Chapter 31.

July 2019

Welcome to the July 2019 edition of DataStax Developer's Notebook (DDN). This month we answer the following question(s);

> I'm confused. I saw a presentation at the 2019 DataStax world conference (Accelerate 2019), detailing how to deliver a product recommendation engine using DSE Graph. I've also seen DSE articles detailing how to deliver a product recommendation engine using DSE Analytics.
>
> Can you help ?
>
> *Excellent question ! As discussed in previous editions of this document; there are 4 primary functional areas within DataStax Enterprise (DSE). DSE Analytics can deliver a 'content-based' product recommendation (aka, product-product). DSE Graph can deliver a 'collaborative-based' product recommendation engine (aka, user-user). Both DSE Analytics and DSE Graph use DSE Core as their storage engine, and DSE Search as their advancedc index engine; a full integration, not just a connector.*
>
> *In this edition of this document we'll detail all of the code needed to deliver the above, and include data. We'll also use this edition of this document to provide a Graph query primer (Gremlin language primer), and answer the nuanced question of; Why Graph ?*

Software versions

The primary DataStax software component used in this edition of DDN is DataStax Enterprise (DSE), currently release 6.8 EAP. All of the steps outlined below can be run on one laptop with 16 GB of RAM, or if you prefer, run these steps on Amazon Web Services (AWS), Microsoft Azure, or similar, to allow yourself a bit more resource.

For isolation and (simplicity), we develop and test all systems inside virtual machines using a hypervisor (Oracle Virtual Box, VMWare Fusion version 8.5, or similar). The guest operating system we use is Ubuntu Desktop version 18.04, 64 bit.

31.1 Terms and core concepts

As stated above, ultimately the end goal is to detail and differentiate the delivery of product recommendation engines ("users who bought (X) also bought (Y); up-sell, increase revenue per transaction), when using DataStax Enterprise (DSE) Analytics and/or DSE Graph.

In this document, the following are held true;

- Really the only prerequisites are that you have access to a working DSE server instance, and DSE Studio Web client. You also need a working Apache Zeppelin, that points to your working DSE.

  Earlier versions of the document in this series detail how to make the above happen. If you have none of the above, and no previous skill to make these prerequisites happen, you're looking at about 1-5 hours worth of work and reading.

- All data and program code are available at,

  `tinyurl.com/ddn3000`

  You want the July/2019 dated set of artifacts.

- In this document, we deliver a product recommendation engine using DSE Analytics (Apache Spark).

- Also in this document, we deliver a product recommendation engine using DSE Graph (Apache TinkerPop/Gremlin).

  This document series has never provided a Gremlin (the query language to Graph) primer, and we use this example to do so.

  In other words; you'll also learn Gremlin in this document.

- Through the work/detail above, we also seek to answer the question; When/Why Graph ?

Machine Learning, is it real

In addition to providing the core Apache Spark data abstraction (DataFrames, Datasets, and RDDs; resilient distributed datasets), Spark streaming, and more, DataStax Enterprise (DSE) also provides 28+ machine learning routines out of the box.

Our favorite definition of machine learning is; "an algorithm whose accuracy improves given more data". (And, we greatly prefer the phrase, machine learning, to its idiot cousin phrase, artificial intelligence (AI)).

As a topic, machine learning first appeared in human history circa 1965 with quotes including:

"Machines will be capable, within 20 years, of doing any work a man can do."

-- Herbert Simon, 1965, Ph.D, IIT

"From three to eight years, we will have a machine with the general intelligence of an average human being."

-- Marvin Minsky, 1970, Ph.D, Princeton, MIT, Turing Award 1969

```
https://en.wikipedia.org/wiki/Herbert_A._Simon
https://en.wikipedia.org/wiki/Marvin_Minsky
```

There have been, since 1965, many "AI Winters", meaning; there have been many false starts between now and then. Machine learning is now a reality due to many factors:

- Availability of lots of data
- Fast, parallel, cheap computing power
- Now a focus on weak/narrow (AI), versus General Intelligence (AGI), which was more the focus in 1965.
- Machine learning (ML) is a focus of many companies, mathematicians, universities, and more, aiding the global economy of scale for companies like; Amazon, Google, US DoD/NSA, Cisco, FedEx, and many more.
- And it can not be understated the impact of open source and social coding in the area of advancing ML.

Machine Learning, the process

Out of the box, DataStax Enterprise (DSE) comes with 28 or more machine learning routines. Learning each routine would take 1-6 hours each (6 or more hours for experimentation), and 8-20 or more pages of documentation each.

In this edition of this document (DataStax Developer's Notebook, DDN), we detail one routine titled; frequent pattern mining. Product recommendation (users who bought (X) also bought (Y)), is a seminal use case for frequent pattern mining.

Figure 31-1 Frequency pattern mining, customers who bought (X) bought (Y)

In general, however, the following process is observed towards delivering a given machine learning routine:

– Generally, machine learning routines fall into one of two cagegories; supervised, or unsupervised.

Note: Coarsely, unsupervised machine learning differs from supervised in that the input columns are not labeled. A seminal unsupervised machine use case is clustering. Given, for example, 20 or so distinct, numeric measures I know about (customers); how are those customers segmented ?

A classic use case for clustering is to discover the potential for a customer to churn; What (cross section of those 20 numeric measures) best identifies customers who wish to discontinue use of my service ?

–

31.2 Complete the following

At this point in this document we have completed a lengthy primer on DataStax Enterprise (DSE) terms, its object hierarchy, history, use, operating conventions, configuration files, and more. 40 pages or so, the above

31.3 In this document, we reviewed or created:

This month and in this document we detailed the following:

- A rather complete primer to DataStax Enterprise (DSE).

  We detailed DSE object hierarchy, history, use, operating conventions, configuration files, and more.
- We downloaded, installed, configured and booted a two node DSE cluster with a NetworkTopologyStrategy.
- We made keyspaces, tables, inserted, updated, and selected data.
- And we ran exercises demonstrating network partition failure tolerance, and eventual consistency.

Persons who help this month.

Kiyu Gabriel, Matt Atwater, and Jim Hatcher.

Additional resources:

Free DataStax Enterprise training courses,

`https://academy.datastax.com/courses/`

Take any class, any time, for free. If you complete every class on DataStax Academy, you will actually have achieved a pretty good mastery of DataStax Enterprise, Apache Spark, Apache Solr, Apache TinkerPop, and even some programming.

This document is located here,

`https://github.com/farrell0/DataStax-Developers-Notebook`

`https://tinyurl.com/ddn3000`

