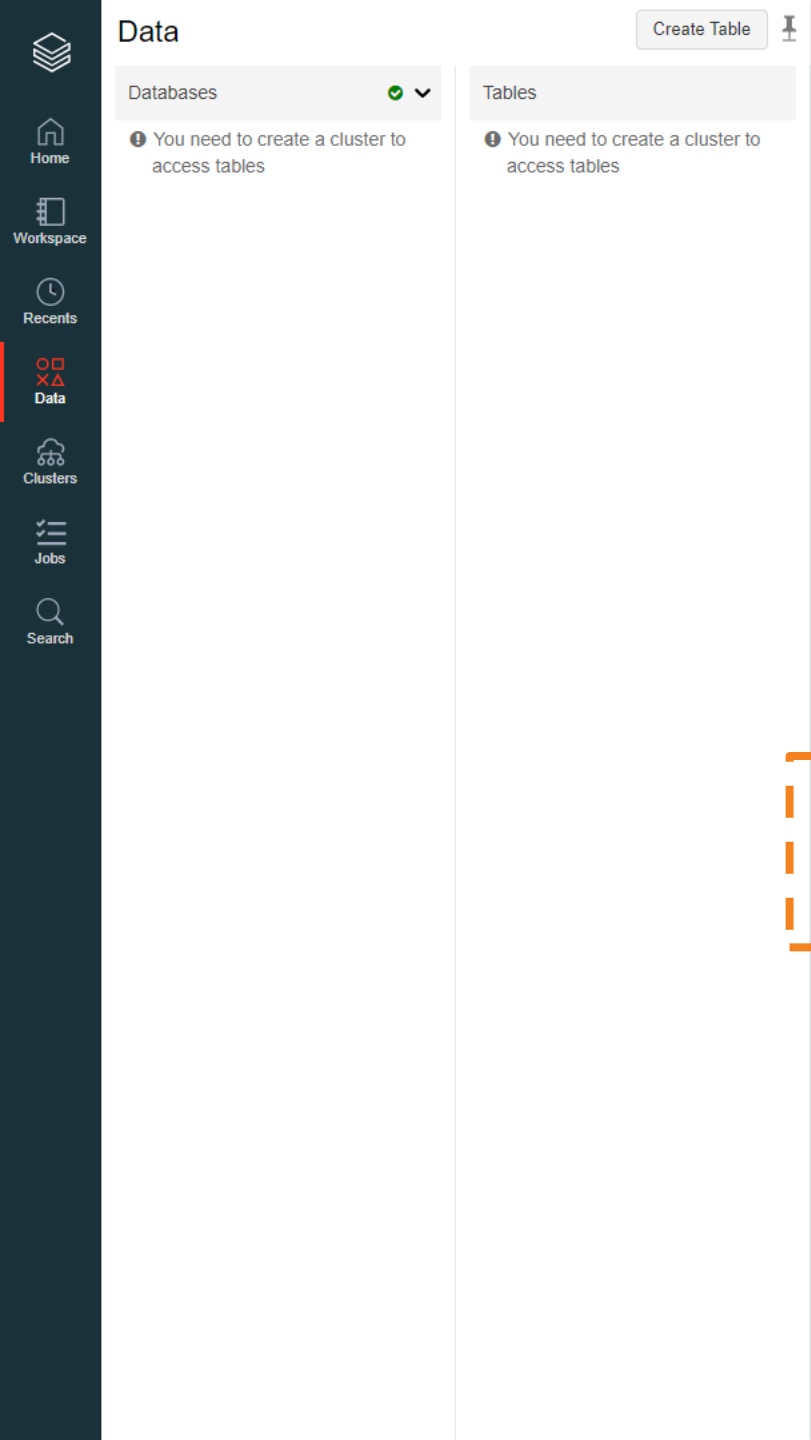
Import Toddler Autism dataset.csv via "Upload File"

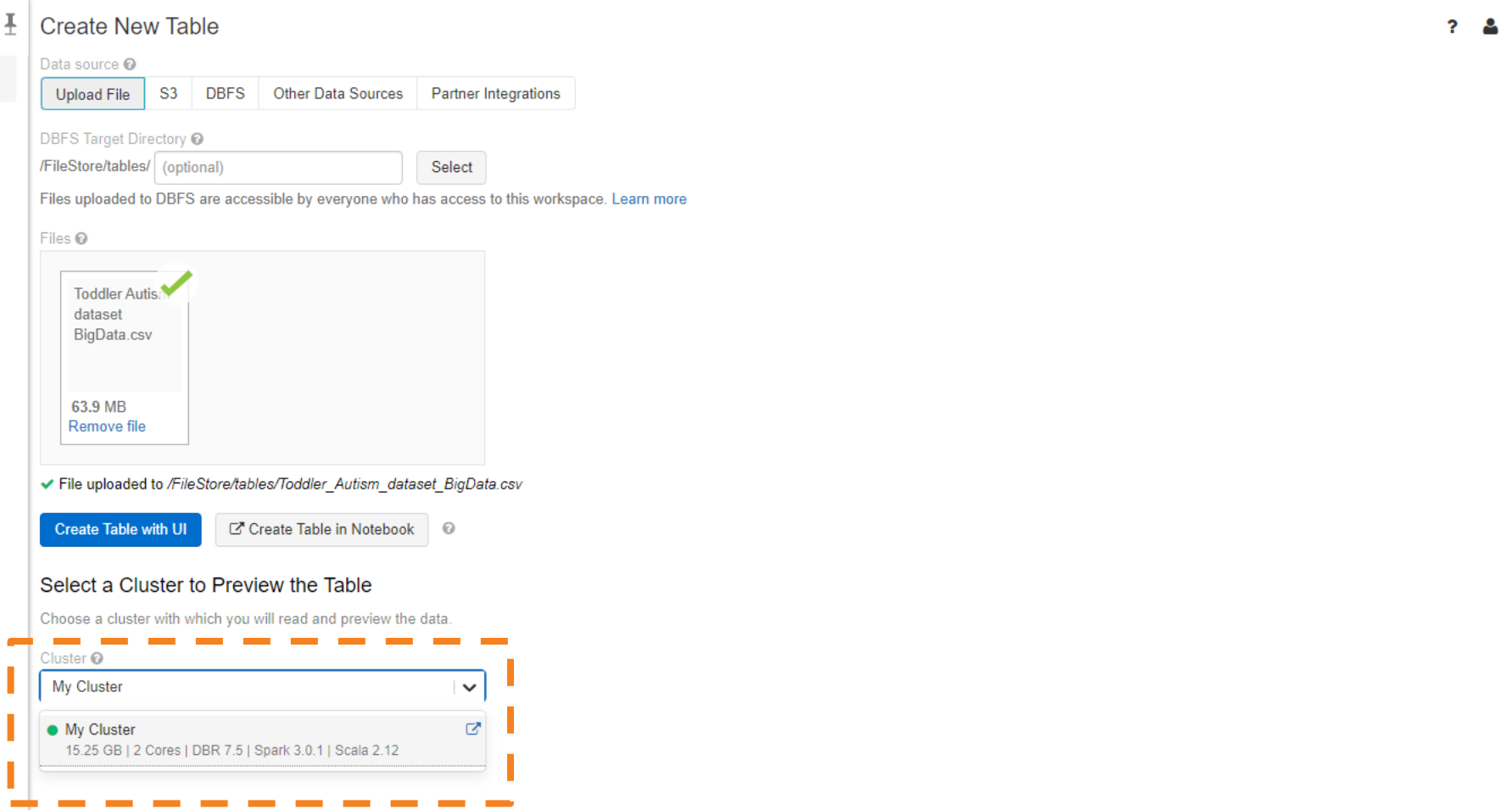


File uploaded to

/FileStore/tables/Toddler_Autism_dataset_BigData.csv

Option 1: Create table with UI





Start a new cluster if needed.

Or use Option 2 to create table

Data

Create Table

Databases

You need to create a cluster to access tables

Tables

You need to create a cluster to access tables

Create New Table

Files uploaded to DBFS are accessible by everyone who has access to this workspace. Learn more

Files

Toddler Autism dataset BigData.csv

63.9 MB

Remove file

File uploaded to */FileStore/tables/Toddler_Autism_dataset_BigData.csv*

Create Table with UI

Create Table in Notebook

Select a Cluster to Preview the Table

Choose a cluster with which you will read and preview the data.

Cluster

My Cluster

Preview Table

Specify Table Attributes

Specify the Table Name, Database and Schema to add this to the data UI for other users to access

Table Name

toddler_autism_dataset_big

Create in Database

default

File Type

CSV

Column Delimiter

,

First row is header

Infer schema

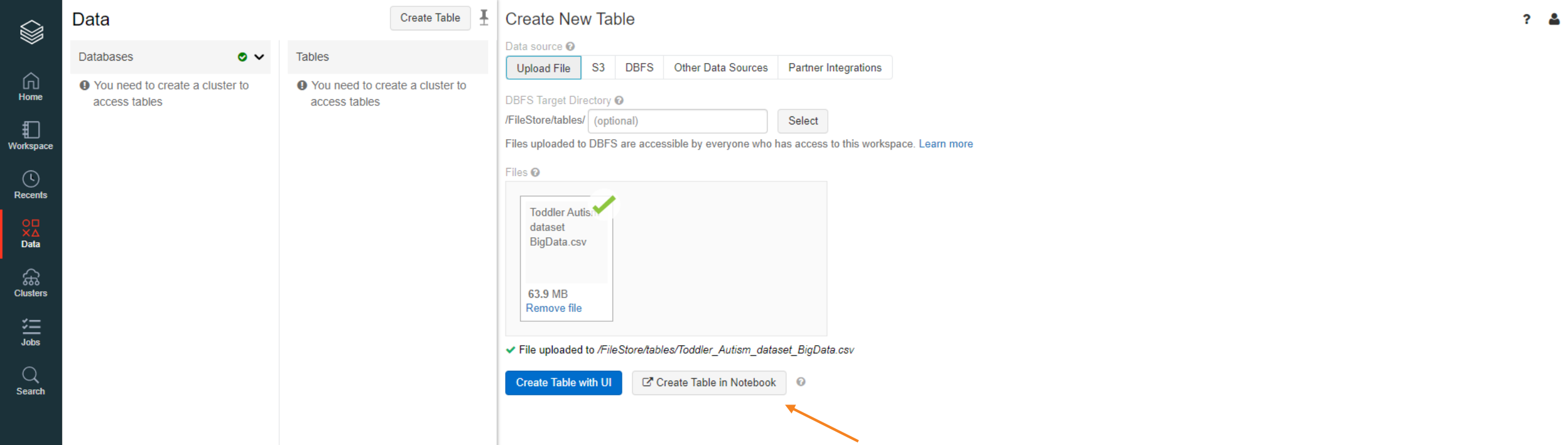
Multi-line

Create Table

Create Table in Notebook

Table Preview

A1	A2	A3	A4	A5	A6
INT	INT	INT	INT	INT	INT
0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	0
1	1	1	1	1	1
1	1	0	1	1	1
1	1	0	0	1	1



Option 2: Create table in Notebook

Workspace

Users

issgz.nus@gmail.com

issgz.nus@gmail.com

2021-02-02 - DBFS Example

ISDemoNotebook

ISDemoNotebookASD

Home

Workspace

Recents

Data

Clusters

Jobs

Search

2021-02-02 - DBFS Example (Python)

Detached | File | Edit | View: Standard | Permissions | Run All | Clear | Publish | Comments | Experiment | Revision history

Cmd 1

Overview

This notebook will show you how to create and query a table or DataFrame that you uploaded to DBFS. DBFS is a Databricks File System that allows you to store data for querying inside of Databricks. This notebook assumes that you have a file already inside of DBFS that you would like to read from.

This notebook is written in **Python** so the default cell type is Python. However, you can use different languages by using the `%LANGUAGE` syntax. Python, Scala, SQL, and R are all supported.

Cmd 2

```
# File location and type
file_location = "/FileStore/tables/Toddler_Autism_dataset_BigData.csv"
file_type = "csv"

# CSV options
infer_schema = "true"
first_row_is_header = "true"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(df)
```

Update:

CSV options

From 'false' to 'true'

(3) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [A1: integer, A2: integer ... 15 more fields]

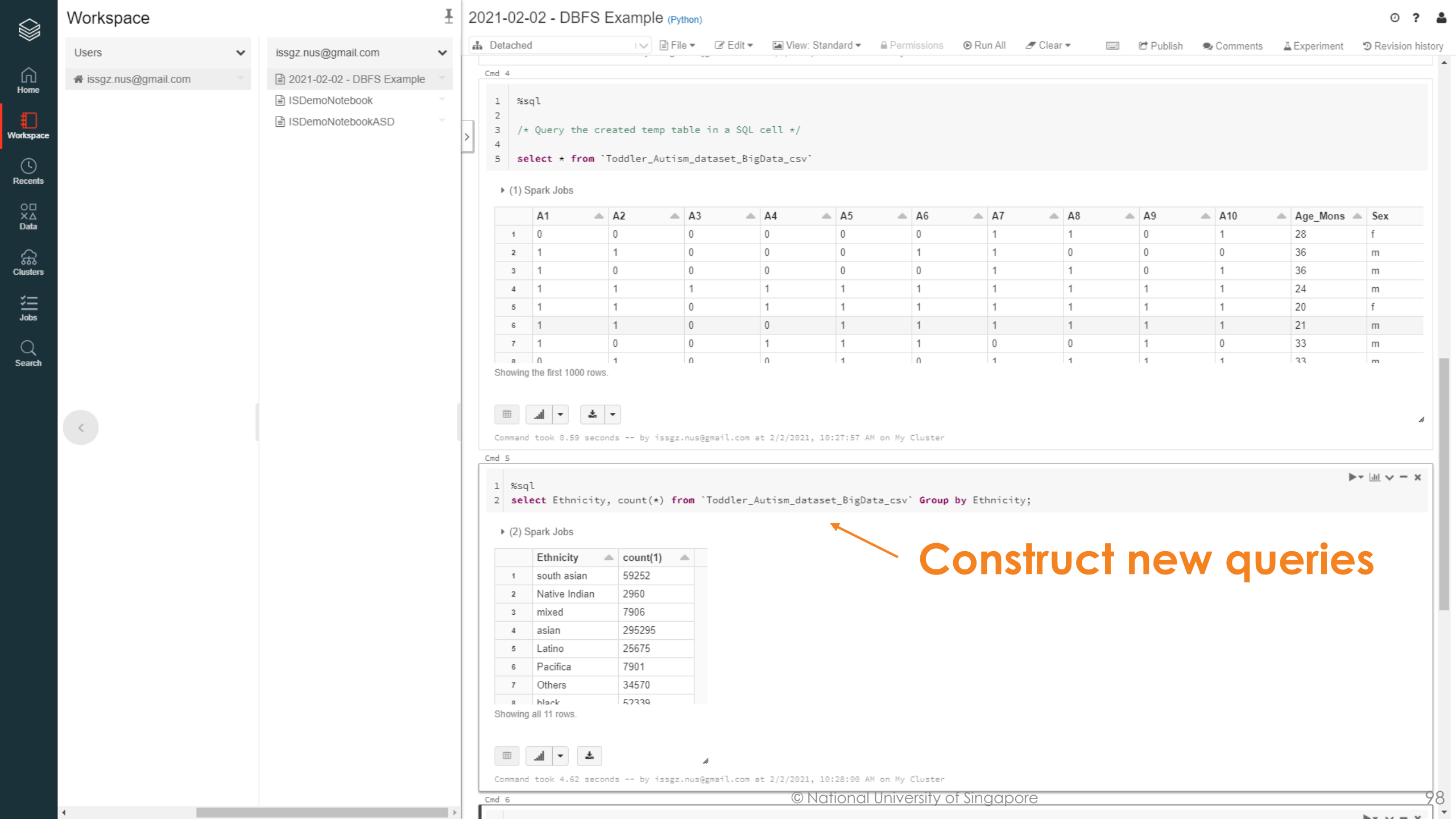
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Sex
1	0	0	0	0	0	0	1	1	0	1	28	f
2	1	1	0	0	0	1	1	0	0	0	36	m
3	1	0	0	0	0	0	1	1	1	1	36	m
4	1	1	1	1	1	1	1	1	1	1	24	m
5	1	1	0	1	1	1	1	1	1	1	20	f
6	1	1	0	0	1	1	1	1	1	1	21	m
7	1	0	0	1	1	1	0	0	1	0	33	m
8	0	1	0	0	1	0	1	1	1	1	33	m

Showing the first 1000 rows.

Command took 19.18 seconds -- by issgz.nus@gmail.com at 2/2/2021, 10:12:40 AM on My Cluster

Then execute Notebook

Cmd 3



Workspace

Users

issgz.nus@gmail.com

issgz.nus@gmail.com

2021-02-02 - DBFS Example

ISDemoNotebook

ISDemoNotebookASD

2021-02-02 - DBFS Example (Python)

Detached | File | Edit | View: Standard | Permissions | Run All | Clear | Publish | Comments | Experiment | Revision history

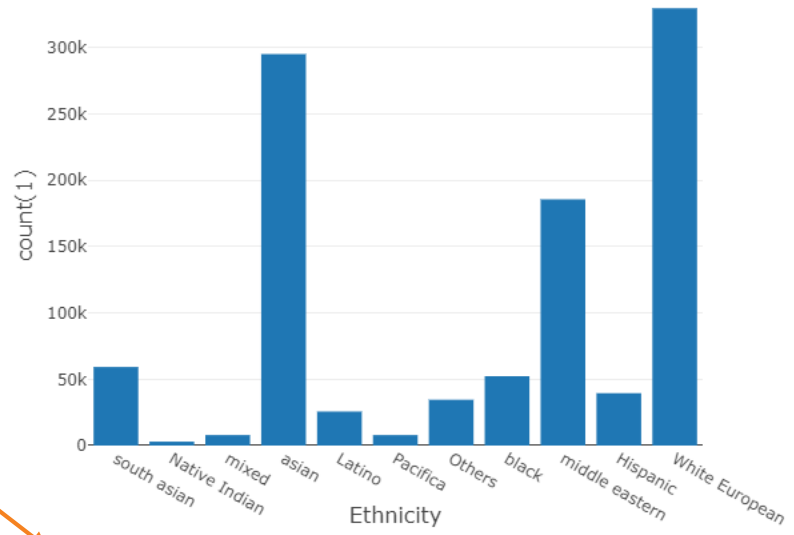
Command took 0.59 seconds -- by issgz.nus@gmail.com at 2/2/2021, 10:27:57 AM on My Cluster

Cmd 5

```
%sql
select Ethnicity, count(*) from `Toddler_Autism_dataset_BigData_csv` Group by Ethnicity;
```

(2) Spark Jobs

Visualization



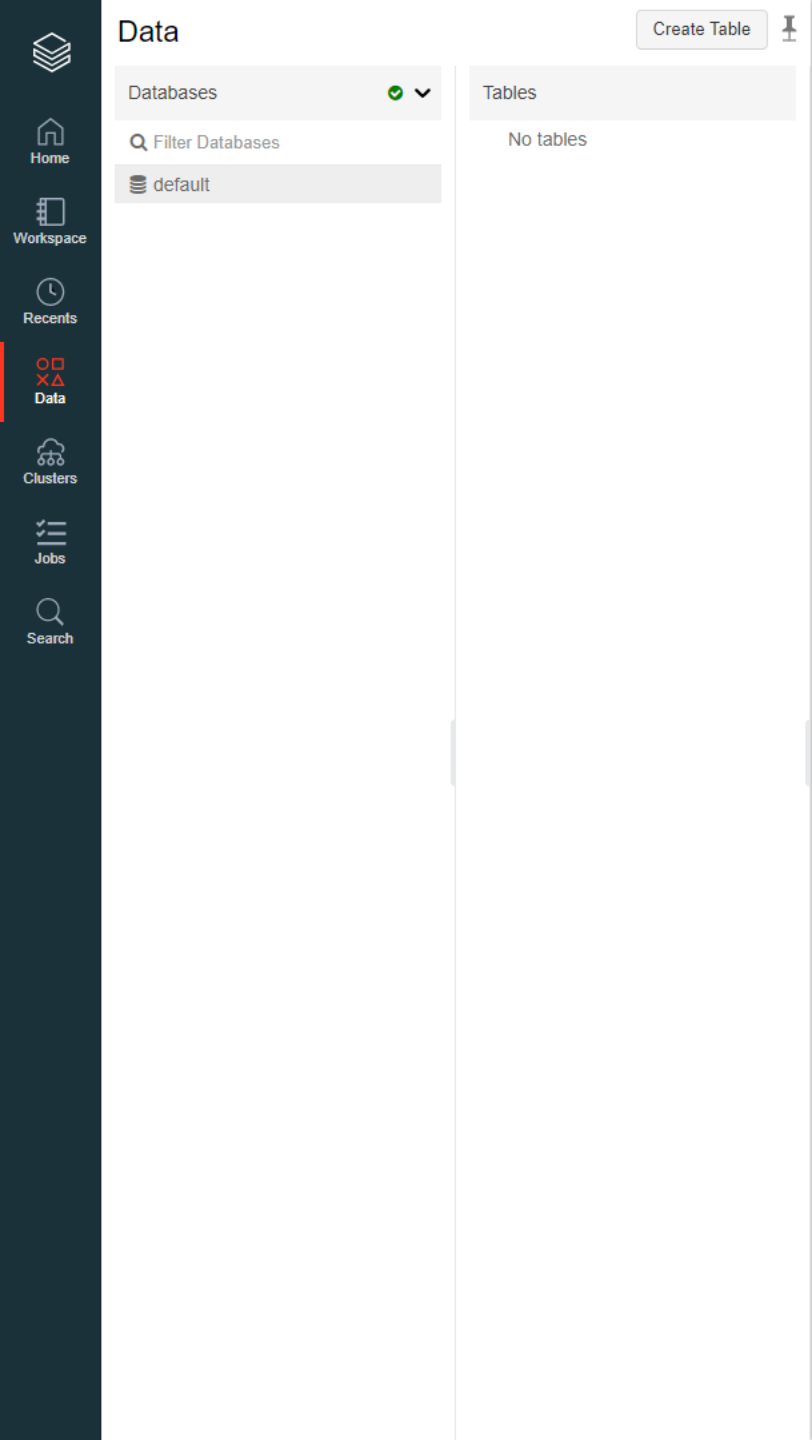
Plot Options...

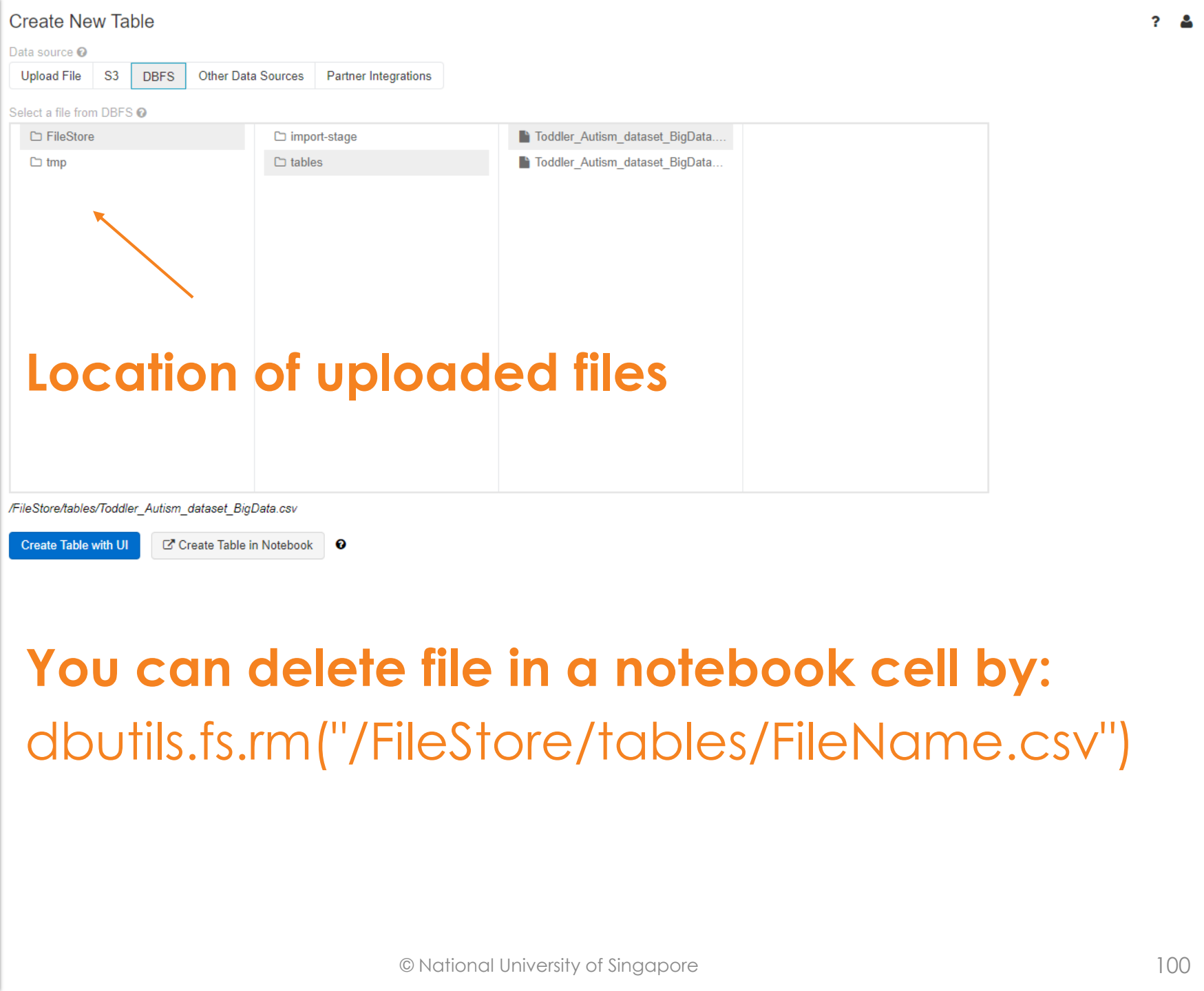
Command took 4.62 seconds -- by issgz.nus@gmail.com at 2/2/2021, 10:28:00 AM on My Cluster

Cmd 6

```
# With this registered as a temp view, it will only be available to this particular notebook. If you'd like other users to be able to query this table, you can also create a table from the DataFrame.
# Once saved, this table will persist across cluster restarts as well as allow various users across different notebooks to query this data.
# To do so, choose your table name and uncomment the bottom line.

permanent_table_name = "Toddler_Autism_dataset_BigData_csv"

# df.write.format("parquet").saveAsTable(permanent_table_name)
```

Shift+Enter to run shortcuts





You can delete file in a notebook cell by:

dbutils.fs.rm("/FileStore/tables/FileName.csv")

[Workshop] ASD Diagnosis in PySpark

- Open the **A1234567X Donald Duck – Spark DTree** PySpark notebook file and complete the workshop;

- Execute notebook to be familiar with PySpark scripts;

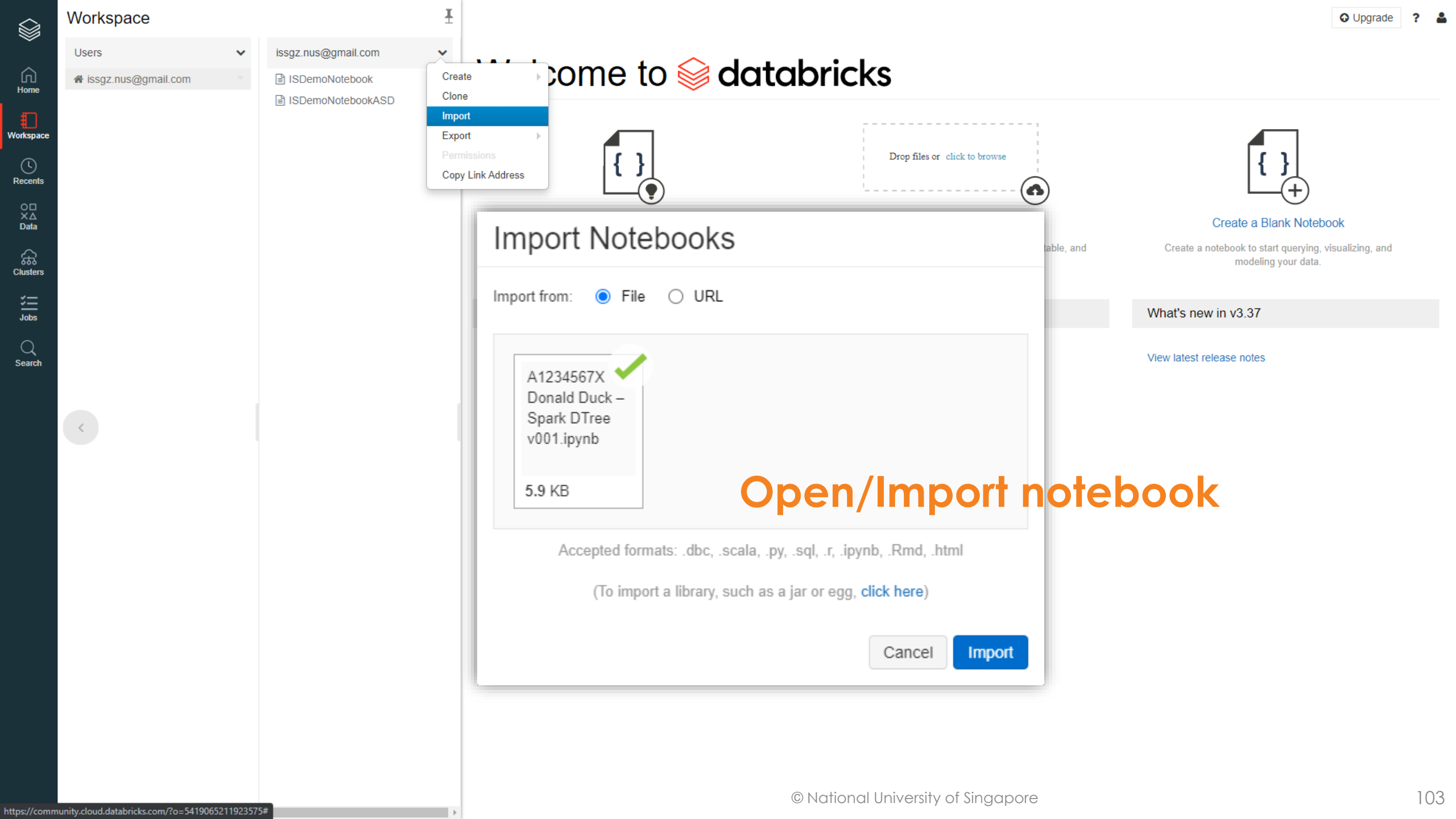
- Interpret decision tree results, e.g. which feature is most important on toddler ASD detection: different ethnicity? age?

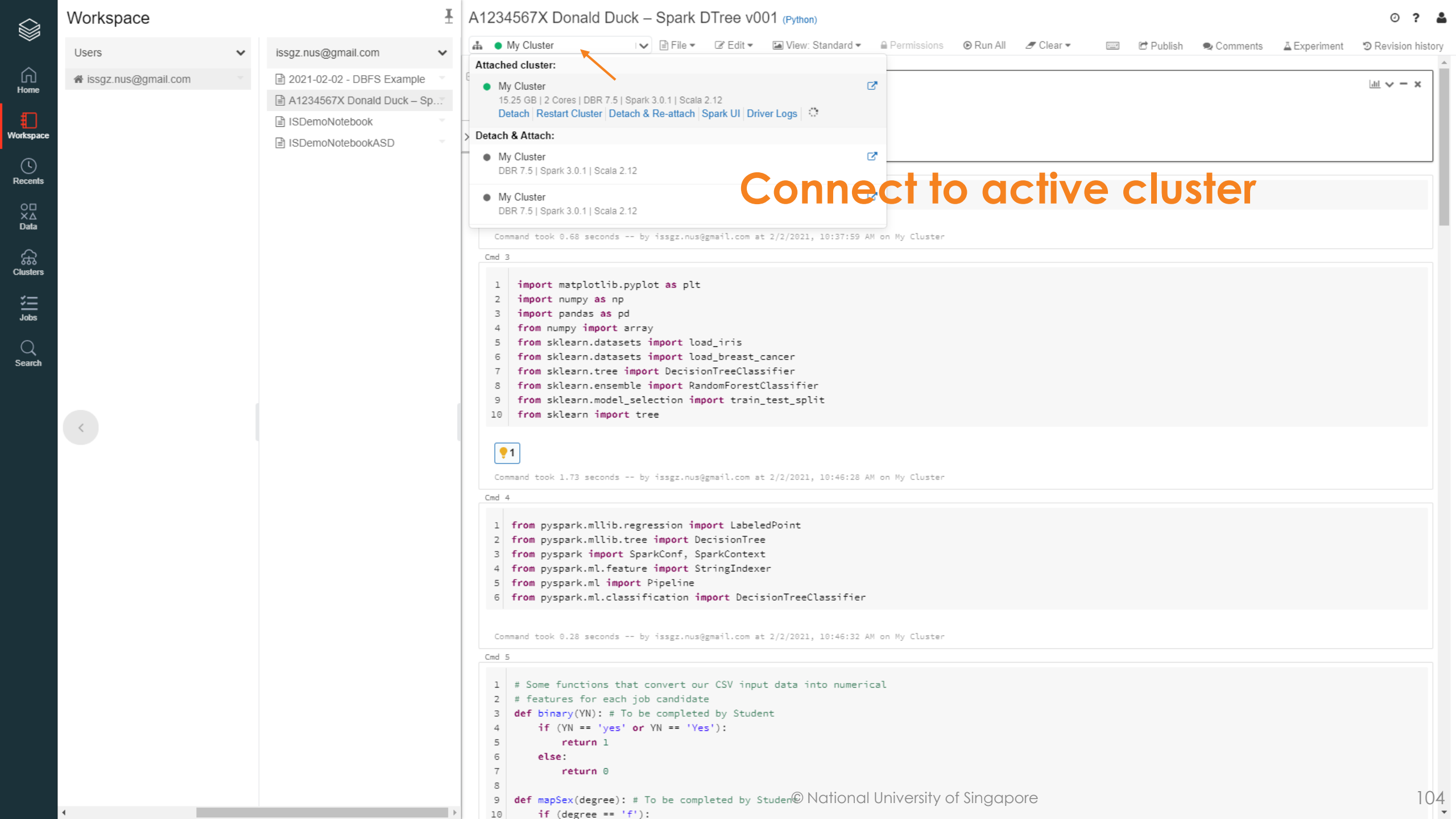
[Workshop] ASD Diagnosis in PySpark

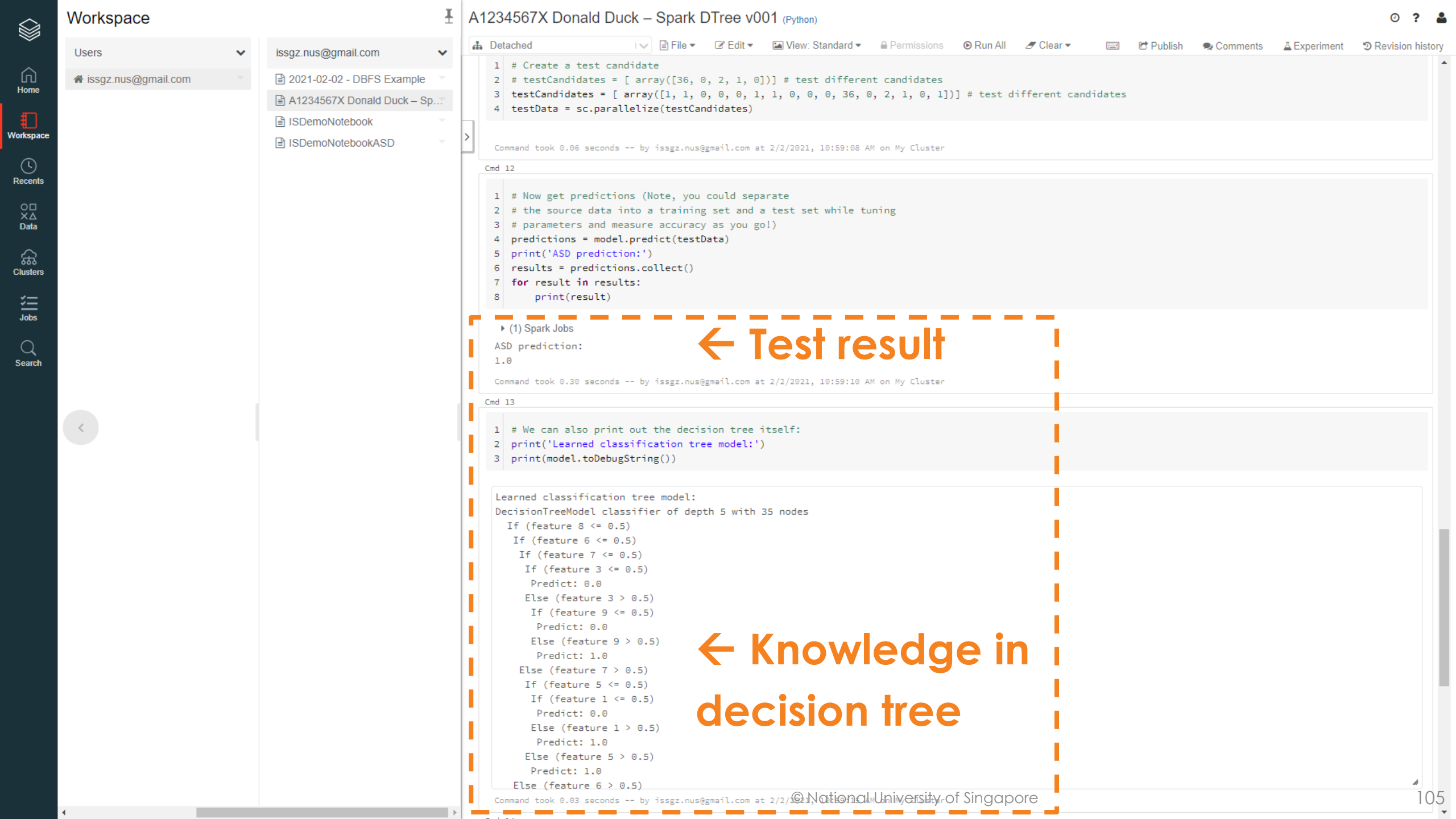
[Optional] Enhancements:

- Evaluate decision tree using confusion matrix;
- Dummy / One-hot encoding of categorical features;
- Feature engineering and selection, e.g. whether A1 – A10 is useful?

```
# Some functions that convert our CSV input data into numerical
# features for each job candidate
def binary(YN): # To be completed by Student
    if (YN == 'yes' or YN == 'Yes'):
        return 1
    else:
        return 0

def mapSex(degree): # To be completed by Student
    if (degree == 'f'):
        return 1
    else:
        return 0

def mapEthnicity(ethnic): # To be completed by Student
    if(ethnic == 'asian'):
        return 1
    else:
        return 0

def mapAssessor(assessor): # To be completed by Student
    if(assessor == 'Health Care Professional'):
        return 1
    else:
        return 0

```







Workshop Submission

- **Naming convention: StudentID YourFullName, e.g. A1234567X Donald Duck – sln – Spark DTree.ipynb/py/zip**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**

4.3 Building Machine Reasoning System [Workshop]

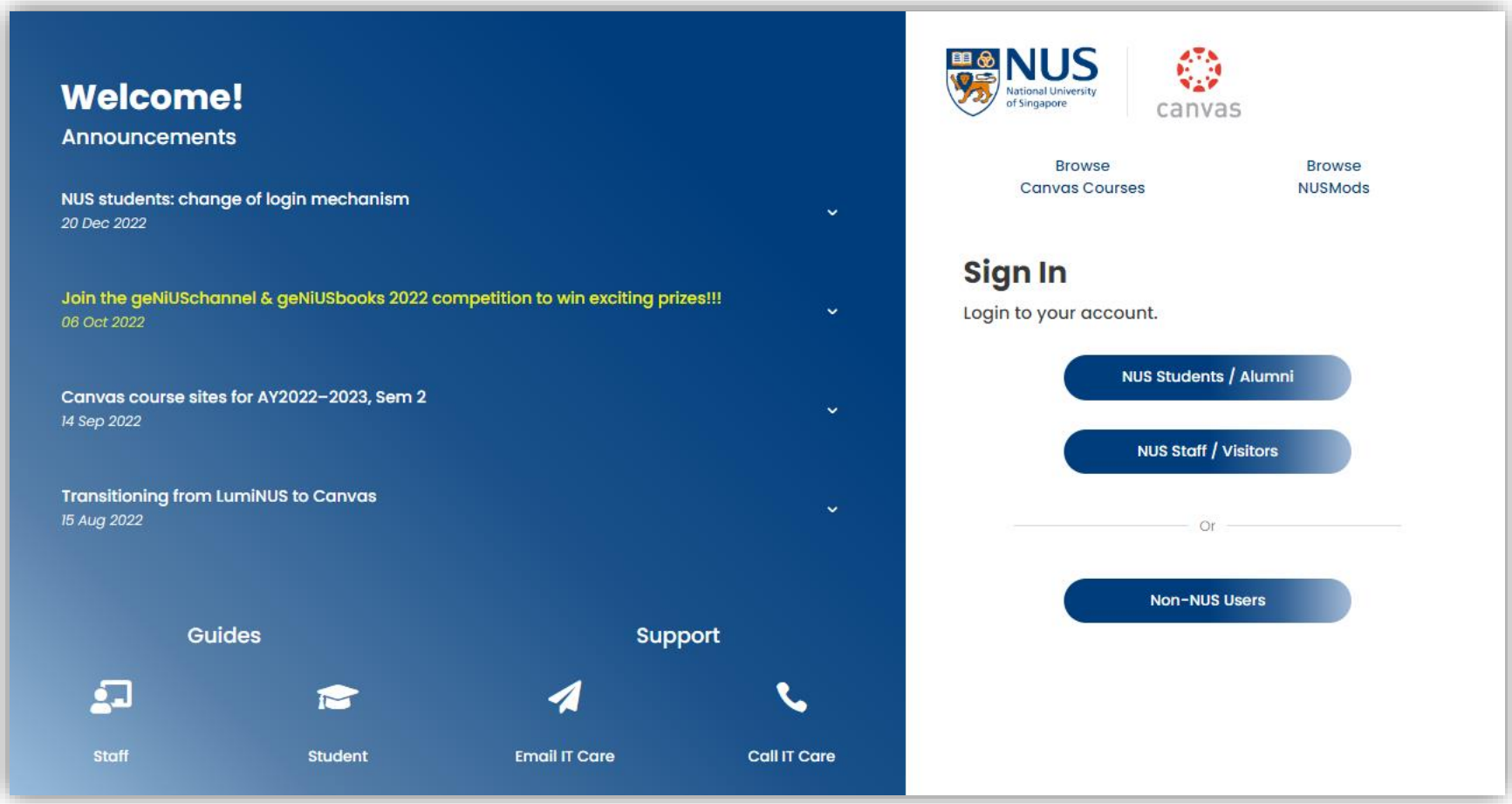
4.3.1 Big Data Machine Learning & Reasoning System

Special thanks to Geet Jethwani (A0215395B) for his contribution.

4.3.2 Workshop Submission

Workshop Submission

- **Naming convention: StudentID YourFullName**
- **Use zip to a single file, then rename, if you plan to submit multiple files.**



4.4 In-class Assessment

4.4.1 In-class Assessment

Machine Reasoning Summary

What we have learnt:

Day 1

1.1 Machine Reasoning Overview

1.2 Reasoning System Architectures

1.3 Rule/Process Reasoning System **Workshop**

Day 2

2.1 Machine Reasoning Enabler: Knowledge Representation

2.2 Machine Reasoning Enabler: Knowledge Acquisition

2.3 Knowledge Representation and Acquisition/Discovery **Workshop**

Day 3

3.1 Deductive Reasoning by Logical Inference

3.2 Reasoning under Uncertainty

3.3 Deductive Reasoning (under Uncertainty) **Workshop**

Day 4

4.1 Knowledge Discovery by Machine Learning (Big Data)

4.2 Contemporary Reasoning Systems (Big Data)

4.3 Building Machine Reasoning System **Workshop**

How can I apply AI to my business problem?

Start

Define & Quantify the business problem/opportunity {Goal, KPI}, e.g. {Increase customer satisfaction, Higher rating}; {Get rich fast, More $ Less time}

Anyone (on Earth) knows how to solve? Or known existing/manual approaches?

False

True

Is **Reasoning/Inference Task**: **Use existing** knowledge to solve problem at scale and speed. (decision automation; optimization)

Is **Leaning Task**: **Acquire new** knowledge to solve the problem.

Has relevant data collection?

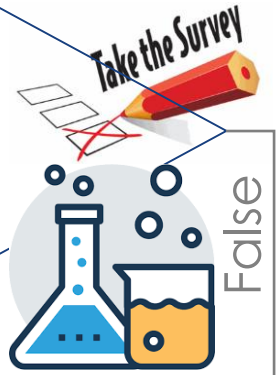
False

True

Use (un)supervised: classification & regression; pattern recognition: clustering; statistical algorithms to **acquire** knowledge from data.

Is a special learning task: **Reinforcement Learning Task**: Use generate & test approach.

Can simulate /generate relevant data? Or conduct experiment?



False

True

Use reinforcement learning algorithms; simulation techniques to **acquire** knowledge, e.g. best series of actions to achieve goal.

Currently not solvable by AI (even human)

End

Future Work (Homework)

Homework Topic	Priority	Example Tools	Reference
From Human Intelligence to Machine Intelligence	1		https://youtu.be/HQUxSi52Ujk
Computer Programming	2	Python Anaconda	https://www.anaconda.com/
Machine Memory: Database & SQL	3	SqLite; MySQL	https://github.com/agarcialeon/awesome-database
Text Processing	4	Python lib: SpaCy	https://github.com/keon/awesome-nlp
Optimization	5	Google OR-Tools; KIE-OptaPlanner	https://developers.google.com/optimization
Cloud Computing	6	GCP; AWS	https://github.com/tmrts/awesome-cloud-computing
Machine Reasoning	7	Semantic Reasoner; KIE-Drools; KIE-jBPM;	https://github.com/semantalytics/awesome-semantic-web
Speech Virtual Assistant	8	MyCroft; RASA;	https://mycroft.ai/
Information Retrieval / Search Engine	9	Chatter-Bot; Lucene; Lemur;	https://github.com/harpribot/awesome-information-retrieval
Knowledge Graph & GraphDB	10	protégé; grakn.ai	https://github.com/totogo/awesome-knowledge-graph
Recommender	11		https://github.com/jihoo-kim/awesome-RecSys
Big Data	12	Spark; Hadoop;	https://github.com/onurakpolat/awesome-bigdata
Machine Learining	13	Scikit-Learn; Orange3	https://github.com/josephmisiti/awesome-machine-learning
Object/Face Detection	14		https://github.com/amusi/awesome-object-detection
Time Series	15	Facebook Prophet	https://facebook.github.io/prophet/
Full Stack	16		https://github.com/kevindeasis/awesome-fullstack

END OF NOTES

APPENDICES

Latest News → Introducing Neo4j Aura Enterprise: The Cloud Graph Database Chosen by Leading Brands

Company Support Contact Us Search

neo4j PRODUCTS USE CASES CUSTOMERS PARTNERS EVENTS & RESOURCES DEVELOPER RESOURCES GET STARTED

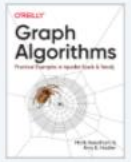
INTRODUCING

Neo4j Aura **Enterprise**

Neo4j's fully managed cloud service – the zero-admin, always-on graph database – now for the global enterprise

Learn More

Allianz Adobe ebay ICIJ UBS

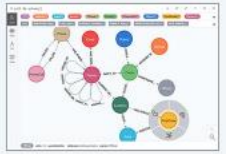


Free O'Reilly Ebook

Graph Algorithms: Examples in Spark and Neo4j

Sample code and tips for over 20 practical algorithms. Find vulnerabilities, detect communities, improve machine learning.

Read Now

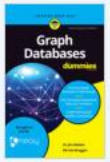


No Download Required

Get Started Quickly with the Neo4j Sandbox

Start using Neo4j in seconds, with built-in guides and datasets for popular use cases. No experience necessary.

Try It Now



Neo4j Special Edition

Graph Databases for Dummies

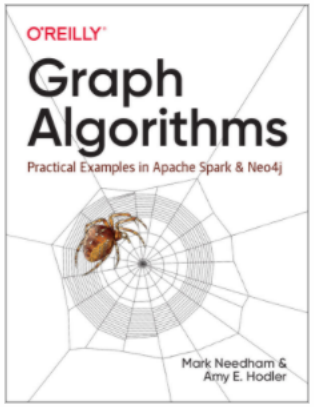
Learn the basics of graph database technology, from building a data model to deploying a graph-powered application.

Download Now

Source https://neo4j.com/

Book

Graph Algorithms: Practical Examples in Apache Spark and Neo4j



By Mark Needham & Amy Hodler

Published by O'Reilly Media

Print Length: 300 pages

Available Formats: PDF - EN US, iBooks, Kindle

Register now for your copy of the O'Reilly book, *Graph Algorithms: Practical Examples in Apache Spark and Neo4j* by Mark Needham and Amy E. Hodler. You'll receive a link to download an electronic version as soon as it's available this spring.

Whether you are trying to build dynamic network models or forecast real-world behavior, this book demonstrates how graph algorithms deliver value – from finding vulnerabilities and bottlenecks to detecting communities and improving machine learning predictions.

We walk you through **hands-on examples** of how to use graph algorithms in Apache Spark and Neo4j. We include sample code and tips for over 20 practical graph algorithms that cover importance through centrality, community detection and optimal pathfinding. Read this book to:

- Learn how graph analytics vary from conventional statistical analysis
- Understand how classic graph algorithms work and how they are applied
- Dive into popular algorithms like PageRank, Label Propagation and Louvain to find out how subtle parameters impact results

Register to Download O'Reilly's Graph Algorithms for Free!

First Name

First Name

Last Name

Last Name

Business Email

Business Email

Company Name

Company Name

Country

Country

Get My Free Copy

The information you provide will be used in accordance with the terms of our privacy policy.

Source https://neo4j.com/graph-algorithms-book

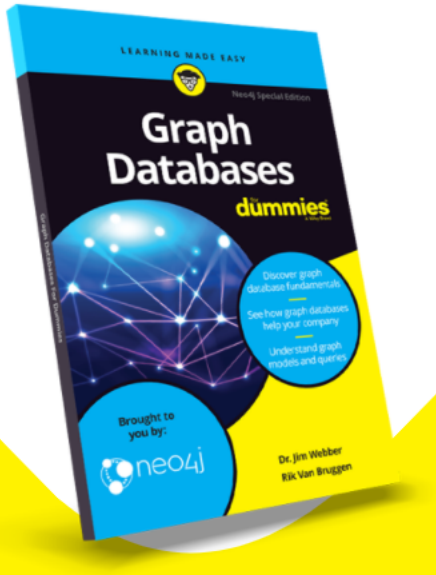
Company Support Contact Us Search

neo4j PRODUCTS USE CASES CUSTOMERS PARTNERS EVENTS & RESOURCES DEVELOPER RESOURCES GET STARTED

FREE BOOK

Graph Databases For Dummies

Graph Databases For Dummies, a Neo4j Special Edition, introduces you to the basics of graph database technology from building a rich graph data model to deploying your first graph-powered application.



Download Your Free Copy

First Name Last Name

First Name | Last Name

Business Email

Business Email

Company Name

Company Name

Country

Country

Download Now

The information you provide will be used in accordance with the terms of our **privacy policy.**

Learn graph database fundamentals

Identify when to use graph technology

Deploy your first graph database

Source https://neo4j.com/graph-databases-for-dummies

END OF APPENDICES