Data Science on the Wild

Luís Sarmento

luis.sarmento@gmail.com
https://www.linkedin.com/in/luissarmento/

February 22, 2017

Setting the Right Expectations

Why "In the Wild"?

- "Data Science" is a relatively new job in the Industry.
- There is a lot of hype about DS:
  - "... data is the new gold..."
  - "... the sexiest job of the decade..." (Sorry Mr. Clooney!)

  .
- But there is NOT a lot of tradition to guide us.

- We are still pioneering ("cowboying") at the level of the:
  - setting the right goals and expectations
  - encoding methodologies and best-practices
  - building, managing and scaling DS teams
  - engaging with other teams

The Industry is "Wild"

- Gathers people with different:
  - backgrounds
  - motivations
  - expectations
  - work philosophies
  - understanding about what the problems to be solved are
- Full of polysemous terms:
  - depending on the people's backgrounds.
  - it is very easy to "agree" on radically different things
- Full of entrenched "methodologies" that may not apply
- This leads to higher risk of insuccess and disapointment...

The Industry is Merciless

- "Why You're Not Getting Value from Your Data Science"
  Harvard Business Review, December 2016
- "150 Data Scientists and still no business value?"
  article on LinkedIn, December 2016

My goal with this talk....

- Is NOT to give you a "Survival Guide for Data Scientists"!
  - I don't yet know all the "tricks"
  - In any case, "ressurrection" is better than "survival"
- Is to tell you about some potential
  - honest but naive mistakes (that I made)
  - disastrous "good intentions" (that I had or was led by)
  - fluffy situations and muddy swamps (that I run into)
  - "land mines" (that I stepped on)

  that can quickly bring you or your teams down...

Data Science Jungle

"We want to hire (a team of)
[type_data_family$_m$]
because we (want|need) to do (research|development) in
[type_trendy_area$_n$]

type_data_family$_m$ $\in$ [Data Scientists, Machine Learning Scientists, Applied Scientists, Research Scientists, Research Engineers, Machine Learning Engineers, Data Engineers, Data Analysts, BI Engineers...]

type_trendy_area$_n$ $\in$ [Data Science, Business Intelligence, Customer Intelligence, Customer Insights, Recommendations, Analytics, Data Mining, Big Data, Machine Learning, Artificial Intelligence, Personalization, Automatic Assistants, Forecasting, Fintech ...]

Data *Science*

- Science is an overloaded word:

> "Science is a *systematic* enterprise
> that builds and organizes knowledge
> in the form of
> *testable explanations* and predictions
> about the universe."
>
> (from Wikipedia)

- A (data) scientist is someone who follows this *entreprise* (?)
- In Academia: sure!
  - It's one of the core jobs of the academic community
  - Well established methodology and check points.

But what about in (most of the) Industry?

Wait... And what about "Research"?

- Research is also a very overloaded word:
  - Market Research
  - Merketing Research
  - User / User EXperience (UX) Research
  - (Talent) Recruiting Research
  - ...

But ...

- Don't think me unkind
- Words are hard to find
- They're only cheques I've left unsigned[1]

[1] Police, The; "De Do Do Do, De Da Da Da", In Zenyatta Mondatta, 1980

Can we define "Data Science"?

Taken literally, "Data Science" would be the "science" of:

- *Data Objects* as entities of first kind:
  - Generic/Abstract: tabular, graphs, sequential, temporal, ...
- *Data Representations*:
  - Encoding, Compression, Selection/Learning of Basis
- *Data Operations*:
  - Classification, Prediction, Mapping, "Translation", Generation...

There is still a whole spectrum of possibilites

- Application-specific *data objects*...
  - music, natural language, images / video, etc.

  ... and *transformation/operations*...
  - parsing, machine-translation

  ... would be considered instantiation of the "generic" concepts.
- A data scientist would be a sort of "renassaince" artist, capable of tackling any problem involving data.
  - This would be desirable... but it is not very realistic.

Spectrum of possibilites - I

1. The person who knows is studying some abstract concept, which fundamentally advances our knowledge:
   - e.g. Clifford Algebras
2. The person who knows all about a very specific technique / method that can be applied in many situations
   - e.g. Convex Optimization in very high dimensional spaces.
3. The person who knows all about a specific modeling technique, applicable in problems of a certain kind in *various* domains
   - e.g. Recurrent Neural Units
4. The person who knows all about a specific application domain, and applies any technique that helps advancing the domain:
   - e.g. Sentence parsing

Spectrum of possibilites - II

5. The person who is confortable with many domains, has system-wide perspective and helps building larger systems:
   - e.g. Search-Engine
6. The person who knows about a customer problem and conceives a customer solution:
   - e.g. Web App
7. ... ...

In the Industry, the work of a Data Scientist lies mostly...

Spectrum of possibilites - III

1. Theory-Driven + Specialized Knowledge
   - e.g. Clifford Algebras
2. Theory-Driven + Experimental + Generic Ap.
   - e.g. Convex Optimization...
3. Performance-Driven + Experimental + Generic Ap.
   - e.g. Recurrent Neural Units
4. Task-Driven + Experimental + Domain Knowledge
   - e.g. Sentence parsing
5. Application-Driven + Experimental + Diverse Knowledge
   - e.g. Search-Engine
6. Business-Driven + Product-Concerns + Human Factors
   - e.g. Web App

... on points (5) with some specific (4) skills:

- with some potential incursions in deeper techniques (3); and,
- very solid understanding of product/business concerns (6)

What sort of "beast" is this?

- More of a Generalist than a Specialist
  - but with ability to specilize fast
- More of a practical contributor than a "theoretical" contributor:
  - closer to an actual "engineer" than to a "scientist"
- But experimentation is a great deal of the work
  - requires a certain mindset that is closer to that of the "scientist" than to that of an "engineer"
- Focused on finding the right "customer problems" (more than finding the solutions?):
  - requires the mindset of a product owner

However...

However...?

In practice, this idealized scenario may not be possible:

- the technical pre-conditions may not be there:
  - "Logs? What logs?"
- the actual needs of a company may actually be different:
  - "Yes, yes yes... That ML thing is great but we need to ship this by end of the week."
- the team may not understand the role:
  - "Can you help me with this Excel formula?"

(more on this later!)

The Role of a Science Team

Complicated vs Complex Systems

- Complicated System:
  - Many parts, but interconnections are known and "linear"
  - There is a natural "ordering" / sequence
  - System dynamics are predictable
  - System is Decomposable / Factorable
  - E.g.: micro-processor, beer factory, accounting software, car, sending the man to the moon
- Complex Systems:
  - Not all parts (and interconnections) are known
  - Circular ordering / Feedback loops
  - Dynamics very hard to predict / Non-Linear / High Entropy
  - Indecomposable
  - E.g.: a living cell, stock markets, path of innovation, social-political systems, natural languages

Complicated Systems

- In general, we are quite good with Complicated Systems.
- We know how to:
  - design them, one part at a time
  - organize teams to build and maintain them
  - delegate / outsource their production
- We have built systems and methodolgies to optimize building complicated systems:
  - Taylorism / Reducionism
  - "Scientific management" and derivatives
- Keywords: Efficiency, efficiency, efficiency...

Complex Systems

- In general, we are very bad with Complex Systems.
- We do not know very well:
  - how to divide them in smaller parts
  - where to start solving problems
  - how to track progress
- We tend to miscalculate
  - degree of complexity
  - the unknowns
  - time to delivery
  - cost of solving problems
- Keywords: Entropy, entropy, entropy...

#1 Goal of a Data Science Team

Transforming
Complexity
in
Complication

...
"Taming the Wild"
...
"Reducing Entropy"

Transforming Complexity in Complication

- Systematize:
  - Help the leadership / product team formalize the problem(s)
    - You can't find the *right* solution if you do not know what is the *right* problem
  - Break the big problems in smaller attackable problems
  - Define the inputs / outputs (the data interfaces)
  - help designing the data infra-structure
- "Incrementalize":
  - Define the performance metrics for evaluation for each of the parts (not "business" metrics)
  - Help building evaluation standards

But how, if the systems are complex in the first place?

Data Science "Methodology"

"Just do it" approach:

- Take the problem and go from "idea" to practice
- Prototype and Succeed or Fail Fast
- Reframe the problem
  - Don't assume there is a clear definition of the problem!
- Prune the "search space of possibilities"
- Reduce Entropy: ask why questions[2]

Fail First:

- Avoid a costly failure from the larger team
- Analogy to Storm Troopers vs Conventional Army

[2] "why" is more important than "how"

A DS team is NOT an Engineering team

- They have intersecting but different missions
- Engineers:
  - focus on defining the *solution* for a problem
  - building well, building fast
  - SLA's, performance, maintainability...
- Data Scientists should
  - focus on defining what the *problems*[3] are, and *sizing* them
  - "failing" well, failing fast
  - definitions, decomposition, evaluation...

[3] Focusing on "Solving" the wrong problem is the #1 reason for potential insuccess of Data Science / Machine Learning teams in the Industry

DS Team vs Eng Team

- Data Science requires a different way of thinking:
  - Different skill set, different tools, different rythms
  - Word of Caution: typical SDE methodologies may not apply![4]
- But we are NOT talking about "hacking" or being sloppy:
  - There is a conscious effort to systematically reduce entropy
  - There is a clear notion of the customer value of the effort
  - There is a constant effort on maintaining the highest standards

[4] Forcing SDE methodologies and managing techniques is #2 reason for potential insuccess of Data Science / Machine Learning teams in the Industry

A DS team is NOT an BI/BA team

- They have intersecting but different missions
- BI/BA (focus on the business):
  - work close to P.O. to define and develop *business* metrics
  - develop for market analysis, forecasting, segmentation,...
  - provide reporting, insights for supporting business strategy
- DS (focus on the product/feature):
  - work close to P.O. to define and decompose customer problems
  - defines *task* metrics and builds evaluation frameworks
  - proposes prototype for new features to improve business

Wait!

This is NOT about "labels" nor "territories":

- Lot of ambiguity in the labels
- People/Teams perform tasks across different missions
- Team Missions are not crystalized or fully compartimentalized
  - Not uncommon for SDE+DS+BI be owned by the same team!

But different missions have different:

- goals and expected results
- operation modes / working methodologies
- balance between need for *eficiency* and need for *robustness*

And as teams grow, such degree of differentiation may be useful

Other Missions of a DS Team (I)

Be Force Multipliers:

- DS team gets temporarily embedded in another team
- Expands the other team with a specific expertise (e.g. ML)
- Combination multiplies the impact of both team
- Once a result is achieved (fast), teams split again
- DS team may then be embedded in another team

Other Missions of a DS Team (II)

Knowledge diffusion and fertilization

- Instead of building a V1 solution, help other teams build it
  - There may be "commodity" software for the problem
  - Motivates other teams, and raises awareness for data
- Prepare short courses on some technology or tool
  - Empower other teams in solving certain V1 situations
  - Buys time for DS to work on V2

This is actually very important:

- Keep DS team focused in better understanding the problems
- Helps the entire team to think in a more sustainable way:
  - logging, gold-standards, performance metrics, feedback loops...

Building a Science Team

Caveat...

I will just give a few hints, because this is huge topic that would require a full day presentation...

What are you trying to build?

We want to build a team with the ability to:

- decompose complex systems into parts
- "formalize" problems
- build different kinds of models
- build prototypes (across domains)
- devise evaluation procedures/frameworks
- teach/communicate with other teams
- adapt to a variety of potential challenges
  - the market is constantly changing

What is the scale?

Very likely, we are starting with a small seed team

- It is very hard and expensive to hire
- But starting small may be an advantage
  - There may still not be a data-infrastruture
  - Team/Company Culture may not be ready

Key "seed" requirements:

- Autonomy
- Ability to Adapt and work across domains
- Robustness

What is the scale?

- If you spend most of your time finding your way in curvy roads
- If you don't have good high-ways, good mechanics, good fuel
- And you still want to move super-fast (and be stylish!)
  - Don't buy a Ferrari!

But thys instead!

The choice of this image has nothing to do with the fact that I own a bike like this.

Generalists vs Specialists (I)

Observation:

- In most cases in Industry, we only need a "good enough" V1
- Trying to solve 95% of the problem be the wrong thing to do!
  - 80% may be enough in practice
  - getting to 95% may be too expensive
  - 95% is irrelevant when there are parts of the system at 10%
  - we may not even be able to reliably measure 95%!!

Generalists vs Specialists (II)

- DS's should "rotate" to help achieving more "80% solutions"
  - Deciding when / where to "rotate" to is key
  - "All Terrain" CS/Eng skills are important
- System-wide perspective is very important:
  - The weakest link defines the product
- DS should mostly be Generalists, that can quicly learn a domain:
  - Autonomous Learning
  - JIT Specialization
- Solid Math + Computer Science Background!
  - You can "easily" learn the core of ML (or some flavors of ML)

A Few more Topics for Other Talks

- Managing Size, Process and Speed
  - Features vs Capabilities
  - Ensuring Speed - Team Mitosis.
- Building an actual Team
  - build momentum / team spirit
  - analogy products vs brands
- Growing Organically vs via Cross-Fertilization
  - De-bottlenecking Strategies

One last quick topic

- DS is about problem formulation, data, models, **evaluation**...
- It is not about platforms, "big-data", tools, clusters, etc
- You don't need supercomputers to solve 98% of the customer problems
  - Most actual problems can be solved with relatively cheap computers and simple (and old) tools
- For you evolution as a DS, focus on learning concepts:
  - not learning the "latest" or "coolest" tools
  - tools are rarely the bottleneck

"There is nothing more practical than a good theory"

One last quick slide

Give back!

Thank you!

luis.sarmento@gmail.com

Ask Me Anything on Data Science!
I will probably not answer your question immediately,
but you will help me thinking about topics for the next presentation, which I will then share.

Connect with me on LinkedIn
https://www.linkedin.com/in/luissarmento/