Project 2 Overview

Computer Science 143 - Spring 2019

This project is still fairly new to CS 143, so we encourage you to comment on this doc, or ask questions on Piazza. I am happy to fix any data issues that may arise.

Just like with Project 1, the point of this project is to teach you something useful, and hopefully have fun.

In this second project, students will have to implement an ETL job using Spark, a very important big data processing system.

You should find part of Project 2 to be similar to Project 1 except with Spark, and a bit of machine learning, but the machine learning will be gentle.

Prerequisite

Prerequisites are the same as with Project 1 (how convenient!). We expect familiarity with the Unix environment as well as the ability to code effectively in Python, or can quickly get up to speed. We expect the resources and links on the project pages will provide you with enough information to get you started even in case you are not familiar with either one.

While Spark was initially developed to work natively with Scala, it was later extended to work natively with Python, so we will use Python for this project.

System Setup

To help students set up the uniform environment for the class project, we will again be using VirtualBox to run the Linux operating system in a virtual machine. VirtualBox allows a single machine to share resources and run multiple operating systems simultaneously. You will need to download the following files

- VirtualBox binary file for your host Operating System if you haven't already
- VirtualBox Image: CS143-P2.ova (requires UCLA BOL login) - This is a very large file (~3-4GB), so it may take a while to download.

As of 5/12/19 1:30pm, the proper hash for CS143-P2.ova is `7158ebb33d11d27665961fb7a58c78e1` (MD5) and `e3ee5b39bf15dc6122a8874556f6d884f71590db` (SHA1).

Please follow our VirtualBox setup instruction to install VirtualBox on your own machine.

The provided virtual machine image is based on Ubuntu 19.04, PostgreSQL 11.2 (for homework), Apache 2.4.27, Python 3.7.3, IPython 7.5.0, Spark 2.4.3 and R 3.5.2.

If you have access to an equivalent machine that has these packages, you may use it instead of the virtual machine image. However, please note that we cannot provide support for systems other than the virtual machine image, and that your project MUST be runnable on the provided virtual machine. We will be using the virtual machine image for grading purposes, and if your submission does not work within this setup, you may get zero points. We cannot make any exceptions to your project schedule for problems incurred by using your own computing facilities.

Project

Reddit (reddit.com) is a hybrid URL bookmark and/or news site, and a forum. Redditors post articles, links, or text on specific topics and other Redditors vote the submission up (meaning "like") or down (meaning "don't like"), and can also write comments in response to the submission. These comments can be nested very deeply in a submission.

Reddit is divided up into thousands of "subreddits" which are essentially forums dedicated to a particular topic.

Politics is the subject that everyone either loves, hates or loves to hate. But, it is also a very polarizing and emotional subject, so it provides a good basis for doing some machine learning. It is also the most active subreddit on Reddit, so there is a lot of data we can use. An election year is coming up, so we will use this Reddit data to find the sentiment across time, across topics, and across states regarding President Trump. I do not know what results we will get. It may be that this data is useless, but we do not know until we try to use it!

Caveat: /r/politics, and Reddit in general, is known to bias heavily towards young (18-29, 59%) males (71%) that lean liberal / progressive / left (47% of overall users, but much higher *anecdotally* in /r/politics), so we are already starting with biased data, but still, let's see what we can get from it.

Jason Baumgartner has collected every Reddit submission and every Reddit comment posted on the site since 2005. This data is available on his website in `bz2` or `xz` format.

Your project is to parse the comments (Python or Scala), use what you have learned in 143 about joins etc. to massage the data into the proper format (Spark and/or SQL in Spark) for training a *sentiment classifier* (Spark MLLib). Then, generate a document containing your findings with plots and answers to some questions.

Part A: Text Parsing

In Part A, you will write a function, in Python to take horribly messy text from Reddit comments, and parse them into a smooth format that we can eventually use to train a classifier.

Due Date: Monday, May 20, 2019, 11:59pm

Part B: Transforming Data, Training and Evaluating a Classifier and

In this part, you will use the function you wrote for Part A to parse the text (if you didn't already do it in Part A), into a data frame containing not only text, but several other features using Spark SQL. You will then train a classifier using a Spark package called mllib. You will then use this classifier to study the wider population on Reddit. You will then prepare a "mini-report" that has several plots and answers questions about the data.

Due Date: Monday, June 3, 11:59pm

~~Part C: Dashboard~~

~~Depending on time, you will then create a dashboard with a few visualizations about sentiment has changed over time, over location, and other interesting findings.~~

~~*Due Date:* Wednesday, June 6, 11:59pm~~

Groups

Students may implement the project individually or in teams of up to two.

An identical amount of work is expected and the same grading scale is used for individual and team projects. Faculty experience indicates that in general it is not necessarily easier or more productive to work in teams of two - it's largely a matter of personal preference and working style. If you choose to work as a team, you are encouraged to make use of collaborative authoring tools for synchronizing your work and ideas, such as version control software (e.g. CVS, SVN, Perforce, Private git) and online document tools (e.g. Adobe Share, Buzzword, Google Docs, Dropbox Paper).

If you work in a team, choose your partner carefully. Teams are permitted to "divorce" at any time during the course (due to incompatibility, one partner dropping the course, or any other reason), and individual students may choose to team up as the project progresses, however students from divorced teams may not form new teams or join other teams. Put another way, if a student turns in any part of the project as part of a team, every later part of the project must be turned in individually or as part of the same team.

Both partners in a team will receive exactly the same grade for each project part turned in jointly. We will not entertain any complaints of the form "I did all the work and my partner did nothing." Choose your partner carefully!

If you work in a team, your work must be turned in jointly, as ONE submission. That is to say, only ONE of you two should submit your work as a team. Your team will get 10 points off as penalty if you violate this rule. Note that teamwork turned in as individual work will be considered as plagiarism and handled through official University channels.

Late Submission Policy

To accommodate the emergencies that a student may encounter, each student (or team) has a 4-day grace period for late submission to use throughout the quarter, 2 per project. If you did not use late days for Project 1, you may use 3 or 4 for this project. Note that the grace period can be used in the unit of one day. even if a student submits a project 12 hours late, he/she needs to use a full day grace period to avoid late penalty.

Electronic submission of Projects

All project submission should be done electronically. Steps for submitting your project electronically is as follows:

1. Visit the online submission page for the particular project, linked from the corresponding assignment page.
2. If you need to resubmit something, just redo these directions. The submission page will notice if you are attempting to resubmit and overwrite your previous entry. Remember that only the very last submission will be graded.

Project References

Unix & VirtualBox

- Unix tutorial

- VirtualBox overview

Python references

- Python 3 Reference
- Python 3 tutorial
- Another great Python 3 tutorial
- Python Regular Expression Tutorial
- Python function reference
- W3Schools Python Tutorial
- "The" Python Tutorial
- Python for Beginners
- The Python Guru

Spark references

- UCLA Brief Tutorial on Spark MLLib (Go Bruins!)
- Spark Quickstart
- Hands On Tutorial of Apache Spark in 10 Minutes
- Machine Learning in Spark (includes info about PySpark)