Homework 4 Part A

10-405/605: Machine Learning with Large Datasets

Due Wednesday, March 22nd at 11:59:59 PM Eastern Time

Instructions: There are two parts to this homework, which will have **different deadlines**.

- Part A (i.e., this document) is due on March 22nd and is worth 20% of the grade. **No Grace days can be used on this part of the homework.**
- Part B is due on March 29th and is worth the remaining 80% of your grade.

Both 10-405 and 10-605 students are required to complete all of Homework 4 Part A.

Submit your solutions to this part to the Homework 4: Part A submission slot on Gradescope, following the template below. You do not need to submit any code for part A.

Submitting via Gradescope: When submitting on Gradescope, you must assign pages to each question correctly (it prompts you to do this after submitting your work). This significantly streamlines the grading process for the course staff. Failure to do this may result in a score of 0 for any questions that you didn't correctly assign pages to. It is also your responsibility to make sure that your scan/submission is legible so that we can grade it.

0 HW4: Machine Learning with the Million Song Dataset

The goal of this assignment is to gain hands-on experience with training a machine learning model on a large dataset on cloud computing services. There are two main parts to the assignment:

1. **Part A: Data Conversion and Preparation:** This part will include setting up Amazon EC2, accessing and converting the Million Song Dataset (MSD) into a usable format, and performing initial analyses on the dataset. Although we give you the code to perform data conversion, please ***START EARLY*** because data conversion might take several hours.

2. **Part B: Modeling:** This part will include setting up Amazon EMR and running Spark to perform exploratory data analysis (EDA), data cleaning, and modeling with some (very limited) starter code. At the end, you will have a chance to optimize your pipeline and model.

0.1 Logistics

For Part A (Data Conversion and Preparation), follow the instructions in Section 1 and the corresponding code `million_song_prep.py`.

For Part B, follow the instructions in the notebook `hw4.ipynb` and implement the missing parts marked with '`# YOUR CODE HERE`'. **Unlike the previous homework, this homework gives you more freedom and less guidance so you may need to complete a cell from scratch.** That said, each cell will be a small task which should be straightforward to implement following our instructions.

Note that we will not autograde your code through Gradescope. Instead, you should submit your code along with your report (including plots, statistics, and short answers). Points will be given according to your answers in the report. We may refer to your code submission in case we have doubts. **You must submit both your report and your code for Part B to receive credit.**

0.2 Getting lab files

You can find the starter code for **HW4**, `million_song_prep.py` and `hw4.ipynb`, in the handout file `hw4.zip` which you downloaded from Github.

0.3 Preparing for submission

There are two separate deadlines for this homework.

- **For Part A (Due 3/22 at 11:59pm):** You must run the million_song_prep.py file on the million song dataset and provide an image of your S3 Bucket containing the converted files with your AWS username visible and the count of the number of datapoints. *Worth 20% of HW4.*

- **For Part B (Due 3/29 at 11:59pm)**: You must complete the hw4.ipynb on an EMR cluster and run your notebook on the full dataset and provide numerous result and graphs and also submit your completed notebook to the programming submission slot. Part B will take the longest to complete! *Worth 80% of HW4.* You must submit your hw4.ipynb to the Homework 4: Programming submission slot.

0.4 Part A Submission

1. Replace the image file myS3bucket.png with a screenshot of your S3 bucket. The image must show your bucket and your username in the upper right corner. This image must be a direct screenshot and should not be altered in any way. You can see an example in the current myS3bucket.png.

2. You only need to submit your completed part A PDF write-up to Gradescope's Homework 4 Part A submission slot.

1 Part A: Data Conversion and Preparation

To get ready for the main part of the homework (Part B), you will first need to set up your AWS account. This section will also guide you through accessing the million song dataset (MSD) and transforming it from `h5` format to `csv`, which will be necessary to complete Part B.

PLEASE PLAN TO START EARLY. One reason is that the entire MSD is 280GB and running data conversion on it might take a few hours. It will be frustrating if your code runs for several hours before the deadline then aborts due to a tiny bug and you have to re-run. You may also run into roadblocks when working with a large dataset on the cloud, especially if this is your first time performing such an exercise. Although we have included several tips in this write-up, you may still encounter some unexpected problems. Luckily, search engines are always your friend. If you have problems with something, someone elsewhere might have encountered the same problem! We recommend that you first try searching for solutions to your problems online (as this will likely be the fastest route to an answer), and if you're still stuck to then post on Piazza and/or come to office hours.

We also recommend that you revisit the lecture on October 5th if you have any issues with this portion of the assignment. The lecture walks you through getting set up on AWS and starting EC2/EMR clusters.

1.1 Setting up AWS

Amazon Web Service (AWS) is one of the most comprehensive cloud computing platforms. Although there are many other options for cloud computing (e.g. Azure, GCP), we ask that you use AWS for this particular assignment. This makes it much easier for us to support the entire class.

1.1.1 Create AWS account & redeem credits

First, if you filled out the AWS credits request form, confirm that the $100 have been added to your account by heading the the Billing Information section and clicking on the credits section.

1.1.2 Cost management

Make sure you manage your cost while doing this assignment. To do this effectively, we highly recommend you learn about EC2 pricing and S3 pricing, and **create an AWS Budget of 30 dollars so that you are alerted when you are close to your budget limit.** CMU will not be responsible for covering extra charges incurred beyond the supplied coupon.

1.1.3 AWS S3

You have to create a S3 bucket in order to store your converted MSD into it.

1.1.4 IAM Role

To have read/write access of S3, you also have to create a role with the `AmazonS3FullAccess` policy attached under `IAM -> Roles -> Create role` (here).

1.1.5 Configure and launch EC2

Configure and launch EC2 instances to develop and run your data conversion. Here are some guidelines:

1. Choose the default image (i.e. something like `Amazon Linux 2 AMI (HVM)`). Then choose `t2.micro` or `t2.medium`. Note that `t2.micro` can be used for development, but might not have enough memory to process your entire dataset.
2. Select a subnet starting with **"us-east-1"**. Remember your choice since you have to select the same subnet for the volume you are going to create.
3. Choose the IAM role you just created.
4. Keep options default and move on. You will need to select a security group (this can be changed after creating the instance). You may select the "default" security group and add an inbound rule to it for the `SSH` service with `source` set to be `Anywhere`.
5. Create a key-pair if you haven't already.

The image might not come pre-installed with Python 3 so you could do so via `$ sudo yum install python37`. You should also install packages you import in the python file. Use `pip3` with the `--user` flag to avoid `sudo`.

Again, cost management is key. You could launch a single `t2.micro` while you are developing to save costs. You should launch up to two `t2.medium` instances while you run the actual conversion. You should stop (**not** terminate, see distinction here) your instances when you are not using them for a while. When an instance is stopped, you will only be charged EBS storage fees of your data but not instance running fees. You should terminate them **after you have downloaded your code** when you are done.

1.1.6 Create MSD volume

Create a volume from the AWS MSD snapshot. Select the same "subnet" as the EC2 instance's, (an example subnet might be "us-east-1d"). **Remember the subnet must start with "us-east-1", or the snapshot of MSD cannot be found**. Attach and mount the MSD volume to the EC2 instance. Remember to delete this volume after you are done with this homework.

1.1.7 Credentials

You may not need to do this in this homework. But if you want to use the AWS command line tool, you may have to configure credential files in order to let your program access AWS services. The easiest way would be to run `$ aws configure`. See Configuration and credential file settings for details.

1.1.8 Development on AWS

You could use whatever IDE/editor you are comfortable with. For starters with remote development, you could try Visual Studio Code with the Remote-SSH extension (see details here) or Vim.

1.2 Converting the Data

The handout file `million_song_prep.py` is used to convert the Million Song Dataset in your S3 bucket.

To run `million_song_prep.py`, you would first need to copy this file into your EC2 instance by using the following command on your local terminal:

```
scp -i <your_keypair> <file> <destination>
```

After that you will need to change the S3 bucket name to be the one you created when you set up your EC2 instance, update the path to the million song dataset and create a folder called `processed`. Finally, install all python packages required and run `million_song_prep.py` to convert the dataset.

[20 points] Update the file myS3bucket.png to be a screenshot of your S3 bucket with your converted data. In this image we should be able to see the name of your account in the upper right corner, the name of your S3 bucket which needs to be globally unique in AWS and a list of the files in the S3 Bucket the image should look exactly like the current image but with your details.

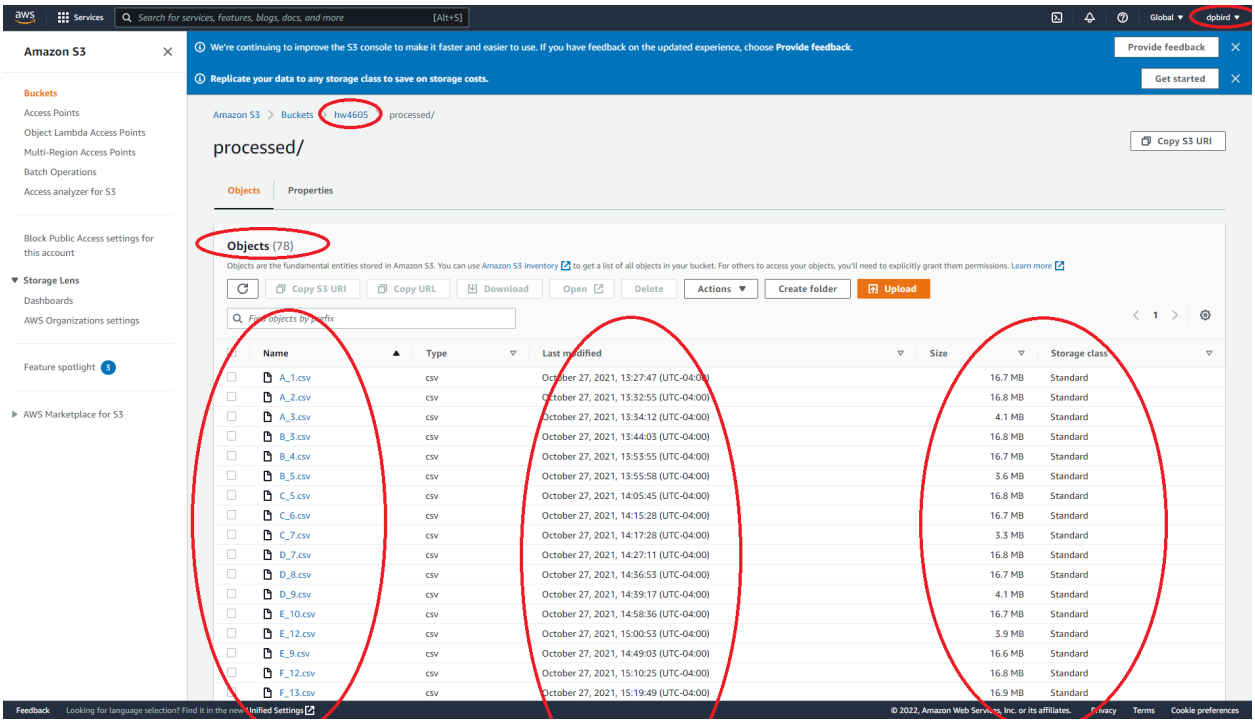

Figure 1: MyS3bucket.png