Homework 4

10-405/605: Machine Learning with Large Datasets

Due Monday, March 21st at 3:00:00 PM Eastern Time

Instructions: Submit your solutions via Gradescope, following the template below. Note that this assignment does not contain a theoretical written part and the programming part is more open-ended than previous homework. Because of this, we will not be autograding your submission. Instead, you will submit a report consisting of responses to questions asked in the notebook. The report is worth 100% of your grade. We may refer to your code submission to verify your work or in case we have any doubts. You must submit your code to Gradescope under the HW4 Programming submission slot to receive credit for the assignment.

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

Although you must run the code that we provide in Part A before you can complete Part B, you are only required to submit the code and write-up from Part B. To submit your work, complete the `hw4.ipynb` and corresponding written questions using this `hw4.pdf` write-up as a template.

0.4 Submission

1. Download the notebook to your local computer by going to `File -> Download as -> Notebook (.ipynb)` and complete relevant written questions in this document.

2. You only need to submit `hw4.ipynb` and this `hw4.pdf` write-up to Gradescope.

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 and redeem your $50 credits. 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 March 2nd 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, create an AWS account if you don't already have one. You will receive an email from the course staff containing a $50 coupon code on AWS. You can redeem it at `Billing Dashboard -> Credits`. **Note that this $50 coupon is meant to cover your costs for ALL OF HOMEWORK 4**.

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 35 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 (note: we covered this process in detail in lecture on Wednesday March 2nd):

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. Finally, install all python packages required and run `million_song_prep.py` to convert the dataset.

2 Part B: Modeling

In this part of the homework, you will perform exploratory data analysis (EDA) and data cleaning, and then train models with the original features. You will then perform feature engineering similar to what we did in Homework 1 (TF-IDF and Bag-of-Words), and then train models with these new features.

2.1 Setting up EMR and Spark

With our data ready in S3, it's now time to configure and create an EMR (Elastic MapReduce) cluster and run Spark, starting from the notebook `hw4.ipynb`. Include at least Hadoop, JupyterHub, and Spark in your cluster software configuration. Set `maximizeResourceAllocation` to true (see here) in software settings. Select your EC2 key pair.

You'll have to do ssh port forwarding (recommended) or attach additional security groups/inbound rules to the master node in order to access the JupyterHub web interface. This step could be tricky and we cover how to do this in detail in lecture. Then, log in to jupyter (learn about login credentials here), upload your notebook, and start working!

Again, cost management is key. Because we might be running a cluster of machines, this could easily blow up your budget. We recommend using at most 1 Driver and 1 Core of type `m5.xlarge` while developing and debugging on a subset of MSD. You could scale this up to multiple Core workers when doing the final run. You may also utilize AWS spot instances to save cost during development.

Note that, although EMR is made up of EC2 instances, unlike EC2, you cannot stop an EMR cluster—you can only terminate it. See here for why this is the case. You should plan your strategy accordingly. **Do not forget to download your code** before you terminate a cluster when you are done.

2.2 Preprocessing

To run the Jupyter notebook, `hw4.ipynb`, for preprocessing, you would first need to copy this file into your cluster by using the following command on your local terminal:

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

After that ssh into your cluster by using the following command:

```
ssh -i <your_keypair> -N -L localhost:<local_port>:<cluster>:9443 hadoop@<cluster>
```

Change the S3 bucket name to be the one you created when you set up your EC2 instance while loading data in Part A.

2.3 Exploratory Data Analysis

(a) *[2 points]* Explain why the two features seem problematic (after performing .summary() operation).

(b) *[3 points]* Histograms (remember to label them)

(c) *[3 points]* Explain what is strange about `year`'s distribution and what might cause this. Describe how you could filter `year` to make its histogram look more balanced.

(d) *[2 points]* New histogram for `year`.

(e) *[5 points]* Provide plots for the three pairs. Describe your findings.

(f) *[3 points]* Think about what simple technique you could use to visualize large datasets while retaining a similar data distribution. Briefly describe what you did.

2.4 Data Cleaning

(a) *[2 points]* Your justification for dropping the two features.

(b) *[5 points]* Compare the two numbers and explain the advantages and potential problem of doing this step. What other techniques could you use to potentially do better?

(c) *[2 points]* State the two features.

(d) *[6 points]* Explain your proposed solution and discuss its pros and cons.

(e) *[2 points]* Report the percentage:

2.5 Baseline

(a) *[3 points]* Explain why treating this as a classification problem might be a sensible choice.

(b) *[2 points]* Report what percentage of songs are assigned the "popular" label.

(c) *[2 points]* Explain why we shift the year.

(d) *[5 points]* Explain what scaling means and why we want to perform scaling before the learning step.

(e) *[5 points]* Explain the difference between these two metrics and when AUC might be more useful than accuracy.

(f) *[8 points]* Calculate the train and test AUC of both models and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

2.6 Featurization: Bag-of-Words and TF-IDF

(a) *[3 points]* Explain what the `vocabSize` hyperparameter means in the context of Bag-of-Words.

(b) *[3 points]* Other than featurizing texts, what other feature engineering would you do on the dataset? Briefly describe one.

(c) *[3 points]* Explain where this number "31" comes from.

2.7 Modeling with New Features

(a) *[8 points]* Evaluate train and test AUC for each model and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

(b) *[8 points]* Include the plot and your explanations.

2.8 Do Your Best

(a) *[2 points]* Your final AUC:.

(b) *[4 points]* Your model and hyperparameters.

(c) *[4 points]* Describe your approach.

2.9 Reflection

[5 points] What challenges did you face in HW4? How did you overcome these challenges? What did you learn from HW4 ?”

3 Collaboration Questions

1. (a) Did you receive any help whatsoever from anyone in solving this assignment?

   (b) If you answered ‘yes’, give full details (e.g. “Jane Doe explained to me what is asked in Question 3.4”)

2. (a) Did you give any help whatsoever to anyone in solving this assignment?

   (b) If you answered ‘yes’, give full details (e.g. “I pointed Joe Smith to section 2.3 since he didn’t know how to proceed with Question 2”)

3. (a) Did you find or come across code that implements any part of this assignment?

   (b) If you answered ‘yes’, give full details (book & page, URL & location within the page, etc.).