BerkeleyX: CS105x Introduction to Apache Spark

Bookmark

Welcome to CS 105.1x and Course Objectives/Prequisites

BERCS1052016-V000400

Download video Download transcript .srt

This course requires a programming background and experience with Python 2.7 (or the ability to learn it quickly). All exercises will use PySpark (the Python API for Apache Spark) and Spark SQL (the Structured Query Language API for Apache Spark), but previous experience with Spark, SQL, or distributed computing is NOT required. Students should take this Python mini-quiz before the course and take this Python mini-course if they need to learn Python or refresh their Python knowledge.

Make sure you join your classmates in the course's Piazza discussion group (access code: **cs1051x**). Piazza provides a a collaborative environment with web-based access and email notifications, along with free Android and iOS mobile applications. We will be using Piazza for announcements, questions and answers, and discussions.

To complete the course lab exercises, you will be using the free Databricks Community Edition platform. By using the free platform, you will not need to install any Spark software on your machine. You can use any Internet-connected computer with the Google Chrome (preferred) or Mozilla Firefox browsers. Note that Internet Explorer, Edge, and Safari are not supported. Databricks Community Edition provides a powerful, complete Spark environment that includes a mini 6GB cluster running on Amazon AWS, and interactive notebook environment with visualizations and dashboards, and public environment to share your work.

QUIZ ON FAQS AND SYLLABUS

To acquaint yourself with the course, we ask that you read the course FAQs and the course syllabus, and then take the following **four** quizzes.

CS105 and CS100

(1/1 point)
If I've already taken CS100.1X will I learn anything new in the course?

- (•) Yes ✔
- ( ) No

EXPLANATION

Yes. This course focuses on learning how you write Spark programs using DataFrames and Spark SQL, a powerful Spark abstraction and programming paradigm for working with Big Data, whereas CS100.1X provided a broader survey of topics related to Big Data and Spark programming using the lower-level Spark abstraction and programming paradigm of Resilient Distributed Datasets. Based on our experiences teaching CS 100.1X (and CS 190.1X), we have created CS 105.1X as an introductory Spark programming course that teaches students how to program using DataFrames. Students who have taken CS100.1X but are unfamiliar with DataFrames or Spark SQL are encouraged to this course.

Supported Software Environment

(1/1 point)

If I want to use my own Apache Spark setup to run the course exercises, can the course instructors and teaching assistants help me install / support my setup?

- ○ Yes
- ◉ No ✔

EXPLANATION

No. We support a single software development environment (Databricks Community Edition) because having all students work in the same environment significantly improves our ability to support many students with limited staff support. For prior courses (CS 100.1X and CS 190.1X), we supported a virtual machine environment, however many students were unable to complete the courses because they encountered significant problems with installing the virtual machine software and downloading the large virtual machine image file. By using a free online software development enviroment that only requires a web browser, we expect all students to be able to quickly start programming in Spark.

Note that we are making all of the lab exercises and data sets available for download should you want to perform the exercises in your own Spark set up, however all submissions must be done through the online software development environment and we cannot provide staff support for issues that you may encounter.

Course Autograder

(1/1 point)

Does the course autograder perform additional tests when grading my labs, beyond the ones given in the assignment?

- ○ Yes
- ◉ No ✔

EXPLANATION

No. The autograder automatically runs submitted notebooks and grades them using the **same** tests that are included in the student notebook (it also requires that the code runs in a reasonable amount of time). Hence, you are strongly encouraged to get your notebooks working in your Databricks Community Edition environment before submitting to the autograder (the autograder will return errors for all segments of the submission that have not been completed). Moreover, before submitting to the autograder, you must publish your notebook, and submit the resulting URL link.

Getting Help

(1/1 point)

Which of the following should you do if you need help with the labs or if you have other questions related to the course?

- [ ] Immediately post a question on Piazza without first checking to see if fellow classmates have asked the same question.
- [x] Check the detailed FAQs, and then head to Piazza and use the "search" box to see if your question has already been answered. ✔
- [ ] Post code snippets of your answers on Piazza.

✔

Note: Make sure you select all of the correct options—there may be more than one!

EXPLANATION

We have created an extensive set of FAQs related to different aspects of the course. If you are having trouble with lab notebooks and/or the autograder, please first check one of these FAQ pages. If these FAQs do not address your questions, you should look on Piazza. When using Piazza, you should search through existing Piazza posts, as others students may very well run into similar problems. It is much more efficent for both you and the course staff to not ask and answer the same questions multiple times. Additionally, you should not post any coding answers or partial solutions for any of the lab exercises. It is perfectly OK to ask for help, but out of respect for fellow classmates and to avoid honor code violations please phrase your questions in a general / abstract fashion.

For more details please check the course FAQs on the course wiki.

Here are the links in this lecture:

- https://spark.apache.org
- http://spark.apache.org/docs/latest/programming-guide.html
- http://spark.apache.org/docs/latest/api/python/
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.Row
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark-sql-module
- https://docs.python.org/2.7/contents.html
- http://ai.berkeley.edu/tutorial.html#PythonBasics
- https://www.google.com/chrome/browser/
- http://www.economist.com/node/16349358
- http://gorbi.irb.hr/en/method/growth-of-sequence-databases/
- http://www.symmetrymagazine.org/article/august-2012/particle-physics-tames-big-data
- http://www.511.org/
- http://en.wikipedia.org/wiki/FasTrak
- http://fortune.com/fortune500/
- http://www.rcsb.org/pdb/explore.do?structureId=3J2T
- http://www.rcsb.org/pdb/files/3J2T.pdb
- http://upload.wikimedia.org/wikipedia/commons/2/23/Lod-datasets_2010-09-22_colored.png

- http://bdgenomics.org/
- https://spark-summit.org/2013/talk/one-platform-for-all-real-time-near-real-time-and-offline-video-analytics-on-spark
- https://spark-summit.org/2014/talk/analyzing-endurance-sports-activity-data-with-spark
- https://spark-summit.org/eu-2015/events/how-spark-enables-the-internet-of-things-efficient-integration-of-multiple-spark-components-for-smart-city-use-cases/
- http://www.slideshare.net/ydn/december-2013-hug-infinidb-for-hadoop
- https://spark-summit.org/2015/events/recommendations-on-with-spark/
- http://ipython.org
- http://pandas.pydata.org/
- https://docs.python.org/2/library/stdtypes.html#dict
- http://pandas.pydata.org/pandas-docs/dev/generated/pandas.Series.html
- http://www.r-tutor.com/r-introduction/data-frame
- http://www.json.org/
- https://parquet.apache.org/
- http://hypertable.org/
- http://aws.amazon.com/s3/
- http://hbase.apache.org/
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.select
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.drop
- https://docs.python.org/2.7/reference/expressions.html#lambda
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.functions.udf

- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.filter
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.where
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.distinct
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.orderBy
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.functions.explode
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.GroupedData
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.groupBy
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.GroupedData.agg
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.GroupedData.count
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.GroupedData.avg
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.show
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.take
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.collect
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.count
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.describe
- http://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark.sql.DataFrame.unionAll
- https://databricks.com/ce
- https://github.com/caesar0301/awesome-public-datasets
- http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4174913/pdf/big.2014.0020.pdf
- https://data.sfgov.org/
- http://www.sf311.org/

- https://databricks.com/blog/2016/05/24/genome-sequencing-in-a-nutshell.html%C2%A0
- https://github.com/bigdatagenomics/adam
- http://www-01.ibm.com/software/ebusiness/jstart/portfolio/usacycling.html
- https://docs.python.org/2/
- https://docs.python.org/2/download.html

This course is sponsored in part by Databricks and UC Berkeley's AMPLab.

The content in this course includes notes and content created by Dan Bruckner, Brian Clapper, John Canny, Sameer Farooqui, Richard Garris, Vida Ha, Michael Franklin, Anthony D. Joseph, Paco Nathan, Kay Ousterhout, Evan Sparks, Shivaram Venkataraman, Patrick Wendell, and Matei Zaharia.

Erik Arvai edited the course videos.


Download on the
App Store
ANDROID APP ON
Google play