

# Statistica e gestione delle informazioni

CLOUD DATA SCIENCE LAB

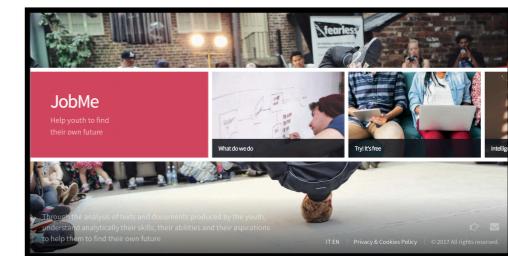
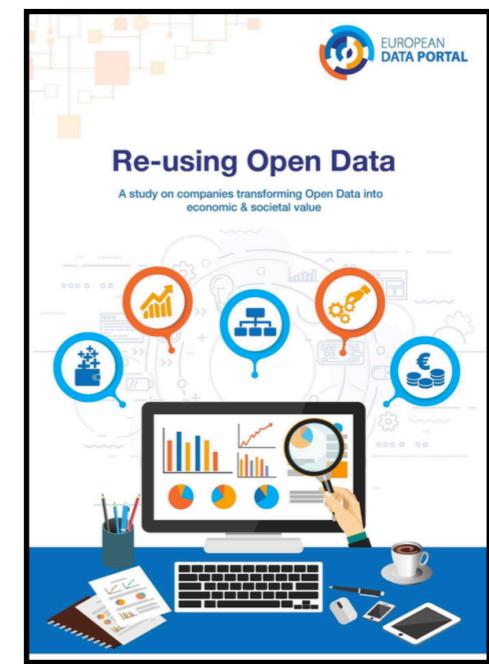
GOOGLE COLABORATORY AND DATABRICKS

[mauro.pelucchi@unimib.it](mailto:mauro.pelucchi@unimib.it)

Mauro Pelucchi

# Conosciamoci

- Mauro Pelucchi mauro.pelucchi@unimib.it mauro.pelucchi@gmail.com
  - Laureato in **Ingegneria Informatica** presso Università degli Studi di Bergamo con una tesi sull'utilizzo degli OpenData ([www.hammer-project.com](http://www.hammer-project.com))
  - Master in **Business Intelligence & Big Data Analytics** (2015/2016, trovate il lavoro finale all'indirizzo web [www.job-me.it](http://www.job-me.it))
  - Mi occupo di **Data Science** e **Data Engineering**: progettazione e messa in qualità di grandi basi di dati, AI, machine learning, estrazione di patterns da dati non strutturati, realizzazione di sistemi direzionali statistici e strumenti per la data visualization
  - In passato ho progettato e realizzato sistemi di Process Integration e Business Intelligence per grandi realtà industriali e per la pubblica amministrazione
  - Main interests: **OpenData e Open Government, Big Data Analysis, Data Presentation techniques, Artificial Intelligence, Machine Learning Modeling & Deep Learning**



# Cosa vedremo?

- Google Colaboratory
- DataBricks

# Google Colaboratory

# DataBricks

Cloud based run time python, R, scala of  
notebooks

# Google Colaboratory

## DataBricks

- Write and execute code in Python, R or Scala
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive or others
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- ~Free Cloud service with free GPU

# Google Colaboratory



# What is Google Colaboratory?



- Free Jupyter notebook environment
  - About 13G memory and 25G disk space limit
- No setup
- 100% Cloud
- Python + R
- Hardware accelerator
  - Graphical Processing Unit (GPU)
  - Tensor Processing Unit (TPU)

# Awesome Sharing Feature



- Save Colab to Google Drive and share.
- Save Colab to GitHub directly.
- Upload / Download Colab to local drive.
- Share with others users
- Control revisions
- Google document
- Install packages

# Cool Features



- If an error occurs, it (usually) automatically provides a link to StackOverflow
- You can add forms for input parameters from user.
- Mount Google Drive and access data from Google Sheet.

# Google Colaboratory



<https://colab.research.google.com/notebooks/welcome.ipynb>

The screenshot shows the Google Colaboratory interface. At the top, there's a navigation bar with the title "Hello, Colaboratory", a "SHARE" button, and a "Sign in" button. Below the title, there's a toolbar with buttons for "CODE", "TEXT", "CELL", "COPY TO DRIVE", "CONNECT", and "EDITING". On the left, a sidebar titled "Table of contents" lists various features: Getting Started, Highlighted Features, TensorFlow execution, GitHub, Visualization, Forms, Examples, Local runtime support, and a "SECTION" button. The main content area displays the "Welcome to Colaboratory!" page, which includes a "CO" logo, a heading, and a paragraph of text. Below this, there's a "Getting Started" section with a bulleted list of links to various resources. At the bottom, there's a "Highlighted Features" section with a dropdown arrow.

Hello, Colaboratory

SHARE Sign in

File Edit View Insert Runtime Tools Help

CODE TEXT CELL COPY TO DRIVE CONNECT EDITING

Table of contents Code snippets Files

Getting Started

Highlighted Features

TensorFlow execution

GitHub

Visualization

Forms

Examples

Local runtime support

SECTION

Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. See our [FAQ](#) for more info.

## Getting Started

- [Overview of Colaboratory](#)
- [Loading and saving data: Local files, Drive, Sheets, Google Cloud Storage](#)
- [Importing libraries and installing dependencies](#)
- [Using Google Cloud BigQuery](#)
- [Forms, Charts, Markdown, & Widgets](#)
- [TensorFlow with GPU](#)
- [TensorFlow with TPU](#)
- [Machine Learning Crash Course: Intro to Pandas & First Steps with TensorFlow](#)
- [Using Colab with GitHub](#)

▼ **Highlighted Features**

# Google Colaboratory

CO

EXAMPLES RECENT GOOGLE DRIVE GITHUB UPLOAD

Filter notebooks

Title	First opened	Last opened	
CO Hello, Colaboratory	Dec 19, 2018	0 minutes ago	

**NEW PYTHON 3 NOTEBOOK ▾ CANCEL**

# Google Colaboratory

CO

The screenshot shows the Google Colaboratory interface. On the left, there's a sidebar with a 'CODE' button and a 'TEXT' button. The main area has a play button icon. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime' (which is currently selected), 'Tools', and 'Help'. A dropdown menu for 'Runtime' is open, listing several options: 'Run all' (⌘/Ctrl+F9), 'Run before' (⌘/Ctrl+F8), 'Run the focused cell' (⌘/Ctrl+Enter), 'Run selection' (⌘/Ctrl+Shift+Enter), 'Run after' (⌘/Ctrl+F10), 'Interrupt execution' (⌘/Ctrl+M I), 'Restart runtime...' (⌘/Ctrl+M .), 'Restart and run all...', and 'Reset all runtimes...'. The 'Change runtime type' option is highlighted with a red box. Below this, another red box highlights the 'Manage sessions' option. To the right, a modal window titled 'Notebook settings' is displayed. It contains a dropdown for 'Runtime type' set to 'Python 3', another dropdown for 'Hardware accelerator' set to 'GPU', and a checkbox for 'Omit code cell output when saving this notebook' which is unchecked. At the bottom right of the modal are 'CANCEL' and 'SAVE' buttons.

Untitled0.ipynb

File Edit View Insert Runtime Tools Help

Run all ⌘/Ctrl+F9

Run before ⌘/Ctrl+F8

Run the focused cell ⌘/Ctrl+Enter

Run selection ⌘/Ctrl+Shift+Enter

Run after ⌘/Ctrl+F10

Interrupt execution ⌘/Ctrl+M I

Restart runtime... ⌘/Ctrl+M .

Restart and run all...

Reset all runtimes...

Change runtime type

Manage sessions

Notebook settings

Runtime type

Python 3

Hardware accelerator

GPU

Omit code cell output when saving this notebook

CANCEL SAVE

# Google Colaboratory R



<https://colab.research.google.com/notebook#create=true&language=r>

# Google Colaboratory R

CO

The screenshot shows the Google Colaboratory interface. At the top left is the 'Welcome To Colaboratory' header. Below it is a navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help' menus. On the far right of the top bar are 'Share', 'Settings', and profile icons. A red arrow points from the top right towards the profile icon. The main content area displays the 'Welcome to Colaboratory!' page, which includes a brief introduction and a section titled 'Introducing Colaboratory'. On the left, a sidebar titled 'Table of contents' lists several sections: 'Introducing Colaboratory' (which is also highlighted with a red arrow), 'Getting Started', 'More Resources', and 'Machine Learning Examples: Seedbank'. There is also a '+ Section' button. At the bottom of the sidebar is a 'Code snippets' tab and an 'X' button.

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive Connect ▾

Table of contents Code snippets Files X

Introducing Colaboratory

Getting Started

More Resources

Machine Learning Examples: Seedbank

+ Section

CO Welcome to Colaboratory!

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

[ ] Introducing Colaboratory

# Google Colaboratory R



## Upload Demo\_R notebook

Examples      Recent      Google Drive      GitHub      Upload

Scegli file nessun file selezionato

NEW PYTHON 3 NOTEBOOK ▾ CANCEL

A screenshot of the Google Colaboratory interface. At the top, there is a navigation bar with tabs: Examples, Recent, Google Drive, GitHub, and Upload (which is underlined). Below the navigation bar is a large dashed rectangular area intended for file upload. In the center of this area, there is a button labeled "Scegli file" and the text "nessun file selezionato". At the bottom of the page, there are two buttons: "NEW PYTHON 3 NOTEBOOK" and "CANCEL".

# Google Colaboratory R



CO Demo.ipynb ☆

File Edit View Insert Runtime Tools Help

Comment Share ⚙️

+ Code + Text Initializing ▾ | Editing ▾

Code without visible output:

```
[ ] a <- 8
```

With visible output:

```
[ ] a + b
```

1. 12  
2. 13  
3. 14  
4. 15

# Databricks



# Databricks

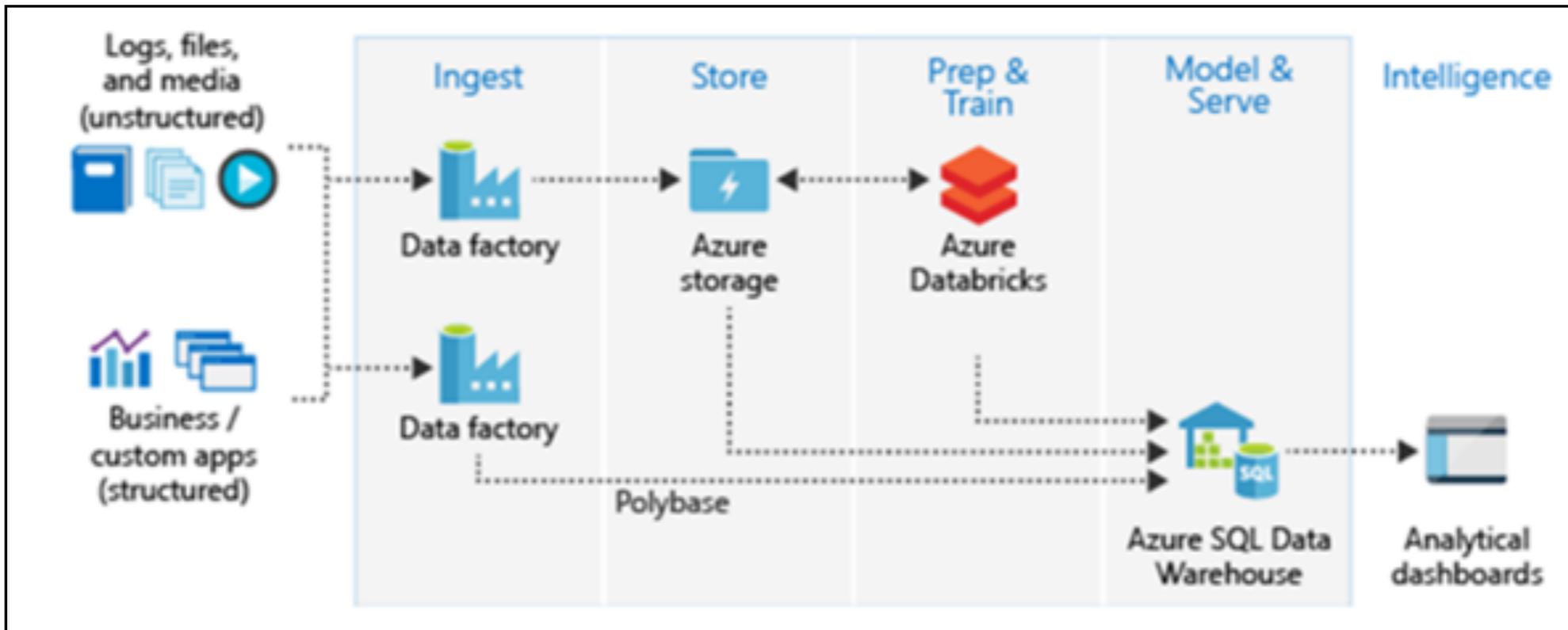
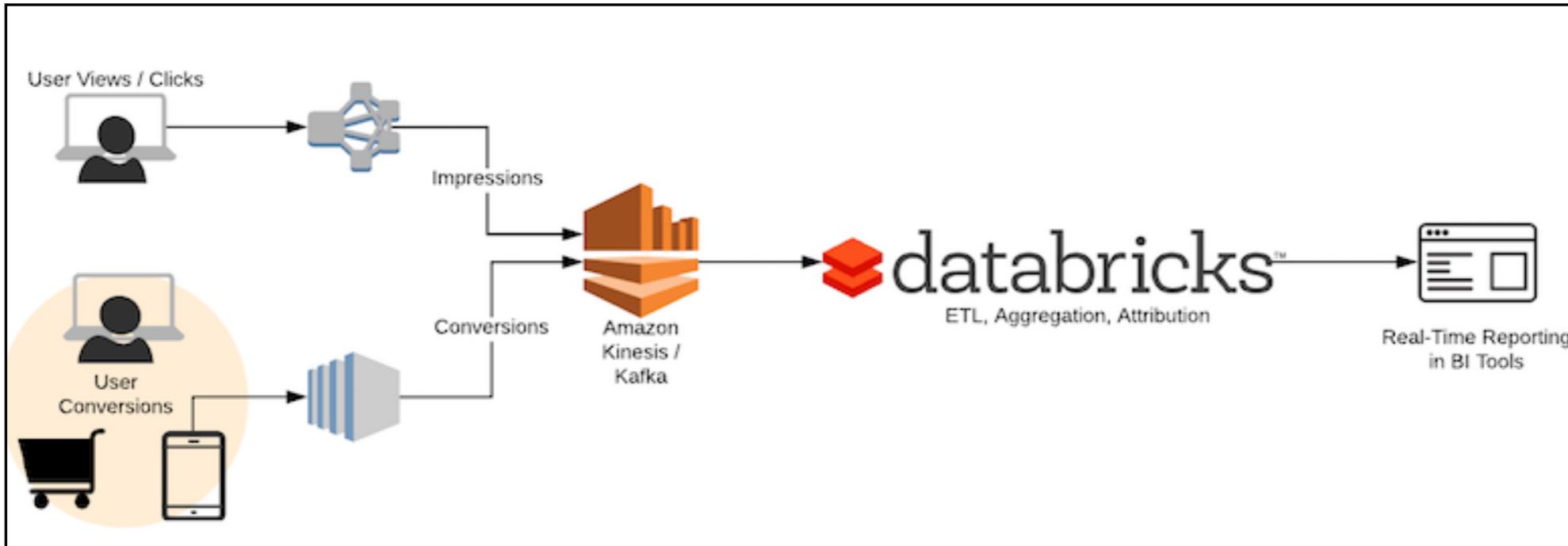


- E' una piattaforma di analytics facile e collaborativa basata su **Apache Spark**
- Tutti i moduli di Spark sono presenti in Databricks (SparkSQL, Streaming, ML, GraphX)
- **“Main goal: to remove all the hardness and complexity to get and manage a Spark cluster”**
- Permette di gestire installazioni e impostazioni con un clic
- Offre flussi di lavoro semplificati e area di lavoro interattiva per facilitare la collaborazione tra data scientists, sviluppatori e business analysts
- Integrazione con i maggiori cloud providers come Amazon AWS and Microsoft Azure

# Environments AWS and Azure



databricks®



# Notebook



- Simile ai notebook di Jupyter o Zeppelin. Linguaggi supportati: R, Python, Scala e SQL
  - Possono essere usati tutti in un singolo notebook.
- La Spark session è già definita per ciascun notebook come variabile globale spark.
- Una volta creato un notebook deve essere collegato a un cluster attivo

# Version Management



- Databricks è una piattaforma di analisi collaborativa dove gli utenti possono condividere workspaces, clusters and jobs attraverso una sola interfaccia.
- E' possibile creare modelli condivisi nello stesso notebook real time, riutilizzare data assets, librerie presenti su uno stesso cluster, o riutilizzare/monitorare scheduled jobs.
- Databricks supporta l'integrazione con Github, Bitbucket Cloud & Azure DevOps Services



## Sign In to Databricks



Email / Username



Password

[Forgot Password?](#)

[Sign In](#)

New to Databricks? [Sign Up.](#)

[Privacy Policy](#) | [Terms of Use](#)

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

The screenshot shows the Databricks Clusters page. On the left is a dark sidebar with navigation icons: Home, Workspace, Recents, Data, Clusters (which is selected and highlighted in blue), Jobs, and Search. The main area is titled "Clusters" and features a blue button labeled "+ Create Cluster". Below it is a section titled "Interactive Clusters" which contains a table with one row. The table has columns for Name, State, Nodes, Driver, and Workload. The single entry is "Esempio" (marked with a green dot), which is "Running" with "1 (0 spot)" nodes, a driver, and a workload. There is also a section titled "Job Clusters" which is currently empty.

Name	State	Nodes	Driver	Workload
Esempio	Running	1 (0 spot)	Community... Cor	

# Creazione del cluster

Create Cluster

New Cluster

Cancel

Create Cluster

0 Workers: 0.0 GB Memory, 0 Cores, 0 DBU

1 Driver: 6.0 GB Memory, 0.88 Cores, 1 DBU 

Cluster Name

Esempio1

Databricks Runtime Version 

Runtime: 5.3 (Scala 2.11, Spark 2.4.1)



Python Version 

3



Instance

Free 6GB 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

Availability Zone 

us-west-2c



# Upload dei dati

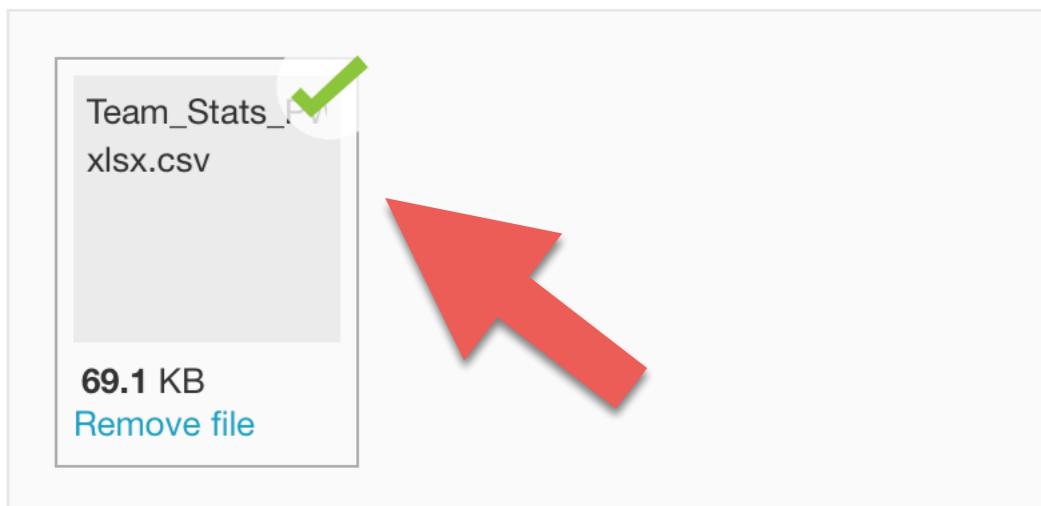
## Create New Table

Upload File S3 DBFS Other Data Sources

Upload to DBFS ?

/FileStore/tables/ (optional) Select

File ?



✓ File uploaded to /FileStore/tables/Team\_Stats\_PW.xlsx-f88a9.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 ?

Esempio



Preview Table

# Upload dei dati

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 ?

team\_stats\_pw\_xlsx\_f88a9\_

Create in Database ?

default

File Type ?

CSV

Column Delimiter ?

,

First row is header ?

Infer schema

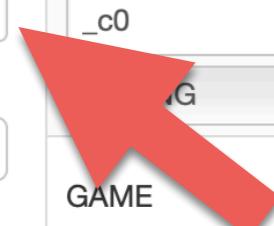
Multi-line

 Create Table

Create Table in  
Notebook

Table Preview

	_c0	_c1	_c2	_c3	_c4
GAME	vs Trieste R12	H	227.0	191.0	7.087384919700637
	vs Trieste R12	H	180.0	60.0	7.995
	Trieste R12	H	59.0	209.0	0.5418053155885424
	vs Trieste R12	H	58.0	240.0	1.0296295450306387
	vs Trieste R12	H	89.0	214.0	1.639856091247034
	vs Trieste R12	H	147.0	236.0	3.9772352206023722



# Upload dei dati

Create New Table

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 

team\_stats\_pw\_xlsx\_f88a9...

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

GAME	LOCATION	COORD_X	COORD_Y	DISTANCE
STRING	STRING	FLOAT	FLOAT	FLOAT
vs Trieste R12	H	227.0	21.0	7.087384919700637
vs Trieste R12	H	180.0	60.0	7.995
vs Trieste R12	H	59.0	209.0	0.5418053155885424
vs Trieste R12	H	58.0	240.0	1.0296295450306387
vs Trieste R12	H	89.0	214.0	1.639856091247034
vs Trieste R12	H	147.0	236.0	3.9772352206023722
vs Trieste R12	H	236.0	243.0	7.445834405894345

# Database

The screenshot shows the Databricks Data interface. On the left, a dark sidebar lists navigation options: Home, Workspace, Recents, Data (which is selected and highlighted in blue), Clusters, Jobs, and Search. The main area is titled 'Data' and shows the 'default' database selected. Under 'Tables', the 'team\_stats\_pw\_x...' table is listed. The table schema is displayed in a grid:

data_type	comment
string	null
string	null
float	null
string	null
string	null
double	null
string	null

# Notebook

The screenshot shows the Databricks workspace interface. On the left, a dark sidebar menu includes icons for Home, Workspace (selected), Recents, Data, Clusters, Jobs, and Search. The main workspace title is "Workspace". A user dropdown shows "mauro.pelucchi@u...". A context menu is open at the top right, with "Create" selected, showing options: Notebook (highlighted with a red arrow), Library, Folder, Clone, Import, Export, and Permissions. Below the menu is a "Drop files or click to browse" area with a cloud upload icon. To the right, there's a "Import & Explore Data" section with a description and a "Create a Blank Notebook" button with a plus sign icon. A "Recents" section shows a placeholder message: "Recent files appear here as you work." A "What's new in v2.101" section lists "Instance Pools" and a link to "View latest release notes".

Workspace

mauro.pelucchi@u...

Create

- Notebook
- Library
- Folder
- Clone
- Import
- Export
- Permissions

Drop files or click to browse

Import & Explore Data

Quickly import data, preview its schema, create a table, and query it in a notebook.

Create a Blank Notebook

Recents

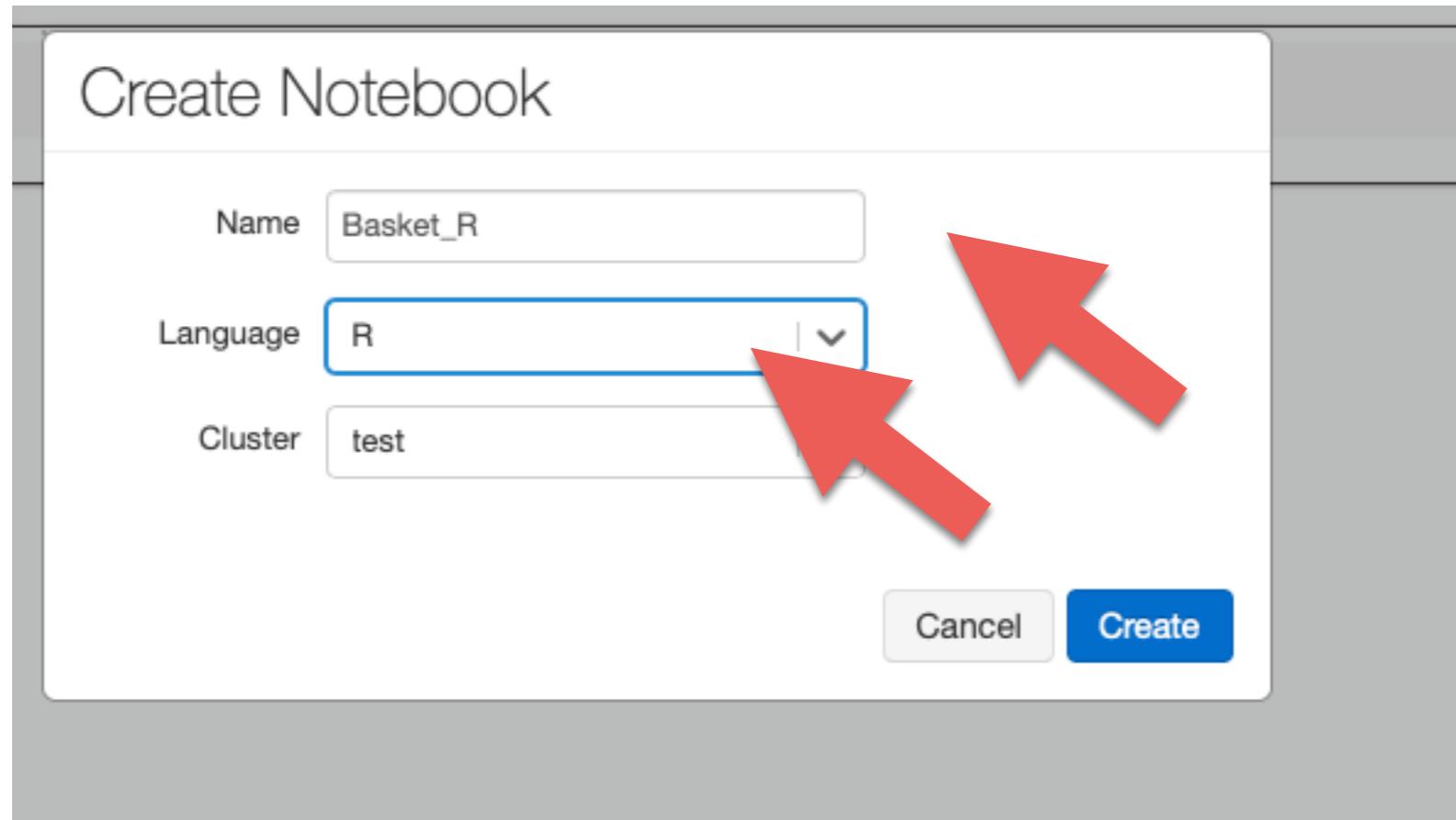
Recent files appear here as you work.

What's new in v2.101

- Instance Pools

[View latest release notes](#)

# Notebook



# Notebook

<https://docs.databricks.com/notebooks/notebooks-use.html#develop-notebooks>

# Notebook - R

Basket\_R (R) ? 👤

test File View: Code Permissions Run All Clear Publish Comments Runs Revision history

Cmd 1

```
1
2
3 library(dplyr)
4 library(caret)
5 library(data.table)
```

▶ ▷ ⌂ ×

Loading required package: lattice  
Loading required package: ggplot2  
  
Attaching package: 'data.table'  
  
The following objects are masked from 'package:dplyr':  
  
between, first, last

Command took 2.51 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 16:29:06 on test

Shift+Enter to run [shortcuts](#)

# Notebook → SQL

Cmd 2

```
1 %sql
2
3 select * from default.basketball_stats_1
```

▶ (1) Spark Jobs

GAME	LOCATION	COORD_X	COORD_Y	DISTANCE	SHOT_CLOCK	DRIBBLES	TOUCHES	AREA	ASSISTS	EXTRA_PASS	FROM	END	TARGET	SITUATION	FINA
vs Trieste R12	H	227	191	7.087384919700637	11	12	4	0	0	0	SHOT_IN	3_SHOT_MISSED	0	P	W
vs Trieste R12	H	180	60	7.995	6	18	5	0	0	0	FT_IN	3_SHOT_MISSED	0	CO	W
vs Trieste R12	H	59	209	0.5418053155885424	21	3	1	1	1	0	STEAL	2_SHOT_MADE	1	FB	W
vs Trieste R12	H	58	240	1.0296295450306387	8	16	4	1	0	0	STEAL+THROW-IN	2_SHOT_MISSED	0	P+I	W
vs Trieste R12	H	89	214	1.639856091247034	23	2	1	1	0	0	STEAL	X_FS	0	FB	W

Command took 1.38 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 16:30:20 on test



# SparkR

Cmd 3

```
1 library(SparkR)
2 basket_data <- sql("select * from default.team_stats_1")
3 display(basket_data)
4
```

▶ (1) Spark Jobs

BLES	TOUCHES	AREA	ASSISTS	EXTRA_PASS	FROM	END	TARGET	SITUATION	FINAL	PERIOD	SCORE	DIFFERENCE	PLAYER	Column	LINK	CHAMP_LE
4	0	0	0		SHOT_IN	3_SHOT_MISSED	0	P	W	1	0--2	-2	Phil Greene IV	null	team/vs Trieste R12/1	A2
5	0	0	0		FT_IN	3_SHOT_MISSED	0	CO	W	1	0--4	-4	Jamal Jones	null	team/vs Trieste R12/2	A2
1	1	1	0		STEAL	2_SHOT_MADE	1	FB	W	1	0--4	-4	Jamal Jones	null	team/vs Trieste R12/3	A2
4	1	0	0		STEAL+THROW-IN	2_SHOT_MISSED	0	P+I	W	1	2--4	-2	Curtis Nwohuocha	null	team/vs Trieste R12/4	A2
1	1	0	0		STEAL	X_FS	0	FB	W	1	2--4	-2	Phil Greene IV	null	team/vs Trieste R12/5	A2

Command took 1.33 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 17:46:50 on test

# Databricks

Cmd 4

```
1 counts <- count(groupBy(basket_data, basket_data$PLAYER))
2 display(counts)
```

▶ (5) Spark Jobs

PLAYER	count
Mitchell Poletti	20
Riccardo Visconti	15
Simone Pierich	15
Andrea Amato	79
Mattia Udom	60
Leonardo Tote	19
Omar Dieng	1
Curtis Nwohuocha	9
Dikil Gomes	07

Command took 1.83 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 17:48:24 on test

Shift+Enter to run [shortcuts](#)

# Databricks

Cmd 4

```
1 counts <- count(groupBy(basket_data, basket_data$PLAYER))
2 display(counts)
```

▶ (5) Spark Jobs

PLAYER	count
Mitchell Poletti	20
Riccardo Visconti	15
Simone Pierich	15
Andrea Amato	79
Mattia Udom	60
Leonardo Tote	19
Omar Dieng	1
Curtis Nwohuocha	9

Detailed View

Command took 1.83 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 17:48:24 on test

Shift+Enter to run [shortcuts](#)

Cmd 4

```
1 counts <- count(groupBy(basket_data, basket_data$PLAYER))
2 display(counts)
```

▶ (5) Spark Jobs

A bar chart titled 'Phil Greene IV: 87.00' showing the count of players for each player name. The x-axis is labeled 'PLAYER' and lists the names: Mitchell Poletti, Andrea Amato, Omar Dieng, and Jamal Jones. The y-axis is labeled 'count' and ranges from 0.00 to 80. The bars are blue. The chart shows that Andrea Amato has the highest count at 79, followed by Phil Greene IV at 87, and Jamal Jones at 60.

PLAYER	count
Mitchell Poletti	20
Andrea Amato	79
Omar Dieng	1
Jamal Jones	60

Plot Options...

Command took 1.83 seconds -- by mauro.pelucchi@unimib.it at 17/1/2020, 17:48:24 on test

Shift+Enter to run [shortcuts](#)