How to be a Data Scientist Impostor?

book, code, sales pitch, and rules

Abstract (first)

In this presentation we give an overview and examples of different philosophical and mathematical methodologies
and software programming techniques that would allow us to practice Data Science
almost as successfully as seasoned practitioners who have solid backgrounds in Statistics or Machine Learning.
(Or better than them.)

1. We start with Data Science market diagnosis and general strategies for problem solving.

2. Then we proceed with the outlines and descriptions of didactic topics for:
   - doing data analysis, an
   - explanations of fundamental Machine Learning (ML) algorithms.

3. Then we give talk about the practical know-how for tackling certain ML problems. (Variations of those problems often occur in "real life.")

4. Finally, we discuss a few “Game Theory” perspectives. That includes knowing what kind of people we are going to collaborate with, argue with, be examined by, be hired by.

The presentation is based on the main ideas, software code, and content of the book "How to be a data scientist impostor?", [1].

References

[1] Anton Antonov, "How to be a data scientist impostor?" book project at GitHub, (2019). (work in progress)

Abstract (new)

I revised the presentation to be more technical.

First part

The book / project description.

- The five areas of focus.
- The book dependencies
- The target audience

Cultural differences: Machine Learning vs Statistics

Learning by comparison.

On Learning Machine Learning.

What are the didactic algorithms?

We will look into one or two of those.

Shock & awe examples

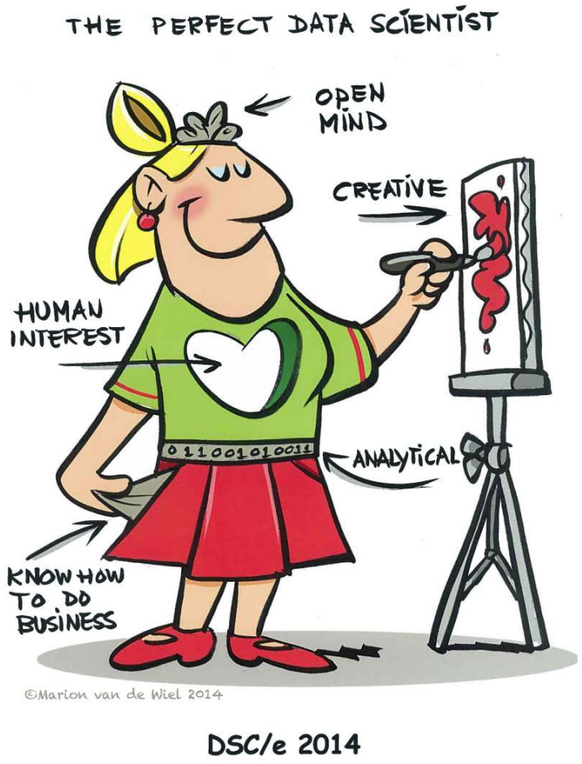
Maybe...

Second part

Let us look into the Morphological Analysis table.

The perfect data scientist

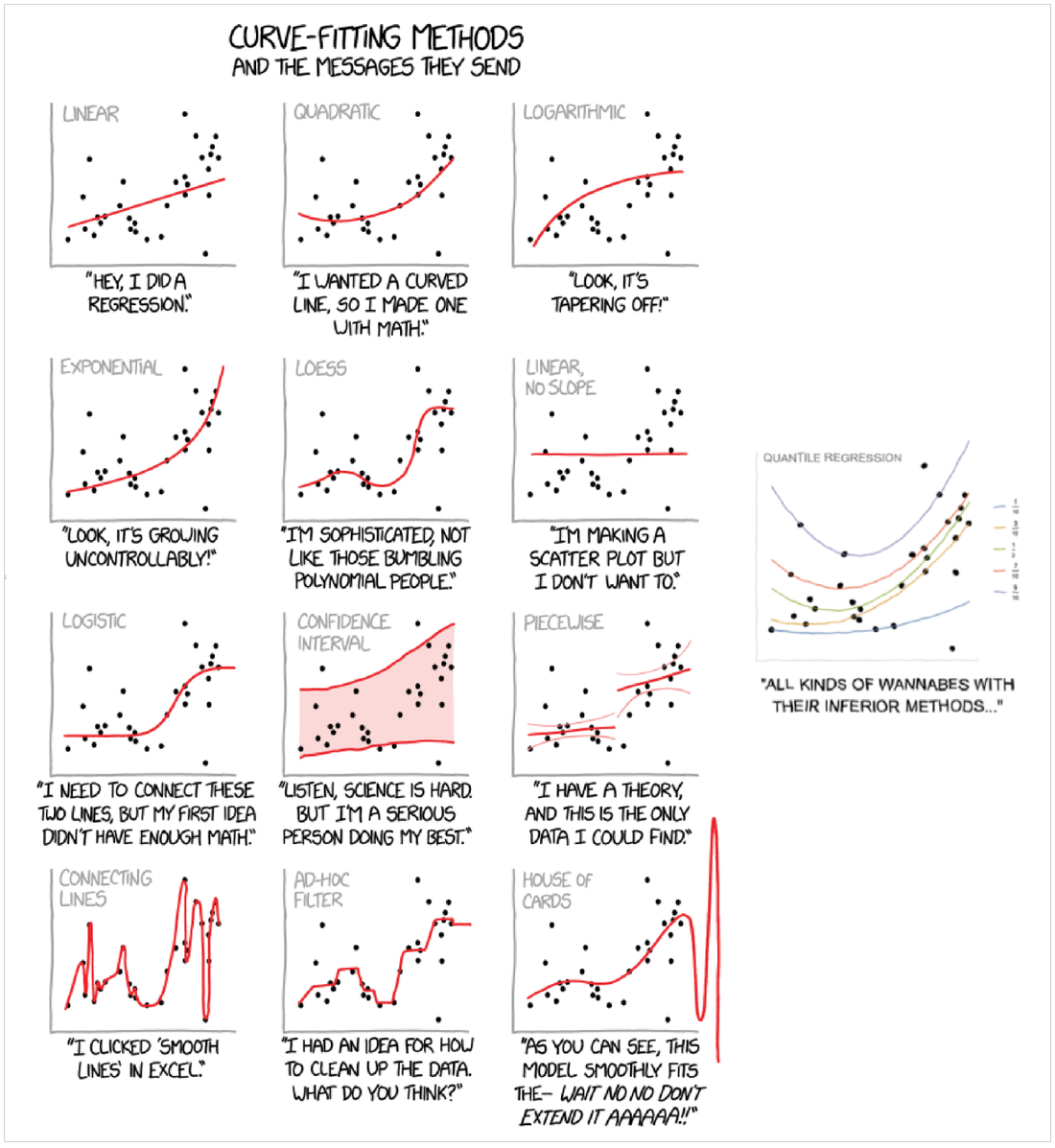
I am somewhat like that...



Please note, that I am not particularly enamored by Data Science, AI, and ML.

From my previous presentation here

Out[•]=



Categories of data scientists

Four types of data scientists:

1. Analytical (e.g. Statistics background).
2. Business oriented.
3. Development-oriented.
4. Data artists.

Two main groups:

- Doing data journalism and data analysis.
- Building data science products

The ideal end result

This is not a book “for dummies”.

Be a fox not (just) a hedgehog

This book can help hedgehogs to become foxes.

See “The signal and the noise” and/or FiveThirtyEight.com.

Jumping into the wanter before learning how to swim

... and the closely related “skin in the game.”

Ideally that is how this book can be utilized.

The practice of data science

Data preparation and analysis

Data transformation

Data visualization and presentation

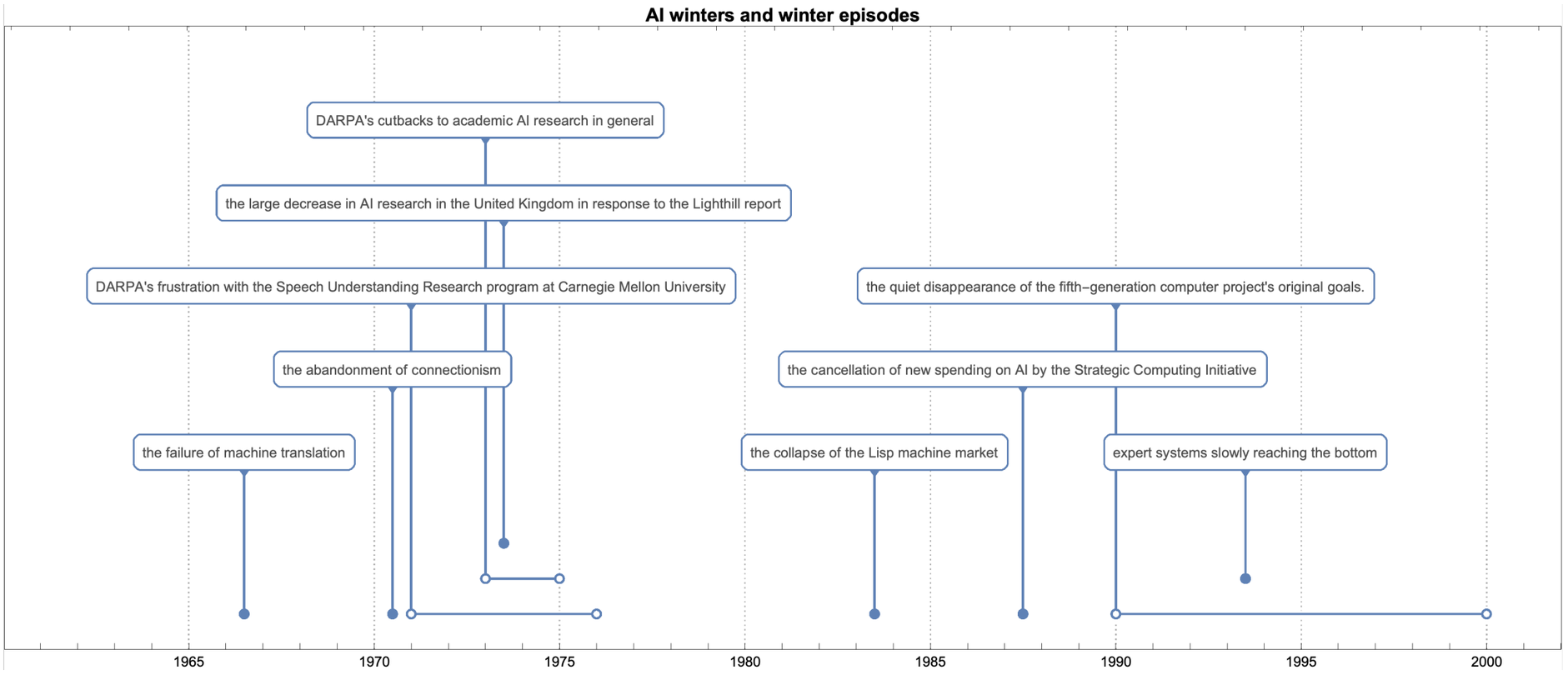
(Data/Statistical) Modeling

Do science about Data Science

Speeding up the coming of the next AI winter

AI winters

Out[•]=



But see "Dynamic Analysis and Replanning Tool".

Why?

So we can get the laypersons out of the way.

The plan to do it

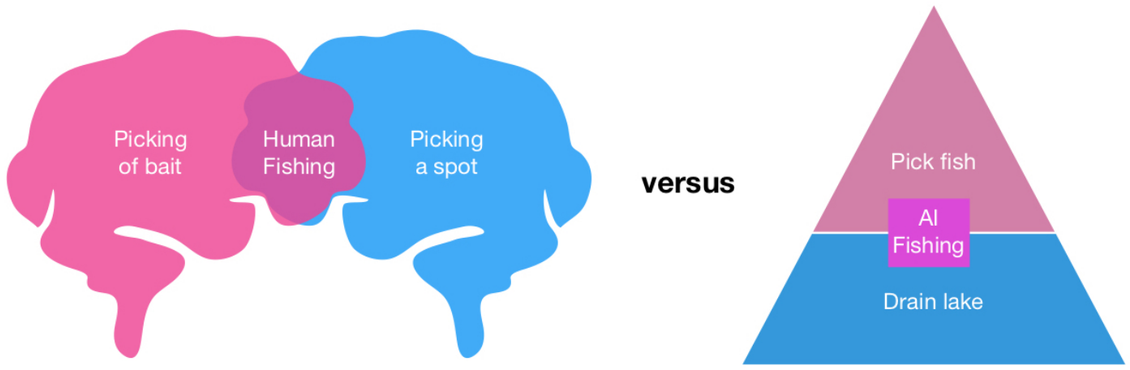
My plan is to provide two types of resources:

1. didactical like this book and the MathematicaVsR projects repository, and
2. making Machine Learning code generation chat-bots.

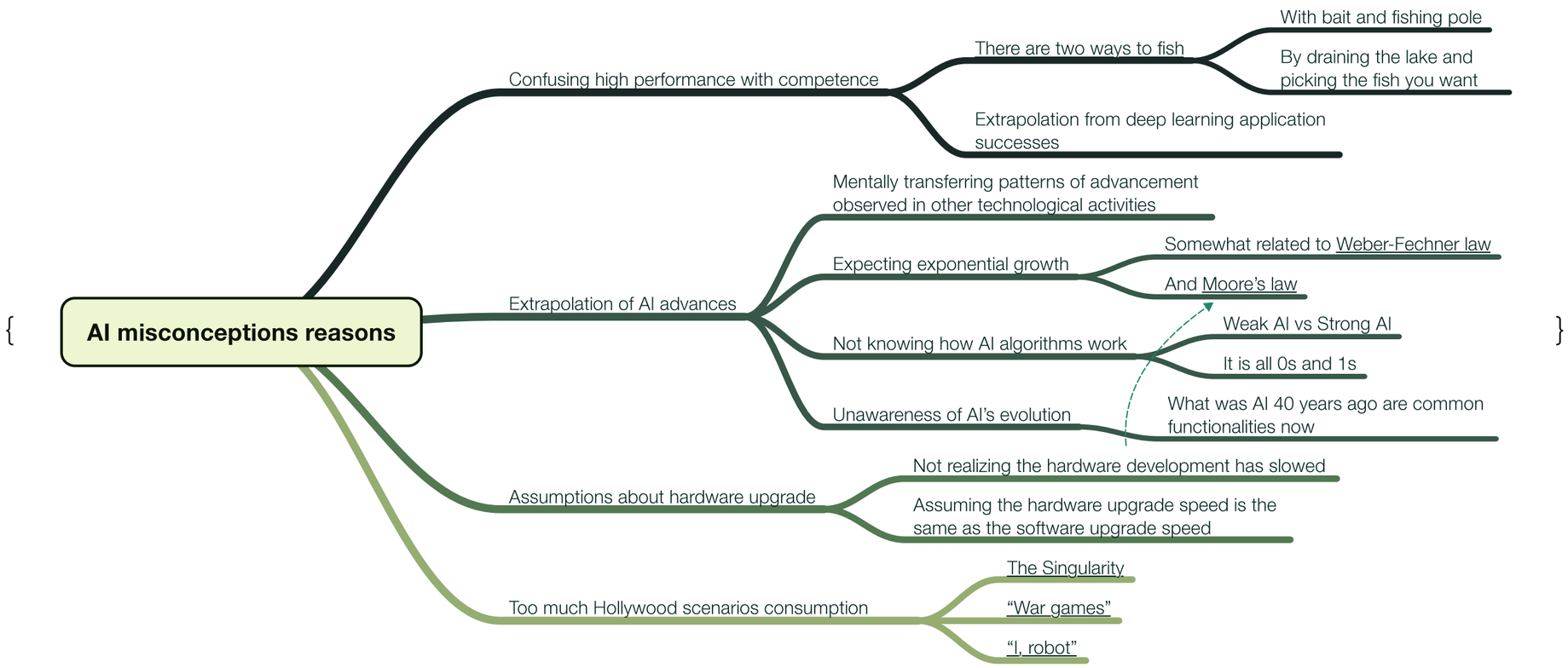
Note that doing the above means true democratization of Data Science and Machine Learning.

No strong AI, just weak AI

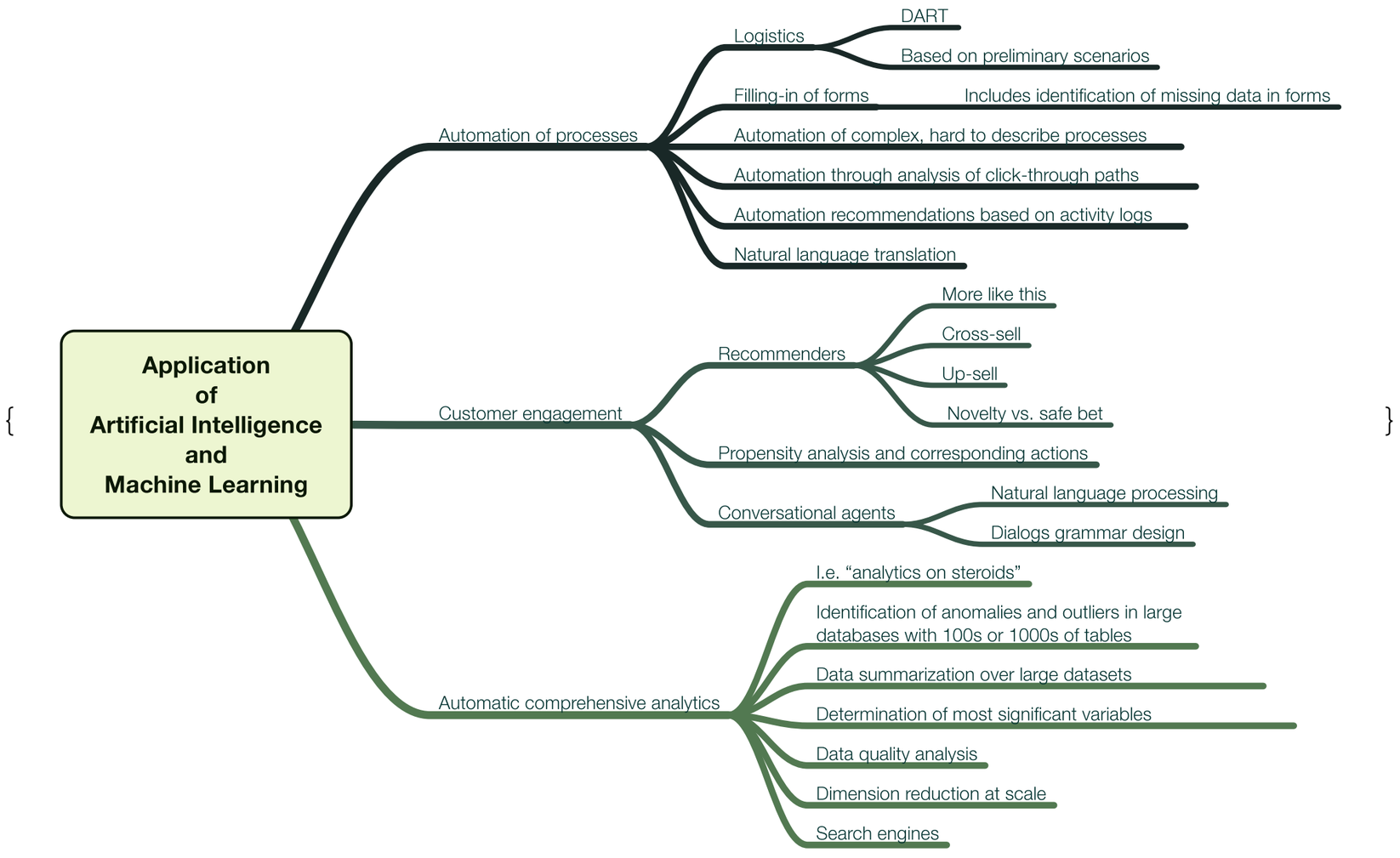
Here is an example: “two ways to fish”



Misconceptions about “AI”



Applications of AI and ML



Making ML code baristas

I wrote/am writing a separate book about it -- "Simplified Machine Learning Workflows".

ML algorithms workflows are some of the easiest ***things*** to automate with ML.

Core workflows

Classification

```
ClConUnit[dfTitanic]⟹CLConEchoDataSummary⟹CLConMakeClassifier[Method->"LogisticRegression"] ...
```

Regression

Latent Semantic Analysis

Additional important workflows

Recommendations

Anomalies detection

Feature engineering for specialized collections

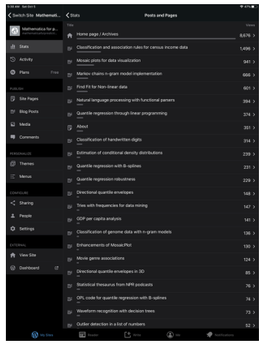
The target audience

- Simple Nuclear Physicist (SNP)
- Common Operations Research Analyst (CORA)
- Standard Machine Leaning Engineer (SMLE)
- Pretending Typical Statistician in Demand (PTSD)
- Technically Literate Opportunist (TKO)

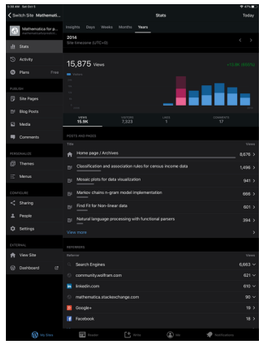
Chapters by the numbers

Let us look into some WordPress statistics.

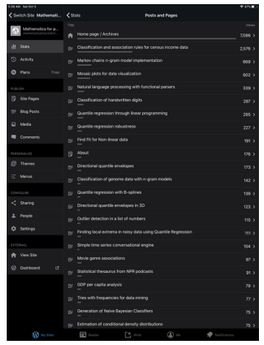
Out[ ]= {



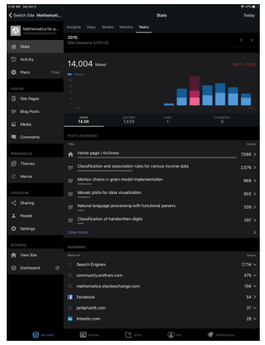
WordPress-2014-stats-detailed.png ,



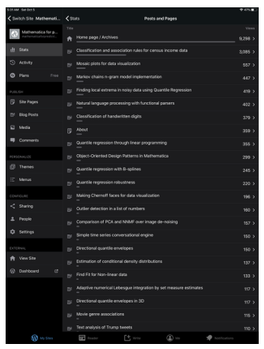
WordPress-2014-stats-general.png ,



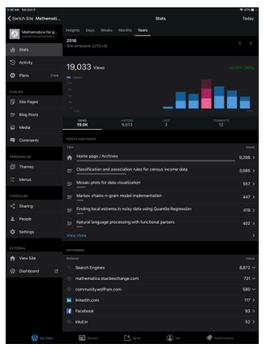
WordPress-2015-stats-detailed.png ,



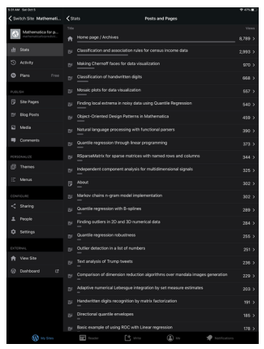
WordPress-2015-stats-general.png ,



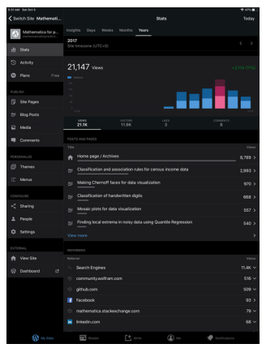
WordPress-2016-stats-detailed.png ,



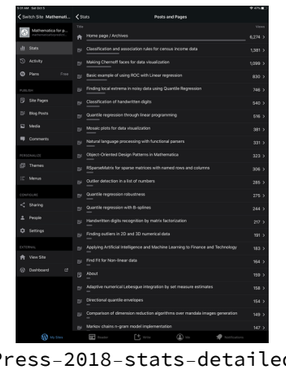
WordPress-2016-stats-general.png ,



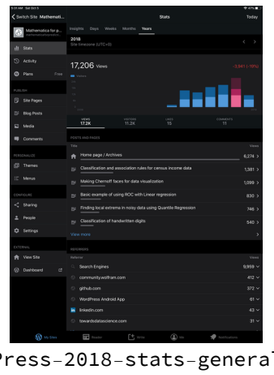
WordPress-2017-stats-detailed.png ,



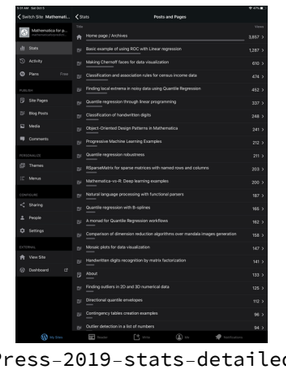
WordPress-2017-stats-general.png ,



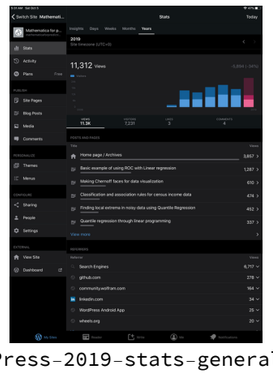
WordPress-2018-stats-detailed.png ,



WordPress-2018-stats-general.png ,



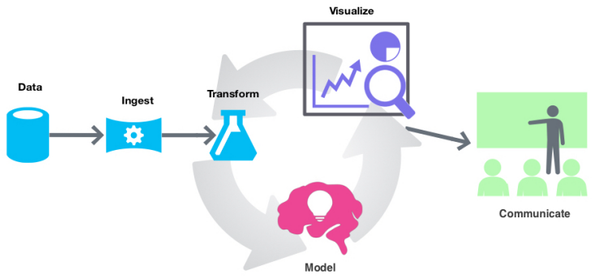
WordPress-2019-stats-detailed.png ,



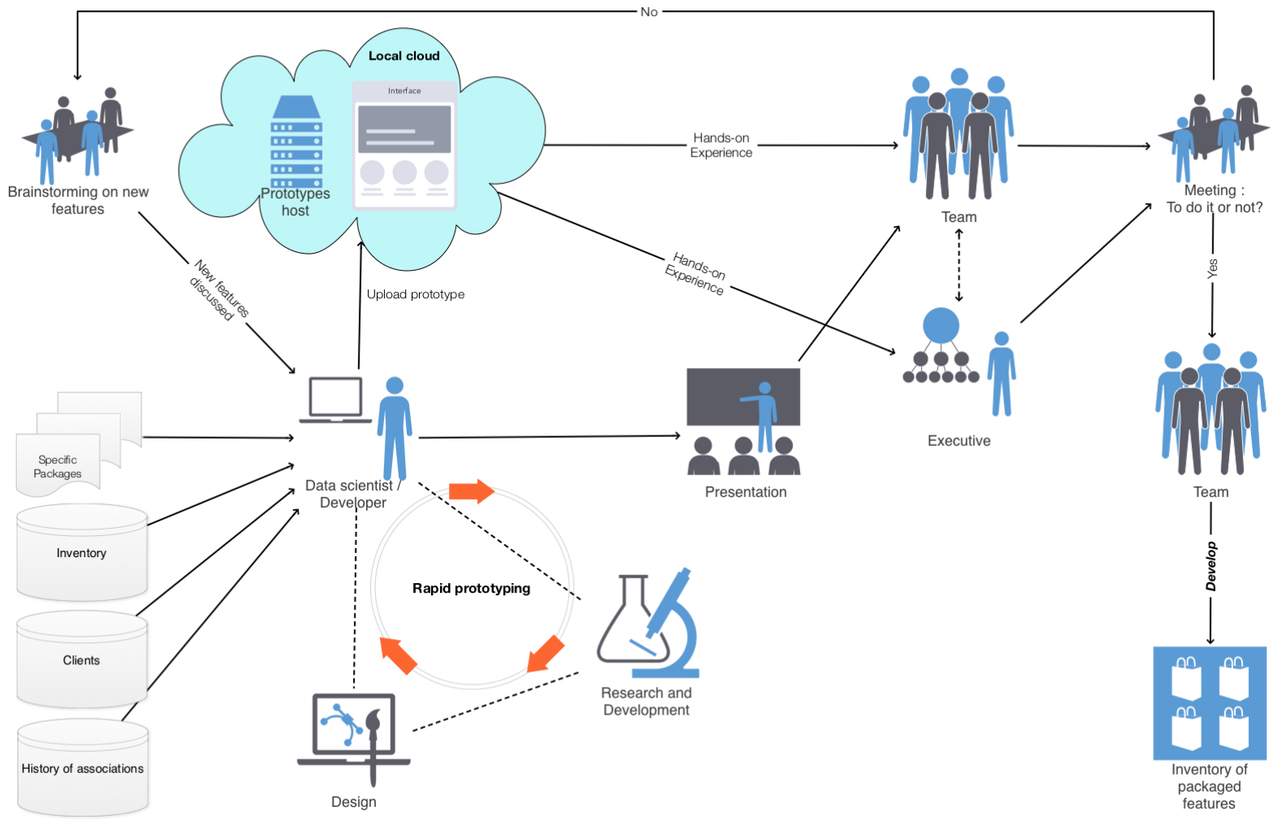
WordPress-2019-stats-general.png }

Typical Data Scientist engagements

The simple general loop



Data Scientist making rapid prototypes

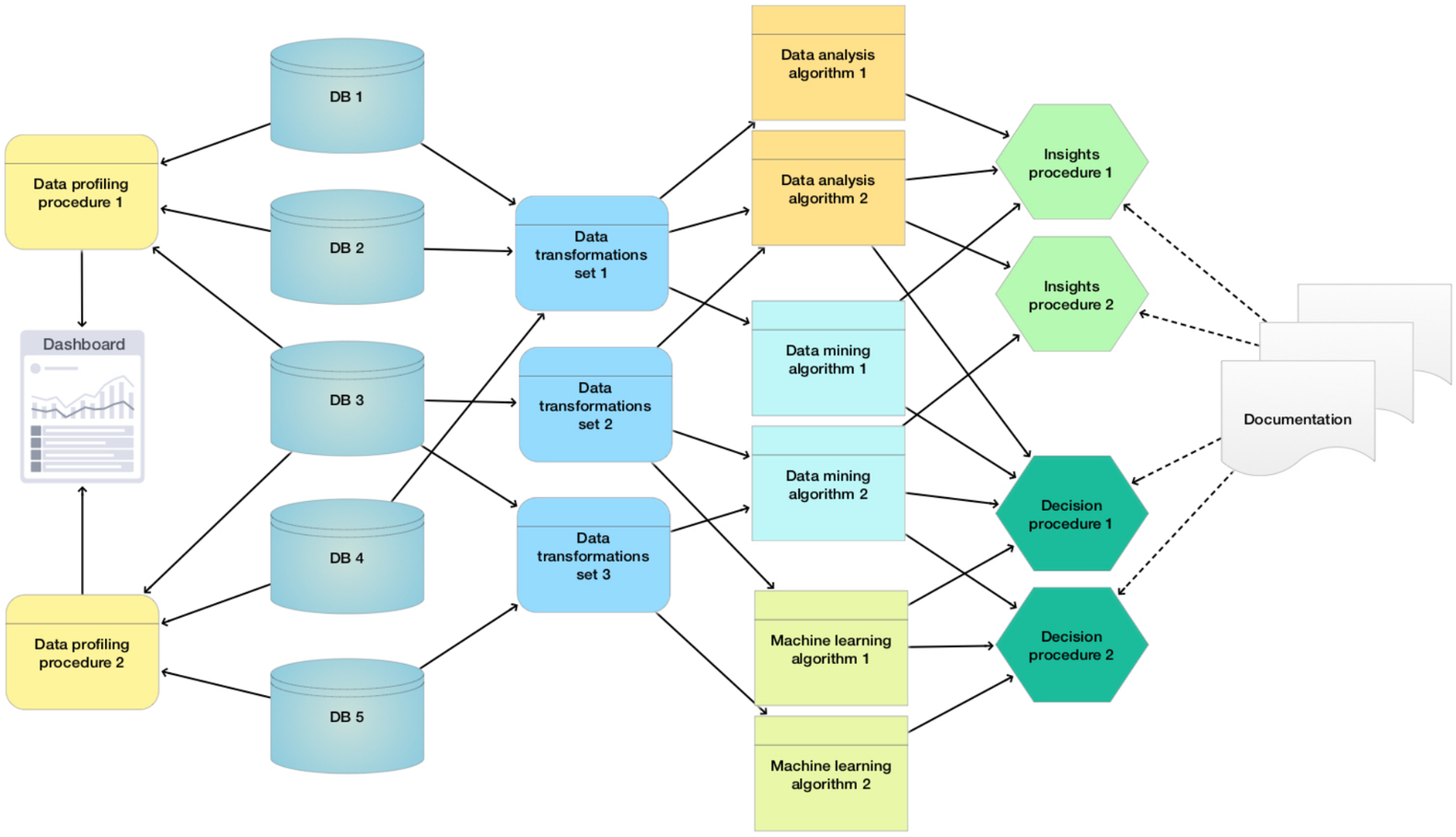


Data driven journalism

For example, "Text analysis of Trump tweets".

For some data scientists this career goal and pinnacle.

General Data Science framework (an instance)



Typical Data Scientist engagements (extended)

Is a “data scientist” a scientist?

- Active participation in the delusions is required.
- Do you want to show scientific integrity or do you want to be out of a job?
- In some job markets the latter is fine — there is high demand for data scientists.
- In other job markets, the "scientific integrity" line of behavior requires some planning.

Not surprising exhibiting integrity is not enough.

- After all, we live in a society in which
  - some entities are too big to fail,
  - the loudest and the whiniest often win.
- See also why the Boeing 737 MAX deaths happened.

Similarly, too many mad scientists and too few hunchbacks.

- Yes, as a data scientist you will be the hunchback (or just one of them.)
- Only if you are lucky you might also play the mad scientist role time-to-time.

Richard Feynman:

- "For a successful technology, reality must take precedence over public relations, for nature cannot be fooled."

Continuous improvement

Apply the usual practice of (professional) self-improvement.

Read about how spectacular predictions were made.

- The great pacific garbage patch.
- BTW, the Indian Ocean garbage patch exhibits the Pareto Principle.

Establish/develop/have notable digital imprint.

- Sort of obvious, but not many of the people I mentor(ed) do it.
- Blogs, communities, StackExchange, online lecture videos, etc.

Reading history of the AI / ML making.

- How the Perceptron was created and critiqued?
- Marvin Minsky and Frank Rosenblatt.
- Similarities between K-means and Perceptron.

Machine Learning vs Statistics

The major methodological differences to watch for

These are the principles and lesson to watch for in this presentation.

ML lessons

As discussed in Leo Breiman's paper [1] the major lessons from ML and the algorithmic culture are:

1. Multiplicity of good models (*Rashomon*).

2. No simple models for good precision results (*no Occam*).

3. Embracing high dimensionality -- not just a curse. (*Bellman*).

ML MO

4. Using simulations instead of models.

5. Mash-up of algorithms.

7. Collaborative filtering application.

Machine Learning vs Statistics (2nd)

Morphological analysis of behavior

Statistics

Frequentist/objective statistics tells us what to do if we have a model. Bayesian/subjective statistics tells us what to do if we know (or have premonition) of the probabilities priors.

1. Gather, transform, clean, normalize data.
2. Look how to apply a model with i.i.d. assumption.
   (Come up with a model.)
3. Extract parameters by some fitting or optimization procedure.
   (Model fitting.)
4. Verify model assumptions from obtained results.
   I.i.d., white noise, quantile plots.
5. Test and verify prediction.
   Goodness of fit tests.

Machine learning

See “Morphological analysis of algorithms vs problems” and “Morphological analysis of algorithms vs data” .

Dedicated presentation (if there is a interest)

"Exemplifying cultural differences between Machine Learning and Statistics.nb"

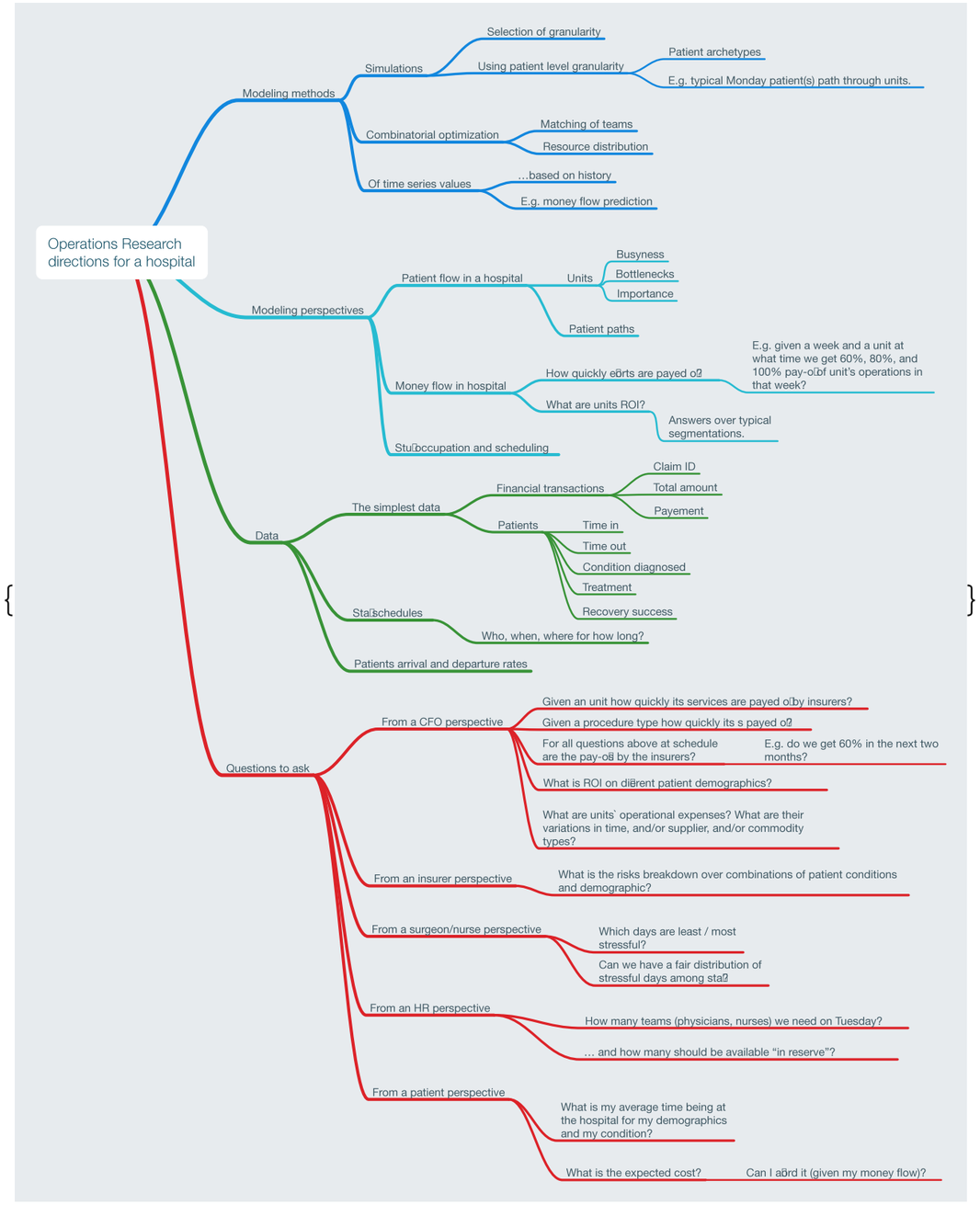
Brainstorming, invention, and exposition techniques

Do Morphological Analysis

Design of conversational agents

Structure it with mind-maps

Take the CFO perspective



Make a Reference Model

Diagrams

Flow-charts

UML

The most important Machine Learning algorithms

... from a didactic and practical perspective.

Clustering / Nearest Neighbors

Dimension Reduction

Decision trees

Naive Bayesian Classifiers

Apriori

Progressive Machine Learning

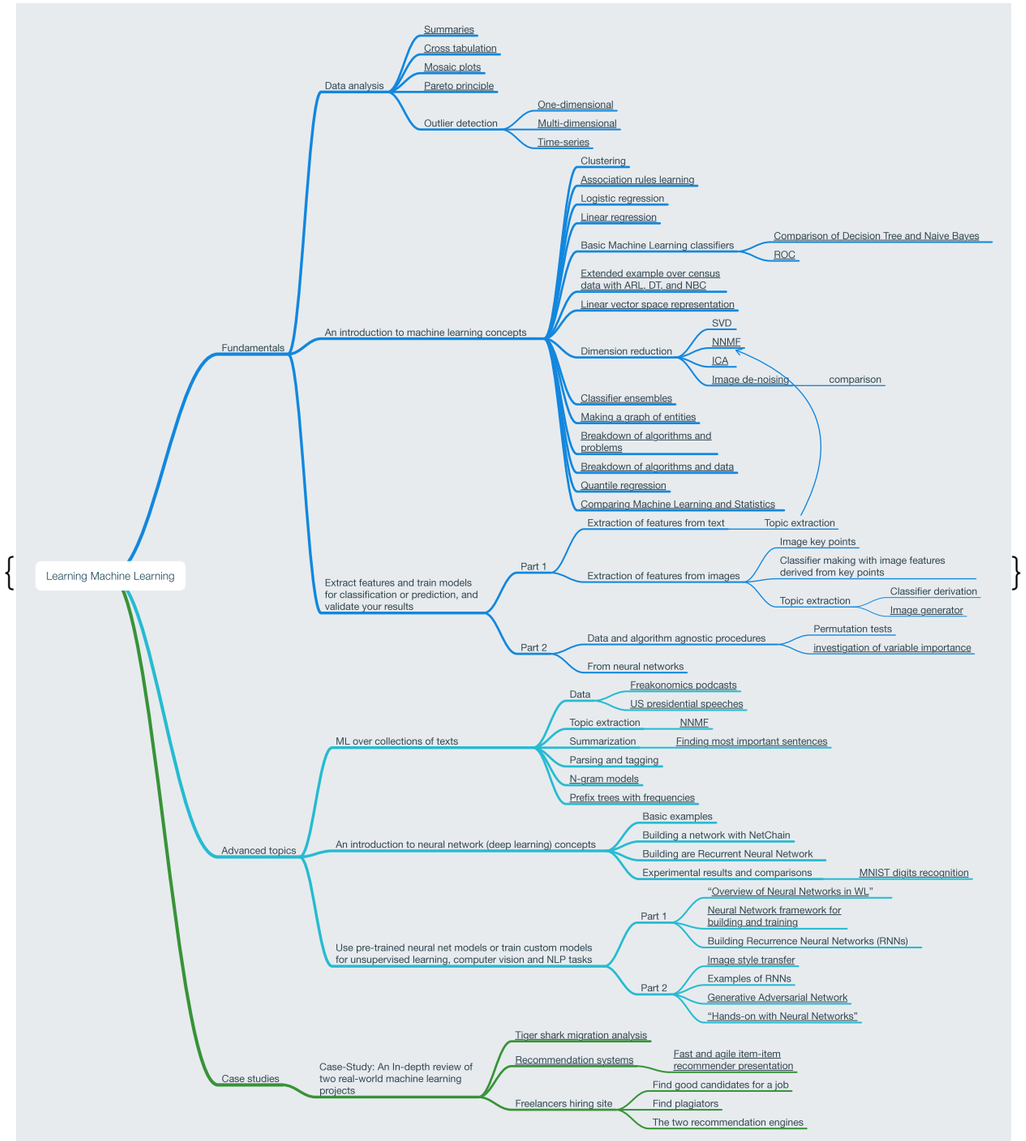
Learning Machine Learning

... by problem solving.

That was an the original idea behind a DS/ML book of mine.

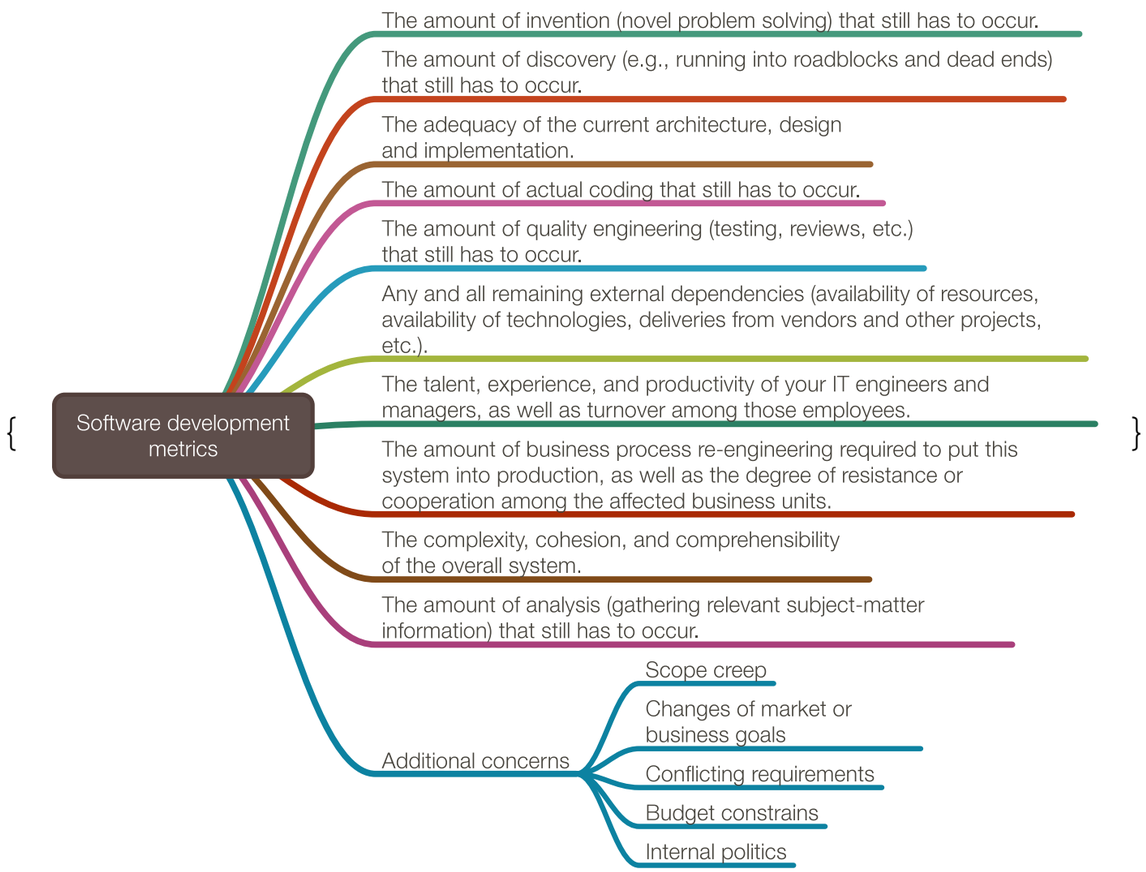
Turned out is simpler to write/finish the presented one.

Here is a mind-map of envisioned course.



Acquire software engineering skills

Software development discipline and management



Architectural

Unit testing

Data expectations.

Algorithms theoretical expectations.

Random data and random pipelines scaffolding.

Data packages

Why “Model management”?

Model management is both a natural thing to do and easy to automate. (As most DS/ML workflows.)

The “Dev-ops” for DS and ML and Model management are needed as separate tools because most data scientists do not know or apply software engineering principles to their DS/ML software work.

The macro rules (1st wave)

There is no substitute for knowledge

William Edwards Deming

Or “There is no substitute for scientific knowledge.”

Examples

The great conversation.

The application of the 103 great ideas by Mortimer Adler for doing LSA of movie summaries.

The best quality of a mathematician is give examples and counter examples

Examples

The “How to be a Data Scientist Impostor?” book and this presentation.

The book "Theorems and Counterexamples in Mathematics".

Search for the invariant

The macro rules (2nd wave)

Data Science is third-best

... when it comes to Mathematical Modeling.

And maybe event the fourth best.

This is closely related to the optimization perspective.

Example

Strategic improvement of

Adopt and apply the optimization perspective

Also applies for the general process.

Examples

Applied in almost anything I do/did but let us discuss this in healthcare.

Thinking technologically

Making analogies a lot and often

Terrorism vs Earthquakes

Work on projects that stir the blood of men

... or find people how work on such projects.

“Make no little plans; they have no magic to stir men’s blood and probably themselves will not be realized. Make big plans; aim high in hope and work, remembering that a noble, logical diagram once recorded will never die, but long after we are gone be a living thing, asserting itself with ever-growing insistency. Remember that our sons and our grandsons are going to do things that would stagger us. Let your watchword be order and your beacon beauty.” -- Daniel Burnham.

The rules (2nd wave) (Data Analysis)

Form a sense of data ownership

Out of sample much?

The OKCupid hack by a mathematician.

“But mathematically, McKinlay’s compatibility with women in Los Angeles was abysmal. OkCupid’s algorithms use only the questions that both potential matches decide to answer, and the match questions McKinlay had chosen—more or less at random—had proven unpopular. When he scrolled through his matches, fewer than 100 women would appear above the 90 percent compatibility mark. And that was in a city containing some 2 million women (approximately 80,000 of them on OkCupid). On a site where compatibility equals visibility, he was practically a ghost.”

Find and explain outliers

Pareto principle everywhere

The rules (3rd wave) (Machine Learning)

Crowdsourcing the intelligence

The Collaborative Filtering trick.

Mash it up

Making ensembles of everything.

The meta-rules (4th wave)

Be a typical outlier

Many ways to do that, mostly from a game theory perspective.

Cultivate high discerning ability

Notice and analyze outliers right away

Looking the part

Because you talking the talk is mostly boring to others.

Find a way to build your stone wall/castle

Future plans

When the book is going to finished?

- Finishing a two other books first.
- Making R-Markdown version of the notebooks.
  - At least 30% are both in Mathematica and R.
  - See MathematicaVsR at GitHub.

Other questions?