Big Data Technologies 774/874

Assignment 3

— UNDER MODERATION —

Overview

Part 1 Goals

- Revisit and further explore the concepts of the lectures.

Part 2 Goals

- Experiment with Spark and perform some basic analysis on data. In the tutorials we worked with aggregating data and using the Spark Structured API. The goal now is to consider an ML workflow with a simple implementation.

Submission

- Answers to be submitted on SUNLearn.

Part 1: Big Data Concepts

Question 1: Model Lifecycle Considerations [6] In the model lifecycle lectures we considered patterns and anti-patterns that can negatively impact models. Consider the following scenario; you have a number of models that draw on a shared feature store.

1. What should you be concerned about and why? [3]

Lastly, reproducibility was mentioned throughout the lecturers;

2. Describe how you would achieve reproducibility as a data scientist when codifying a model for sharing with your team (you may discuss this from a patterns perspective or technology perspective, or both)? [3]

Question 2: CAP Theorem [10] Consider the CAP theorem of the lectures.

1. Describe in a single paragraph the central statement of the theorem (as you understand it, and in your own words). [2]
2. Identify a distributed storage system where 'C' is sacrificed and explain why 'C' is not met in this system. [2]

3. Which pairs of CAP properties do the majority of NoSQL systems typically adhere to (in a big data setting)? [2]
4. Suppose you have a usecase where it is necessary that the same answer/value is returned to all clients (and your clients are located globally). Describe which of the properties you would require of your underlying database management system? [2]
5. Google Spanner claims to be able to offer effective CA when network outages occur. Explain how this is possible. [2]

Question 3: Spark [13]

1. What are the properties of the Spark Structured API that makes it particularly well suited to big data and to data science analysis? [3]
2. How are operations like `COUNT DISTINCT` managed on truly *massive* datasets? [2]
3. How is fault tolerance handled in Spark? [2]
4. What operations are subject to lazy evaluation and what is the utility of it? [2]
5. Suppose you wish to join a very large dataset (say 1 TiB) with a very small one (less than 100MiB); how would you approach this in order to conserve cluster resources? [2]
6. Explain why GroupByKey is an undesirable operation. Suggest an alternative approach and explain why it is better. [2]

Question 4: An Example Architecture [8] Consider the Netflix's data platforms in the year 2013 and 2017 (Figure 1 and 2 respectively).

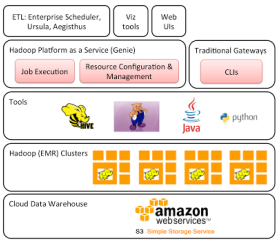

Figure 1: Architecture in 2013

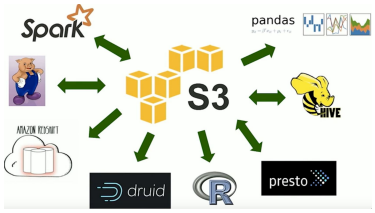

Figure 2: Architecture in 2017

1. Identify which technology provides large scale storage in these architectures? [1]
2. Which two technologies could be used for ETL tasks and which technology for machine learning [3]
3. Why does Netflix use Presto (what purpose does it serve within their platform) and could we use Spark for the same task? [2]
4. Suppose you were to migrate from Amazon Web Services to Google Cloud platform, what technologies would you use for storage, to run Spark and

to replace Hive? [2]

Part 2: Analysis Tools

Question 5: Spark [10] The data that you are working with is growing in size and your team has decided to consider technologies capable of large distributed workloads as a way to scale your machine learning. To this end, you have been tasked with exploring Spark and what would be required to use this technology from a Data Scientist's point of view.

Consider this Python Sklearn Random Forest implementation here. Your task is simply to port it/reproduce the implementation in Spark making use of the Spark API (your organisation uses R and Python, so you may choose the language that suits you best).

1. Reproduce the implementation using only the Spark API. Submit your code and demonstrate that you can calculate the accuracy metric (you may submit a notebook with output intact). [8]
2. Is the accuracy metric a good choice of metric for this problem (the way it is implemented here)? Motivate your answer. [2]

Further information: These data are available on Kaggle; the basic problem is also described there. This is a simplified solution modified from one of the community notebooks. You need not improve the current approach, simply port it.

For the master's level, Big Data Technology (Eng) 874 (not 774), students

Question 6: H2O [13] You have been tasked with investigating integration with open source H2O as your team is currently using an on-premises Cloudera stack and performing all their ML in SparkML.

1. What characteristics make H2O worth investigating considering your current technologies and big data in general? [3]
2. Modify your Spark implementation in Question 5 to integrate with H2O. Submit your code (you may submit a notebook with output intact). [8]
3. Is a random forest a good choice of model for this problem? Motivate your answer. [2]

Hint: H2O provides a server with an IP address then you launch the `jar`. You will require Java to be installed and will need to be able to connect to the port that the server provides.