

How to Clean a Dataset Using Databricks, Amazon Web Services, and SQL

Jake Kravetz

Abstract—This paper presents tools such as Databricks, Amazon Web Services, and SQL that can be utilized to manipulate large datasets. This paper will go step by step on how to create accounts for Databricks and Amazon Web Services as well as how to use them to clean raw data. The explanations and instructions that follow are intended for an audience that has little to no technical expertise with using these services. A requisite knowledge of computing, SQL, and relational databases is needed to fully understand and follow the information below. I have provided all code and screenshots based off a modified version of a project I am working on.

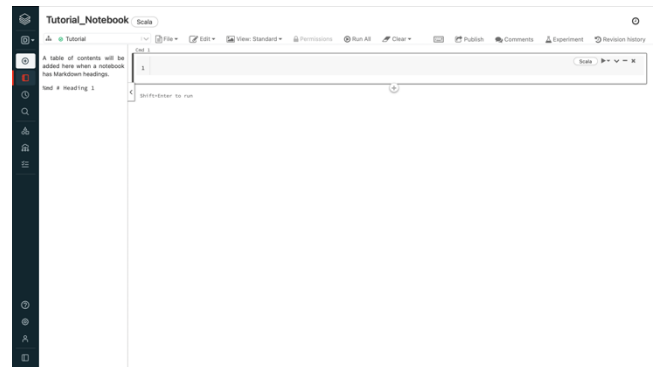
I. INTRODUCTION

The world is full of data. Wherever you go and whatever you can see, can be data that can be collected. Things such as your daily eating habits, spending habits, and workout routines are all things that can be considered data. When that data is processed it becomes usable information. Information can be incredibly useful to any person or organization looking to make calculated decisions based on factual data. For example, if you oversaw a business that sold basketball sneakers and were looking to advertise to a specific demographic of people, you can collect the data of individuals who buy sneakers. You can then take that data, process it, and then analyze the information to see what age group buys the most basketball sneakers. Afterwards you would be able to target your advertising to that specific age group so that your business can make the most profit. That is the power of data in the modern world. So, how would one go about processing data? To understand that question you first need to understand what raw data is. **Raw data** is the data that is initially taken from a source before it is manipulated in any way. This data can have issues such as having null values, typographical errors, unnecessary values, etc. The first step of processing data is to clean the raw data of its issues so that it can become viable for use. There are many other steps that are taken before data becomes usable information such as transforming and visualizing, but in this paper the focus will be on cleaning. The act of cleaning raw data can be done using tools such as **Databricks**. Databricks is a program used for manipulating large datasets and we will be using it to learn how to remove the issues that can plague raw data. So, where are we going to store the datasets before and after the data is manipulated? That is where Amazon Web Services come in. Amazon Web Services or AWS provides the servers we can use to store our data in the cloud. Data can get very large and messy, and it is important to have it all backed up in the cloud. In the next segments, you will learn step by step how to setup accounts for both services so that we use them to clean a raw dataset.

II. Procedure to Create a Databricks Account

For the purposes of this tutorial. We will be using the community edition of Databricks so that we can use the service for free.

1. To sign up for a free edition of Databricks you need click this link: <https://databricks.com/try-databricks> You then must go through the sign-up process to activate your free account.
2. Once you are logged in to Databricks you will see a black sidebar that has the option to create with a plus icon next to it. Click on create.
3. You then should click on cluster so that you can create a new cluster. (A cluster is a set of resources used to help us so we can process our data). Submit the cluster name as Tutorial and leave the runtime version and availability zone as it is. Click create cluster. This should take a few minutes.
4. After your cluster is created, click on create again. This time click on notebook. Submit the notebook name as Tutorial_Notebook. Click on the drag down menu on cluster to select the Tutorial cluster you created before. Leave the default language as it is. Click create.
5. You should then see an interface that looks like this.

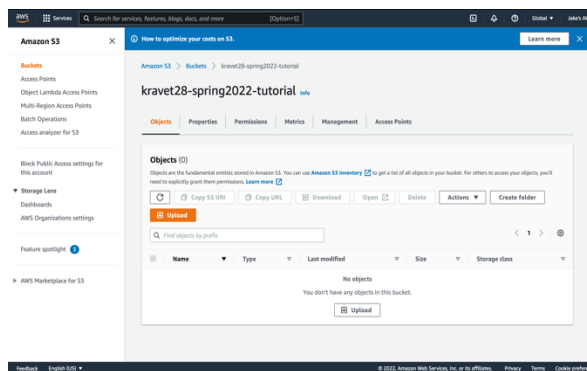


6. Your Databricks account is now ready to clean raw data!

III. Procedure to Create an AWS Account

For the purposes of this tutorial. We will be using the free tier of AWS.

1. To use the free tier of AWS, click on this link to register:
https://portal.aws.amazon.com/billing/signup?refid=em_127222&redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation%2Faccount
2. After you are registered and logged in to AWS, click on the search bar at the top of the web page. Type in S3 and click on the S3 result that comes up. (Amazon S3 is a program used to store data in “buckets”)
3. Click on the orange button that says create bucket. For the bucket name submit something unique. A good way to choose a name is to do: *the beginning of your email - semester date - what we are trying to accomplish*. For example, yours could look like John1234-spring2022-tutorial. Leave the rest of the information as default and click create.
4. You should then have an interface that looks like this:



5. You are now going to create a folder called datasets so that you have a location to upload a raw dataset to the bucket. Inside of the datasets folder create two more folders called bronze and silver. You are going to create tables in Databricks and put them in these folders. The bronze folder will house the raw data and the silver folder will house the clean data.
6. The final step before you can connect your Databricks account with your AWS account is to get the access key and secret key from the IAM console. Search IAM in the search bar on top and click on the result.
7. In the IAM console create a new root user with any name you like. After you create the new user, click on your user and then click on the security credentials tab. Select create access key to get an access key as well as a secret key. DO NOT forget your secret key as it only appears once, so keep it somewhere safe.

IV. Procedure to Connect AWS to Databricks

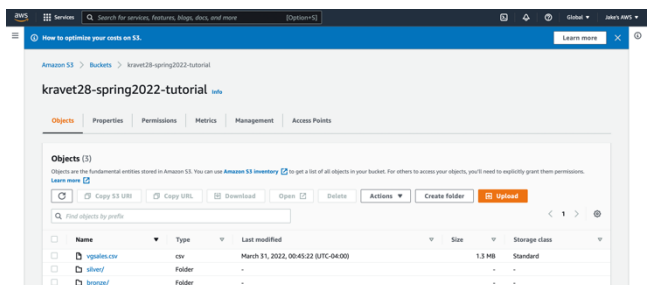
1. Now that you have setup your AWS and Databricks accounts, you are ready to connect them. Have your S3 bucket name, secret key, and access key ready.
2. In your Databricks Tutorial notebook click on the plus tab to add a command. Now follow what is done in the picture below to mount your notebook to your S3 bucket.



3. Your notebook should now be mounted to AWS! You are now ready to select a dataset to clean.

V. Selecting a Raw Dataset and Uploading to AWS

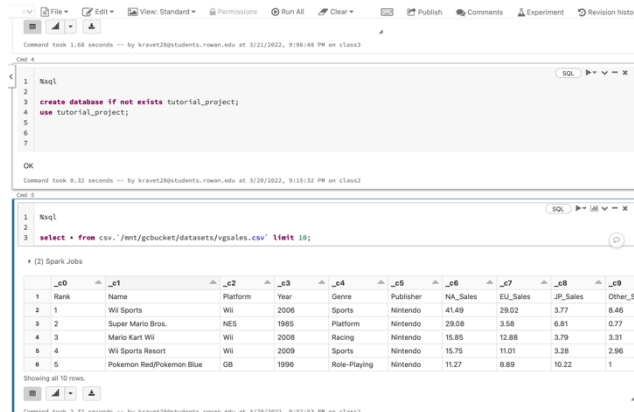
1. A great website to look for datasets is kaggle.com. As you are first starting, it would be best to download a dataset that is not too large as it may be harder to manipulate it. The dataset that will be used in this tutorial will be of video game sales from the past 5 years. It's also recommended that you use a .csv file for this tutorial. You can follow the steps done below with the dataset that you choose. You should be able to come up with your own ways to clean and query through your dataset as you should already know how to use SQL and relational databases.
2. Upload your csv file to your datasets folder in your S3 bucket.
3. Your S3 bucket should now look like this:



4. You are now ready to clean your raw data!

VI. Creating a Table for Your Raw Data

1. The first thing you are going to do is create a database to use for your table. You can do so by using SQL which will be demonstrated below. To select SQL to use for your command, type in %SQL on top. Also, you can keep clicking the plus button to create new commands.



```
1 %SQL
2
3 create database if not exists tutorial_project;
4 use tutorial_project;
5
6
7
8 OK
9 Command took 0.32 seconds --- by braver128@students.rowan.edu at 3/29/2022, 9:15:32 PM on class3
10
11 %SQL
12
13 select * from csv.`/mnt/gcubucket/datasets/vgsales.csv` limit 10;
14
15 (2) Spark Jobs
16
17 Rank Name Platform Year Genre Publisher NA_Sales EU_Sales JP_Sales Other_Sales
18 1 Wii Sports Wii 2006 Sports Nintendo 41.49 29.02 3.77 8.46
19 2 Super Mario Bros. NES 1985 Platform Nintendo 29.08 3.58 6.81 0.77
20 3 Mario Kart Wii Wii 2008 Racing Nintendo 15.85 12.88 3.79 3.31
21 4 Wii Sports Resort Wii 2009 Sports Nintendo 15.75 11.01 3.28 2.96
22 5 Pokemon Red/Pokemon Blue GB 1996 Role-Playing Nintendo 11.27 8.89 10.22 1
23
24 Showing all 10 rows.
```

As you can see above, a database named `tutorial_project` has been created. To test it, you can run a simple select statement that pulls the data from your S3 bucket.

2. Now you are going to create a table for the raw data and store it in your datasets folder.



```
1 %SQL
2
3 create table rawdata_vgsales using csv
4 options (path = '/mnt/gcubucket/datasets/vgsales.csv.bz2'
5          ,header = 'true',inferSchema = 'true');
6
7 (2) Spark Jobs
8
9 OK
10 Command took 3.93 seconds --- by braver128@students.rowan.edu at 3/21/2022, 8:44:38 PM on class3
```

Follow the code above to create a table that also removes the header.

3. After you have created a raw data table for the dataset, you are going to use a Databricks Delta table so that you can start cleaning. (Delta tables make it easy to manage the data). Follow the code below to create a delta table and place it in your bronze folder in S3.

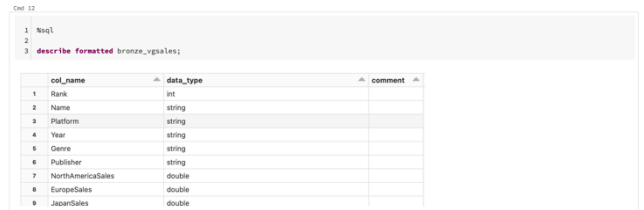


```
1 %SQL
2
3 create table bronze_vgsales using delta
4 location '/mnt/gcubucket/datasets/bronze/'
5
6 as
7 select Rank, Name, Platform, Year, Genre, Publisher, NA_Sales as NorthAmericaSales, EU_Sales as EuropeSales, JP_Sales as JapanSales
8 --Removing Other_Sales and Global_Sales as those columns are unnecessary.
9 --This dataset should tell me the different genres that each of North America, Europe, and Japan likes best.
10 from rawdata_vgsales;
11
12 (4) Spark Jobs
13
14 Query returned no results
15 Command took 18.55 seconds --- by braver128@students.rowan.edu at 3/21/2022, 8:15:36 PM on class3
```

Part of cleaning data is removing unnecessary columns that do not contribute to what you want to use the data for. For example, in the picture above, certain columns were not included in the delta table for this purpose. Another part of cleaning data is using acceptable column names. In the example above, the names of some of the columns are changed to be made clearer.

VII. Cleaning the Data

1. In the last segment, you have begun to clean the data by creating a fixed table. Now you are going to check the data types for each column to make sure they are correct. If there is a problem with your data types, you can write a SQL command to fix it.



```
1 %SQL
2
3 describe formatted bronze_vgsales;
```

col_name	data_type	comment
Rank	int	
Name	string	
Platform	string	
Year	string	
Genre	string	
Publisher	string	
NorthAmericaSales	double	
EuropeSales	double	
JapanSales	double	

In this case, there are no problems with the data types, but your dataset may have some.

2. Another way you can clean your data is to fix typographical errors. The video game sales dataset used in this tutorial misspells “3DS” and the code below shows how to fix it.



```
1 %SQL
2
3 update bronze_vgsales
4 set
5 platform = REPLACE(platform, '3DS', 'DS');
6
7 (4) Spark Jobs
8
9 num_affected_rows
10 1 16598
11
12 Showing all 1 rows.
```

This is a good method to use in the dataset you choose.

3. The next step you can do is check for null values in all your columns. You can remove the null values to clean the data. The code below checks for null values in one column. You should check all columns and remove values as needed.



```
1 %SQL
2
3 select * from bronze_vgsales where Name is not null
4
```

4. Finally, you can come up with your own creative ways to fix issues in your dataset. After you are done cleaning your raw data, you will be ready to create another table.

VIII. Creating a Table for Your Cleaned Data

1. Now that your data is cleaned, you can create a new delta table for it. You should place the new table in your silver folder so that you keep your raw data and cleaned data in separate folders.

```
1 sql
2
3 create table silver_vgsales using delta
4 location '/mnt/gcbucket/dataset/silver/'
5
6 as
7 select Rank, Name, Platform, Year, Genre, Publisher, NorthAmericaSales, EuropeSales, JapanSales
8
9 from bronze_vgsales;
```

2. You have now successfully cleaned a dataset using Databricks, AWS, and SQL. What more is there to do now? Just because the data is cleaned does not mean you are done with it. Now that your data is clean, you are free to move on to other steps in the process such as visualizing the data and analyzing the data. Hopefully you can take away something from this tutorial and if it all goes well from here you will have information worth using.

IX. References

- [1] N. Esther, *How to Create an AWS Account*, Medium, Oct. 6, 2021. Accessed on: March 30, 2022. [Online]. Available: <https://aws.plainenglish.io/how-to-create-aws-account-b85738e59b39>
- [2] P. Dubey, *Free Community Edition Databricks Account*, Medium, June 30, 2020. Accessed on: March 30, 2022. [Online]. Available: <https://medium.com/@prateek.dubey/free-community-edition-databricks-account-b53f3e2e94b8>
- [3] P. Dubey, *Databricks on AWS*, Medium, July 9, 2020. Accessed on: March 30, 2022. [Online]. Available: <https://medium.com/@prateek.dubey/databricks-on-aws-ee043ae937f>