Student Guide for CS4225/CS5425 Assignment 2

Table of Contents

1 Problem Statement

1.1 Task Description

Every month, millions of people use bike-sharing systems to navigate cities and towns around the world. New York City's Citi Bike program launched in 2013 with 6,000 bikes, and in 2020, the system saw its 100 millionth ride. Citi Bike provides freely accessible data about its system operations that is both real-time and historical.

Your task is to predict the daily trip counts using citi bike data and other supporting datasets. For example, weather is an important factor affecting the daily trips in bike-sharing business, so we will be combining weather data with the citi bike data to get some important features for making predictions.

1.2 Datasets

1.2.1 Citi Bike Data

Citi Bike data are available at: https://s3.amazonaws.com/tripdata/index.html

The data includes:

- Ride ID
- Rideable type
- Started at
- Ended at
- Start station name
- Start station ID
- End station name
- End station ID
- Start latitude
- Start longitude
- End latitude
- End Longitude
- Member or casual ride

Detailed information can be found at https://citibikenyc.com/system-data

We will be using Year 2022 New York data as training data to build ML model:

- from 202201-citibike-tripdata.csv.zip to 202212-citibike-tripdata.csv.zip
- **Hint:** two files have typos on file names

We will be using Year 2023 (Jan to Jul) data as testing data to evaluate the model:

- from 202301-citibike-tripdata.csv.zip to 202307-citibike-tripdata.csv.zip

1.2.2 Weather Data

Weather data is available at: https://www.visualcrossing.com/weather-data

The data dictionary is available at:

https://www.visualcrossing.com/resources/documentation/weather-data/weather-data-documentation/

You need to register an account to download the weather data and each day you can only download 1000 records (the limit for the free account).

1.2.3 Other Data

Feel free to explore other data you think relevant to this problem.

1.3 Tasks and Marking Scheme (a total of 25 marks)

For this assignment, we will be using Databricks (easy setup and allows editing code through notebook interface, similar to Google Colab), a web-based platform for working with Spark. You can also easily track the runtime for each cell and view Spark UI for each job.

The assignment is designed to be solved using Spark. You can use Spark in a language of your choice (Python, Scala, SQL etc.) but you have to use Spark to complete all the tasks. It is NOT allowed to use other packages (e.g. pandas, sklearn, pytorch, tensorflow and etc.) in this assignment.

A template notebook is provided for you to solve this problem following the below steps:

1.3.1 Data Preparation (12 marks)

Step 1 (5 marks): Download and preprocess CitiBike data

Download and unzip CitiBike data files using codes. Read all the csv files into a Spark DataFrame, aggregate the data into daily records, i.e. each row represents the events happened in a day with:

- "**trip_count**" column: record the daily trip counts (must have)

- **“datetime”** column: record the date (must have)
- **the rest** columns: extract the useful features to predict trip_count. These features should not contain trip_counts information directly or indirectly.

Step 2 (2 marks): Download and preprocess the weather data

- Download the weather data into a csv file, upload it to databricks FileStore (details in Section 2) and read it into a Spark DataFrame.
- Preprocess the data, e.g. handle the missing values, select the relevant weather features and etc.

Step 3: Feel free to explore and download other data which is helpful for you to solve this problem. If the data file size is huge, you should provide the codes of downloading the data. No marks are allocated in this step, but if the extra data helps to improve your model performance you will get the better performance marks in Section 1.3.3.

Step 4 (4 marks): Combine all the data into a final table / dataframe with “trip_count” column as target and the rest columns as the features you may use to train your machine learning model. Prepare the data ready for modelling.
Save the aggregated table into DBFS (Databricks Filesystem, details in Section 2) and read the table from DBFS before running the rest of steps. (**Hint:** this will speed up the cells in the following steps as spark only needs to load this small table from DBFS, instead of reading the original citibike data from the beginning).

Step 5 (1 mark): Filter the final table / dataframe into training data and testing data:

- Training Data: Year 2022 (Jan to Dec) data <300 plus rows>
- Testing Data: Year 2023 (Jan to Jul) data <200 plus rows>

1.3.2 Build ML Pipeline using Training Data (5 marks)

Build a machine learning pipeline and train the pipeline using the training data.

The ML pipeline is likely to include the below steps:

- Encode the categorical features
- Scaling down the numerical features
- Select the relevant features
- Build machine learning model

1.3.3 Evaluate ML Pipeline using Testing Data (8 marks)

Evaluate the performance of the trained ML pipeline using testing data. You need to at least provide the performance metrics of Mean Absolute Error (MAE) and R squared value.

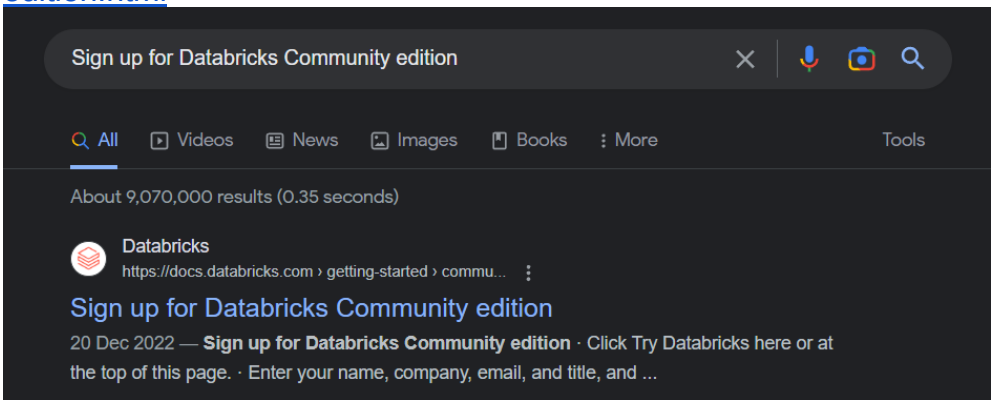
- Correct codes **(1 mark)**
- A valid evaluation with reasonable testing performance **(3 marks)**
    - Testing MAE < 15000
- Better Testing Performance **(4 marks)**
    - Testing MAE < 12000

2 Using Databricks

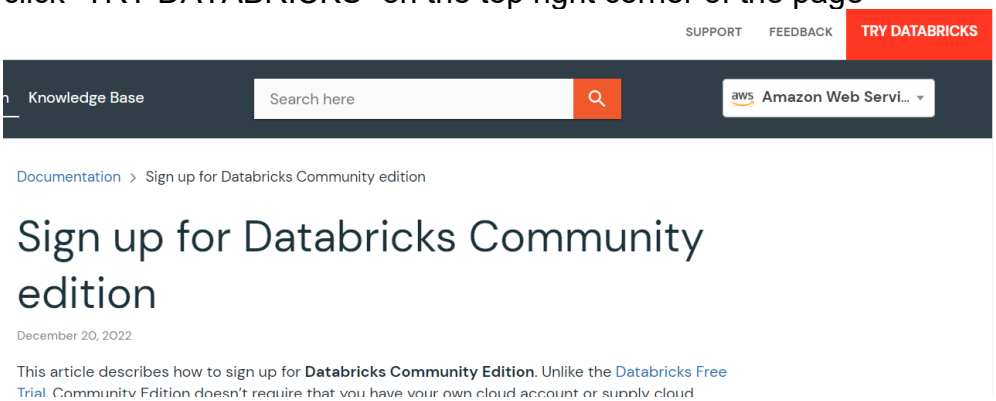
2.1 Databricks Community Edition Registration

We will be using the free version of databricks, called databricks community edition. You need to register for this service.

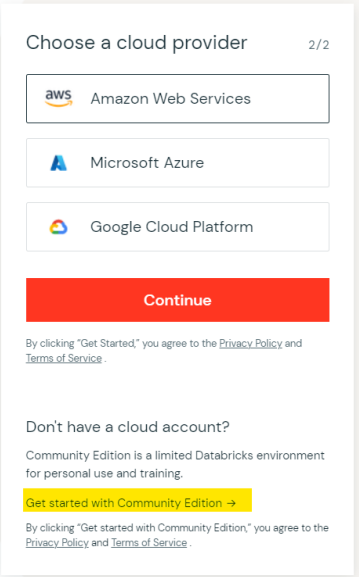
1) Google “Sign up for Databricks Community edition” and click on the first link or click https://docs.databricks.com/getting-started/community-edition.html



2) click “TRY DATABRICKS” on the top right corner of the page



3) Fill in the necessary details and click continue to create an account
4) You will reach the page below and since we are using the Community Edition, please click on the portion highlighted below:

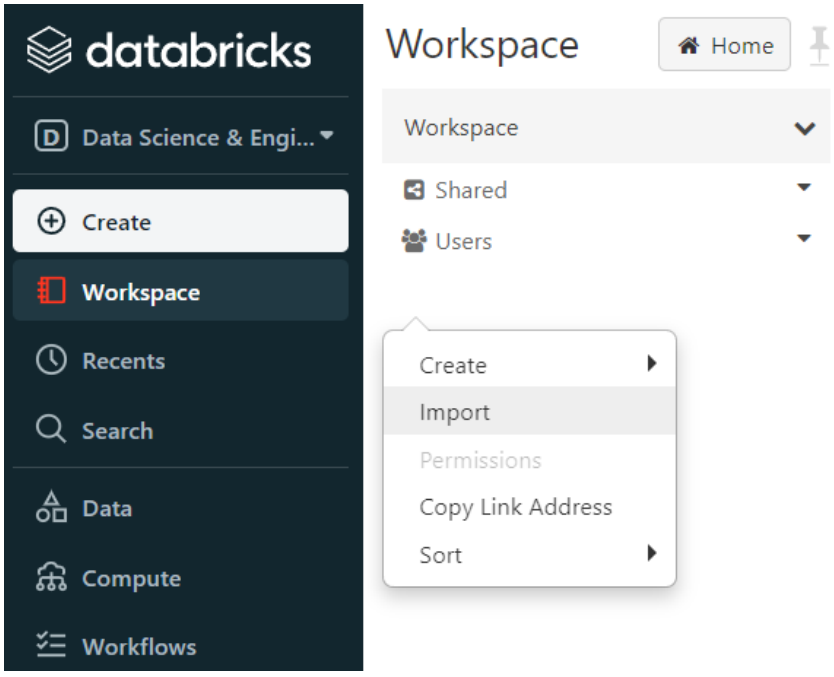


5) Validate your email address and login to databricks

2.2 Upload and Run Notebook in Databricks

Once you log in into databricks, you can start by importing the provided template notebook into the workspace.

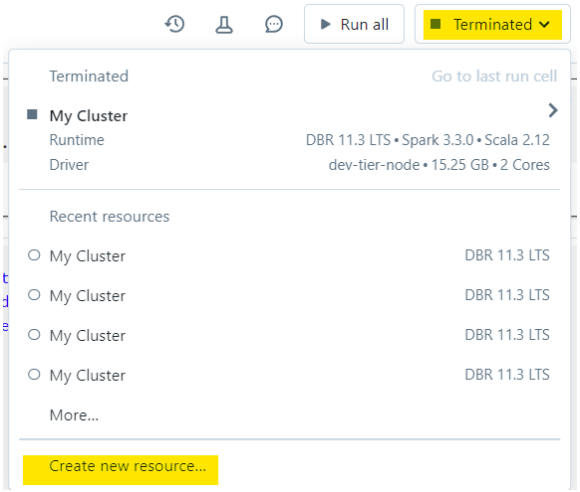
Click on Workspace > right-click > import to import the template notebook into databricks workspace.



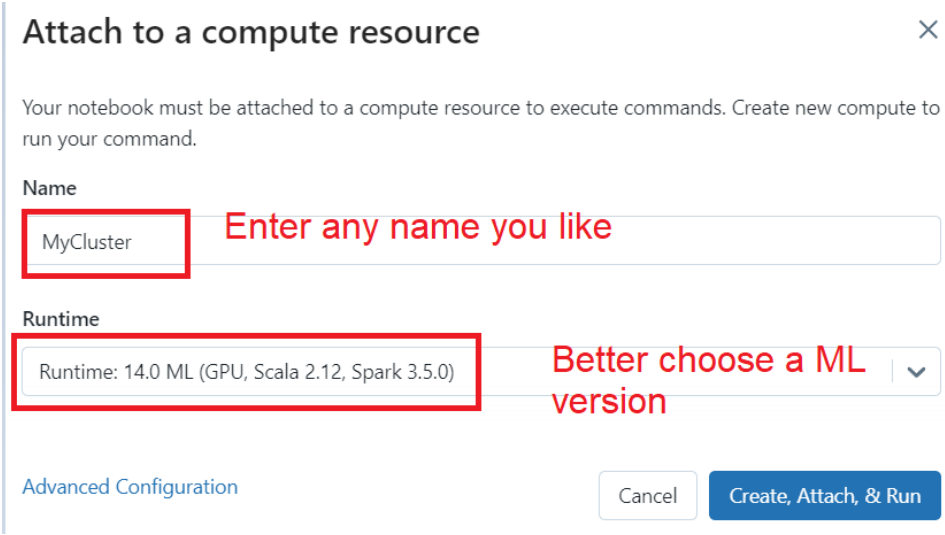
Once the notebook file is opened, you would need to create and attach a cluster to run your code. Note that every cluster you create will automatically terminate after an idle

period of two hours for the Community Edition. Hence you would need to re-create a new cluster to run your code if your cluster has terminated.

1) On the top-right of the notebook if it is not connected, click on it to create new resource

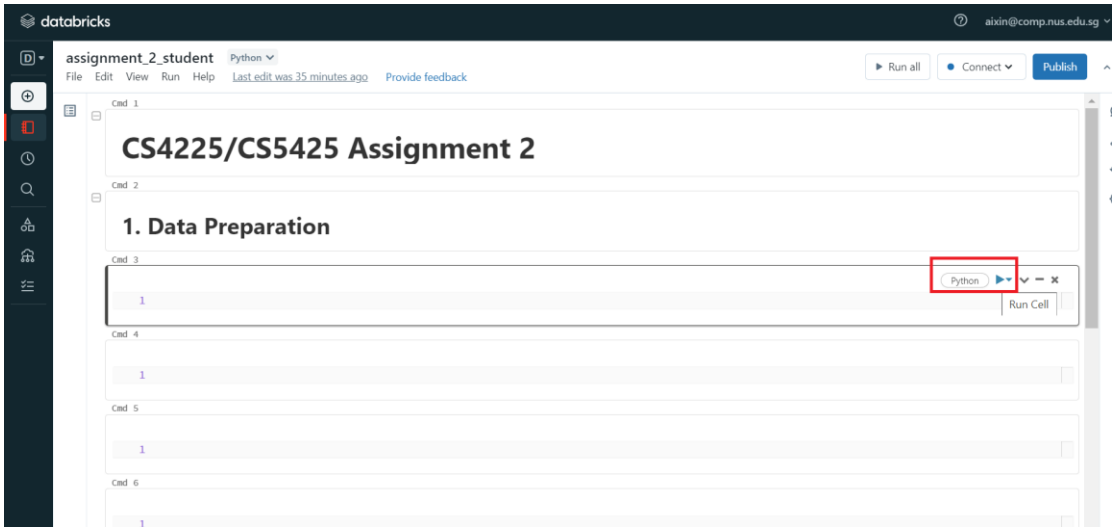


2) Click on Create, Attach, & Run and you would be ready to run your code



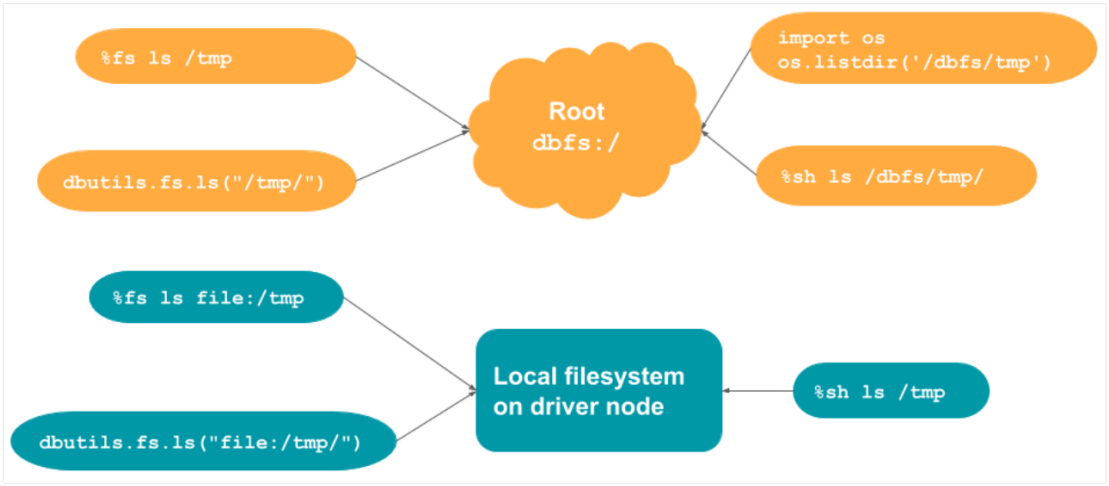
Better choose a ML runtime version for your cluster as it has more packages installed.

Now you can start to enter codes into each cell and run cell by clicking the play button at upper right corner of each cell.



2.2 How to work with files on Databricks

You can find detailed information in the below link.
https://docs.databricks.com/en/files/index.html

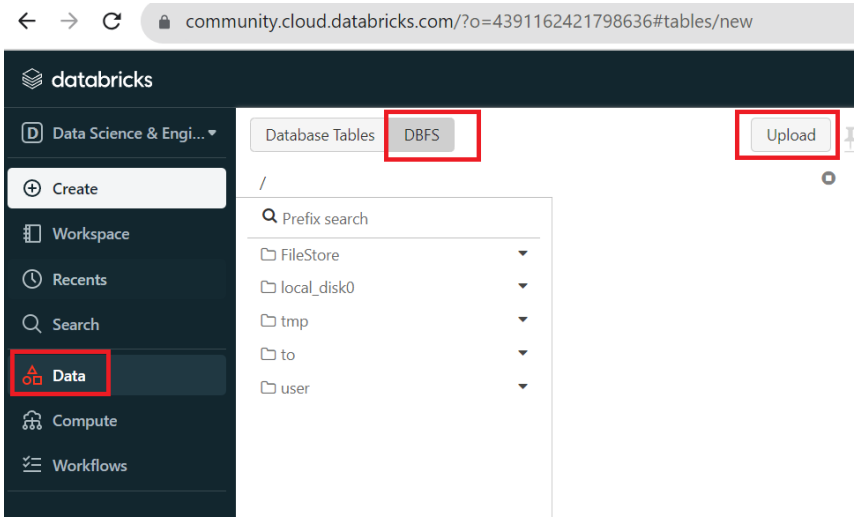


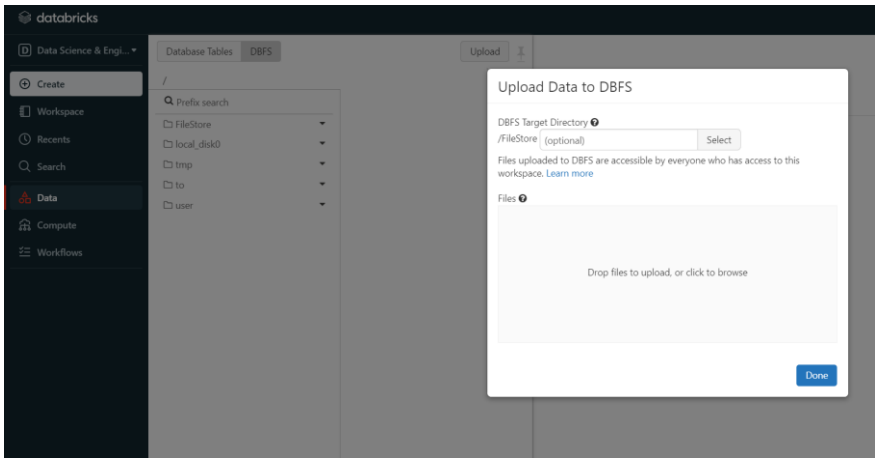
- Local filesystem: all the data will disappear after terminate / re-create the cluster.
- DBFS: persistent cloud storage. Data will still be there after terminate / re-create the cluster.

You can follow the suggestion in the below link to download zip files, unzip to csv files, move the csv files to DBFS (this may take a while) and then load them into Spark DataFrame. https://docs.databricks.com/en/files/unzip-files.html
Feel free to use other bash commands or python codes to do the same.

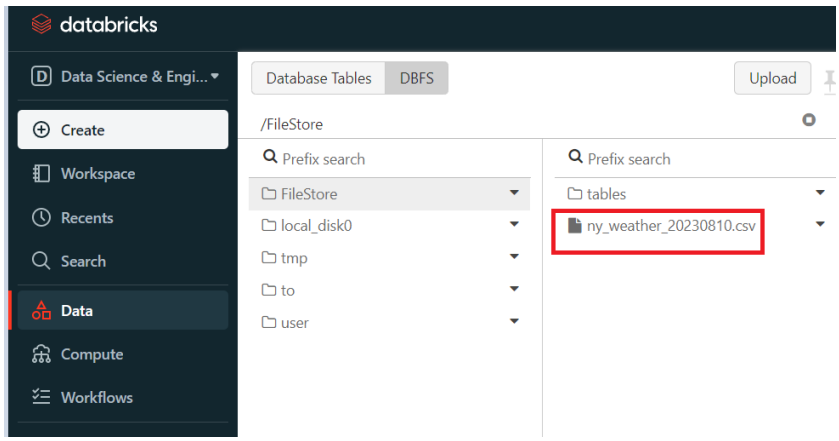
Note: *In a real cluster with a driver and multiple worker nodes, Spark can only load data from DBFS or other supported persistent storage. But for Databricks community edition, our cluster has one driver ONLY, so Spark can load data directly from driver's local filesystem. This can help you save some time, i.e. skip the step of moving the csv files into DBFS. But do bear in mind the data stored in the local filesystem will disappear after you terminate /re-create the computing cluster.*

You can upload the small data files (e.g. weather data) into databricks FileStore located at DBFS. Click on Data > DBFS > Upload to upload the file into FileStore.





Once uploaded, you can see the file under DBFS > FileStore



You can read this csv file into a spark dataframe using the below path:

```
'/FileStore/ny_weather_20230810.csv'
```

4 Submission

Download your completed Databricks notebook by clicking File > Export > IPython Notebook. We will run your .ipynb file to ensure that your code generates your given answers.

Zip all the files (as shown below) into a zip file with file name “asg2_Student Name_Matric No.zip”

- assignment_2_student name.ipynb
- weather_data.csv
- other small data files you used in your notebooks (if applicable)

Submit the zipped file to Canvas by Nov 19, 2023, Sunday 11:59pm.

Links and References

For a basic guide + API reference for Spark, see https://spark.apache.org/docs/latest/sql-getting-started.html, or https://spark.apache.org/docs/latest/api/python/getting_started/index.html

for PySpark.

For a PySpark ‘cheatsheet’, see https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PySpark_SQL_Cheat_Sheet_Python.pdf

For the information related to DBFS, see https://docs.databricks.com/en/dbfs/index.html