FOURKIND

Machine Learning in practice - considerations, examples and some rambling

Nemanja Cvijanovic & Bruce Ferguson

- Originally from Belgrade, Serbia

- Studied "Technische Informatik" at the RWTH University in Aachen

- PhD at the Philips Research Laboratories and the TU/e
  - Topic: Speech communication systems in realistic environments
  - Basically making processing, understanding and use of various types of data: audio, speech, ultrasound, behavioral, physiological...

- Worked in DS consulting for about 2.5 years (0.5@Fourkind)
  - Cool DS/ML or even easier DE projects when lucky
  - Business development
  - Project/team management

- Originally from Toronto, Canada
- Machine Learning Consultant, Partner, Fourkind B.V.
- Background in Geophysics
- 28 years of experience in designing and implementing top-tier algorithmic solutions. Previously Senior Data Scientist /Machine Learning Engineer at KPN. Prior to that was a Quant Strategist in Investment Management and Capital Markets industries, and prior to that a Geophysicist in the Canadian Oil & Gas Industry, employing ML techniques throughout career.

Basically **some details about a lot of things** to inspire you to potentially try cool stuff during isolation (including a free to watch movie recommendation!)

- Some practical considerations for ML projects

- Examples
  - Algorithmic trading
  - Generative design
  - Reinforcement learning

What and why?

AI
FOR CREATIVITY

Augment the expert.

https://ai-whisky.com/

FOOD & WINE

TechCrunch

THE TIMES

POPULAR MECHANICS

Forbes

TNW

THE Sun

NEW YORK POST

Practical considerations

What problem are you trying to solve?

Business: "I secured budget for a project for you. I'd like you to build me an attribution model so that I can see what percentage of conversions each channel is responsible for."

Data Scientist: "What will you do with that information? "

Business: "Well allocate my marketing budget of course."

Data Scientist: "Let's assume my model allocates 30% of the credit to channel A, 30% to channel B and 40% to channel C. What do you do with that information?"

Business: "Hmm"

Data Scientist: "..."

What are you trying to achieve by solving it?

- Startups often kick-off with an amazing idea, a cool highly complex product and tech.
- However, a lot of startups also need to adapt their original idea/product in order to get an investment.

- Why? - Misunderstanding of the market and the consumers needs...

What are you trying to achieve by solving it?

- Me (a wine enthusiast): "Hey we've used algorithms to help develop a whiskey recipe at my company before. I'm sure I can come up with some ML use cases to make wine production more efficient or even design a new recipe. What do you think?"

- French vigneron in Beune: "Get off my property!"

Is the world/your potential customer ready for it?

Do you need ML to solve it?

"Rule #1: Don't be afraid to launch a product without machine learning.

Machine learning is cool, but it requires data. Theoretically, you can take data from a different problem and then tweak the model for a new product, but this will likely underperform basic heuristics. If you think that machine learning will give you a 100% boost, then a heuristic will get you 50% of the way there."

Rules of Machine Learning: Best Practices for ML Engineering (Martin Zinkevich et al), http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf

In many cases, you don't need machine learning in order to solve a problem

Data Scientist: “we built a model for predicting the channel (email, phone, contact form…) customers contact us in”

PO: “awesome, let’s take this to production!”

Data Scientist: “great, I’ll work with our engineers to make it happen”

(development continues)

Engineer: “why don’t we just collect the correct channel information when someone calls or emails us?”

PO: “…”

Data Scientist: “…”

There might be simpler, faster and better solutions

Business: “We want to use image processing to link production parameters (mostly machine settings) to the final product and its imperfections. We'll start to log the production parameters and take photos of a sample product from each batch in our labs.”

Data Scientist: “Sounds good, when can I get access to some sample data?”

Business: “Oh one more thing. Since the photos taken will be different every time as they we'll have different people take them taken from different devices, angles, lighting conditions etc., can your algorithm include an image preprocessing stage where it deals with all that?”

Data Scientist: “Or you could just standardize the way the photos are taken and communicate that before collecting the data?”

Often, the problem can be simplified

Do you have data you need for your ML algorithm?

Client: “we want to be able to predict who is most likely to be our customer in the future”

Data Scientist: “OK, for whom would you like to be able to predict that?”

Client: “for all people that aren’t already our customers”

Sometimes, the data you already have may not be useful

Client: “we want to be able to predict anomalies in our products during production as they raise an alarm and cause the whole production line to stop.”

Data Scientist: “OK, what does the anomaly data look like?”

Client: “Well the dimensions of the product are well outside the specifications. ”

Data Scientist: "Understood, and you have all those logs labeled as anomalies?"

Client: "No we discard them and the products as we know that they are wrong."

Sometimes, the data you already have may not be useful

Client: “We want to build an attribution model using Google Analytics data.”

Data Scientist: “It captures the full customer journey right? ”

Client: "Well, it doesn't really work if the user uses multiple devices, if they disable cookies, clear their browser history or come from Facebook. They can also see an add but not click on it, we can't see that either."

Sometimes, the data you have is extremely noisy

Not understanding technical constraints can make a machine learning project fail

Business: “let’s use machine learning to automatically assign tickets to the proper technician”

Data Scientist: “sounds plausible, I’ll get to work”

(development continues)

Data Scientist: “here’s the best model I could make. In simulation, it’s only wrong 0.1% of the time”

Business: “that’s unacceptable – it can’t assign work to the wrong technician”

Data Scientist: “but it’s function approximation…by definition, it can’t–”

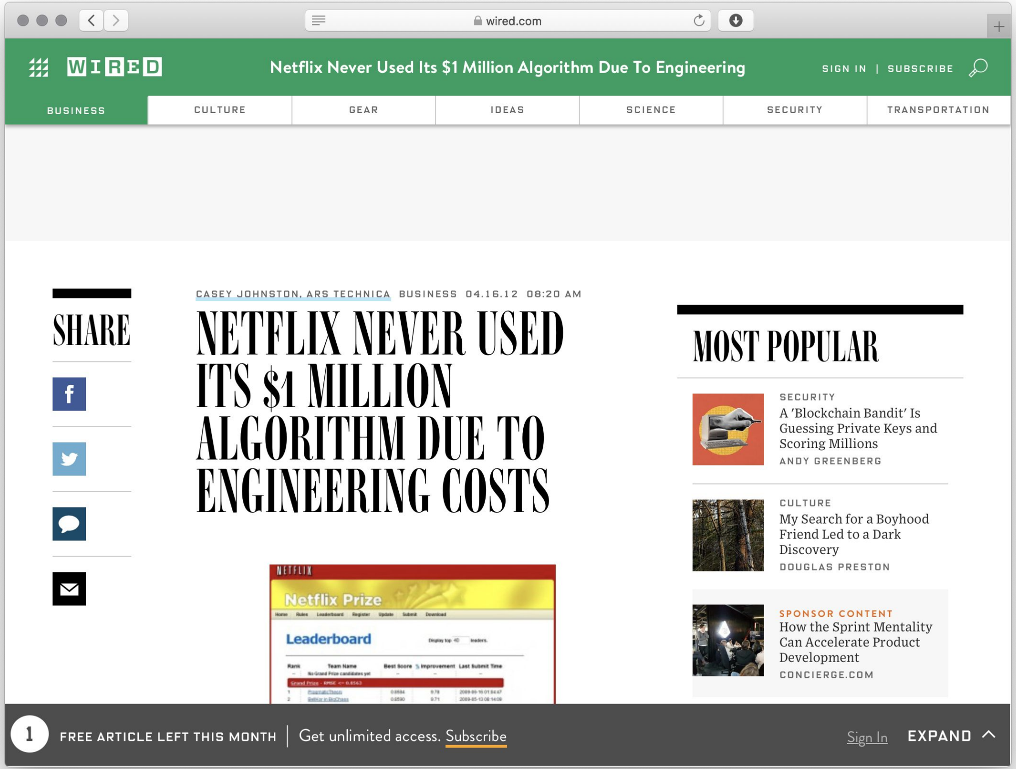
Business: “no exceptions”

Data Scientist: “…”

Are you aligned with the management's expectations?

https://www.wired.com/2009/09/bellkors-pragmatic-chaos-wins-1-million-netflix-prize/

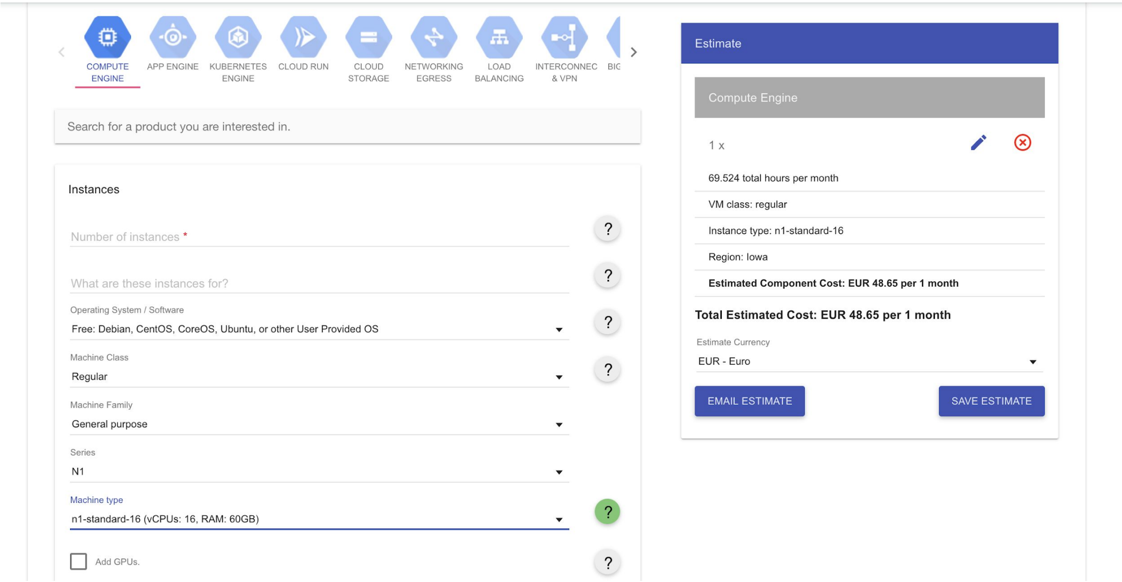
Designing algorithms without technical constraints in mind...



https://www.wired.com/2012/04/netflix-prize-costs/

… can lead to brilliant but not viable solutions.

Nowadays, storage and processing power are accessible and cheap



*Google GCP pricing calculator and play around a bit. This is no Google endorsement - you can use AWS, Azure or any other provider you like. Each one of them provides free credits to start and student discounts (GCP gives you 300$ of credit for example).

Why use the cloud?

- Speed of development is unparalleled
  - You can get a server in seconds
  - You can utilise services & components that are highly developed
  - Services managed by the cloud provider
  - (Auto-)Scaling
- Low initial cost - low threshold for experimenting & starting something new
- Many tutorial and learning resources available

Example 1: Software update requires setup to change

What do I need to know?

Jobs will start failing in February 2020 as a way to notify all users of the requirement to migrate affected pipelines to supported SDKs before March 31, 2020. Adding your project to the allow list lets us know that you are working on migrating your projects before the March deadline. **After March 31, 2020, any job still running on Apache Beam or Cloud Dataflow SDK versions 2.4.0 or earlier will fail.**

Your projects listed below will be affected by this change:

- [redacted]

What do I need to do?

1. To exempt jobs running affected SDKs from failure between February and March 2020, request that the project ID(s) be added to the "allow" list. **Requests must be submitted by January 31, 2020.**

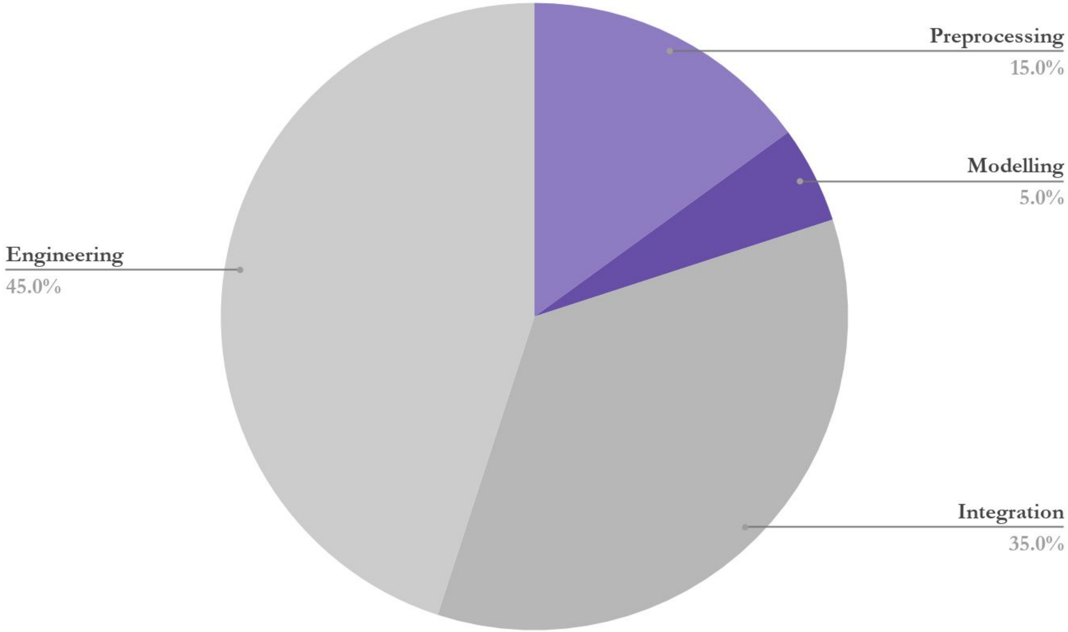
Example 2: Ill-advised behavior

Plot: Bob is interested in tech and wants to learn about stream processing. Sets up a process he found in a tutorial. Forgets to close the job, enables autoscaling and leaves on vacation.

Result: Due to a setup error the job triggers the use of ~700 virtual machines and incurs about 80000$ in cost in 10 days. I noticed it during a random check. Don't be like Bob.

Make sure you or someone on your team understands what's going on and monitors your cloud setup

Always make a proof-of-concept



*may not apply in research

Always make a proof-of-concept

Always make a proof-of-concept

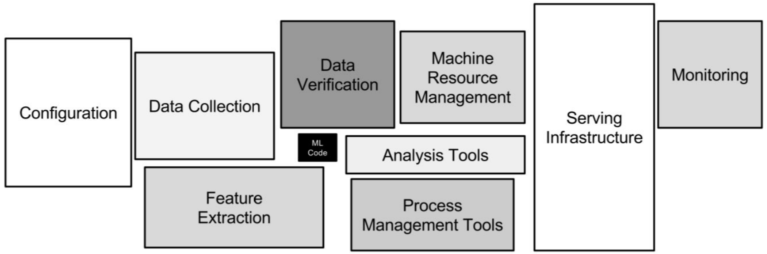


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Machine learning projects can, and will, fail from time to time. To start, make the simplest model possible, and test its effectiveness using the simplest possible process. Adding surrounding infrastructure without validating the approach first is asking for trouble.

Hidden Technical Debt in Machine Learning (D. Sculley, Gary Holt, Daniel Golovin et al), http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Machine Learning: The High-Interest Credit Card of Technical Debt (D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov et al), https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43146.pdf

When the machine learning part of a machine learning project fails, it's because of bad features/feature engineering

When a model fails to predict something, it's because the information used to train it lacks predictive power.

This, in turn, is because either the information used is wrong, or not engineered into useful features.

Applied machine learning is basically an exercise in feature engineering (note: feature engineering is hard).

Garbage In, Garbage Out.

Good feature engineering + a naïve learning algorithm trumps bad engineering + a sophisticated learning algorithm 99% of the time.

Data Scientists that understand this are far more effective than those who don't.

Focus on good features rather than a complex algorithm

At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used. If you have many independent features that each correlate well with the class, learning is easy. On the other hand, if the class is a very complex function of the features, you may not be able to learn it. Often, the raw data is not in a form that is amenable to learning, but you can construct features from it that are. This is typically where most of the effort in a machine learning project goes. It is often also one of the most interesting parts, where intuition, creativity and "black art" are as important as the technical stuff.

A Few Useful Things to Know about Machine Learning (Pedro Domingos), https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf

When the machine learning part of a machine learning project fails, it's because of bad features/feature engineering

-https://www.featuretools.com/

Python library for automated feature engineering

Think about updates, monitoring, performance measurement and changes in the data your algorithm relies on

Good metrics are:

- \+ Measurable
- \+ Objective and unhackable*
- \+ Derived from strategy
- \+ Describe what you want and need to know
- \+ Are usable in every-day work
- \+ Understanded and accessible by everyone
- \+ Validated regularly

Bad metrics are:

- \- Unmeasurable
- \- Subjective and/or hackable
- \- Derived from coffee table conversation
- \- Chosen because they were easily available
- \- Too big to have an impact on or too narrow to describe different cases
- \- Unknown to other stakeholders and in worst case even to you
- \- Not trusted or fully understood

Credit: Jan Hiekkaranta

Make sure you have a solid metric to measure success/performance

In many projects, we see algorithms in production being updated/retrained daily - to include the new training data. Be careful when doing this!

Example - churn management in a telco. You craft a model that predicts churn. You go live with it and the company starts acting on that information. The top 10% of possible churners are being contacted and some are won back and some are not. You retrain your model daily and slowly unlearn your original signal. The next iteration of that model will predict something entirely different and it only gets worse at it as time goes on.

https://medium.com/value-stream-design/the-unlearning-machines-876f320ff7b8

How do you adapt your model?

- **Batch vs stream processing**

https://medium.com/value-stream-design/architecting-a-scalable-real-time-learning-system-95623d27dd15

- **Concept drift** - Concept drift primarily refers to an online supervised learning scenario when the relation between the input data and the target variable changes over time. Example: spam filtering

https://dl.acm.org/doi/10.1145/2523813

How does your model deal with new data?

Use case: ML for algorithmic trading

Some Definitions

Algorithmic Trading: Automated trading driven by an algorithm (in this case ML)

Buy Side: Hedge Funds, Pension Funds, Proprietary Trading firms

Sell Side: Brokerage Firms, Investment Banks

Order Book: The structure that holds the active buy and sell quotes for each stock

Brokerage Fee: The cost incurred of crossing the spread to take liquidity

Rebate: The credit gained from posting liquidity (when the market comes to you)

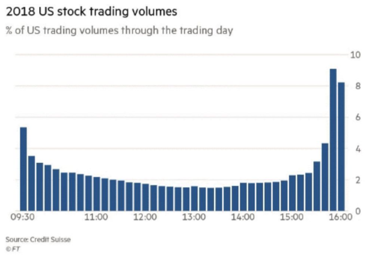
You can build a Deep Neural Network that perfectly predicts the future, but if you can't execute on it, it's useless. In this context, this might mean that the predictions just aren't produced quickly enough to be useful, or the production environment is unstable.

For HFT, the predictions need to be lightning fast, so models should have as few computations as possible. However, you still need to be right the majority of the time. Otherwise, you're just wrong faster, which can lead to hemorrhaging from brokerage fees.

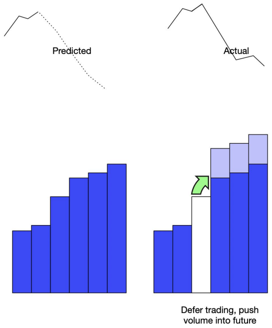
You need to be crystal-clear about requirements and your environment, and burden your system accordingly

Example: A client sends a huge order to a trading desk and tells it to get VWAP (volume-weighted average price) or better. The trading is spread out over the day so that it's size minimally impacts the market and the price is determined by:

$$VWAP = \frac{\sum_{i=1}^{N}(Volume_i * Price_i)}{\sum_{i=1}^{N} Volume_i}$$



Buy Order



Sell Side (Brokerage Firm)

Example: Market-making strategy. Trading both sides of market trying to capture the spread. The goal is to try to passively place orders at levels that the market will touch (in order to capture the rebate), trade as many shares as possible, and be flat at the end of the day.

Buy Side
(Hedge Fund or Prop Trading Firm)

Tech Specs

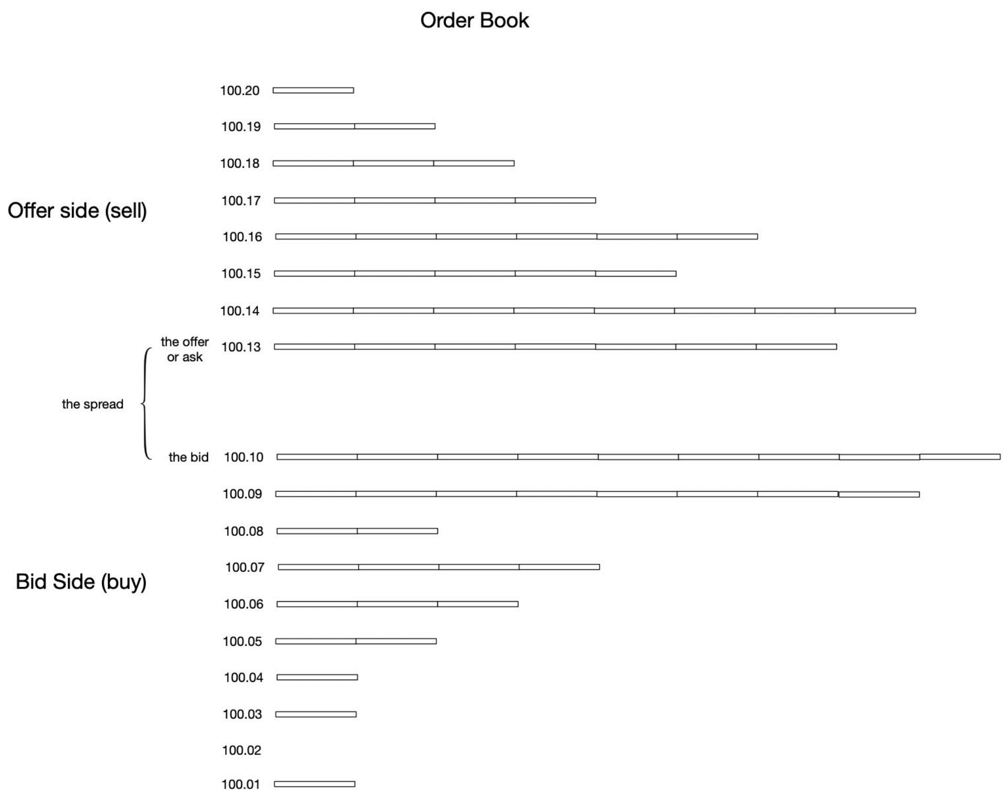
48-core HP Proliant Servers, Co-located on the NYSE
Separate Network Interface Controller (NIC card)
Platform & Model Language: Scala
Messaging: Akka

Data

Data Sources

Data Sources:

- Order Book
  - Activity
  - Imbalances
  - Patterns
- Sector ETFs
  - Other stocks in group
- Macroeconomic data
  - Regime classification
- Twitter Feed & RSS News Feeds
  - Chatter



The Order Book

Order Book imbalances can be an indication of an impending directional move. For example, if a disproportionately large volume of orders begins to appear on one side of the book, the implication is that the market will try to move away from that, as counterparties seek to improve their pricing. Thus, you will often see a bunch of order cancellations on the other side.

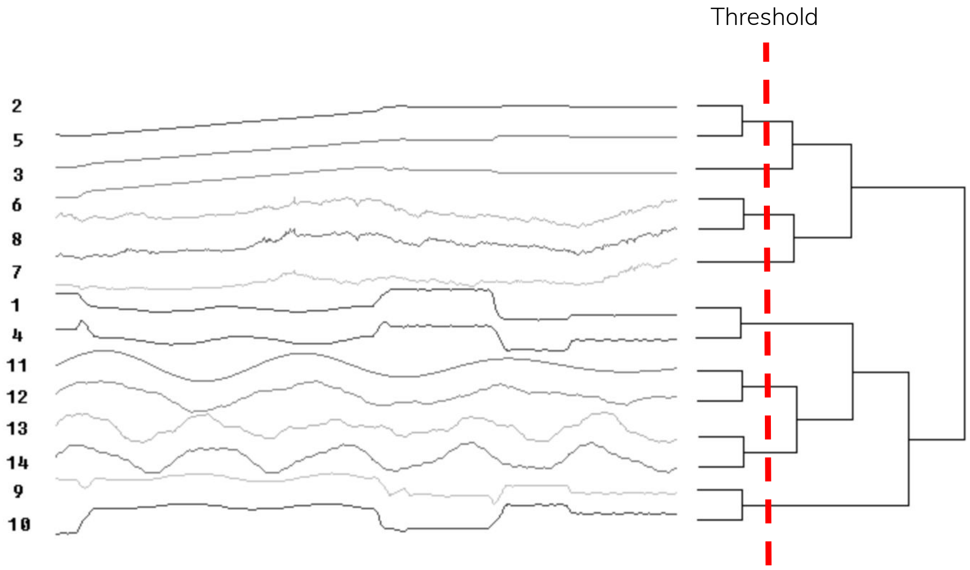
Book Pressure is an example of a feature that is indicative of impending movement. It calculates the ratio of the value held within N levels on the bid side versus the ask side. Values greater than 1.0 suggest upward pressure, and values less than 1.0 suggest downward pressure. How this varies over time holds some information.

$$BookPressure = \frac{(\sum_{i=1}^{N} |\, price - mid \,| \; * \; size)_{bid}}{(\sum_{i=1}^{N} |\, price - mid \,| \; * \; size)_{ask}}$$

Feature Engineering

Dimensionality Reduction

In Capital Markets, time series data are often highly correlated. Including highly correlated data will lead to higher than necessary degrees of freedom, which will severely hurt performance out-of-sample. As such, dimensionality reduction should be performed.



One needs to think about how to mix features with different frequencies, what it means with respect to the forecast horizon, etc…

The forecast horizon should be commensurate with the sample interval. If there are multiple sample intervals, one should consider multi-scale approaches.

As the frequency increases, the distribution of returns becomes more leptokurtic.

Correlation and Scale

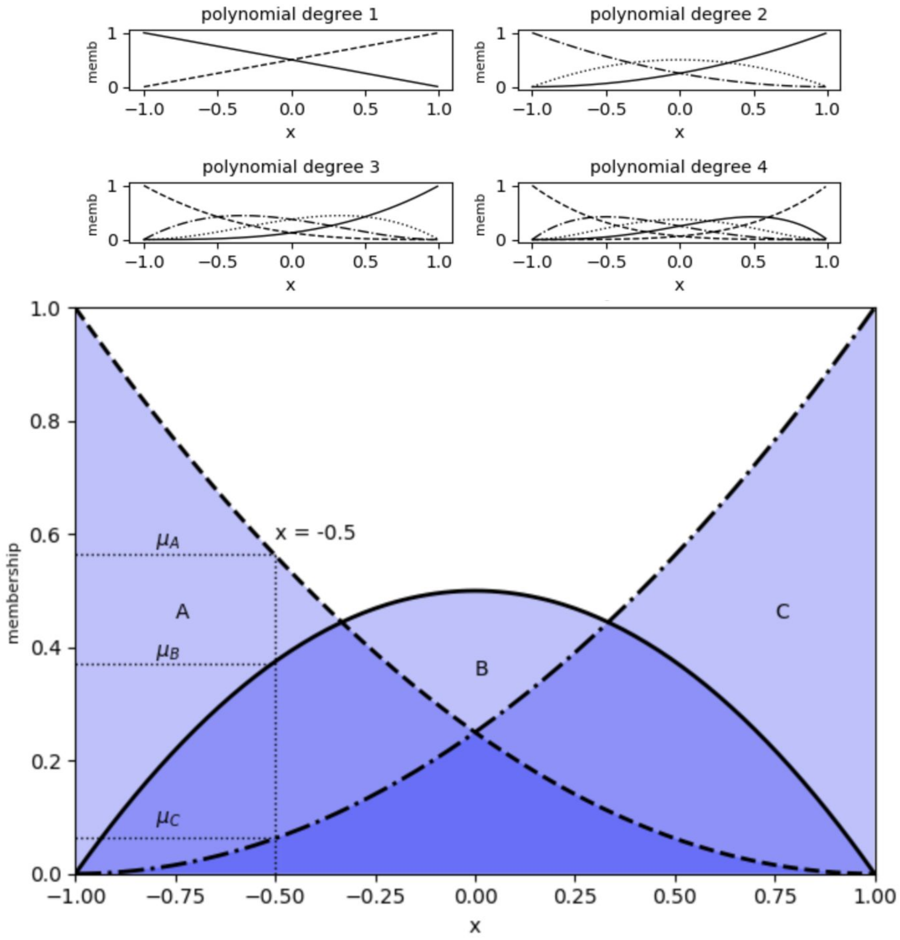
Volume Binning

In the case of HFT, trading too frequently/aggressively can lead to excessive brokerage fees. Trading too infrequently/passively can lead to a huge opportunity cost if the market suddenly moves away. This is essentially the *Trader's Dilemma*.

Being appropriately active: One possibility is to bin by volume. What this does is create more samples during high volume/volatility times, and fewer during low volume/volatility times. From the algorithm's perspective, this is like time dilation. It makes an otherwise highly leptokurtic (fat-tailed) distribution more Gaussian, which helps most ML algorithms.

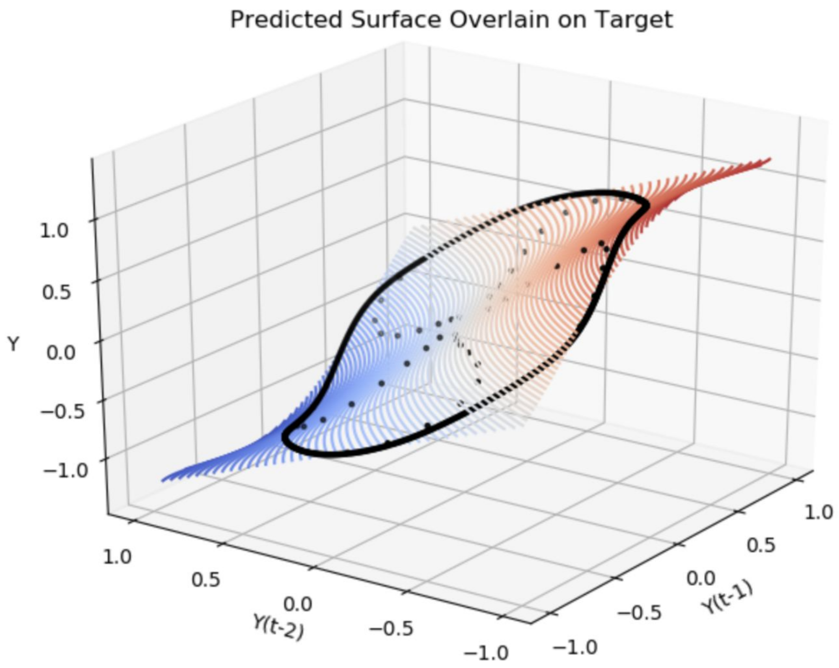
Volume can be a proxy for volatility to some degree.

Some ML



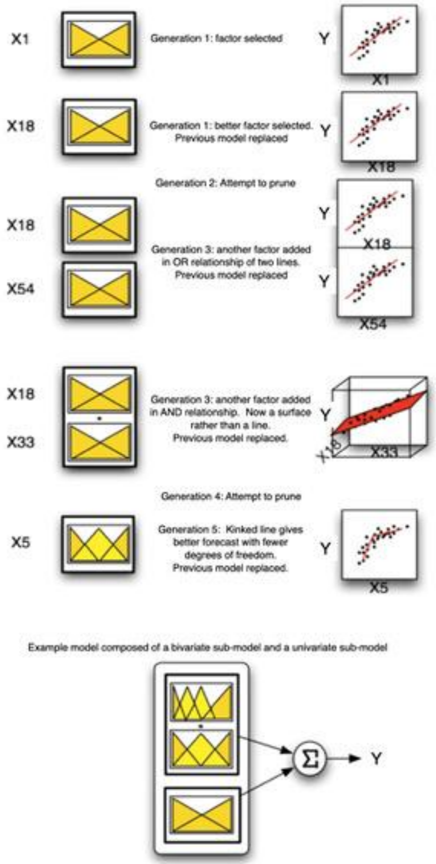
Generalized Additive Models

$$
\begin{aligned}
y(t) = {} & [0.8 - 0.5\exp(-y^2(t-1))]y(t-1) \\
& -[0.3 + 0.9\exp(-y^2(t-1))]y(t-2) \\
& +0.1\sin(\pi y(t-1))
\end{aligned}
$$



The red and blue surface is formed by 6 degrees of freedom

Building a Parsimonious Model



Model Induction

$$MDL = \underbrace{\log(1/N\sum_{i}^{N}(y_i - \hat{y}_i)^2)}_{\text{MSE}} + \underbrace{p * \log(N)/N}_{\text{Penalty for Complexity}}$$

$$\begin{aligned} &where: \\ &\quad p = degrees\ of\ freedom \\ &\quad N = number\ of\ samples \end{aligned}$$

There are many choices of Information Criteria (AIC, BIC, MDL, ...)

Portfolio of Strategies

Model induction brings with it a number of opportunities:

- Generating models in the presence of pre-existing models in order to minimize correlation
- Generating models with specific attributes in mind
- Matching the performance of competitor strategies in order to infer information

Generative modelling - augment design processes and create new products

What's a generative model?

- **A generative model** describes how a dataset is generated, in terms of a probabilistic model. By sampling from this model we are able to generate new data.
- Most of what you have seen so far probably falls under discriminative modelling. A key difference is that in discriminative modelling, each observation in your data set has a label. Example: build a binary classifier to detect Van Gogh paintings.
- Discriminative modelling estimates $p(y|\mathbf{x})$ - the probability of a label y given an observation $\mathbf{x}$.
- Generative modelling estimates $p(\mathbf{x})$ - the probability of observing observation $\mathbf{x}$ (generate a new painting that looks like a Van Gogh), or $p(\mathbf{x},y)$ if you have labels
- Another way to look at it is that discriminative models draw boundaries in the data space (boundaries between classes) and generative models model the distribution of classes

- A good generative model
  - Generates samples that appear to have been drawn from the original data distribution.
  - Generates samples that are sufficiently different from the observations in X.

Picture: *These people are not real – they were produced by our generator that allows control over different aspects of the image.*

https://github.com/NVlabs/stylegan

Hyper-realistic human faces

Text completion

SYSTEM PROMPT (HUMAN-WRITTEN)

A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

MODEL COMPLETION (MACHINE-WRITTEN, FIRST TRY)

The incident occurred on the downtown train line, which runs from Covington and Ashland stations.

In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.

“The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.”

The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials.

The Nuclear Regulatory Commission did not immediately release any information.

According to the release, the U.S. Department of Energy’s Office of Nuclear Material Safety and Security is leading that team’s investigation.

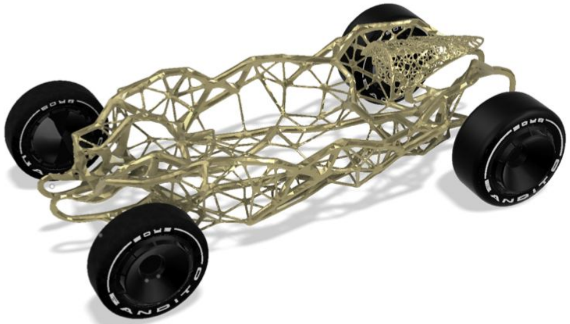
“The safety of people, the environment and the nation’s nuclear stockpile is our highest priority,” Hicks said. “We will get to the bottom of this and make no excuses.

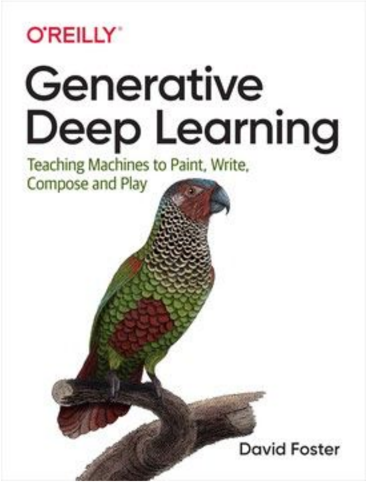
https://openai.com/blog/better-language-models/

Is artificial intelligence set to become art’s next medium?

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie’s becomes the first auction house to offer a work of art created by an algorithm

https://www.nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html





What about whiskey?

1.

DEFINE A TARGET

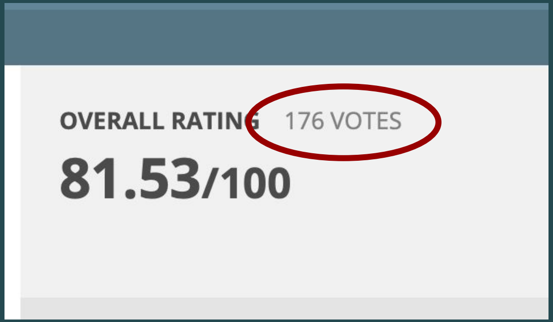
WHAT'S GOOD?

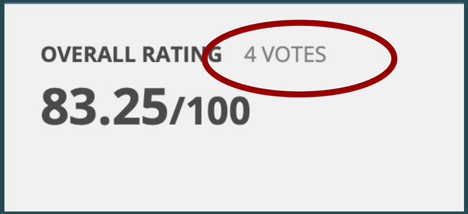
"Performance metric" uncertainty

Recipe A:

Customer review: 0.8, Expert review: 0.81, Mackmyra review: 0.82, Awards ratio: 0.80

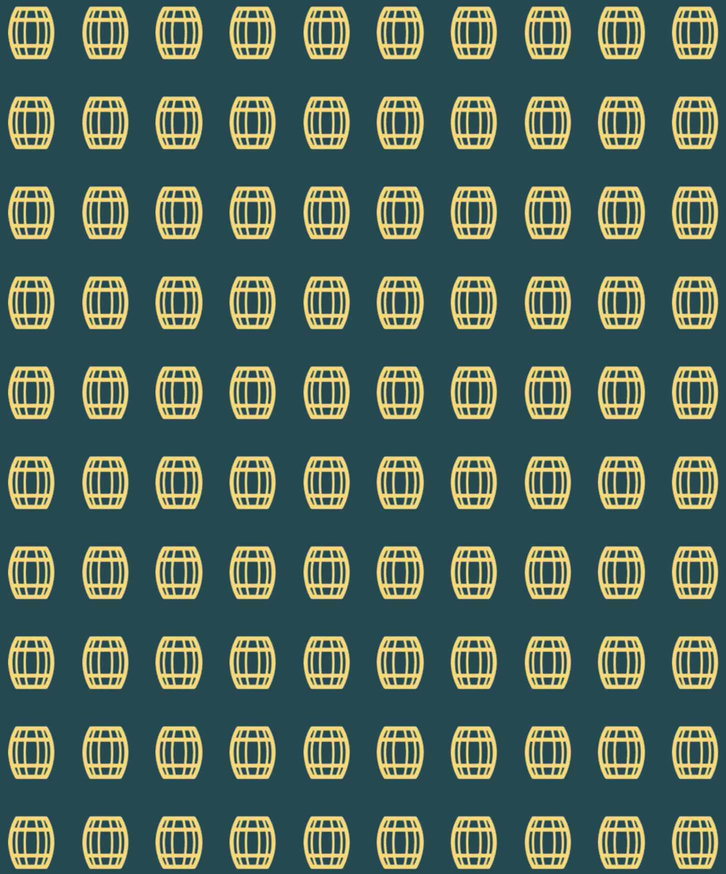
However: Experts rated 40 out of 75 recipes, customers 70 out of 75...



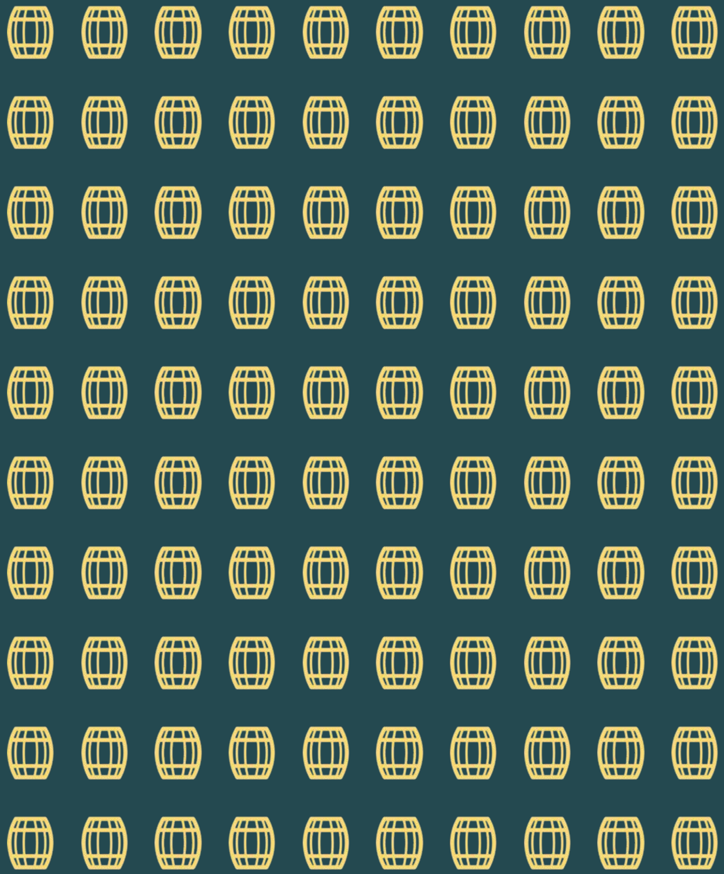


2.

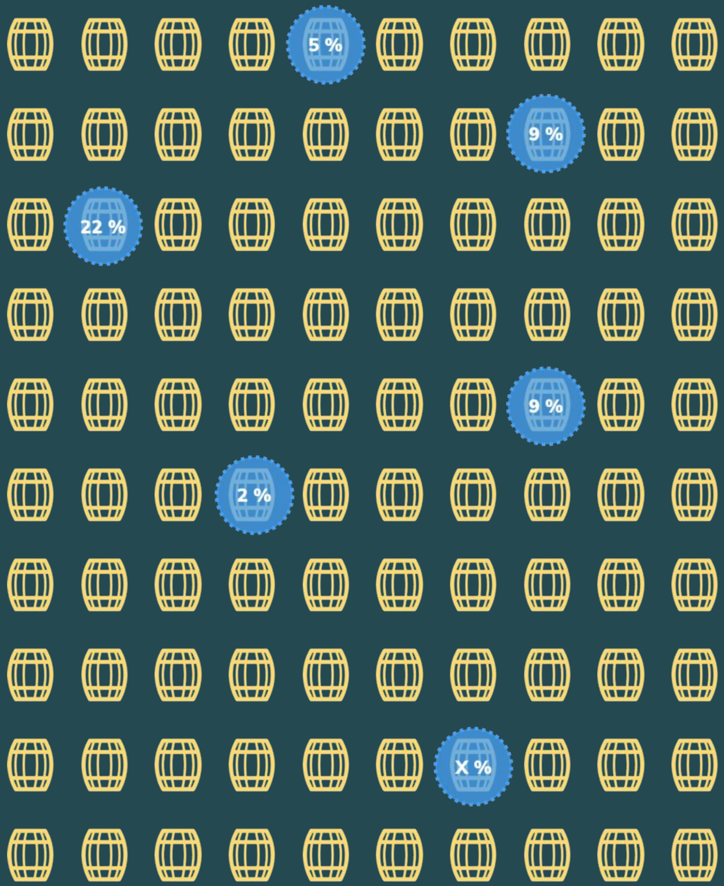
BUILD A GENERATIVE SYSTEM



Here are 100 casks



Your job is to pick 1-20 of them and choose how much you will use, in percentages, so that the recipe will sum to 100.



You realize you are a computer nerd
and need to solve this
without taste and in python.

Also: there's an infinite amount of
possibilities.

Generative adversarial network (GAN)

- GANs are a model architecture for training a generative model and consist of two adversarial parts
    - **A generator** - The generator creates samples that are intended to come from the same distribution as the training data. The generated instances become negative training examples for the discriminator.
    - **A discriminator** - The discriminator examines samples to determine whether they are real or fake. The discriminator learns using traditional supervised learning techniques, dividing inputs into two classes (real or fake). It penalizes the generator for producing implausible data.
- The generator is penalized for producing a sample that the discriminator network classifies as fake.
- The discriminator is penalized for misclassifying a real instance as fake or a fake instance as real.

https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
NIPS GAN tutorial - https://arxiv.org/pdf/1701.00160.pdf

Generative adversarial networks (GAN)

- The generator and the discriminator are trained in separate training processes. The generators aim is to fool the discriminator and it learns from its failures (discriminator feedback).

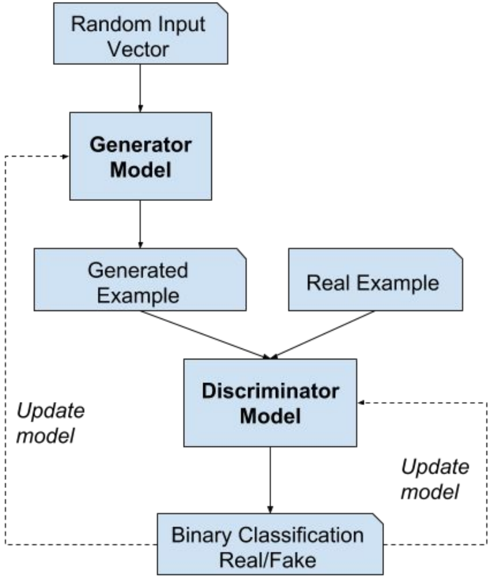
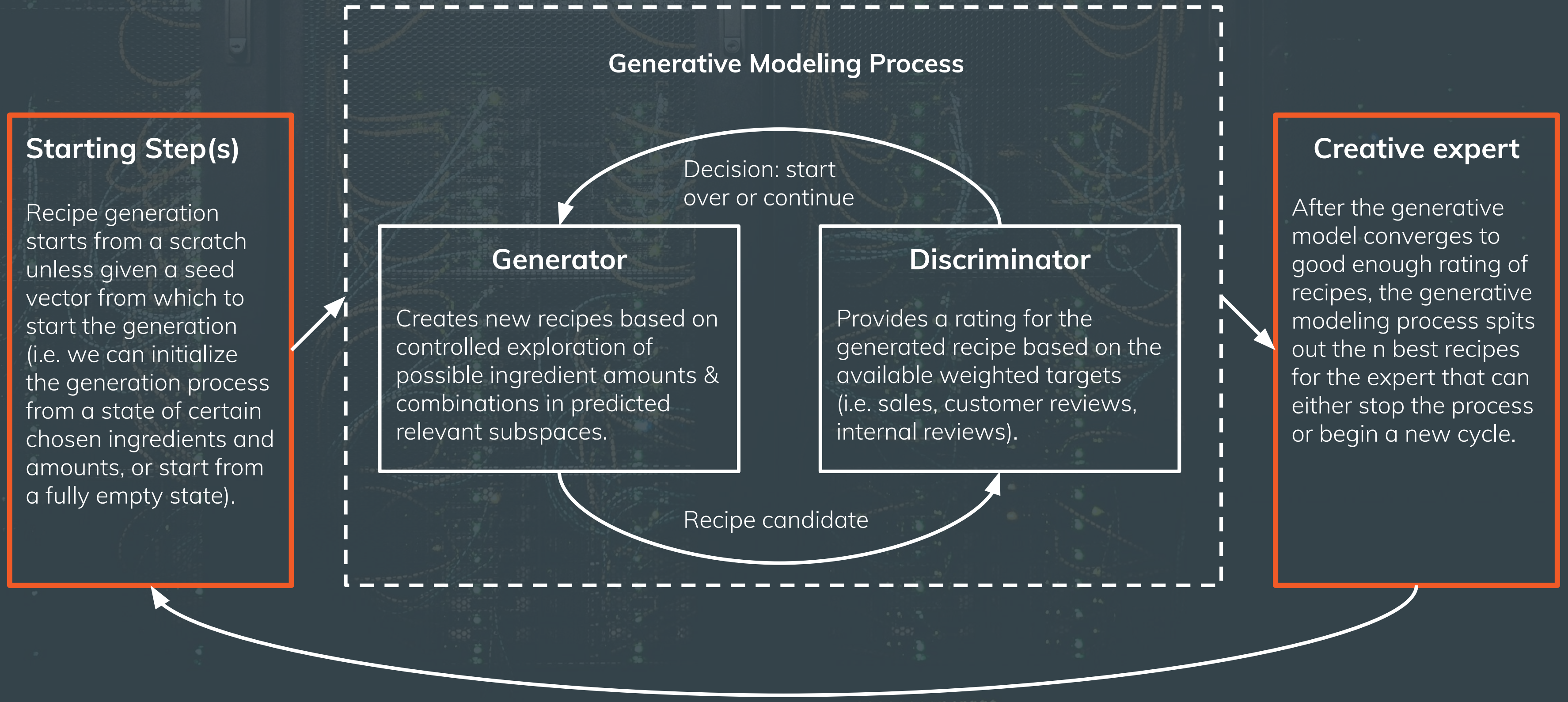


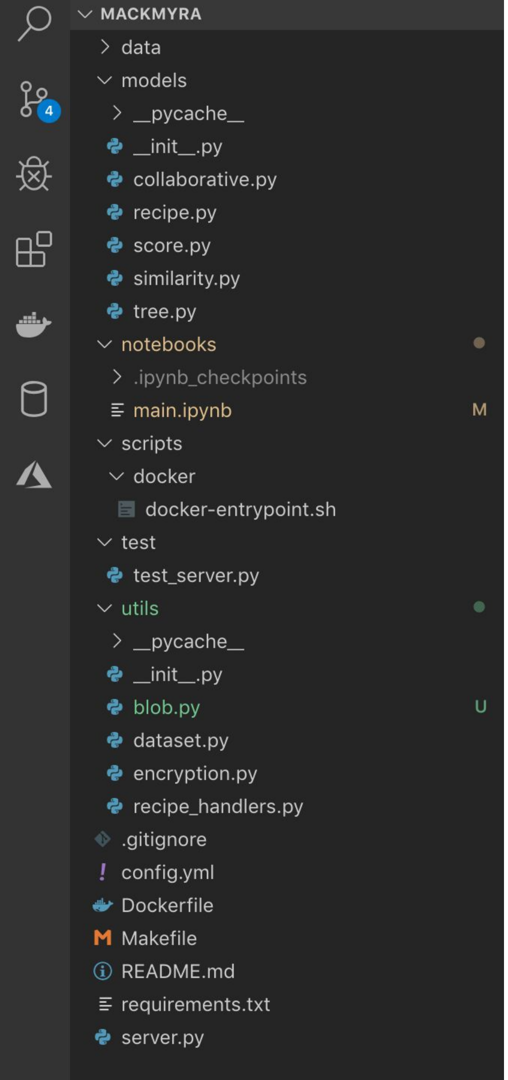
Diagram from - https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

3.

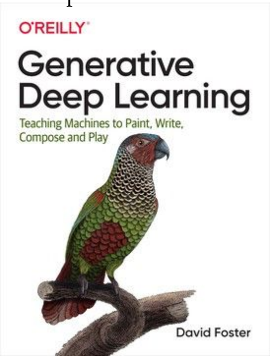
MAKE SURE IT WORKS
END TO END

Technical principle is based on two independent models: One is taught to generate new recipes and the other one is taught to determine whether they are good or not





Check for theory, examples and code



Ultimately we needed:

Decision trees and their traversed form

Singular value decomposition

Regression techniques for scoring

Different probability distribution sampling techniques & uncertainty handlers

Hyperparameter samplers (!)

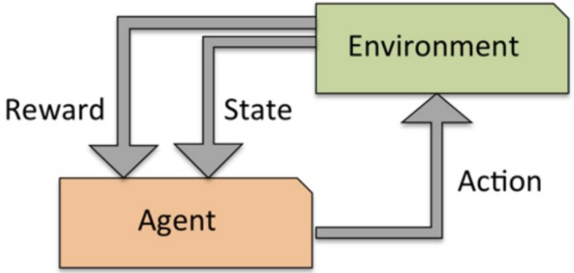
Lots of numpy/scipy custom glue-stuff

A custom generator-discriminator frame

Reinforcement learning use case - Dynamic pricing

Reinforcement learning

- Develop an agent that improves its performance based on interactions with the environment
    - Example: games like Chess, Go,...
- Search a (large) space of actions and states
- *Reward function* defines how well a (series of) actions works
- Learn a series of actions (policy) that maximizes reward through exploration



Reinforcement learning

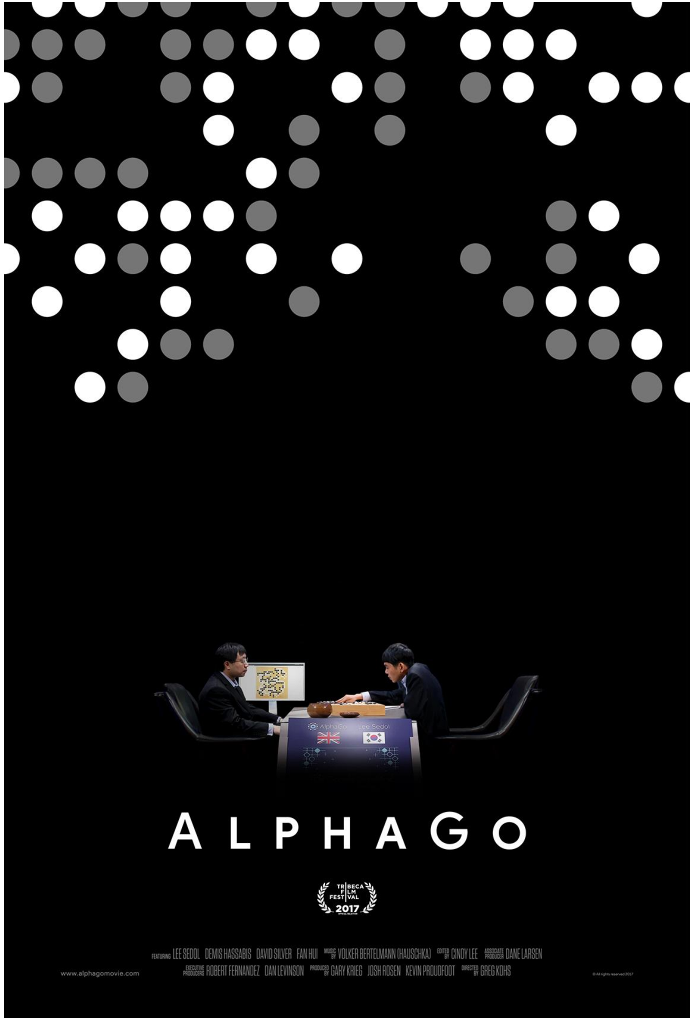
Some resources to get into RL

Presentation about reinforcement learning by a colleague of ours -
https://www.slideshare.net/MaxPagels/realworld-reinforcement-learning

Reinforcement learning literature:
http://incompleteideas.net/book/RLbook2018.pdf
https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf

David Silvers lectures at UCL
http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
https://www.youtube.com/watch?v=2pWv7GOvuf0&feature=youtu.be&list=PL5X3mDkKaJrL42i_jhE4N-p6E2Ol62Ofa

And his paper about mastering the game of Go in Nature
https://www.nature.com/articles/nature16961



Movie recommendation

What's dynamic pricing?

Dynamic pricing is the practise of dynamically adjusting the price of a product or service in order to increase a target metric (e.g. turnover) over time

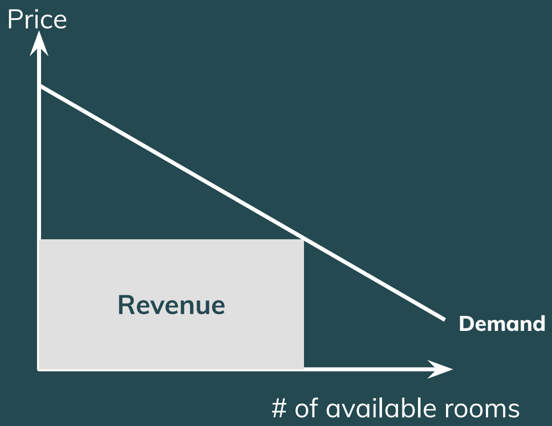
Dynamic pricing can be done:

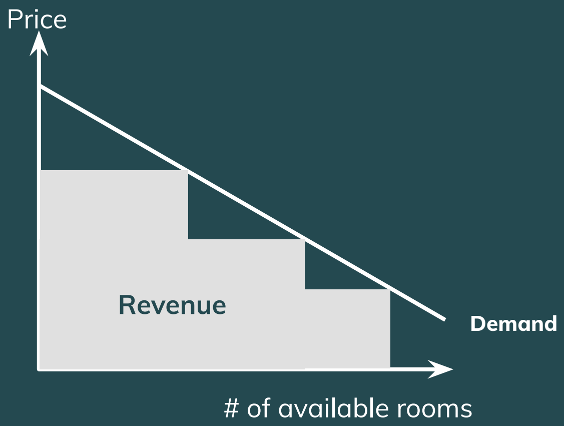
- On a global level: what is the globally optimal price for this movie ticket?
- On a global level, with side information: what is the globally optimal price for this movie ticket on Mondays at 9pm?
- On an individual level, with side information: what is the best price for this movie ticket, if it's 9pm on Monday and the customer is Bruce?

The upper bound on how well dynamic pricing works grows from 1) to 3).

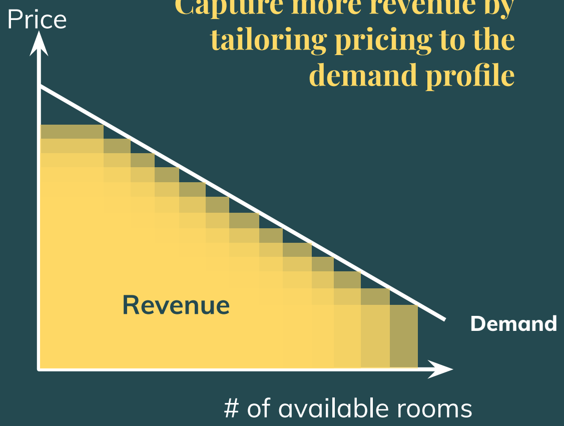
Dynamic Pricing for hotel rooms

Static Pricing vs Manually Managed Pricing vs Dynamic Pricing





Capture more revenue by tailoring pricing to the demand profile

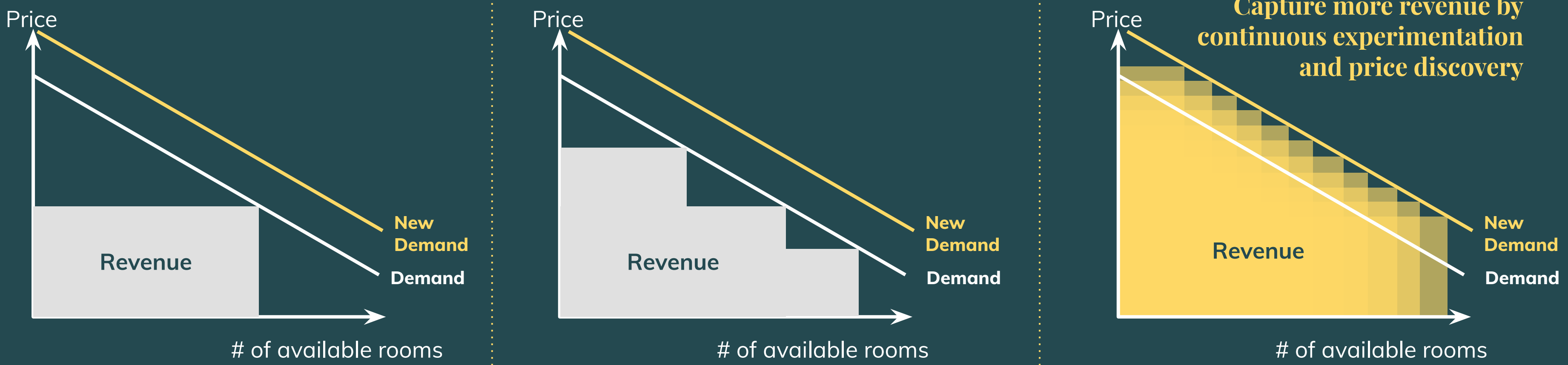


F & F

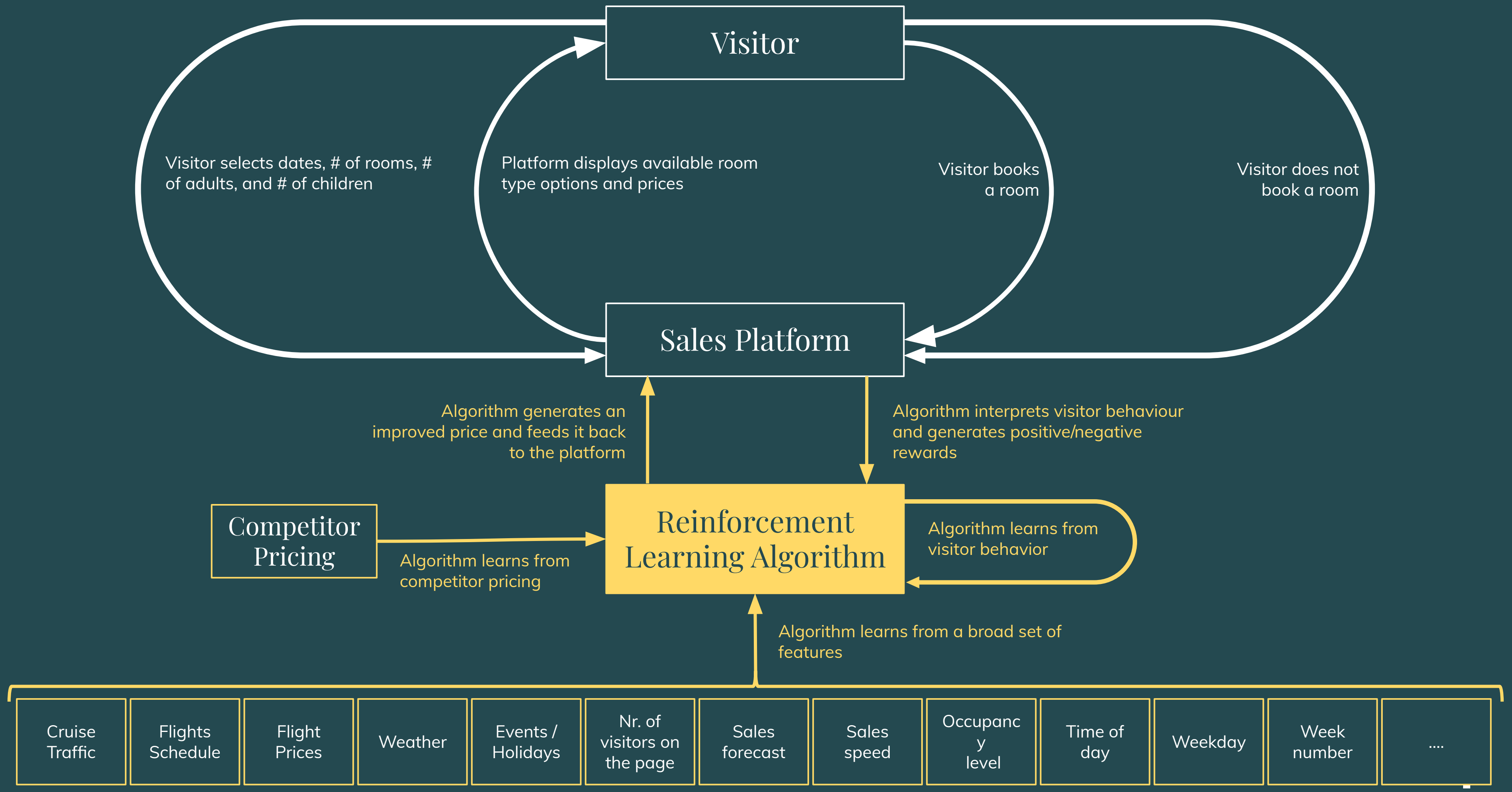
React to changes in the environment

The world is constantly changing, and so do customer demand profiles. Your solution needs to deal with this unexpected change effectively by experimenting with prices continuously and taking advantage of new opportunities in situations where a human would not have awareness of such change.

Static Pricing vs Manually Managed Pricing vs Dynamic Pricing



F & F


Visitor
Visitor selects dates, # of rooms, # of adults, and # of children
Platform displays available room type options and prices
Visitor books a room
Visitor does not book a room
Sales Platform
Algorithm generates an improved price and feeds it back to the platform
Algorithm interprets visitor behaviour and generates positive/negative rewards
Competitor Pricing
Reinforcement Learning Algorithm
Algorithm learns from visitor behavior
Algorithm learns from competitor pricing
Algorithm learns from a broad set of features
Cruise Traffic
Flights Schedule
Flight Prices
Weather
Events / Holidays
Nr. of visitors on the page
Sales forecast
Sales speed
Occupanc y level
Time of day
Weekday
Week number
....

Forenom

We helped Forenom significantly increase their revenue per available room through a reinforcement learning based, dynamic pricing model.

Short term rental apartment provider Forenom wanted to maximize revenue per available room and increase operational efficiency without sacrificing customer satisfaction. We created a self-learning and dynamic pricing algorithm that both cuts down the planning time and optimizes pricing: for Forenom and the customer.

13%
increase in revenue per available room

Growth
Enables Forenom to grow their business faster

Automation
Saves work hours through efficient pricing

Business advisory ◇ Reinforcement learning ◇ Pricing strategy ◇ Production deployment ◇ Cloud architecture

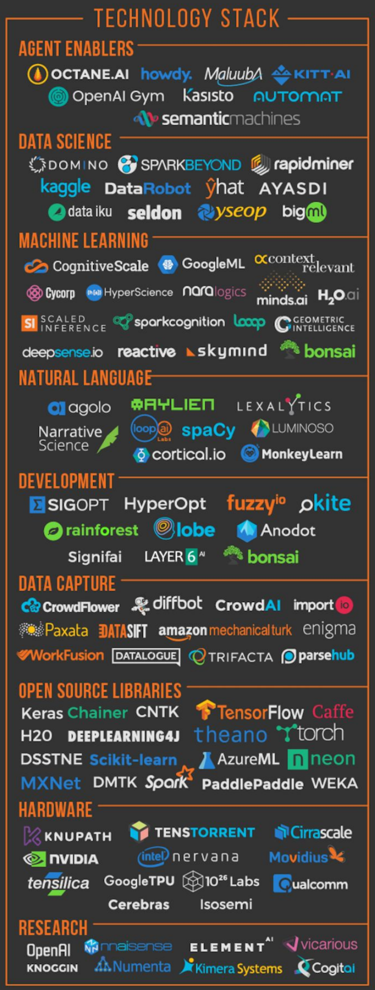
fourkind.com/work/forenom-pricing/

Thank you! Questions?

A special thanks to Max Pagels, Jarno Kartela & Jan Hiekkaranta,
for letting us borrow some slides from their talks.

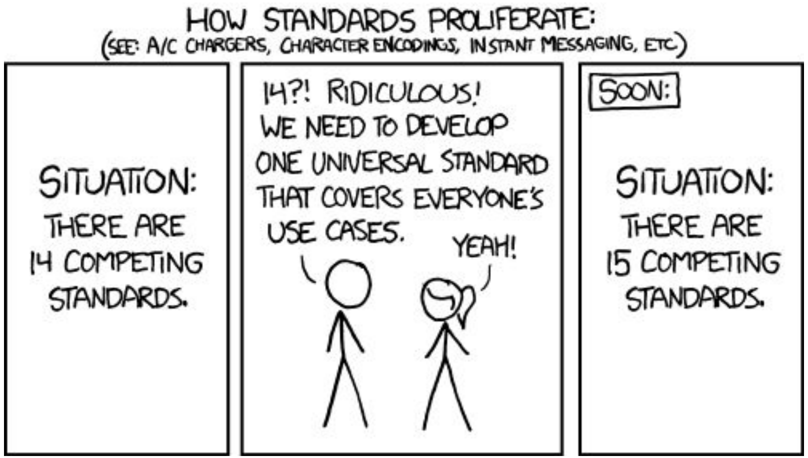
nemanja.cvijanovic@fourkind.com
bruce.ferguson@fourkind.com
www.fourkind.com

The tool & service ecosystem for machine learning is fragmented, non-standardised, and fragile



The tool & service ecosystem for machine learning is fragmented, non-standardised, and fragile

Current status of model exchange formats



The tool & service ecosystem for machine learning is fragmented, non-standardised, and fragile

Sometimes, Data Scientists make good models using learning algorithms they don’t fully understand

Me: “neural networks learn through backpropagation, which adjusts weights based on the chain rule and the partial derivative of the loss function with respect to the weights in each layer. Initialisation must however be symmetry-breaking…”

Me: “gradient boosted trees learn using a set of weak learners”

Me “Random Forests are made up of trees”

Me: “what’s an SVM?”

Sometimes, Data Scientists make good models using learning algorithms they don’t fully understand

Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn’t be

"The license for this platform cost us 1.2 M€, so it should be our primary platform going forward."

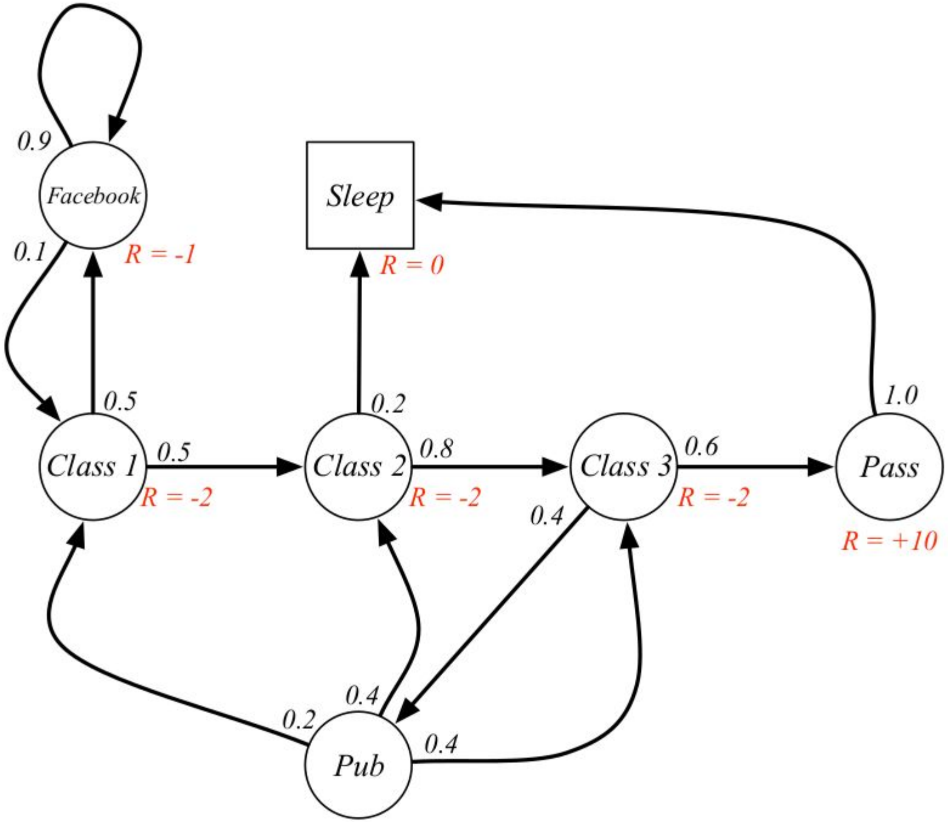
Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn't be

Internal thinking: “developing this model & A/B test took 4 months, so we’re definitely taking it to production”

Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn’t be

In 1968 Knox and Inkster,[2] in what is perhaps the classic sunk cost experiment, approached 141 horse bettors: 72 of the people had just finished placing a $2.00 bet within the past 30 seconds, and 69 people were about to place a $2.00 bet in the next 30 seconds. Their hypothesis was that people who had just committed themselves to a course of action (betting $2.00) would reduce post-decision dissonance by believing more strongly than ever that they had picked a winner. Knox and Inkster asked the bettors to rate their horse's chances of winning on a 7-point scale. What they found was that people who were about to place a bet rated the chance that their horse would win at an average of 3.48 which corresponded to a "fair chance of winning" whereas people who had just finished betting gave an average rating of 4.81 which corresponded to a "good chance of winning".

Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn't be



Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn’t be

Though from a different domain, adapting the **Markov property** is a good rule of thumb.

"The future should be independent of the past given the present"

Sunk costs are almost always taken into account when productionising machine learning projects, but they shouldn't be