Homework 4 Part B

10-405/605: Machine Learning with Large Datasets

Due Wednesday, March 29th at 11:59PM Eastern Time

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

- Part A 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 (i.e., this document) is due on March 29th and is worth the remaining 80% of the grade.

IMPORTANT: Be sure to highlight where your solutions are for each question when submitting to Gradescope otherwise you will be marked 0 and will need to submit regrade request for every solution unhighlighted in order for fix it!

Note that Homework 4 Part B consists of two parts: this written assignment, and a programming assignment. Remember to fill out the collaboration section found at the end of this homework as per the course policy.

10-405 Students are required to complete the following:

1. All of Homework Part A, which must be submitted by 3/22 *[20 points]*
2. All of Section 3 *[65 points]*

Your homework score will be calculated as a percentage over the maximum points you can earn, which is 85.

10-605 Students are required to complete **all** sections of both parts of the homework. Your homework score will be calculated as a percentage over the maximum points you can earn, which is 100.

Programming: The programming in this homework is **NOT** autograded however you are required to upload your completed notebooks to Gradescope, otherwise you will not receive credit for the programming sections.

1 Part A: Data Conversion and Preparation

Please complete Homework 4 Part A following it's write-up before attempting this part.

2 Part B: Written

2.1 Distributed Decision Trees

****10-605 Students Only****
We are given the below dataset. It contains 8 instances, each having 3 features (X_1,X_2,X_3), and output Y. We are also given $M = 2$ worker machines. The first four instances are placed on worker 1, and the second four instances are placed on worker 2.

	X_1	X_2	X_3	Y
1	1	2	3	1
2	4	5	6	2
3	7	8	9	3
4	10	11	12	4
5	12	11	10	5
6	9	8	7	6
7	6	5	4	7
8	3	2	1	8

You are planning to create a row-partitioned decision tree using your distributed training data. As a first step, you are considering using the $X_1 > 5$ splitting rule in the root node, and you want to calculate the information gain of this predicate.

Your goal in this exercise is to i) calculate the messages that the worker machines will send to the driver machine, and then ii) calculate the information gain of this splitting rule.

As a reminder, the information gain of the splitting rule $X_k > a$ in a node is defined as

$$Gain(X_k > a) \doteq D\,\mathrm{Var(Y)} - D_L\,\mathrm{Var}(Y \mid X_k > a) - D_R\,\mathrm{Var}(Y \mid X_k \leq a),$$

where D is the size of the training data in the node, and D_L and D_R are the number of the training points in the left and right branches of the node, respectively.

To calculate the gain of the $X_1 > 5$ splitting rule, we will use the map-reduce based algorithm discussed during the lecture. In what follows, we will use the notation introduced in the lecture slides.

2.1.1 Task 1

[4 points] Since we want to calculate the gain of the $X_1 > 5$ predicate, let $k = 1$ and $a = 5$.

Map step: Calculate the values of the following function for all the 8 training instances:

$$g([x_1, x_2, x_3], (a, k)) \doteq (1, y, y^2, R(x_k), R^2(x_k), L(x_k), L^2(x_k), 1_L(x_k), 1_R(x_k))$$

The $R(x_k)$, $L(x_k)$, $1_L(x_k)$, $1_R(x_k)$ functions are defined in the lecture slides.

To complete this task, fill out the below tables:

Task 1:

On machine 1:

	1	y	y^2	R	R^2	L	L^2	1_L	1_R
1									
2									
3									
4									

On machine 2:

	1	y	y^2	R	R^2	L	L^2	1_L	1_R
5									
6									
7									
8									

2.1.2 Task 2

[2 points] Calculate the messages of the two workers to the driver: $g_j \doteq \sum_{(x,y)\in\mathcal{I}\cap\mathcal{J}} g(x,(a,k))$

As discussed during the lecture, for a node i, define $\mathcal{I}$ as the set of instances $\{(x,y)\}$ that belong to this node. Let $\mathcal{J}$ denote the set of instances $\{(x,y)\}$ that worker j contains. Fill out the below tables:

$g_1 =$

1	y	y^2	R	R^2	L	L^2	1_L	1_R

$g_2 =$

1	y	y^2	R	R^2	L	L^2	1_L	1_R

2.1.3 Task 3

[2 points] **Reduce Step**: The server uses a reduce operation to aggregate the messages.

$$g^* = \sum_{j=1}^{M} g_j$$

Calculate g^*. Fill out the below table:

Task 3:

$g^* =$

1	y	y^2	R	R^2	L	L^2	1_L	1_R

2.1.4 Task 4

[5 points] Using the aggregated message g^*, calculate the D, D_L, D_R, $\text{Var}(Y)$, $\text{Var}(Y \mid X_k > a)$, $\text{Var}(Y \mid X_k \leq a)$ quantities.

2.1.5 Task 5

[2 points] Finally, calculate the information gain with the above terms:

$$Gain(X_k > a) \doteq D\,\mathrm{Var(Y)} - D_L\,\mathrm{Var}(Y \mid X_k > a) - D_R\,\mathrm{Var}(Y \mid X_k \leq a)$$

3 Part C: 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.

3.1 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.

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. You should plan your strategy accordingly. **Do not forget to download your code** before you terminate a cluster when you are done.

3.2 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.

3.3 Exploratory Data Analysis

(a) *[1 points]* Explain why the two features seem problematic (after performing .summary() operation).

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

(c) *[2 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) *[1 points]* New histogram for `year`.

(e) *[3 points]* Provide plots for the three pairs. Describe your findings.

(f) *[1 points]* Think about what simple technique you could use to visualize large datasets while retaining a similar data distribution. Briefly describe what you did.

3.4 Data Cleaning

(a) *[1 points]* Your justification for dropping the two features.

(b) *[3 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) *[1 points]* State the two features.

(d) *[3 points]* Explain your proposed solution and discuss its pros and cons.

(e) *[1 points]* Report the percentage:

3.5 Baseline

(a) *[2 points]* Explain why treating this as a classification problem might be a sensible choice.

(b) *[1 points]* Report what percentage of songs are assigned the "popular" label.

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

(d) *[2 points]* Explain what scaling means and why we want to perform scaling before the learning step.

(e) *[3 points]* Explain the difference between these two metrics and when AUC might be more useful than accuracy.

(f) *[4 points]* Calculate the train and test AUC of both models and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

3.6 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) *[2 points]* Explain where this number "31" comes from.

3.7 Modeling with New Features

(a) *[4 points]* Evaluate train and test AUC for each model and report them.

Models	Train AUC	Test AUC
Logistic Regression		
Random Forest		

(b) *[6 points]* Include the plot and your explanations.

3.8 Do Your Best

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

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

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

3.9 Reflection

[3 points] What challenges did you face in HW4 Section 3? How did you overcome these challenges? What did you learn from HW4?

4 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.).