Machine Learning & Google Big Query

Data collection and exploration – notes from the field
• Limited to support of Machine Learning (ML) tasks
  – Review tasks common to ML use cases
    • Data Exploration
    • Text Classification

• Field notes on BigQuery (BQ):
  – Has limitations which make it unsuitable for many data service use cases
  – Be patient with Cloud SDK & AppEngine APIs
    • Badly documented APIs and Google training is not up to the task
    • Stark absence of players (internet search results mostly limited to Google docs)
    • Plan on spending lots of time reverse engineering these APIs
  – Unstable APIs
    • Changes are improvements, mostly
    • Upgrade breakages were common-place

Introduction
• Origins are from the Dremel project at Google
  – Highly parallelized
    • Across commodity servers
  – Column storage
  – No indexes
    • Single scan of disk

• Top level structure is a Project
  – Billing, users/groups
  – Dataset(s) belong to Project
    • Lowest level of access control

• Accessed via
  – BigQuery web GUI console
    • High latency when interacting with list of UI objects
  – Command line scripts
  – REST API via Google client

• SQL?
  – BQ SQL is not the SQL your are used to
    • In some regards that is a good thing, but perhaps the losses are too big

Google BigQuery
- No updates of data
  - Must handle updates outside of Big Query
  - Random writes require very complex custom code
    - There are conventions that simplify the complexity
    - No need to purchase proprietary ETL toolset
      - None of them scale, none are fault tolerant
      - Put a software engineer on the team
      - Adopt suite of open source tools
      - Remember you are creating mini software products
    - Run on-premise or on GCP?
      - (N) Google Compute Engine (GCE) + SSD + Docker containers of ETLs

- No deletes of data
  - Same issues as updates

- High batch latency between source and BQ
  - Queries will be out of date by hours

- Purchase of Google Cloud Storage service is necessary
  - Functions as the staging data source service to BQ

- Arbitrary queries are not supported
  - Indexes are not supported so neither are random reads
  - Scan of everything always happens
  - Have to create custom pre-computed materialized views
    - These are always specific to a given use case
    - Moreover, always have to re-compute them programmatically, on schedule, and fault-tolerant

- BQ SQL
  - Don’t even think about doing a port to BQ

Big Query when viewed from an RDBMS mindset
• You can
  – Bulk load new data into new table
  – Append new data to existing table

• Is a fully managed data service

• Very fast read access to 10s of TBs
  – De-normalization might not be necessary
    • JOIN 10s of 3NF tables
    • Avoid sub-selects
  – At most, just ‘Write it the way you need to read it’
    • Columnar data model

• Powerful text query via regular expression statements
  – BQ SQL + regular expressions = actionable text information

• Nested entities intrinsic to Business domain
  – Big Query entity mirrors business domain entity

• Join together disparate sources easily
  – Cast into standardized ML model

Big Query Features
ML Business Use Cases in the cloud must be grounded

• Do you really know your business use case(s)?
  – Stop and think
    • It’s the only way to avoid buying fool’s gold
  – Understand the explicit business objective(s)
    • Domain manager and domain subject matter expert define and prioritize objective(s)
      – If they can’t make the objective explicit then the initiative will fail for lack of leadership
        • No need to hire a business analyst

• Written definition clearly describes: functionality/information valuable to consumer
  – Uses language that is ubiquitous to the business

• Absolutely critical to managing project risks
  – Presence of use cases differentiates the principled professionals from those who are not
    • Absence of use case indicates small degree of trust in themselves, their products, services and organization, as well as in the domain manager and domain SME.

• Proof of Concept? Proof of Value?
  – Without a use case you’ve proven nothing, you’ve proven no real business value.
• $ value of use case is well understood
  – The shared business value creates the *us*, the team
  – Know your ROI, even if it is an R&D exercise
    • Tie it to acquisition, preservation or growth of capital
  – Staff team only with those who can maintain that focus and shoulder that obligation

• Written description that can be used for
  – Project planning, and
  – Conversations about the business use case

• Tests that can be used to determine when the solution to use case is both complete and valid
  – If you do not know what the finish line is, then you will never cross it

• Technical team determines how to develop & operate solution to use case

Criteria of good Business Use Case(s)
• Domain model is exactly how the business works (or will work)
  – Team - domain management, domain subject matter experts, technical members
    • Continually trying new approaches to see what will work best
    • Unless domain managers want to innovate, ‘actionable insights’ will remain just another over-used marketing slogan
  – Minimize scope, get small, really small, and very focused
    • Machine Learning (ML) system must change quickly in the face of changing needs
    • ML must be easy and inexpensive to change
  – Testable, can be empirically proven to meet core business objective(s)

• Think of the DM as an ontology
  – Things of interest to you, and their characteristic properties, as well as their relationships and the properties of those relationships

• Often implemented as an object model
  – Literal and accurate business meanings assigned to data as well as behavior(s)
    • JSON document captures definition of Big Query entity/table
    • Flattening object model may not be needed
  – Business finds it easy to understand the DM and easier to query the DM than a domain data cast into a relational schema

Domain Driven Design and ML
• From Google Cloud Storage (GCS), (truncate) bulk upload static snapshots into Big Query table/entity
  – Method supported via Big Query web-based console
    • BQ console is too labor intensive for operations, fine for exploration
  – Programmatically control batch upload processes
    • Most stable portions of APIs
  – JSON representation of Big Query table/entity
    • Business domain model with nested entities are supported
  – CSV input file with header line or JSON format input file
    • JSON rep + Header useful specs for extractor

No updates! No deletes! There’s only snapshots
• Preprocess data sets prior to uploading into GCS
  – Partition data sets by time (_CCYYMMDD for a given day)
    • 1 partition per slice in time
      – In BQ 1 partition = 1 table/entity
    • Create View over partitions to provide consumer(s) with 1 entity to query in BQ
  – Google AppEngine (GAE) kills process running for >10 seconds
    • GAE is a no go when updating large data volumes
    • Google Dataflow?
      – It is an Idempotent RESTful service
        • Client makes the same call repeatedly while producing the same effect
        • ETLs can rarely be applied repeatedly without effecting results (negatively)
      – Documentation looks weak, small population of users (once again)
  – Update/delete processes are typically disk I/O intensive
    • Run update/delete processes on-premise, or on GCP instance, use (local) SSDs
    • Co-locate process with the data file(s), avoid moving data over network to the process
  – Big Query Streaming API was under construction
    • Did not function as documented (once again)

Remediation via Conventions
• Avoid merging business channel domains
  – Learn the lessons from enterprise DW and BI debacles
  – Intra domain use cases are doable
    • But only if the will to act exists within leadership
    • Real change is always disruptive
  – Blending business channel domains greatly increases probability of failure
    • Scope creep
    • Lack of shared incentives across business channel domains are the invisible barriers
    • Culture stresses competition over collaboration

• If new use case just queries existing schema then you do not need DDD task

Business Domain Model as Big Query Schema
• Use to transform large volumes of raw data into a representation suitable for ML models

• Big Query Console and APIs can support exploratory steps in data analysis
  – Storing very large numeric summary tables
    • Collect classic statistics output
  – Data source for basic visualization to search for patterns in the data
  – Typical suite of mathematical functions supported in BQ SQL
    • Natural logarithm, Base-2, Base-10, radians, etc.

• Correlation research
  – Examine many variables simultaneously
    • Relatively easy inclusion of many variables
    • Join many tables in a single query

• Passive confirmatory data analysis
  – Tables support tests of formal model on the pattern you think you found in the data

• Representative Sampling
  – Less time consuming on BQ, therefore may be less expensive relative to alternative columnar data services
  – Increases chances of being able to generalize the results from the population
  – Common batch processing approach when model is re-trained using all data

Data Exploration
• Classify by searching for hidden patterns in unstructured text
  – Regular expressions are supported in Big Query SQL
    • Search through text for terms using REGEX_MATCH()
      – Can be used on integer and float data cast to string
    • Be sure to remove new line from within uploaded text, else load will fail!
  – Build vocabulary from very large corpus of text
    • Store counts of the # of times the term occurs
      – Counts can be used to support Naïve Bayes classifier
• Know your ML business use case
  – Experiment with BQ’s ability to support ML task(s)
  – Avoid hard project deadlines (bad docs; very small community of users; volatile APIs)

• Understand the limitations and features of BigQuery
  – Are you ready to pay the design, development and testing of custom update and delete code?
  – Have you priced in the cost of the update/delete custom code?

• Well suited for:
  – Transforming very large data sets into ML models
  – Exploring very large data sets
  – Text classification

• Is BigQuery ready for the enterprise?
  – Best kept in department (or R&D) for the time being
  – Consider cloud-based Mesos + Apache Hadoop + Spark + MLlib + Dremel + Parquet
    • Very well documented + large dev and ops populations

• Questions?

Closing