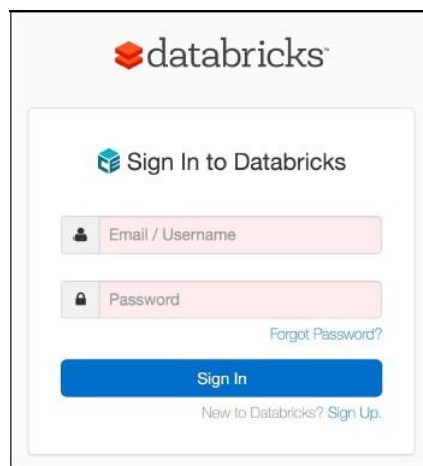


## Exercise 1 :

Databricks is the company behind Spark. It has a cloud platform that takes all the complexity out of Spark deployment and provides you with a ready-to-use environment with notebooks for different languages. Databricks Cloud also offers a community edition which provides a 15 GB RAM instance with 2 cores (terminating after 2 hours).

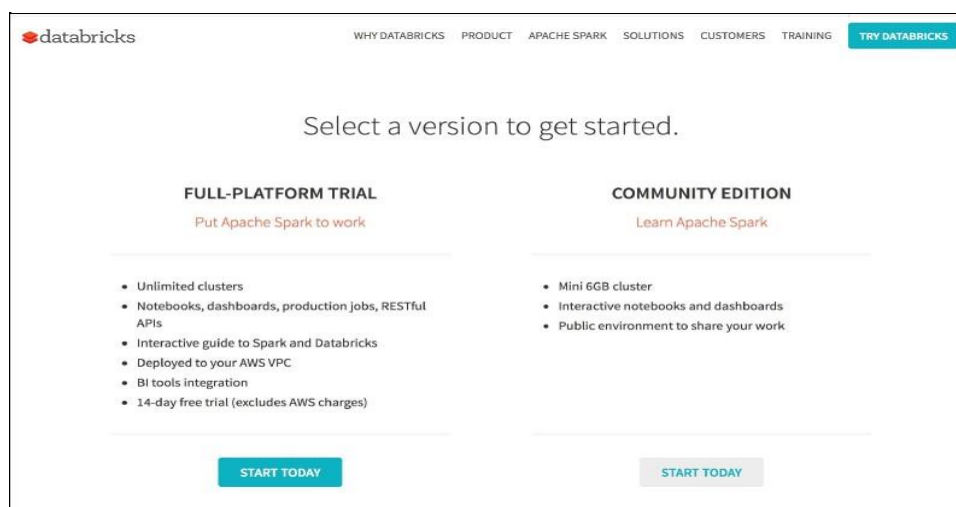
To start, go to <https://community.cloud.databricks.com> :

1. Create an account by clicking on “sign up”



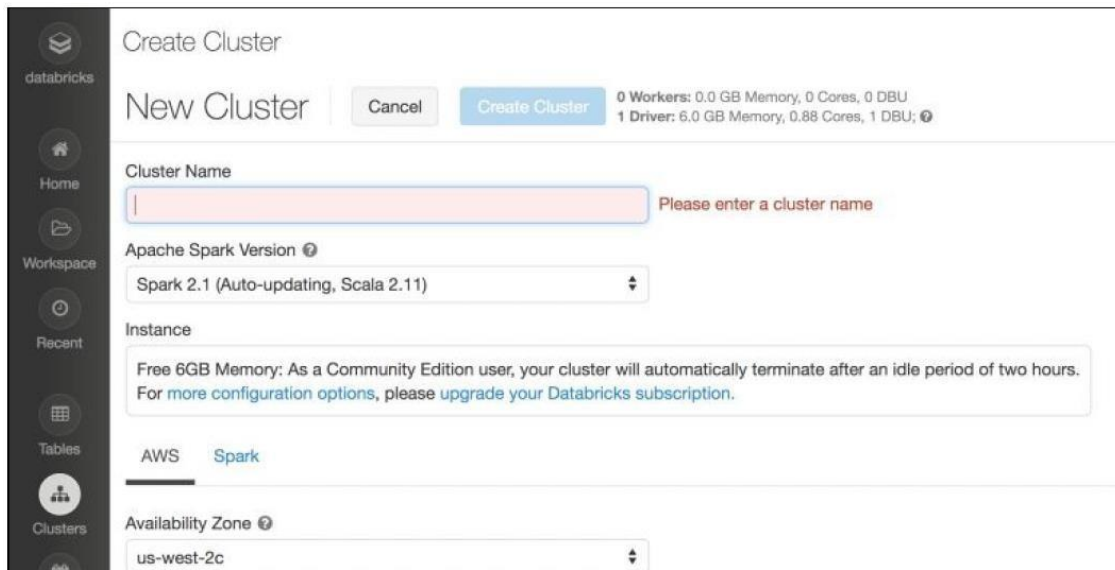
The image shows the Databricks sign-in interface. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box are two input fields: "Email / Username" and "Password". There is a "Forgot Password?" link next to the password field. Below the fields is a blue "Sign In" button. At the bottom of the box is a link that says "New to Databricks? Sign Up."

2. Choose “COMMUNITY EDITION”, then complete the form



The image shows the "Select a version to get started" page on the Databricks website. The page has a navigation bar at the top with links: WHY DATABRICKS, PRODUCT, APACHE SPARK, SOLUTIONS, CUSTOMERS, TRAINING, and a "TRY DATABRICKS" button. The main heading is "Select a version to get started." Below this are two columns. The left column is titled "FULL-PLATFORM TRIAL" with the subtext "Put Apache Spark to work". It lists features: Unlimited clusters, Notebooks, dashboards, production jobs, RESTful APIs, Interactive guide to Spark and Databricks, Deployed to your AWS VPC, BI tools integration, and 14-day free trial (excludes AWS charges). At the bottom of this column is a "START TODAY" button. The right column is titled "COMMUNITY EDITION" with the subtext "Learn Apache Spark". It lists features: Mini 6GB cluster, Interactive notebooks and dashboards, and Public environment to share your work. At the bottom of this column is a "START TODAY" button.

3. On the home page, click on “clusters”, then “create cluster”



4. Next, import the dataset and create an empty notebook.



In the Lab folder, you will find the “apache.access.log” dataset and its corresponding notebook. Open the notebook locally and try to copy-paste each cell into the new notebook.

## Exercise 2:

For this exercise, we will be using real-world data from Last.fm<sup>1</sup> to see how Collaborative Filtering can be used to recommend artists to users.

### Dataset

We start with the data from Table 1. The table shows the play counts for each band in the data set of the 10 users – assume empty cells to correspond to no (0) plays for the user-artist pair. The number of times a user has played a song by an artist is used as an implicit rating (rather than asking them to explicitly rate the artists).

---

<sup>1</sup> <http://ocelma.net/MusicRecommendationDataset/lastfm-360K.html>

	The Beatles	Radiohead	Coldplay	Pink Floyd	Muse
User 1	39655				
User 2		903	962		44076
User 3		489	6051		47468
User 4	14975			31957	
User 5	31526			5882	
User 6					42970
User 7	33685	2304		2351	
User 8		18652	31121		690
User 9	4		118857		
User 10			168		44036

We are next given the ratings for two users for whom we want to make recommendations:

	The Beatles	Radiohead	Coldplay	Pink Floyd	Muse
User 21	3344	?	?	22458	?
User 101	?	6293	2286	?	5156

## Tools

To generate recommendations to the two users, we will use the excel spreadsheet cf-extended.xlsx. The spreadsheet contains the data set as well as an implementation of user-based and item-based collaborative filtering.

### Task 1: User-based collaborative filtering

Examine the excel by understanding the difference of simple and advanced prediction, which considers the average rating behavior of the users. Can you find a band, which receives a much worse result, when the rating behavior of the users is considered? What is the reason for this observation?

Next we are interested in examining different ks for the nearest neighbor selection. What is the impact of setting  $k=10$ ? What is a proper value for  $k$ ?

### Task 2: Item-based collaborative filtering

Examine the excel by understanding the difference of cosine and adjusted cosine similarity, which considers the average rating behavior of the users. Can you find a band, which receives a much worse result, when the rating behavior of the users is considered? What is the reason for this observation?

Next we are again interested in examining different ks for the nearest neighbor selection. What is the impact of setting  $k=10$ ? What is a proper value for  $k$ ? What is the impact of setting a threshold instead of choosing  $k$ ?

### **Exercise 3:**

Open the “MovieLens Exploration/Exploring MovieLens Dataset.ipynb” notebook.

### **Exercise 4:**

Open the “Introduction to Surprise - User & Item based Collaborative filtering .ipynb” notebook.