Business Intelligence

Concepts

Why Should We Care?

Why should a data scientist care about
BI (business intelligence) or DW (data warehousing)?

- Big companies usually have large BI/DW departments.
- BI/DW departments are typically much larger and established long before data science group.
- Data science departments often have to play well with BI/DW and leverage existing BI/DW resources.

BI: Business Intelligence

- Business-focused
  - Understand the business and how to apply analytics to solve business problems
  - Math not as important—simple basic statistics
- Data science light with a business focus
- Older terms
  - OR (operations research)
  - DSS (decision support systems)

Beginnings of BI

Late 1980s through mid-1990s

- 1987: stock market crash and recession
- Most big companies had an OR (operations research) department creating DSS (decision support systems)
- Reporting databases
  - Nightly dumps of production data to a reporting database
- AI (artificial intelligence) lab
  - AI very popular—most companies had an AI lab
  - Smaller number of personnel than OR
  - Due to hardware and software limitations, very little progress
  - High expectations

Dot-Com Era

Mid-1990s through 2000

- Dot-com era
- Focus on internet
- Huge sums of money coming in—less pressing need to analyze and improve the business
- AI systems had failed to produce anything meaningful—most AI labs shut down

Dot-Com Bubble Burst

- 2000: stock market crash and recession
- Back to analyzing the business
- Bad memories of AI and OR failures
- More focus on understanding the business and how to improve the business than on math
  - BI (business intelligence)
  - DW (data warehousing)

Data Science/AI

- 2009: another stock market crash and recession
- Big companies had large BI and DW departments
- Starting to realize that they needed more math focus
- Data science
- 2015 or so: AI comes back

Metrics

- Ideal value to compare with and measure difference
- Analyze business
  - Establish metrics
  - Compare actual performance to metrics
  - Determine good or bad performance
  - Performance scorecards

Conflicts

- Department metrics often conflict
- What is good for one department is often not good for another department
- Need enterprise focus, not a local focus
- Finance often source of most conflict

KPIs: Key Performance Indicators

- New idea and term to replace department-level metrics
- Enterprise focus
- KPIs can be decided by executive level if there are department conflicts

HR Issues

- One manager wants it done one way
- Another manager wants it done another way
- Measuring managers against KPI helps ensure corporate policies are followed
- Fairness to employees—lessens subjective evaluation

SMART Methodology

- Specific
  - Target a specific area for improvement
- Measurable
  - Quantify an indicator of progress
- Assignable
  - Specify who is responsible
- Realistic
  - Achievement must be realistic given available resources
- Time
  - Specify when the result should be achieved

Common Types of KPIs

- Quantitative
  - Can be described with a number
- Qualitative
  - Cannot be described with a number
- Leading
  - Predict the outcome
- Lagging
  - Present success or failure after the event
- Actionable
  - Within our control to make change

The End

Business Intelligence

Business Cases

Conflict

- Big box store registers
- Store manager want lines to be short since customers complain if lines are long
- Finance department want lines to be long because it saves labor cost on cashiers

Conflict (cont.)

- Create a KPI
  - All registers should be checking out a customer
  - 75% of registers should have one person in line
  - At 25%, close some registers
  - At 100%, open more registers

Marketing and Sales KPIs

- New customer acquisition
- Customer attrition
- Attributes of potential customers
- Levels of approval, rejections, pending numbers

Debt Collection KPIs

- Late balances by customer attributes
- Terms of payment by customer attributes
- Write-offs by customer attributes

Manufacturing KPIs

- OEE (overall equipment effectiveness)
  - OEE = availability x performance x quality
  - Availability = run time / total time
  - Performance = total count / target count
  - Quality = good count / total count
- Cycle time
  - Beginning to end of process
  - Cycle time ratio = target cycle time / actual cycle time

Data Center KPIs

- Availability and uptime
- Mean time between failure
- Mean time to repair
- Unplanned unavailability

Software Engineering KPIs

- Earned value
- Estimate to complete
- Labor spent per month/labor budget per month
- Nonlabor spent per month/nonlabor budget per month
- Average time to delivery
- ROI (return on investment): how many months until the system pays for itself in savings
- Planned milestone date vs. actual milestone date

Supply Chain KPIs

- Sales forecast
- Inventory
- Procurement and suppliers
- Warehousing
- Transportation
- Reverse logistics

HR KPIs

- Employee turnover
- Employee performance indicators
- Cross-functional team analysis

Fast Food KPIs

- How many people in line inside
- How many cars in the drive-through
- How many cars do not get items at window and have to park
- How long to complete a regular order
- How long to complete a special order
- How many returns for food preparation errors
- How many returns for missing items

The End

Data Warehousing

Concepts

Siloed Reporting Databases

1990s

- Companies started dumping production databases into reporting databases
  - Typically restore a nightly backup to a different reporting database
- Reporting databases were siloed (stovepipes)
  - What if we need data from two or three or more different reporting databases?
- Hardware/software limited on database size
  - Hard to put multiple reporting databases in one database
- Current data
  - No historical data

Single Reporting Database

2000

- Dot-com bubble burst, stock market crash, recession
- Focus no longer on internet
- Back to business
- Scaled up SQL relational databases are coming on the market
- Dump all the reporting databases into a single database

Issues With Single Reporting Database

3NF (third normal form) not a good fit

- Supports current data, not historical data
- 3NF style joins are too intensive—more than hardware of the era could handle
- Normalized data is not convenient for analytics

New Methodologies Needed

- Separate reporting into two focus areas
  1. Data warehouse
  2. Data mart
- New data modeling methodology
  - Dimensional data modeling

Analogy of Warehouse and Mart

- Company has a warehouse to store finished goods
  - Assume we have only one warehouse that serves all marts
- As marts (stores) need finished goods, they are moved from the warehouse to the mart
- Warehouse stores like goods together in one place, compact format
- Marts can store the same good in several places, convenient format for customers

Data Warehouse and Data Marts

- Data warehouse
  - Enterprise-wide data from all production systems
  - Compact format
  - Would not be convenient for users
- Data mart
  - Functional area focus
  - Data stored multiple times in multiple ways
  - Convenient for users

Common Methodologies

- Inmon’s method
- Kimball’s method
- Stand-alone data mart

Inmon’s Method

- Data warehouse
  - 3NF
- Data marts
  - Dimensional data model
  - Virtual
- Issues
  - 3NF does not easily store historical data
  - Hardware of the era not able to handle virtual

Kimball’s Method

- Data warehouse
  - Dimensional data model
- Data marts
  - Dimensional data model
  - Copy from data warehouse
- Easier to store current and historical data
- Issues
  - Uses a lot more storage
  - Long development time frames

Stand-Alone Data Mart

- No data warehouse
- Data marts
  - Dimensional data model
  - Data loaded from production
- In practice, the most common type of data mart
- Issues
  - Lose enterprise focus—silo or stovepipe
  - Considered a hack

Treated as Data Marts

Most data science, big data, AI, ML, DL, data lakes, etc. are treated as stand-alone data marts at a lot of companies.

Issues With DW

- Long and expensive time frames for development
  - New data takes months to show up in DW
- Rework for business changes can be very hard
  - Can take months or years to show up in DW
- Dimensional modeling can leave gaps
  - The data we need is in production but did not make it to the DW
  - Data is aggregated and we need it at a lower granularity
- After we have our data mart ready to go, we can still end up with a silo (stovepipe), which is what we started out to avoid

The End

Data Warehousing

Business Cases

Big Company

- Established BI/DW department
- Most production system data dumped into the DW
- Lots of resources
  - BI personnel, tools, and expertise
  - Data visualization personnel, tools, and expertise
  - Reporting personnel, tools, and expertise
- Policies, procedures, audits, etc. well-defined

Big Company (cont.)

- Data science department is only a few years old and very small compared to the BI/DW department
- Leverage existing BI/DW resources
  - Pull data from the DW when possible
  - Classify our systems as data marts and leverage existing policies, procedures, audits, etc.
  - Software integration tools can connect our systems to BI, reporting, etc. systems to leverage existing tools and talent

Other Business Cases?

- In the era of data lakes and serverless SQL, it’s very difficult to justify developing a new traditional DW
- DW typically limited to:
  - Existing systems
  - Modifications to existing systems

The End

Dimensional Theory of Data

Concepts

Facts and Dimensions

- Facts
  - Quantitative data
- Dimensions
  - Qualitative data
  - Place facts into a context

$2.29

- $2.29: price of a gallon of gas in Berkeley on July 8, 2020
- $2.29: price of a package of doughnuts in Los Angeles on March 23, 2020
- $2.29: price of a cup of decaf coffee in Seattle on April 14, 2020

Facts and Dimensions

Fact	$2.29	$2.29	$2.29
Dimensions	• Gallon • Gas • Berkeley • July 8, 2020	• Package • Doughnuts • Los Angeles • March 23, 2020	• Cup • Coffee • Decaf • Seattle • April 14, 2020

Dimensional Theory of Data

- Data can be expressed in terms of:
  - Facts: quantitative data
  - Dimensions: qualitative data that places facts into context
- Date/time is almost always a dimension
  - Allows current and historical data
- Same data can be expressed in multiple dimension sets
  - Convenient for analytics

The End

Dimensional Data Models

Concepts

3NF: Third Normal Form

- Data is stored exactly once
  - No duplication
  - Only one place for updates
- Best for current data, not good for historical data
- Requires many joins
- Great for transactional databases

Dimensional Data Modeling

- Data is organized according to the dimensional theory of data
  - Fact tables
  - Dimension tables
- Analytical convenience
  - Data is duplicated
  - Data is presented from multiple points of view
  - Denormalized

Star Schema

- Fact table surrounded by dimension tables
- Dimension tables
  - Points of the star
  - Primary key is a surrogate key to allow historical data
  - Rich attributes
- Fact table
  - Center of the star
  - Child table of all the dimensions
  - Primary key is a composite key of the primary keys of the dimensions

Querying a Star Schema

- Join the fact table to one or more of its dimension tables
- Fast performance
- With large scale-up databases, this was the only way it would work in the early age of DW

Drilling Across

- Suppose we want to join two star schemas
- Issue is with joining two fact tables
- Recall when we discussed relational database joins
  - Usual case is to join primary key to foreign key
  - Two fact tables will not have primary key/foreign key relationship
  - Dangerous join
    - Extra rows problem
    - Missing rows problem

Drilling Across (cont.)

- Possible solutions
  - Create another star schema
    - Duplicating data is normal and expected in DW
  - Drill across
    - Fancy way of saying write functional code to perform the join manually that handles any extra row or missing row anomalies
  - Hybrid
    - Drill across for short-term
    - Consider creating another star schema for long-term

Normalizing Star Schemas

- Dimensional modeling purists would say everything needs to be a star schema
- Two common exceptions
  1. Snowflake schema
     - Dimensions are children of other dimensions
     - Parents are called outriggers
  2. Bridge schema
     - Dimensions are parents of other dimensions
     - Association table usually built
       - Both dimensions parents of the association table

Normalizing Star Schemas (cont.)

- Using snowflakes and bridges totally contradicts what we are trying to accomplish with dimensional modeling
- If we are using snowflakes and bridges, it’s probably better to just use 3NF

Idea Whose Time Has Passed

- Dimensional data modeling is an idea whose time has probably passed
- Modern hardware now makes 3NF possible for large databases
- Tools on top of 3NF can be just as convenient as a dimensional model
- Serverless SQL tools are probably just as convenient as a dimensional data model
- Legacy: DW will be with us for years to come before it’s finally retired

The End

Cube Theory of Data

Concepts

Cubes

- 1D: one dimensional = row or column
- 2D: two dimensional = table
- 3D: three dimensional = cube
- >3D: x dimensional = hypercube
- When we say cube, we generally mean hypercube

Cube Theory of Data

- Similar to dimensional theory of data
- Facts
  - Inside the cube
- Dimensions
  - Edges of the cube
  - High dimensions mean many edges

Cubes Shine

- Rollup by dimensions at various levels of granularity is where cubes shine
- Store facts at the finest grain level to allow for maximum rollup

Star Schema/Cube Conversions

- Easy to convert a star schema to a cube
  - Each row from the fact table gets a slot inside the cube
  - Each dimension from the dimension tables gets an edge
- Easy to convert a cube to a star schema
  - Each slot inside the cube becomes a row in fact table
  - Each edge becomes a dimension in a dimension table

No Snowflakes

- Outriggers from snowflakes would be just the equivalent of adding another dimension to a cube
- Snowflakes not part of cube theory

Bridges

- Bridges are not considered part of cube theory.
- Some cube products support them.
  - Customers with bridges in star schema will see it as a gap.
- If you need a bridge, a cube is not a good fit, as there are better ways.

Love/Hate

- All or nothing
  - When cubes are a fit, they fit well
  - When cubes are not a fit, they are basically useless
- Culture of love/hate with cubes
- Not realistic to expect a cube to work for everything
- High granularity into cube
  - People do not know/understand that data in cubes has to be fine grained to work
  - Ends in disaster
  - Blame cubes in general

The End

Data Lakes, Data Warehousing

Business Case Comparison

Data Warehousing Development Cycle

- Takes months
  - Dimensional model and schemas must be designed and built
  - E**T**L cannot start until the dimensional model is in place
  - E**T**L processes must be designed and built
- We may not get to look at data for weeks or months into the process
  - Find out too late data is not what we need
- Gap are a big risk
  - We go into production and discover a gap
  - Weeks or months to retrofit a solution to the gap

Data Lake Development Cycle

- Quickly
  - Load data into the data lake
  - Use tools to execute serverless SQL against the data
  - Explore the data—see if it’s what we need
  - Identify and fix gaps
- Management can typically get some basic analytics within a few days

New Startup Without DW

Best to just go with data lakes and not try to build out a traditional data warehouse

Legacy Company With BI/DW

- Big company has a long legacy of BI/DW for over 20 years
- Lots of BI, reports, etc. depend on the DW and its current structure, queries, etc.
- Huge effort to move everything to a DL (data lake)

Legacy Company With BI/DW (cont.)

- Strategies
  - New datasets
    - DL
    - May have to design E**T**L to push to DW if needed
  - Migrate existing from DW to DL
    - Dump DW to DL—store both places
    - Start moving analytics from DW to DL
  - Leverage existing tools and talent
    - Connect BI tools to DL
    - Connect reporting tools to DL
    - Make SQL as easy and as transparent as possible

The End