Reinforcement learning in Finance

Playing Atari vs Playing Markets

Alex Honchar, ODSC Europe 2021

Github -> Rachnog -> RL in Finance -> ODSC

About me

- ex-independent AI consultant | startups, patient monitoring | risk management

- UK consulting firm co-founder and ML director | Neurons Lab

- lecturer | European conferences and universities | Medium 1M+ views

Plan for today

From 0 to 1 © in financial RL

- ML in Finance: quick recap on why we need it
- RL 101: environment and agents on cosine function
- RL 101-2: OpenAI's gym and market data
- RL 101-3: Do you trust it? What could go wrong? Everything!
    - Measuring financial metrics
    - Measuring probabilistic interpretation
    - Measuring overfitting probability
    - Measuring strategy and multiple testing risks
- Now, when we trust the pipeline, it's time for the cool stuff :)

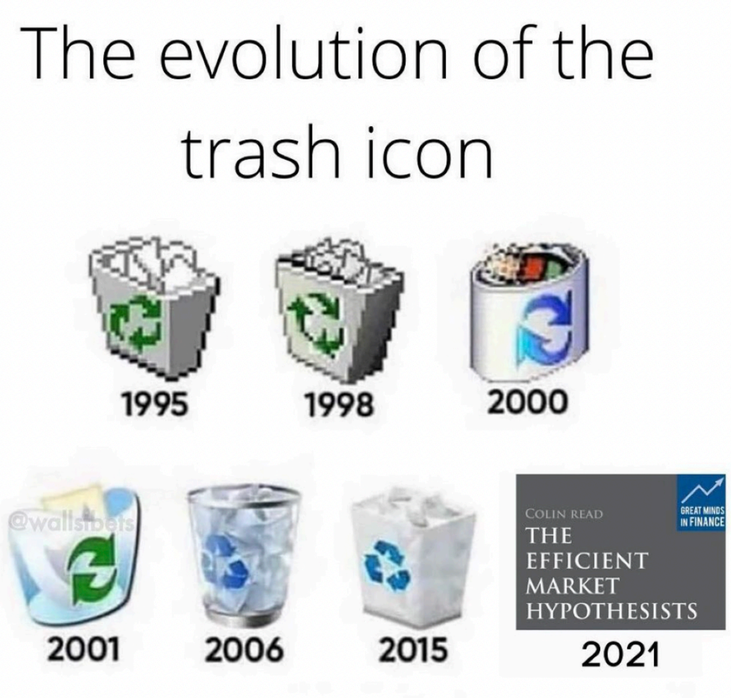
Not a plan for today

Getting rich in financial RL

- Not cool deep learning architectures
- Not state-of-the-art RL algorithms
- Not huge HFT datasets
- Not making easy money while chillin 😎

ML in Finance

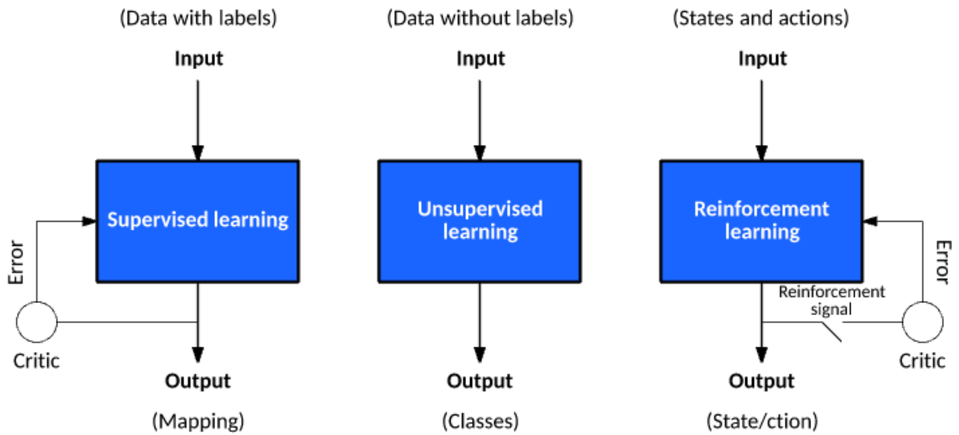
- Relationships in finance are non-linear / threshold / hierarchical
- High dimensionality, low number of samples, no structure
- Mathematical models are designed only to explain in-sample patterns



	Playing Atari	Playing Markets
Environment	Created by human game designers, there are limits and clear rules	Constantly evolving, changing themselves and rules as well
States	Coming from the game only	Can come outside of the game
Signals	Clear and easy to interpret, high signal-to-noise ratio	Can have various interpretations in different times, low signal-to-noise ratio
Rewards	To win regardless the reward scheme provided	Depending on the investor needs and risk profile
Impact	Your actions lead to next states, but don't change the game	Your actions also change the environment itself

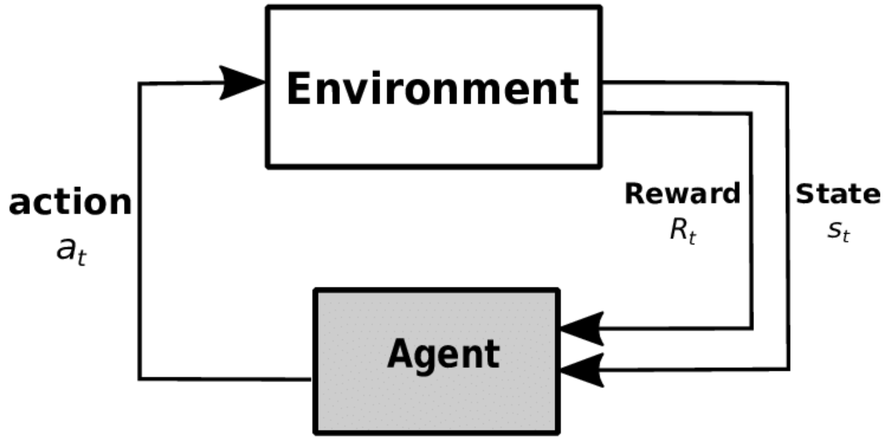
FINANCIAL ML ≠ ML ALGORITHMS + FINANCIAL DATA

RL vs *L



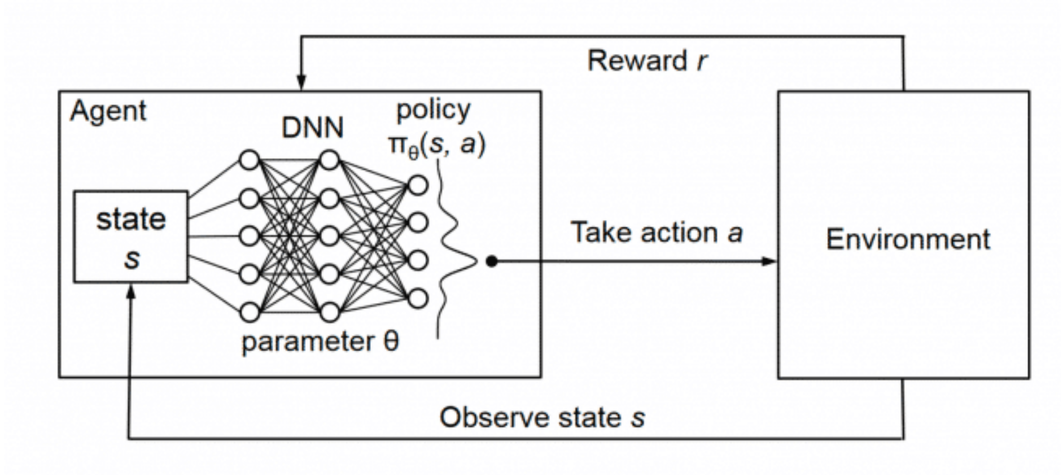
https://docs.paperspace.com/machine-learning/wiki/supervised-unsupervised-and-reinforcement-learning

RL



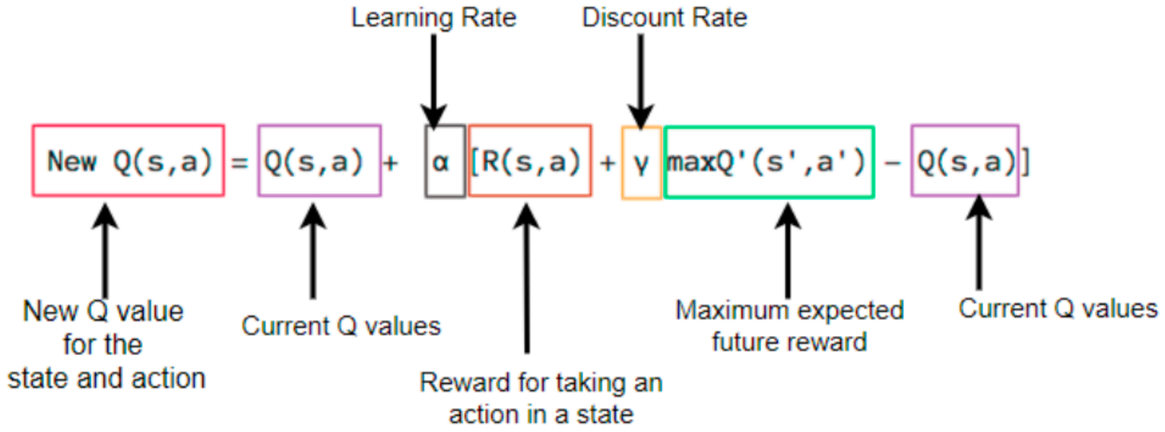
https://www.researchgate.net/figure/Reinforcement-Learning-Agent-and-Environment_fig2_323867253

Q Learning



https://www.novatec-gmbh.de/en/blog/deep-q-networks/

Learning Q



https://www.mygreatlearning.com/blog/simplified-reinforcement-learning-q-learning/

Playground #1: shorturl.at/bpBIR

Problems spotted: short-term, simple data, no CV

Playground #2: shorturl.at/tvCQ5

Problems spotted: “inverse optimal behavior”, random seed? Overfitting

Financial metrics

- Better than benchmark = **Information ratio**
- Return + risk = Risk-adjusted return (**Sharpe ratio**)
- Skewness + kurtosis adjustment = **Probabilistic Sharpe ratio**

$$IR_A = \frac{\overline{R}_A - \overline{R}_B}{\sigma_{A-B}}$$

Information ratio

$$SR = \frac{\mu}{\sigma}$$

Sharpe ratio

$$\widehat{PSR}(SR^*) = Z\left[\frac{\left(\widehat{SR}-SR^*\right)}{\hat{\sigma}(\widehat{SR})}\right] = Z\left[\frac{\left(\widehat{SR}-SR^*\right)\sqrt{n-1}}{\sqrt{1+\frac{1}{2}\widehat{SR}^2-\gamma_3\widehat{SR}+\frac{\gamma_4-3}{4}\widehat{SR}^2}}\right]$$

Probabilistic Sharpe ratio

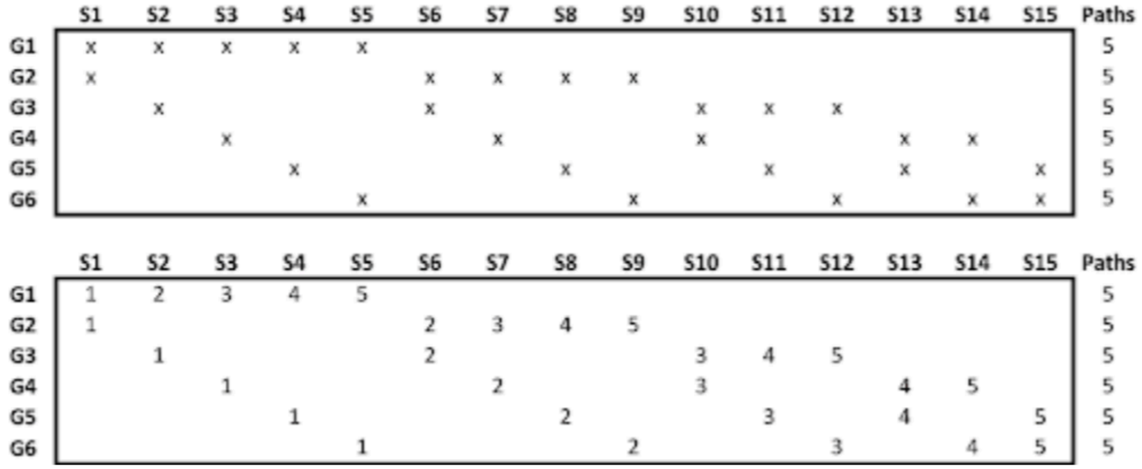
$$(\widehat{SR}-SR) \xrightarrow{a} N\left(0, \frac{1+\frac{1}{2}SR^2-\gamma_3 SR+\frac{\gamma_4-3}{4}SR^2}{n-1}\right)$$

Martens standard deviation estimate

Playground #3: shorturl.at/ijxH6

Probabilistic metrics

- A single test set is just a single realization of a stochastic process
- We could do a lot of simulations, if we knew for sure the nature of the process, but we don’t!
- We can re-shuffle our dataset combinatorially - same nature, different train and test pairs



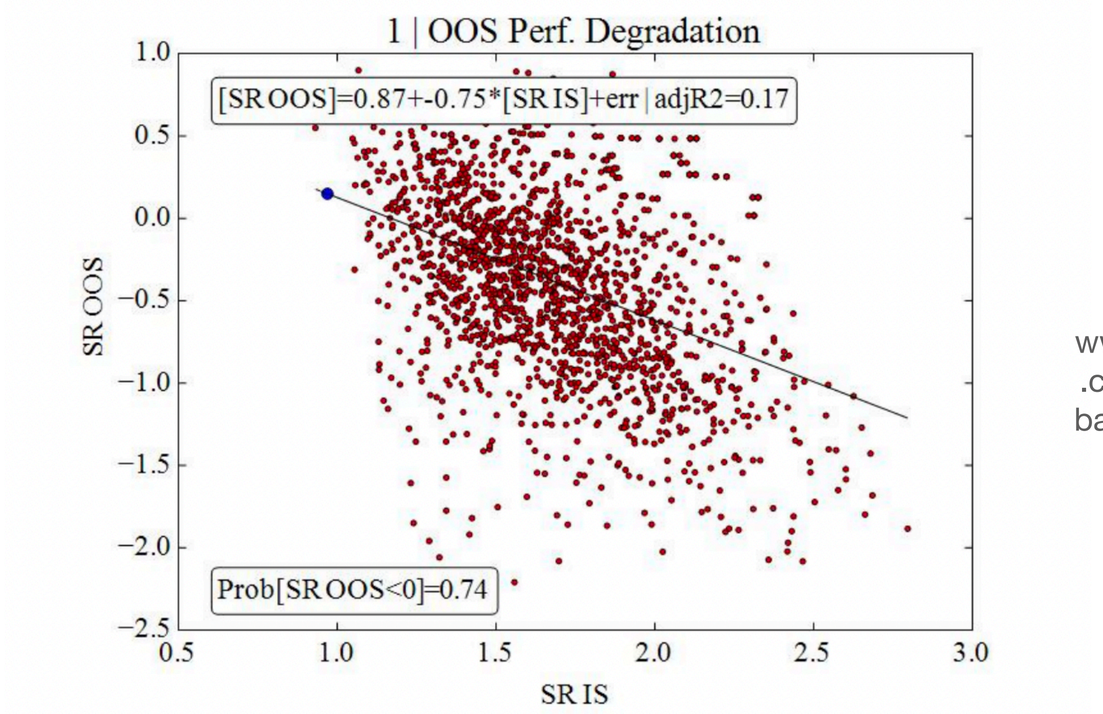
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	Paths
G1	x	x	x	x	x											5
G2	x					x	x	x	x							5
G3		x				x				x	x	x				5
G4			x				x			x			x	x		5
G5				x				x			x		x		x	5
G6					x				x			x		x	x	5

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	Paths
G1	1	2	3	4	5											5
G2	1					2	3	4	5							5
G3		1				2				3	4	5				5
G4			1				2			3			4	5		5
G5				1				2			3		4		5	5
G6					1				2			3		4	5	5

https://blog.quantinsti.com/cross-validation-embargo-purging-combinatorial/

Overfitting probability

- The better we can model in-sample data, the better should be out-of-sample performance, if not - we just overfit the data
- We can study this effect combinatorially as well



https://www.davidhbailey.com/dhbpapers/backtest-prob.pdf

Probability of failure

- Sharpe ratio depends on the our precision (probability to take profit), expected returns, number of bets per year
- We can inverse this formula:
  what precision do we need to get target Sharpe given other params?
- The probability to fall below this precision and wipe out our returns - this is a strategy risk, i.e. probability of failure

$$\theta(p, n, \pi_-, \pi_+) = \frac{nE[X_i]}{\sqrt{nV[X_i]}} = \frac{(\pi_+ - \pi_-)p + \pi_-}{(\pi_+ - \pi_-)\sqrt{p(1-p)}}\sqrt{n}$$

Theta — Sharpe Ratio; n — number of bets per year; pi+ and pi- are take-profit and stop-loss targets, i.e. our returns; p — the probability of taking profit

$$p = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$
$$a = (n+\theta)^2(\pi_+ - \pi_-^2)$$
$$b = [2n\pi_- - \theta^2(\pi_+ - \pi_-)](\pi_+ - \pi_-)$$
$$c = n\pi_-^2$$

What probability to get a positive return (i.e. precision of our classifier) do we need to get target Sharpe ratio theta and doing n bets per annum?

Multiple testing

- The Sharpe ratio is computed as a function of mean and standard deviation of returns
- Deflated Sharpe Ratio (DSR) uses additionally skewness, kurtosis, variance of population of Sharpe ratios, number of independent trials
- DSR can tell us if the “best” obtained strategy is indeed the best or it’s just a spurious correlation

$$\widehat{DSR} \equiv \widehat{PSR}\left(\widehat{SR}_0\right) = Z\left[\frac{\left(\widehat{SR} - \widehat{SR}_0\right)\sqrt{T-1}}{\sqrt{1 - \hat{\gamma}_3\widehat{SR} + \frac{\hat{\gamma}_4 - 1}{4}\widehat{SR}^2}}\right]$$

https://www.nomura.com/events/9th-annual-global-quantitative-investment-strategies-conference/resources/upload/10_00_Marcos_Lopez_de_Prado_20150510.pdf

$$\widehat{SR}_0 = \sqrt{V\left[\widehat{SR}_n\right]}\left((1-\gamma)Z^{-1}\left[1 - \frac{1}{N}\right] + \gamma Z^{-1}\left[1 - \frac{1}{N}e^{-1}\right]\right)$$

Expected maximum Sharpe ratio, where γ is the Euler-Mascheroni constant (approx. 0.5772), Z is the CDF of the Standard Normal and e is Euler’s number

A note on the random seeds

- RL suffers from reproducibility and we need to run multiple random seeds to see the whole spectrum of potential performances
- DFR and other multiple testing tools can help to estimate if the parameters of the algorithm are robust enough
- Welch test can help us to understand if one algorithm is better than the other based on the multiple runs of both

$$\overline{x} \hat{=} \sum_{i=1}^{n} x^i, \qquad s \hat{=} \sqrt{\frac{\sum_{i+1}^{N} (x^i - \overline{x})^2}{N-1}},$$

https://openlab-flowers.inria.fr/t/how-many-random-seeds-should-i-use-statistical-power-analysis-in-deep-reinforcement-learning-experiments/457

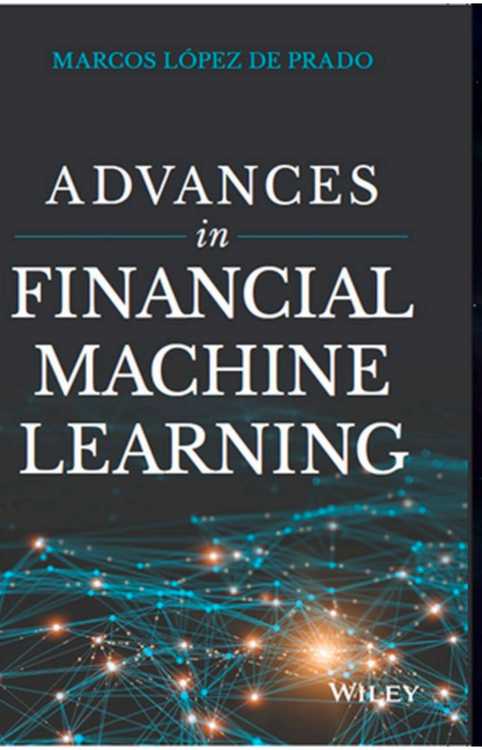
- $H_0 : \mu_{\text{diff}} = 0$
- $H_a : \mu_{\text{diff}} \neq 0$

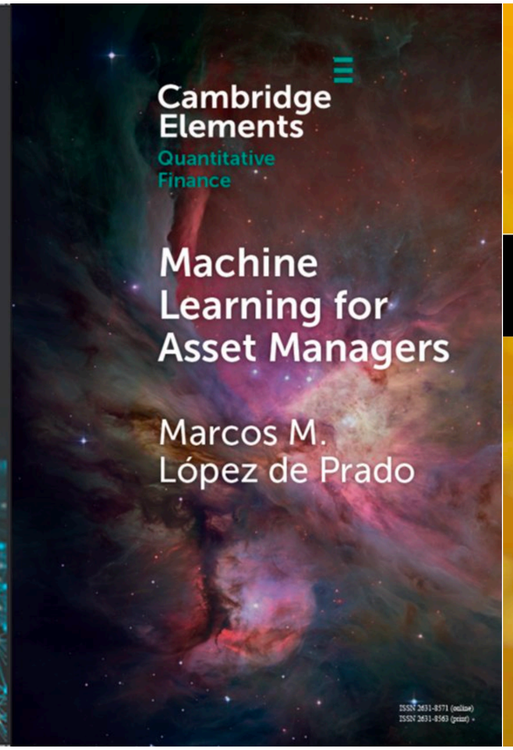
$$t = \frac{x_{\text{diff}}}{\sqrt{\frac{s_1^2 + s_2^2}{N}}}, \qquad \nu \approx \frac{(N-1) \cdot \left(s_1^2 + s_2^2\right)^2}{s_1^4 + s_2^4},$$

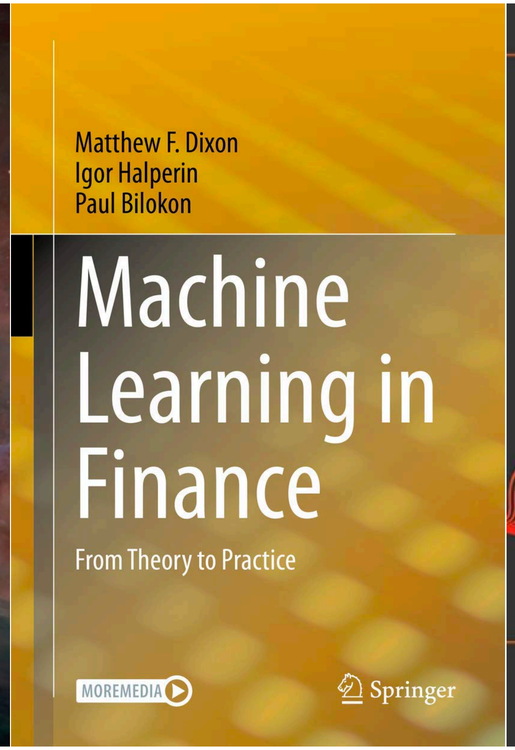
$$p\text{-value} = P(X_{\text{diff}} \geq \overline{x}_{\text{diff}} \mid H_0),$$

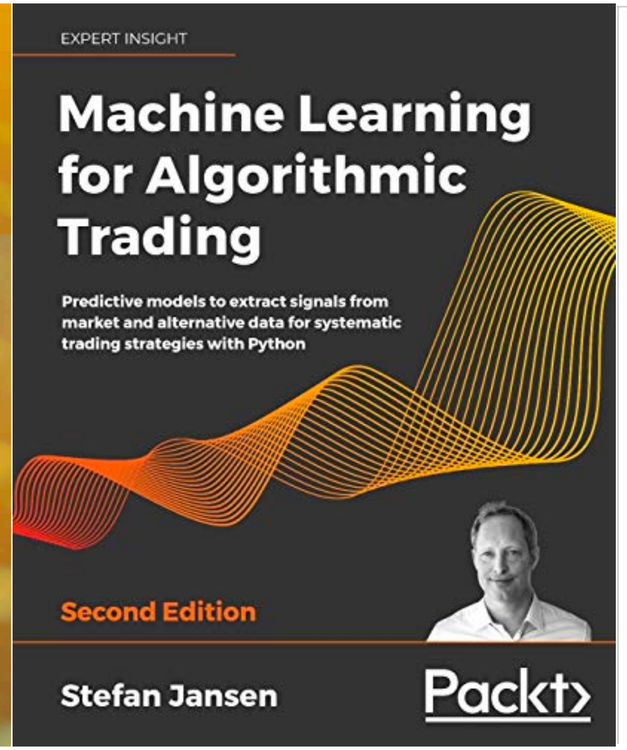
Next steps

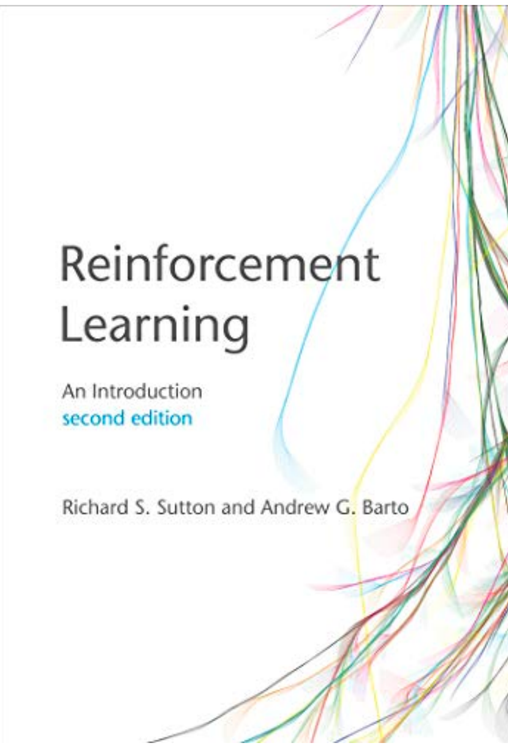
Education and practice resources











Medium: @alexrachnog
Github: @Rachnog
Linkedin: Alexandr Honchar