Chapter 5 and 3 material; material covered in quiz 1

Chapter 5

* Basic Unix commands (ssh, ls, rm, cd, vi, etc.)
* Interpret basic SQL like in the homework examples
* Compare and choose more efficient code (e.g. parallel vs. split-apply-combine)
* Basic understanding of big matrix and why useful

Chapter 3

* Understand basic processing steps followed in the spam detect examples (conceptual level)
* Understand and recognize key-value pairs
* Code modularity-
  + why it’s useful
  + how to create utility functions out of part of a larger function
  + how to use functions in combination to get desired result
* Basic regex (e.g. which expressions would or would not match a particular pattern)
* Naïve Bayes interpretation and formulas
* Type I and Type II error analysis
* Accuracy and computational efficiency of formulas (e.g. better to use log sums or
* products, order of operations, etc.) -> pick between formulas
* Interpretation of recursive partitioning
* Interpretation of visualizations shown in Chapter 3
* Common code errors and debugging of short functions
* Be able to read function definitions and explain what the code should return

Hadoop and MapReduce

Material covered in Quiz 2

Includes material covered in class from Chapters 1, 2, 7 and 4 of the Hadoop Book text

Chapter 1

* Benefits and issues of parallel computing
* MapReduce paradigm -> what are valid combinors?
* Comparisons of MapReduce with other systems -> databases, etc.

 Chapter 2

* What are Hadoop and MapReduce
* Different interfaces (Java, Streaming) and associated programming languages -> streaming allows you to read from stdin and write to stdout
* Problems solved by MapReduce
* General steps of MapReduce framework
* How keys and values are used
* How to run at command line
*  Data to HDFS -> put data in, run map + reduce, combine results, put data back and forth
*  Map script
*  Reduce script
*  Output directory
*  Reading the results
*  Combiners
*  Terminology and managing data flow
* Importance of load balancing -> share tasks across nodes (not sure what to ask about this)
* Identifying valid combiner functions
* How streaming works
* What map and reduce do

Chapter 3 -> do not need to know this in very much detail (just the basics)

* Elements of design
* Times to not use HDFS
* Terminology and what elements do
* Resiliency and federation
* High-Availability
* Failover and Fencing
* Chapter 7
* Components/entities and setup of YARN
* Failure cases and what happens
* General flow from submission to completion

 Chapter 4 -> very important!!!

* Schedulers
  + Be able to answer multiple choice, matching, True/False, short answer about the topics above
  +  Be able to identify issues with Hadoop run and how to fix
  + Know basics of HDFS/Hadoop commands (file system commands and map-reduce runs like we have done in class

Pig (Chapter 16)

* Pig material covered in Quiz 2
* Motivation for Pig -> easier for data transformation, easier for programmer than map reduce, tradeoff is compute time
* What is Pig?
* General process
* Benefits of Pig
* Trade-offs relative to direct use of MapReduce in Hadoop
* How to run (locally and on hdfs) -> command line stuff
* Comparison with databases -> database has structured data in a database but in pig can define schema as you go along and can make those as you want
* Data types in Pig
* Basic data types
* Complex data types (bags, tuples, maps)
* Schema: understand and read from code
* Positional and named field references ->dollar or number?
* Null values
* Macros: read and understand -> quiz question
* Parameter substitutions: read and understand -> quiz question
* Be able to answer multiple choice, matching, True/False, short answer about the topics above
* Be able to read code and fix basic code errors (e.g. invalid function use, typos, missing ;’s in
* code, invalid comments)
* Be able to complete code segments -> fill in the blanks with code
* Be able to read and determine what code should return

Hive (Chapter 17)

* Includes Hive material covered in Quiz 4 as well as code exams (homework assignments or quizzes)
* Motivation for Hive -> looks like SQL but works with map reduce (easy for people who know SQL), less efficient than map reduce but saving human time
* What is Hive?
* Differences between Pig and Hive -> database (constraints) to get subsets rather than data transformation
* Comparison with SQL -> boils down to size of data and SQL does not operate well on large data, schema validation different (when is it validated), hive more expressive
* Benefits and drawbacks
* -e and –S options
* Comparison to traditional databases -> Hive (schema on read) when data is read or queried does it actual check for valid data, SQL (schema on write) when data is put into database
  + Don’t need to go through entire pile of data
* Schema validation and tradeoffs between validation on read and validation on write
* Data types in Hive
* Type conversion and hierarchy -> conversions of types
  + Subsetting types are easily convertible (int -> bigint but not bigint -> int)
* Casting types
  + Force conversion of like a bigint to int if they are in range
* Differences between managed and external tables (maybe a multiple choice)
* What are partitions and buckets and why are they useful
  + Partition -> first create table, predefined sets (subsetting on front end)
  + Buckets -> after data has already been written in
  + Work with smaller amount of data
* Types of Joins -> outer join, semi join, inner join, map join (might show up in code!!!)
* What are views and how are they used
* What are subqueries and how are they used
* Be able to answer multiple choice, matching, True/False, short answer about the topics above
* Be able to read code and fix basic code errors (e.g. invalid function use, typos, missing ;’s in
* code, invalid comments)
* Be able to complete code segments
* Be able to read and determine what code should return

Additional Visualization

* Be able to choose an appropriate type of visualization based on a description of what is to be shown in the visualization (e.g. something like “Select the best type of visualization to show XXXXX .” or “Which of the following graphics is the clearest visual representation of XXXX?”)
* Be able to recognize and interpret the advanced visualizations we saw (heat maps, tree maps, circular graphics, stream graphs, maps, network graphs, kernel density plots), including geometric interpretation (e.g. impact of circular graphics on rectangular plots)
* Be able to answer multiple choice, matching, True/False, short answer about the topics above (no code at all!!)
* Ex. Movie data -> number of movies in a genre per year
  + Heatmaps -> individual variables across each other
  + Flow of information (networks or diagrams)
  + Changing geometry
    - Circular graph changes straight line into an arc
    - Kernel density
    - Treemap -> color and size of squares

20-25 multiple choice

12 short answer