

# **Financial data structures**

or why machine learning in finance is nothing like Kaggle competitions /  
research papers / industrial best practices

# About speaker



## Entrepreneur

creating and selling AI solutions worldwide as Chief AI Officer and Co-founder at Neurons Lab



## Practitioner

7 years building data-driven products, top-10% at the hardest data science contest Numerai



## Educator

Medium with >1M views, scientific articles >150 citations, taught at UNIVR, UCU, KPI

# About you...?

let's get to knowing each other

# What we are not going to discuss

- Technical analysis with patterns, golden ratios, fractals and similar things. Leave it for amateurs :)
- Fundamental analysis with reading reports, book-to-price ratios, assets and liabilities analysis
- Complex financial mathematics, advanced probability and stochastic processes theory (although I highly recommend to follow-up after this course exactly with stochastic calculus)
- Deep learning for price prediction and getting rich ;)

# Day 1

Safe path to save money for your retirement days

- Algorithmic investments: roles and goals
- Financial data structures: from retail to alternative data
- Criteria of success in algorithmic investment
- Classical approaches to algorithmic investment [PRACTICE]
- “Learning”-like approaches to algorithmic investment [PRACTICE]

# Day 2

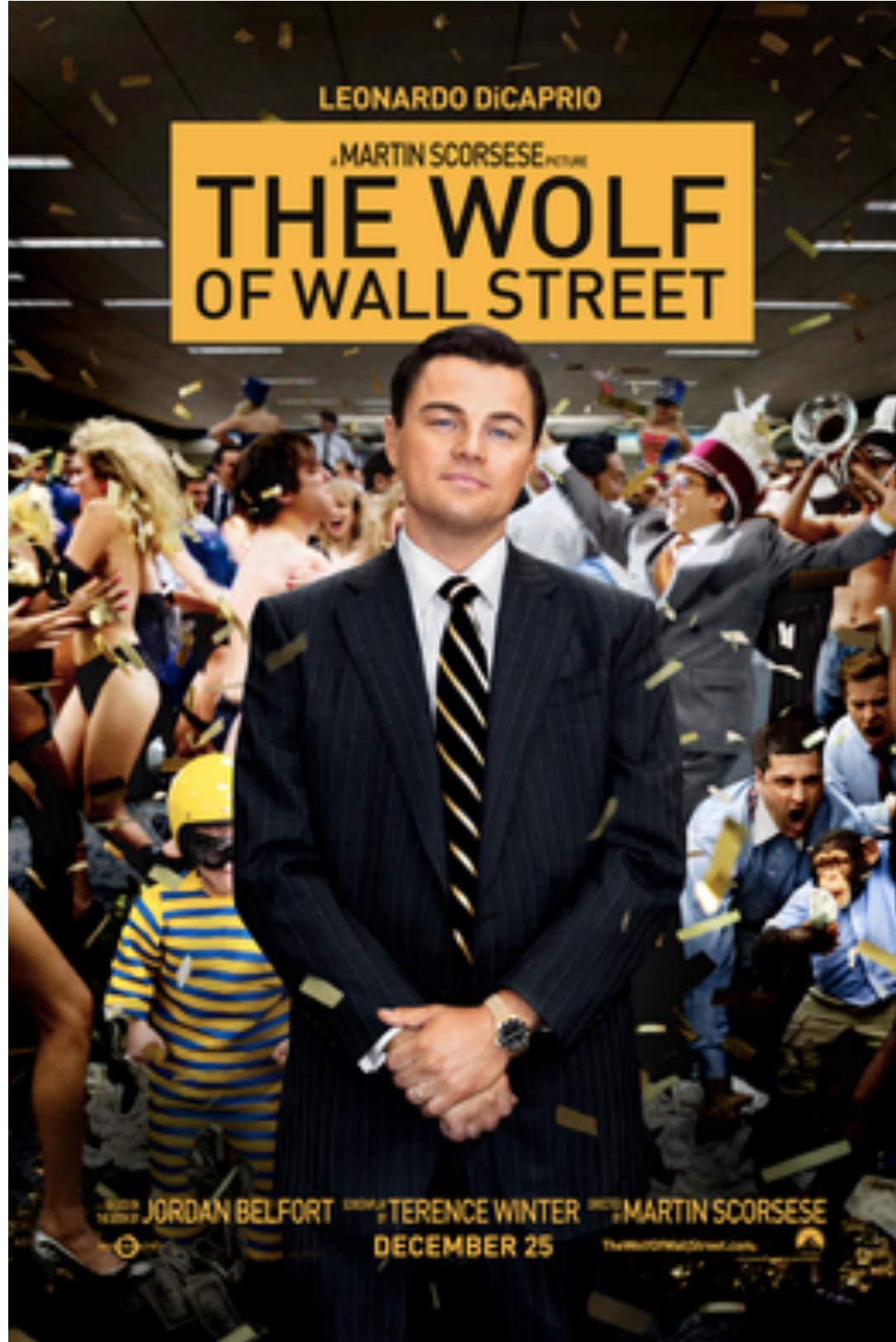
Hazardous path of speculations and wild guesses with ML

- Active investment management: what is that alpha?
- Why financial machine learning is different from regular one?
- Predicting asset prices in Python in a Kaggle way [PRACTICE]
- Fixing inputs and outputs with respect to financial logic
- Predicting asset prices in Python as you never did before [PRACTICE]
- Why theoretical physicists success in this job and what to do next?

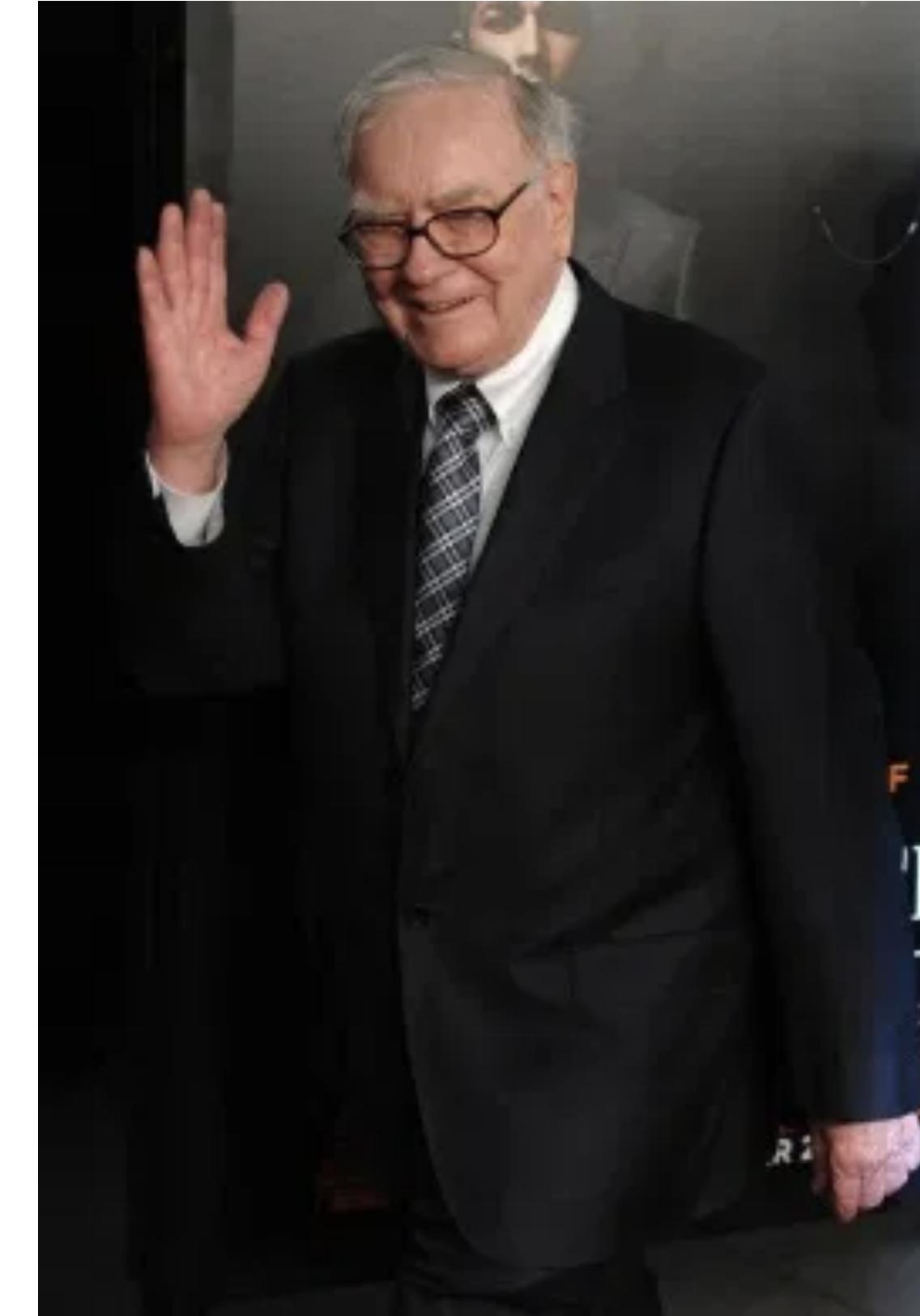
# **Roles and Goals**



Bobby Axelrod,  
“Billions”, HBO



Jordan Belfort,  
“Wolf of Wall Street”



Warren Buffet,  
Berkshire Hathaway



Ray Dalio,  
Bridgewater

# Jobs Types

<http://isomorphisms.sdf.org/maxdama.pdf>

	Front Office	Back Office
Buy Side	<b>Asset management</b> at a big bank. <b>Hedge fund</b> (strategies constrained to prospectus) <b>Prop trading</b> (fastest moving) <i>Matlab, Java, Functional Languages.</i> 70-100k+large bonus	<b>Data scraping and maintenance, Execution, Server administration</b> <i>Bash, SQL, SVN, Linux, C++.</i> 90-100k+small bonus
Sell Side	<b>Sales &amp; Trading</b> at a big bank (taking & executing orders, creating derivatives by client reques, execution algos) <i>Excel.</i> 70-80k+medium bonus	<b>Technology, Operations, or Risk Management</b> at a big bank (hard to transition to front office) <i>C++, internal language, hacky code.</i> 90-100k+small bonus

# Applied directions

<b>Banking</b>	<b>Asset Management</b>	<b>Trading</b>
Retail operations	Representation learning	Price forecasting
P2P operations	Portfolio optimization	Optimal execution
Lending, credit scoring	Risk management	Optimal strategy development

# Algorithmic investments dream team

From the idea to the execution

- Analysts
- Data curators
- Feature analysts
- Strategists
- Backtesters
- Deployment team
- Portfolio managers

# 10 ML applications in Finance

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3257415](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3257415)

# **Finance after COVID-19**

<https://ssrn.com/abstract=3562025>

# Financial data structures

# Inputs

What we are basing our ideas on?

- Banking data: clients information, transactions...
- Fundamental data: assets, liabilities, sales, earnings...
- Market data: price, volume, dividends, open interest...
- Analytics data: recommendations, credit rankings, news sentiment...
- Alternative data: CCTV, satellite images, Amazon, Twitter...

# Outputs

Problems we're trying to solve

## Regression:

- earnings prediction
- credit loss forecasting
- price forecasting

## Classification:

- rating prediction
- default modeling
- credit card fraud
- anti-money laundering

## Clustering:

- customer segmentation
- stock segmentation

## Representation learning:

- factor modeling
- denoising
- regime change detection

# **What do we optimize for?**

focusing on trading / investment management

# What we optimize for in normal ML?

- “**Accuracy**”-like surrogate on OOS data  
accuracy of prediction? “understanding the market?” do I really want to be exactly accurate in random conditions?
- **Mathematical loss function** on OOS data  
logloss? MSE? of what? for what? how do I measure something in the future?
- **Business metrics**
  - returns on investment? some risk? maximal potential loss? diversification?

# One step back to the process

Portfolio selection for the retirement

- **Select desired universe of financial instruments**  
it should grow  
it should be diversified  
it should be “calm”, predictable
- **Select desired factor exposure**  
they should explain the market now and in the future
- **Select preferred allocation scheme**  
“not bet everything on one horse”
- **Select final investment goal and metric**  
“maximize returns but minimize risks”
- **“Optimize” to have that all in the real future, not just in the Jupyter Notebook**

# Growth and security

We want to grow profits but feel safe same time

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Returns

$$\sigma^2 = w^T \sum w$$

Variance

# Explainability by the factors

We want to predict growth based on something useful

$$R_i = R_f + \beta_i * (R_m - R_f)$$

Capital Asset Pricing Model

$$r = R_f + \beta_3(K_m - R_f) + b_s \cdot SMB + b_v \cdot HML + \alpha$$

Fama-French Model

# Allocation schemes and diversification

Don't put all eggs in a single basket

- Cap-based
- Equal allocation
- Risk diversification

# Final metrics

How do we value a constructed portfolio?

$$SharpeRatio = \frac{E[Returns] - RiskFreeRate}{\sqrt{V[Returns]}} * \sqrt{n}$$

Normally, some form of risk-adjusted returns. What will be the risk?

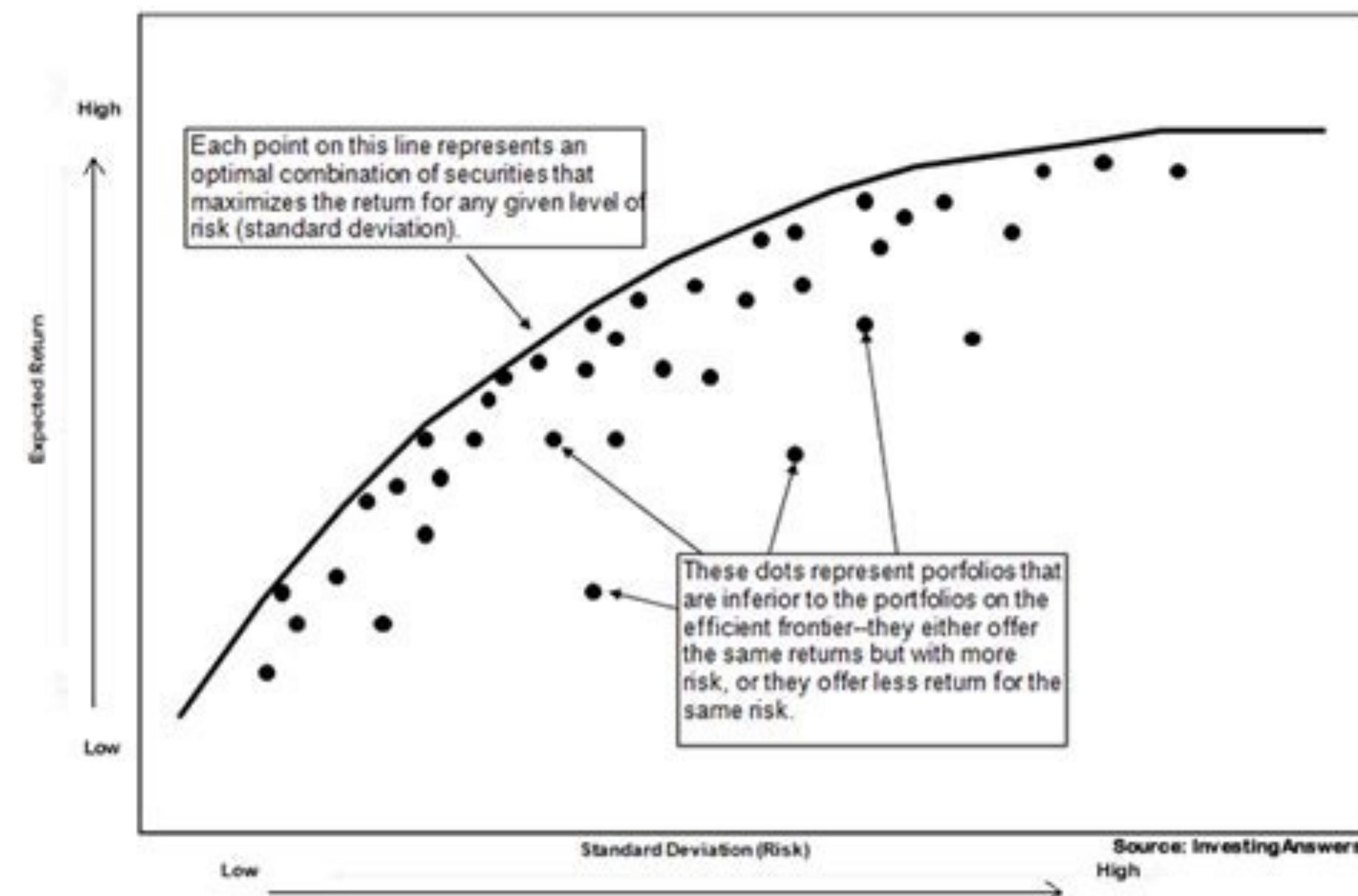
# **Classical approaches**

directly optimizing for the metrics based on the data

# Markowitz portfolio

The only free lunch in Finance

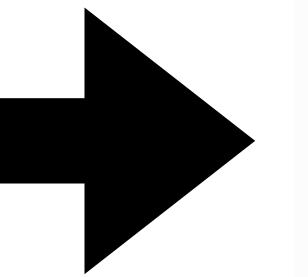
- De-correlated assets combined together in a portfolio get lower volatility



# Markowitz portfolio

Mathematical optimization formulation

$$\begin{array}{ll}\max_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & g(\mathbf{x}) \leq 0 \\ & h(\mathbf{x}) = 0\end{array}$$

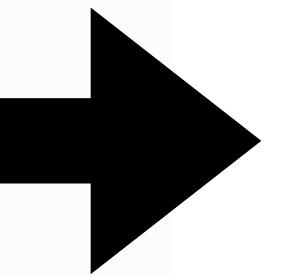


$$\begin{array}{ll}\min_{\mathbf{w}} & \mathbf{w}^T \sum \mathbf{w} \\ \text{s.t.} & \sum_{i=1}^n w_i = 1 \\ & \mu^T \mathbf{w} = \mu_t\end{array}$$

# Maximal Sharpe Portfolio

Mathematical optimization formulation

$$\begin{aligned} \text{maximise}_w \quad & \frac{\mu^T w - R_f}{(w^T \Sigma w)^{1/2}} \\ \text{s.t.} \quad & \sum_{j=1}^n w_j = 1 \\ & w_j \geq 0, j = 1, \dots, N \end{aligned}$$



$$\begin{aligned} \text{minimise}_w \quad & y^T \sum y \\ \text{s.t.} \quad & (\mu^T w - R_f)^T y = 1 \\ & \sum_{j=1}^N y_j = \kappa, \\ & \kappa \geq 0, \\ & w_j = \frac{y_j}{\kappa}, j = 1, \dots, N \end{aligned}$$

# Naive diversification

Don't put all eggs in a single basket

- How to measure number of baskets?
- It's not just the numbers of assets in the portfolio. Why? Because you can have several concentrated bets on small amount of assets and almost nothing on all the others
- Effective number of constituents (ENC) measures it and can be optimized directly

$$\text{ENC}_\alpha(\mathbf{w}) = \|\mathbf{w}\|_\alpha^{\frac{\alpha}{1-\alpha}} = \left( \sum_{k=1}^N w_k^\alpha \right)^{\frac{1}{1-\alpha}}$$

alpha >= 2

# Risk-Parity Optimization

Baskets of risks, not dollars

- Imagine a portfolio with 50% allocated in risky asset with 30% volatility and not-risky bond with 10% volatility (both not correlated)
- ENC = 2, but are these two baskets actually equal from the risk point of view?
- Instead of stacking and normalizing for weights for the asset, we can do the same for the volatility per asset

$$q_k = \frac{w_k [\Sigma w]_k}{w' \Sigma w}, \text{ where } \sum_{k=1}^N q_k = 1$$

$$\text{ENCB}_\alpha(\mathbf{w}) = \|\mathbf{q}\|_\alpha^{\frac{\alpha}{1-\alpha}} = \left( \sum_{k=1}^N q_k^\alpha \right)^{\frac{1}{1-\alpha}}$$

# Other options

Same optimization process :)

$$w^{MV} = \arg \min w^T \cdot \Sigma \cdot w$$

min variance

$$w^{MD} = \arg \max \frac{w \times \sigma}{\sqrt{w^T \cdot \Sigma \cdot w}}$$

max diversification

$$w^{MDec} = \arg \min w^T \cdot A \cdot w$$

max de-correlation

$$\sum_i^N w = 1$$

# Ensuring that it wasn't just an overfit

Alternative useful metrics to track

- Information ratio - how much (if we do at all) we perform better than a benchmark
- Maximal drawdown and corresponding periods
- Factor exposure: how much we “overfit” to some feature (factor models + ML feature importance)

$$InformationRatio = \frac{E[Returns - Benchmark]}{\sqrt{V[Returns - Benchmark]}} * \sqrt{n}$$

# Ensuring that it wasn't just an overfit

Corrections for distributions and trials

- Probabilistic Sharpe: adjusted estimate of SR, by removing the inflationary effect caused by short series with skewed and/or fat-tailed returns.
- Deflated Sharpe: a PSR where the rejection threshold is adjusted to reflect the multiplicity of trials.
- Minimum track record: “How long should a track record be in order to have statistical confidence that its Sharpe ratio is above a given threshold?”

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

$$SR^* = \sqrt{V[\{SR_n\}]((1 - \gamma) * Z^{-1}[1 - \frac{1}{N} + \gamma * Z^{-1}[1 - \frac{1}{N} * e^{-1}]]}$$

$$MinTRL = 1 + [1 - \gamma_3 * SR + \frac{\gamma_4 - 1}{4} * SR^2] * (\frac{Z_\alpha}{SR - SR^*})^2$$

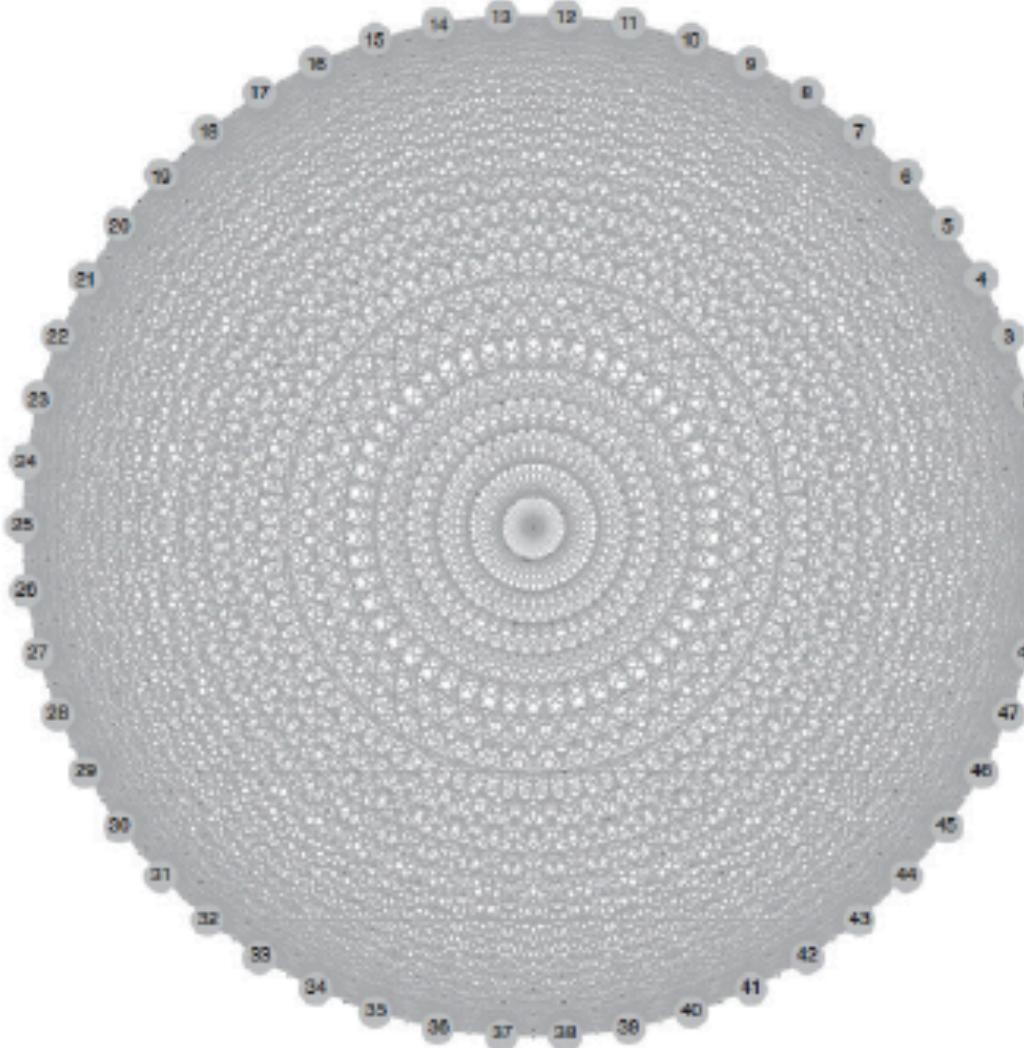
**Let's put it together into a  
strategy**

# **ML-based approaches**

curse of dimensionality and overfitting

# Optimization problems

## Dimensionality curse



complete graph  
of assets

$N(N-1)/2$  correlations



hierarchical tree  
of assets

- increase sample period of frequency
- decrease number of parameters
- impose a structure

# Other problems of Markowitz and Co

- Small deviations in the forecasted returns will cause very different portfolios
- Quadratic programming methods require the inversion of a positive definite covariance matrix (all eigenvalues must be positive)
- Markowitz's curse: The more correlated the investments, the greater the need for diversification, and yet the more likely we will receive unstable solutions. The benefits of diversification often are more than offset by estimation errors.
- Anyway need to rebalance, because correlations change all the time

# Constant correlation model

Reducing dimensionality of the model

- We can impose a model of the same correlation between the assets
- $N(N-1)/2$  parameters -> 1 parameter
- But meanwhile, we are introducing a model specification risk!

$$\sigma_{ij} = \begin{cases} \sigma_{ii} = \sigma_i^2 & \text{when } i = j \\ \sigma_{ij} = \rho\sigma_i\sigma_j & \text{when } i \neq j \end{cases}$$

a parameter ***rho*** in covariance matrix stays the same for all ***i,j***

# Robust Risk Estimation

i.e. regularization of the model

- Sample risk suffers from “overfitting” and curse of dimensionality
- Model risk comes from mainly from misspecification
- A tradeoff between sample views and model views: shrinkage, the sample covariance matrix is ‘shrunk’ towards the structured estimator

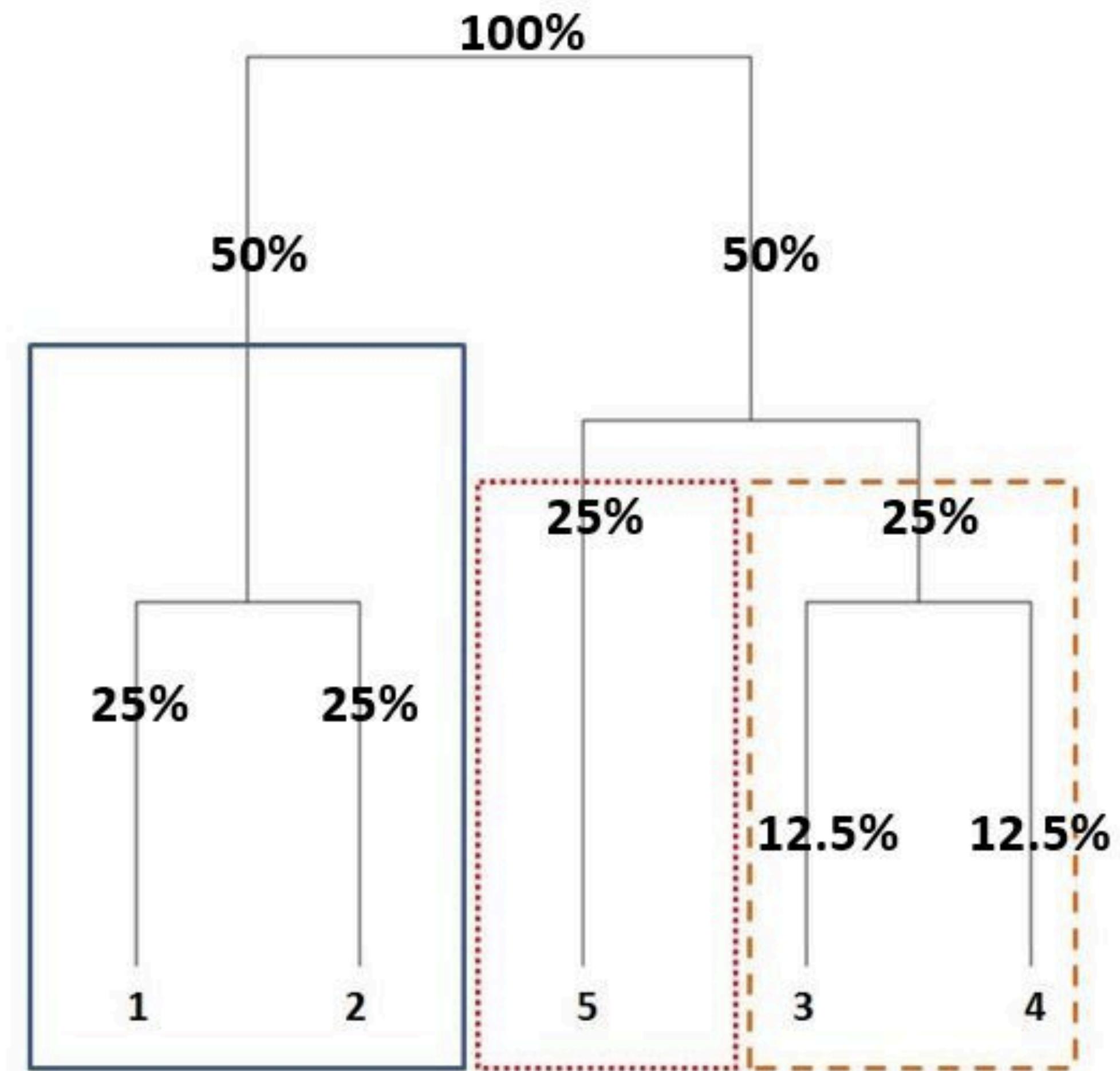
$$\hat{\Sigma}_{Shrink} = \hat{\delta}^* F + (1 - \hat{\delta}^*) S$$

**S** - sample cov, **F** - model

# Hierarchical Risk Parity

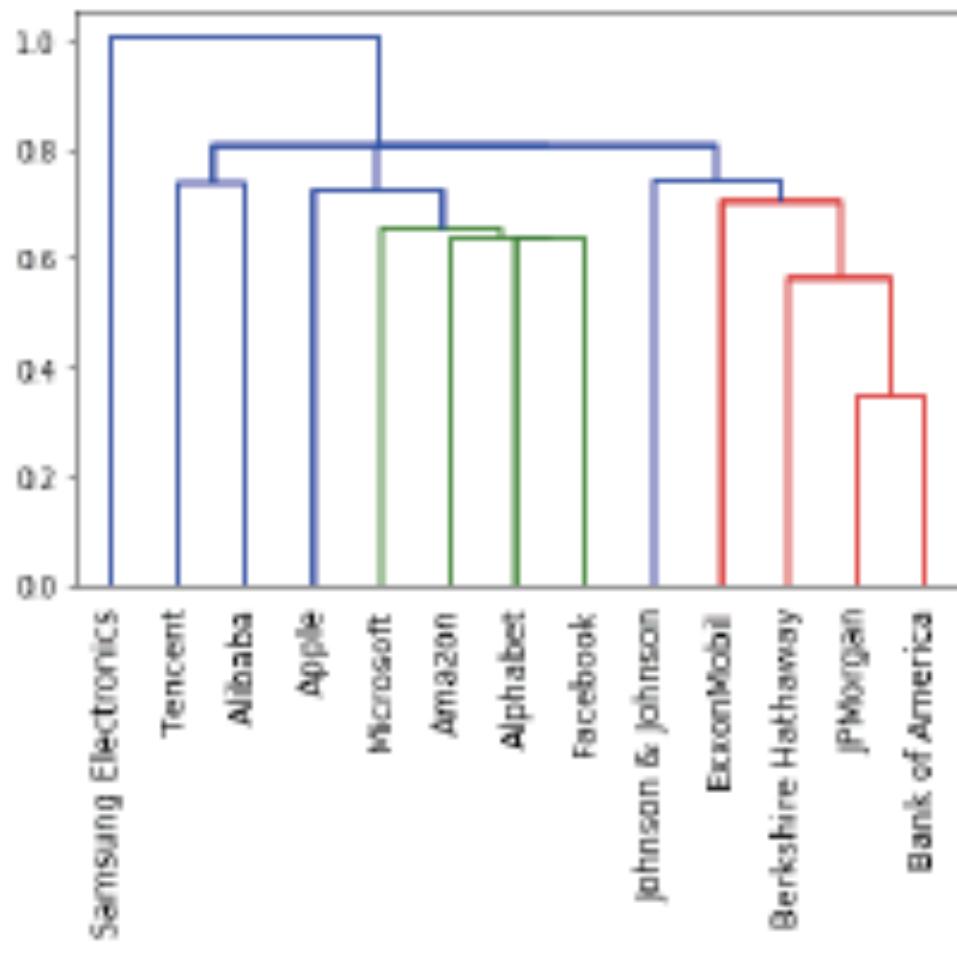
Imposing structure in the model

- Hierarchical clustering
  - Selecting the optimal number of clusters
  - Capital is allocated across clusters
  - Capital is allocated within clusters
- In the Equal Weighting portfolio, all the clusters are weighted equally in terms of risk to construct our optimal portfolio.
- The following image shows how asset allocation works with a small example

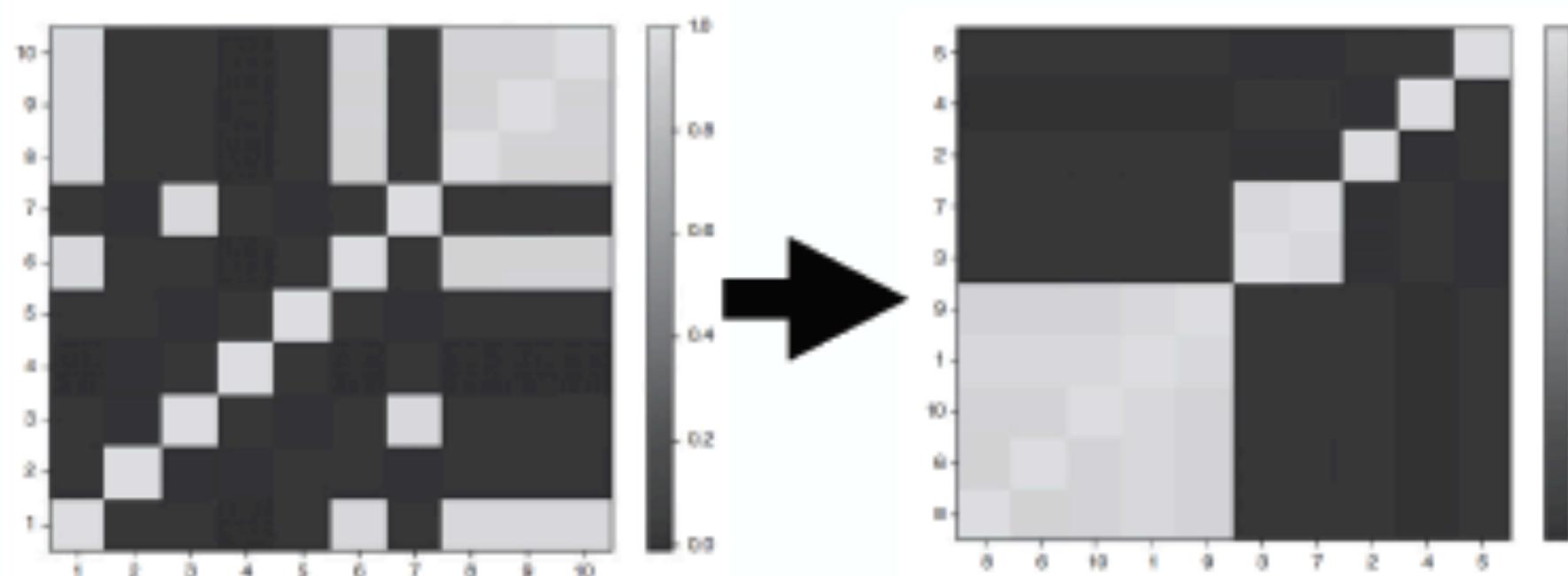


# Hierarchical Risk Parity

Imposing structure in the model



**Perform tree clustering of the covariance matrix**



**Regroup covariance matrix based on the clusters and quasi-diagonalize it**

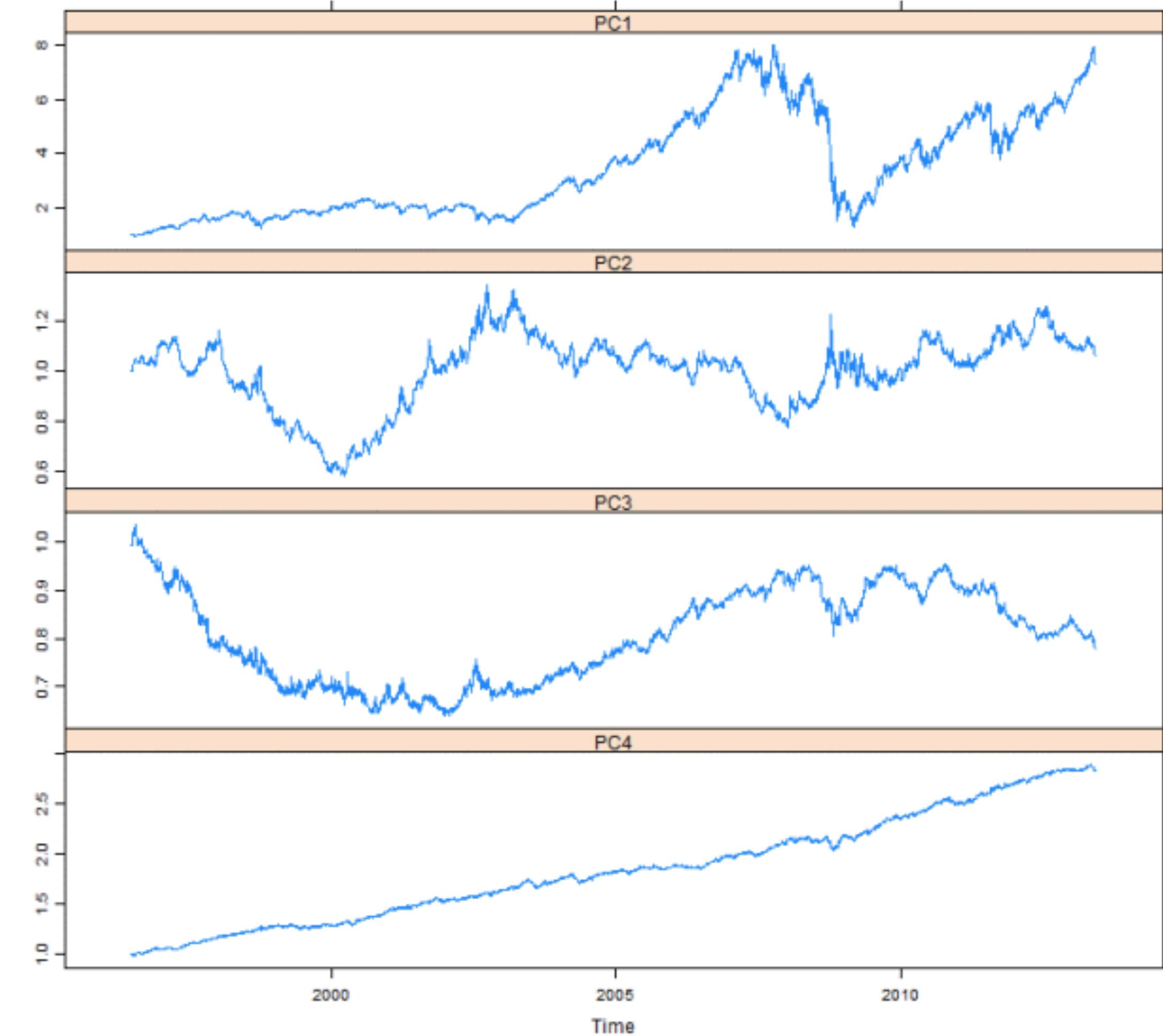
**RECURSIVE ALLOCATION**

**Recursively apply weights to the assets**

# PCA and Autoencoders

Imposing structure in the model

- Your portfolio is a linear combination of the assets
- CW, EW, GMV portfolios are good benchmarks, “proxies” to the market
- How to select such proxy based on the explanation of the variance of the market?
- Or based on the explanation of the predictability of the market?



**Let's do the horse racing**

# Why we did it?

Sometimes it's even worse than equal allocation...

- As we discussed above, everything starts with selection of the trading universe and strong underlying factors
- Yes, we really want to be active and predict things before other participants of the market figure that out, so we get advantage and extra-profit
- But first, we want to build a baseline of our market, a fundament that will be stable long-term by itself. That's why everything today was about some sort of “unsupervised” learning of the market structure
- We will allocate X% of our assets there to cover our liabilities and to be secure, and put the rest (1-X)% to the performance-seeking part, which we are going to discuss tomorrow

# Future improvements

For the portfolio optimization

- Using expected returns: teaching models to forecast prices, merging together with baselines using Black-Litterman portfolios
- Working on the risks: correlation is the most primitive measure of similarity of the assets. How about non-linear measures? Statistical measures?
- Rethinking the whole setup. We don't construct portfolio only once and hold it forever. Time-to-time we refresh information about the market and update the weights. It already looks a bit more like optimal control / reinforcement learning, rather a single function optimization

# Day 2

Hazardous path of speculations and wild guesses with ML

- Active investment management: what is that alpha?
- Why financial machine learning is different from regular one?
- Predicting asset prices in Python in a Kaggle way [PRACTICE]
- Fixing inputs and outputs with respect to financial logic
- Predicting asset prices in Python as you never did before [PRACTICE]
- Why theoretical physicists success in this job and what to do next?

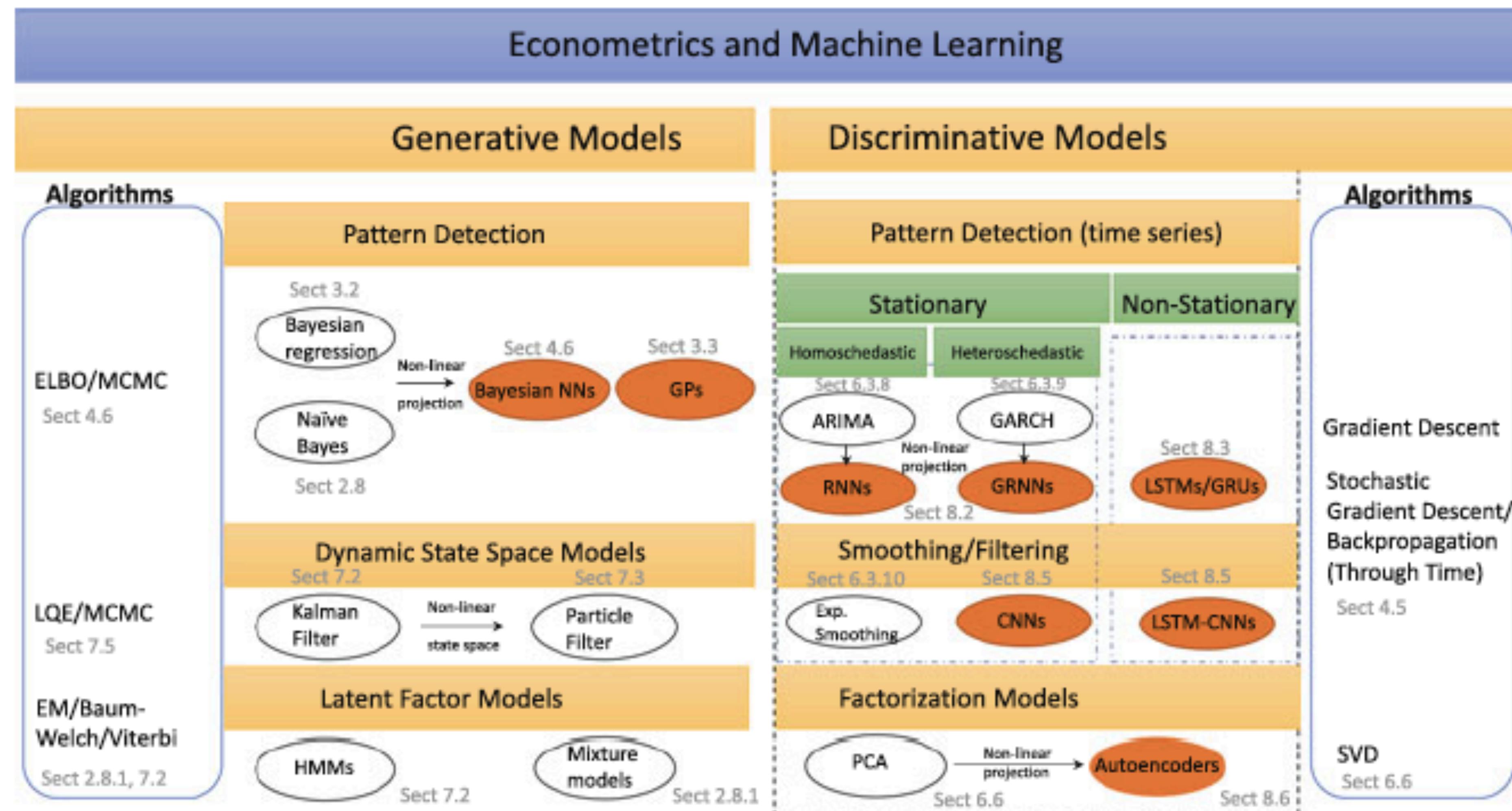
# Why financial ML is different?

prices != pixels

# Hello to econometrics

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3373116&download=yes](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3373116&download=yes)

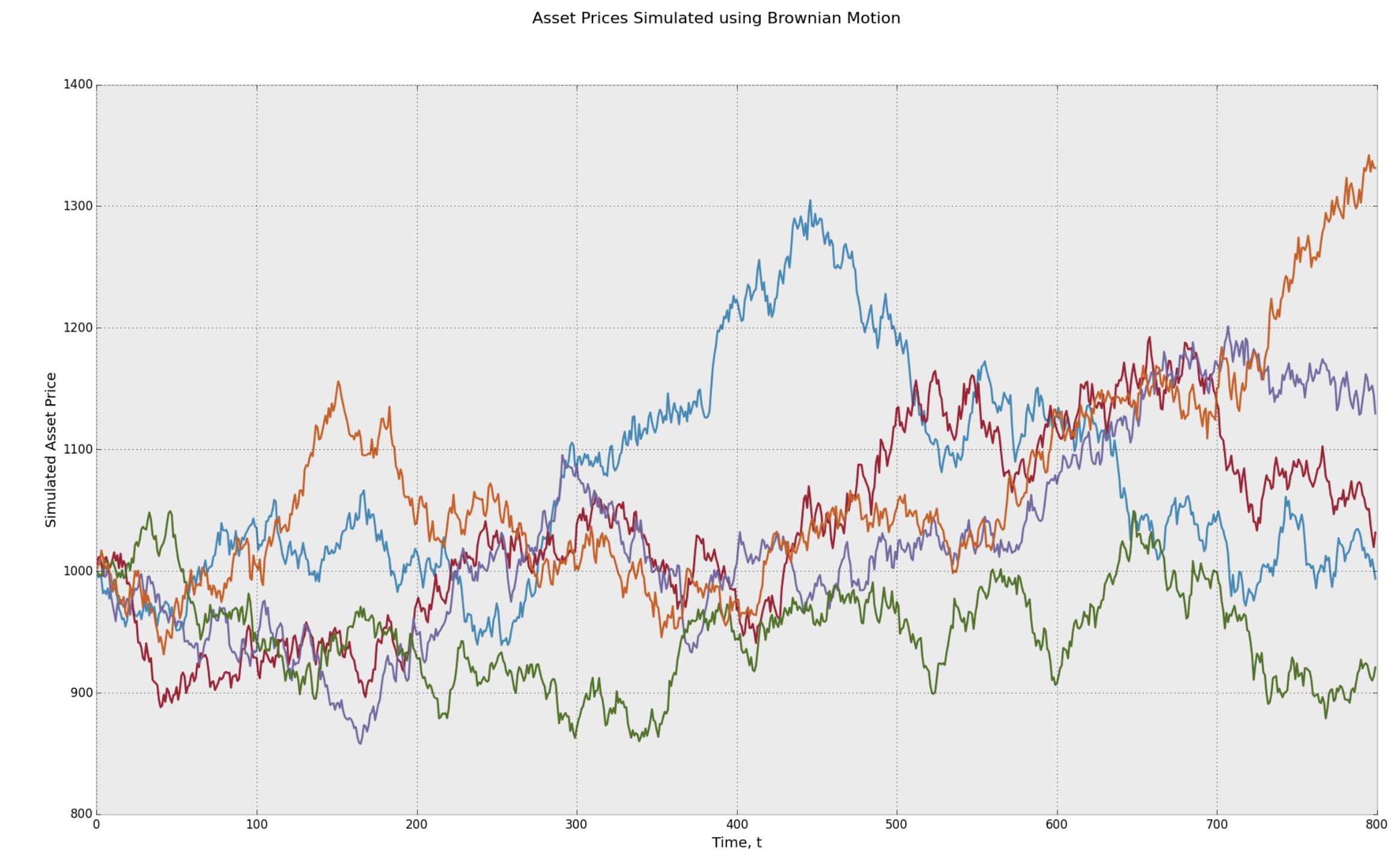
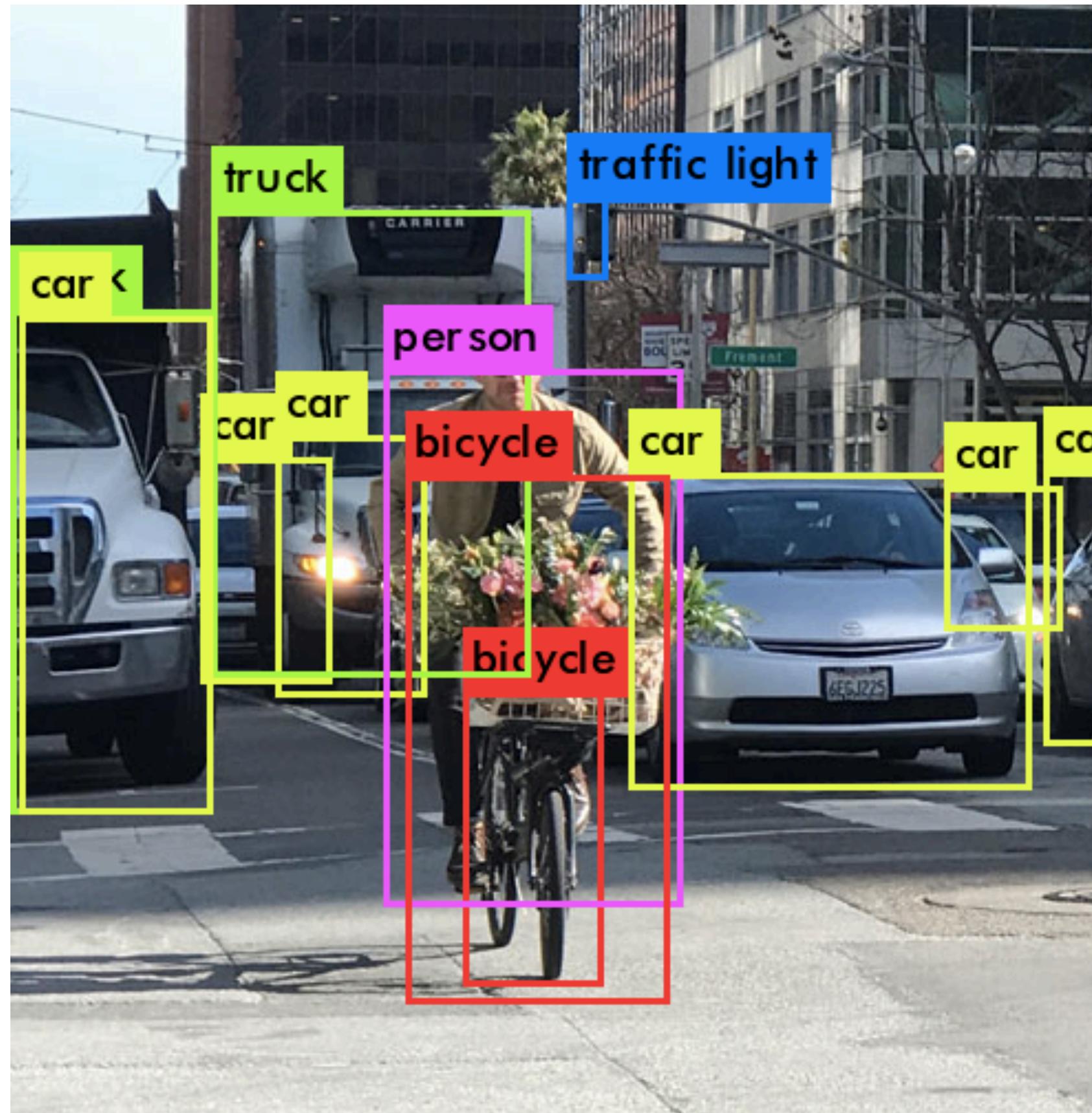
# Econometrics -> ML



**Let's train the models!**

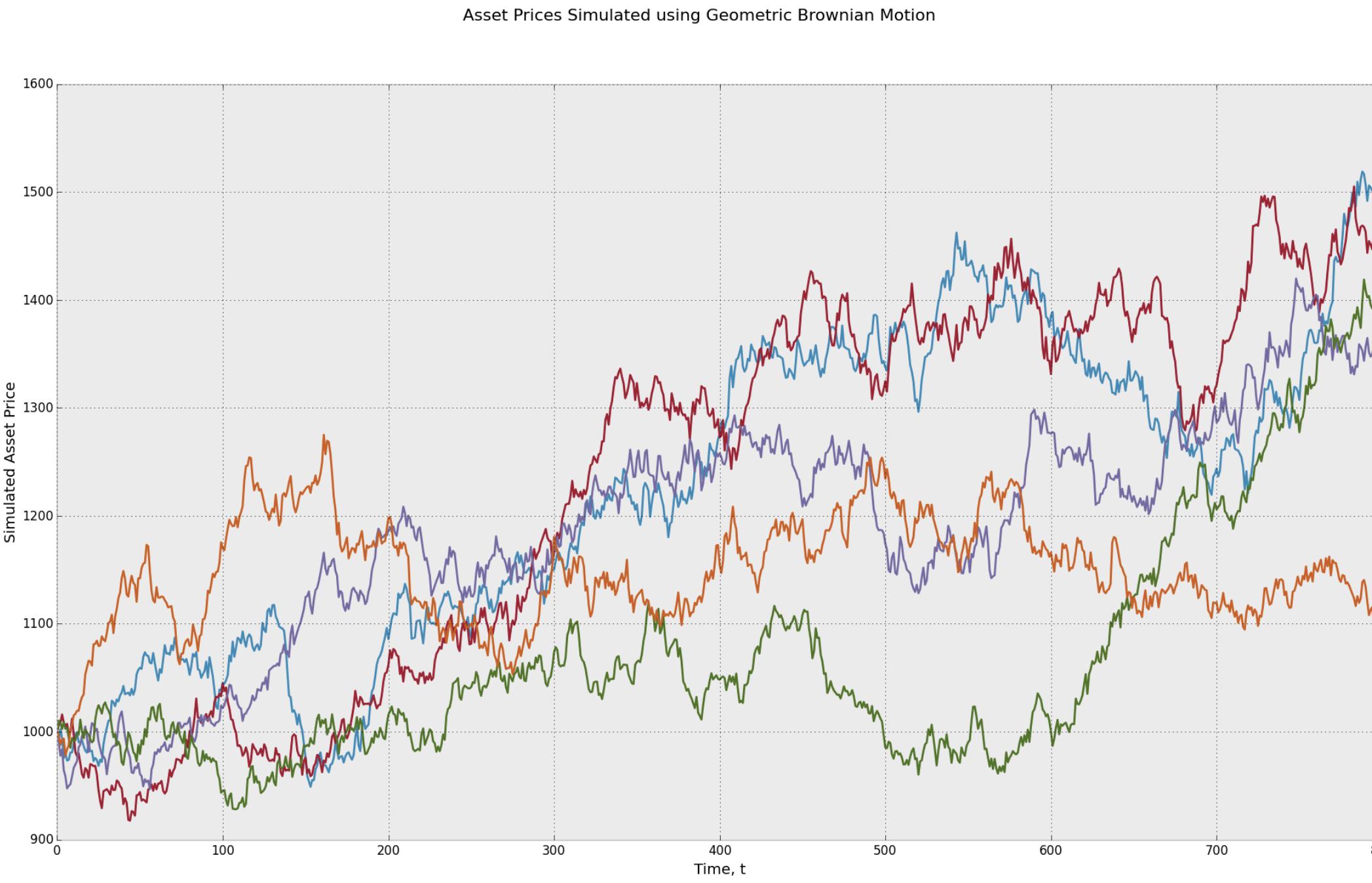
# Fixing inputs and outputs

# Treating it as a regular ML exercise



# Treating it as a regular ML exercise

$$dS_t = \mu S_t dt + \sigma dS_t W_t$$



$$dS_t = \mu S_t dt + \sigma S_t dW_t + dJ_t$$

$$dJ_t = S_t d\left( \sum_{i=0}^{N_t} (Y_i - 1) \right)$$

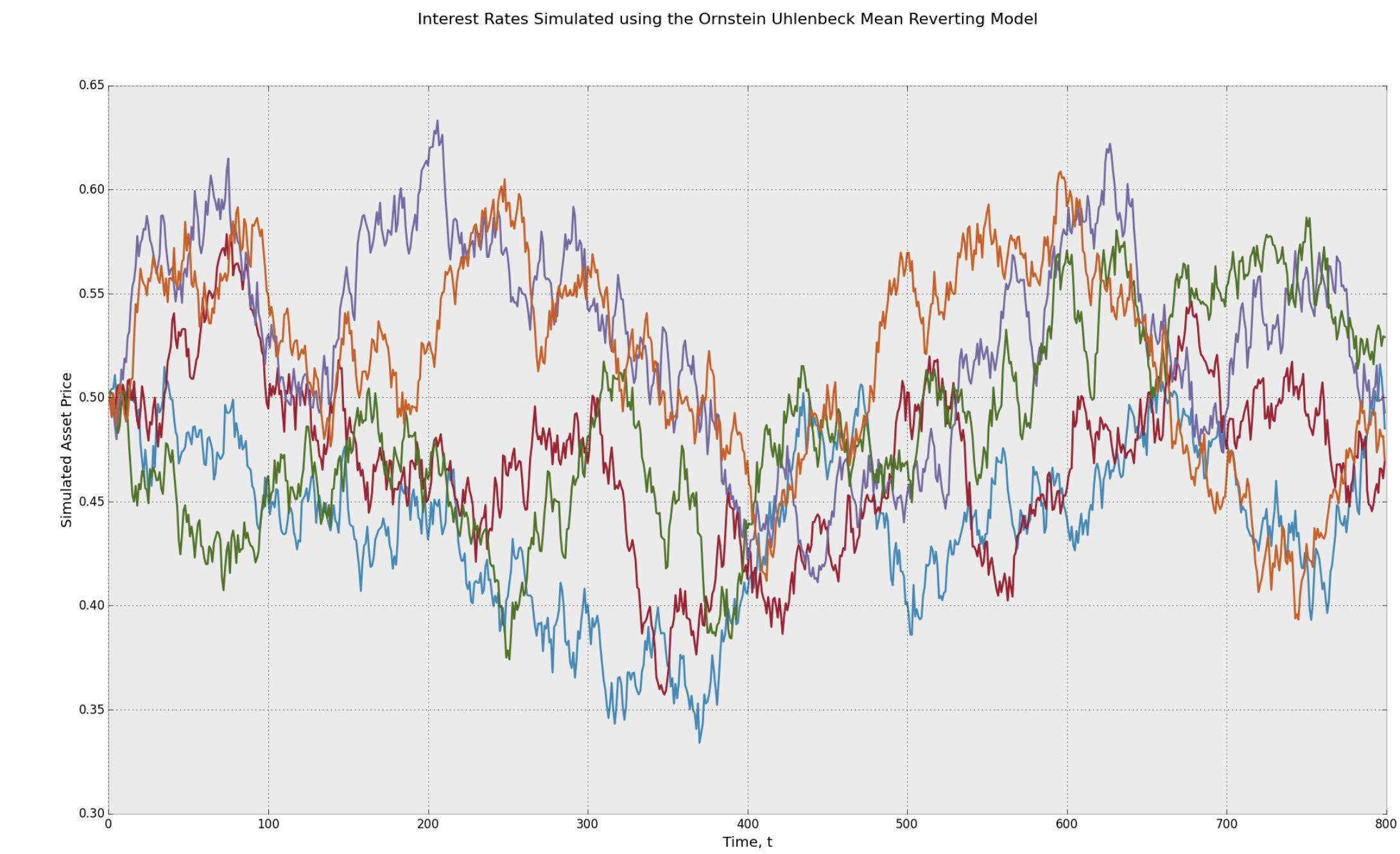
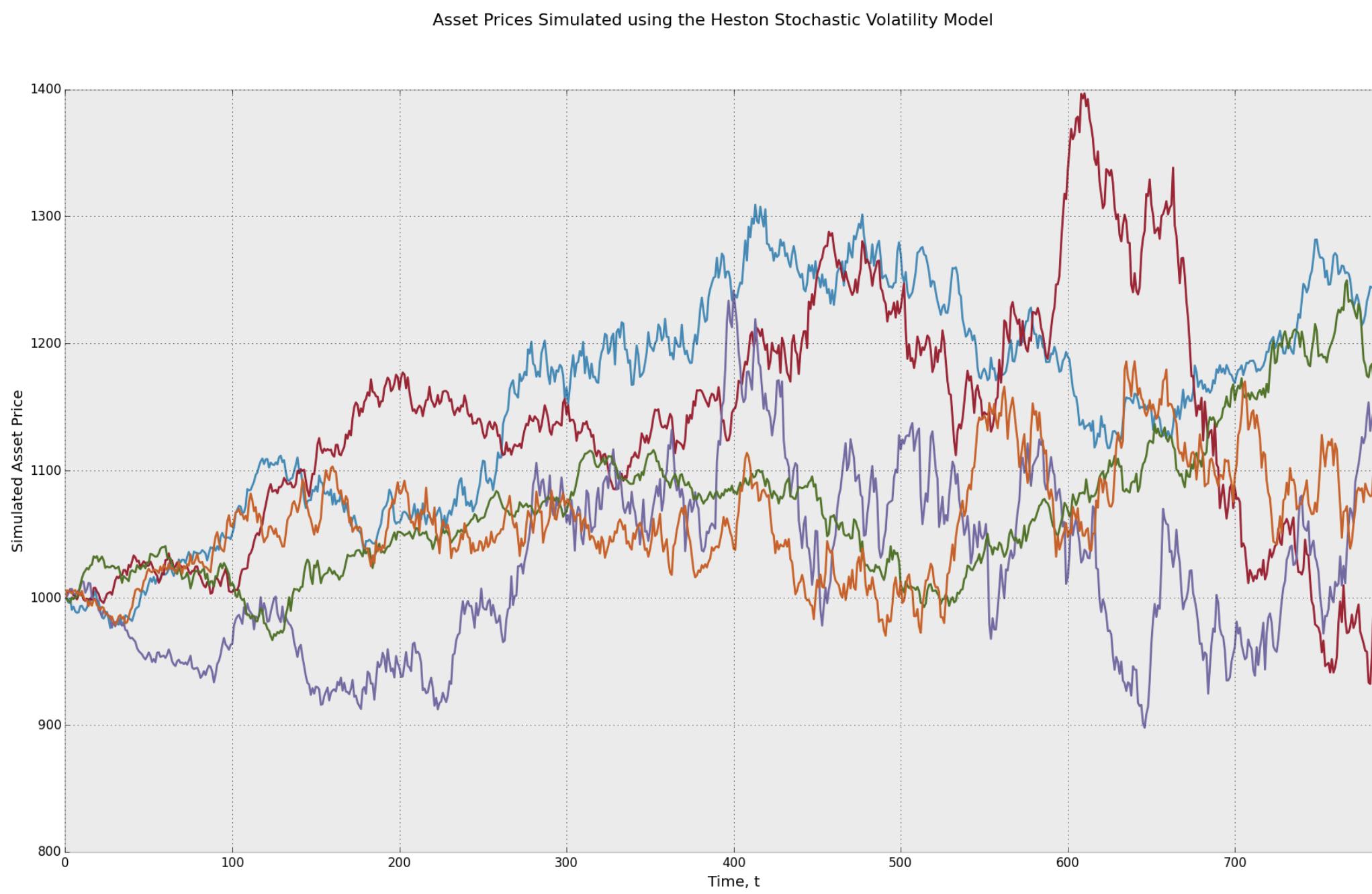


# Treating it as a regular ML exercise

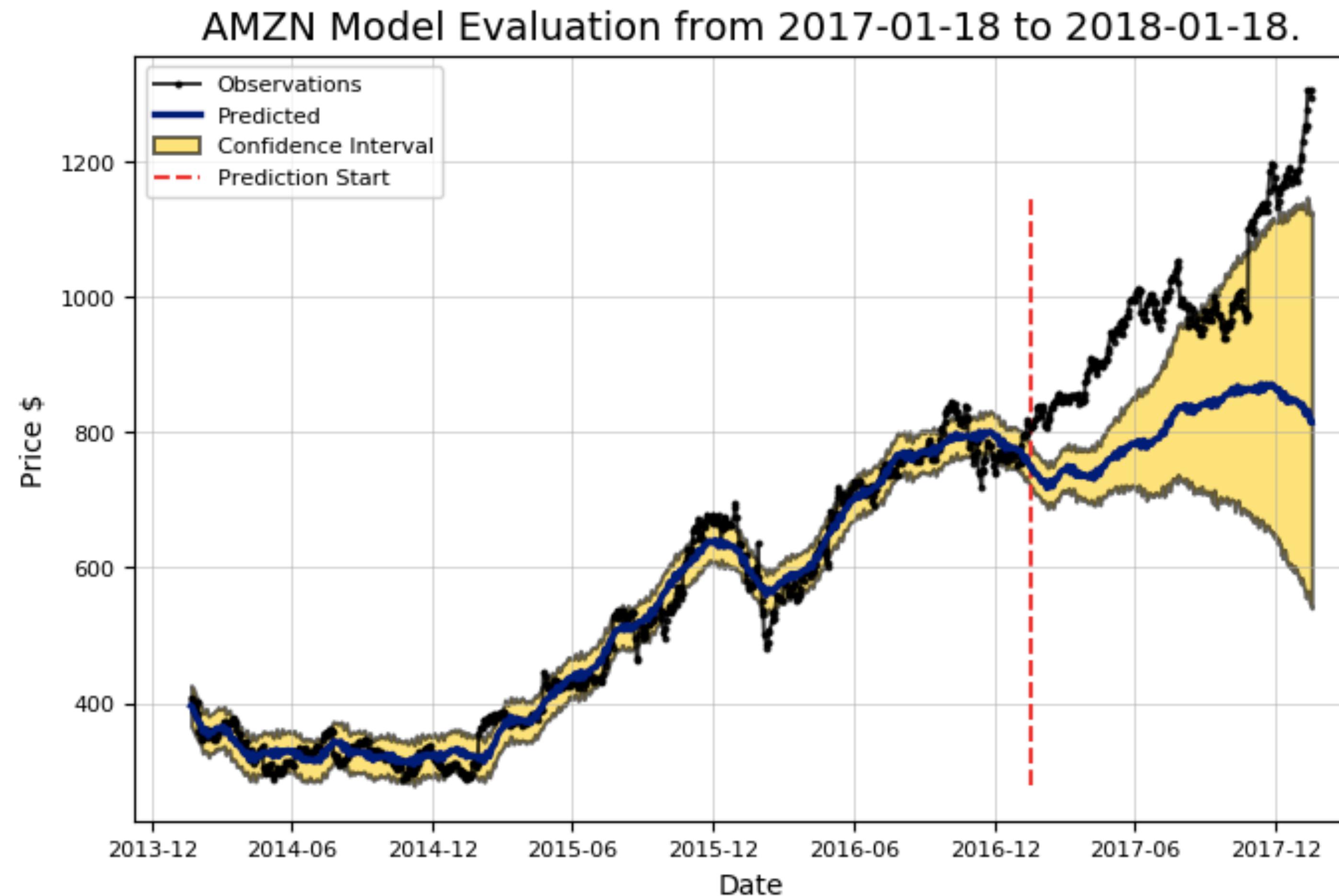
$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S \text{ where}$$

$$dv_t = a(b - v_t)dt + \sigma \sqrt{v_t} dW_t^v \text{ (Cox Ingersoll Ross),}$$

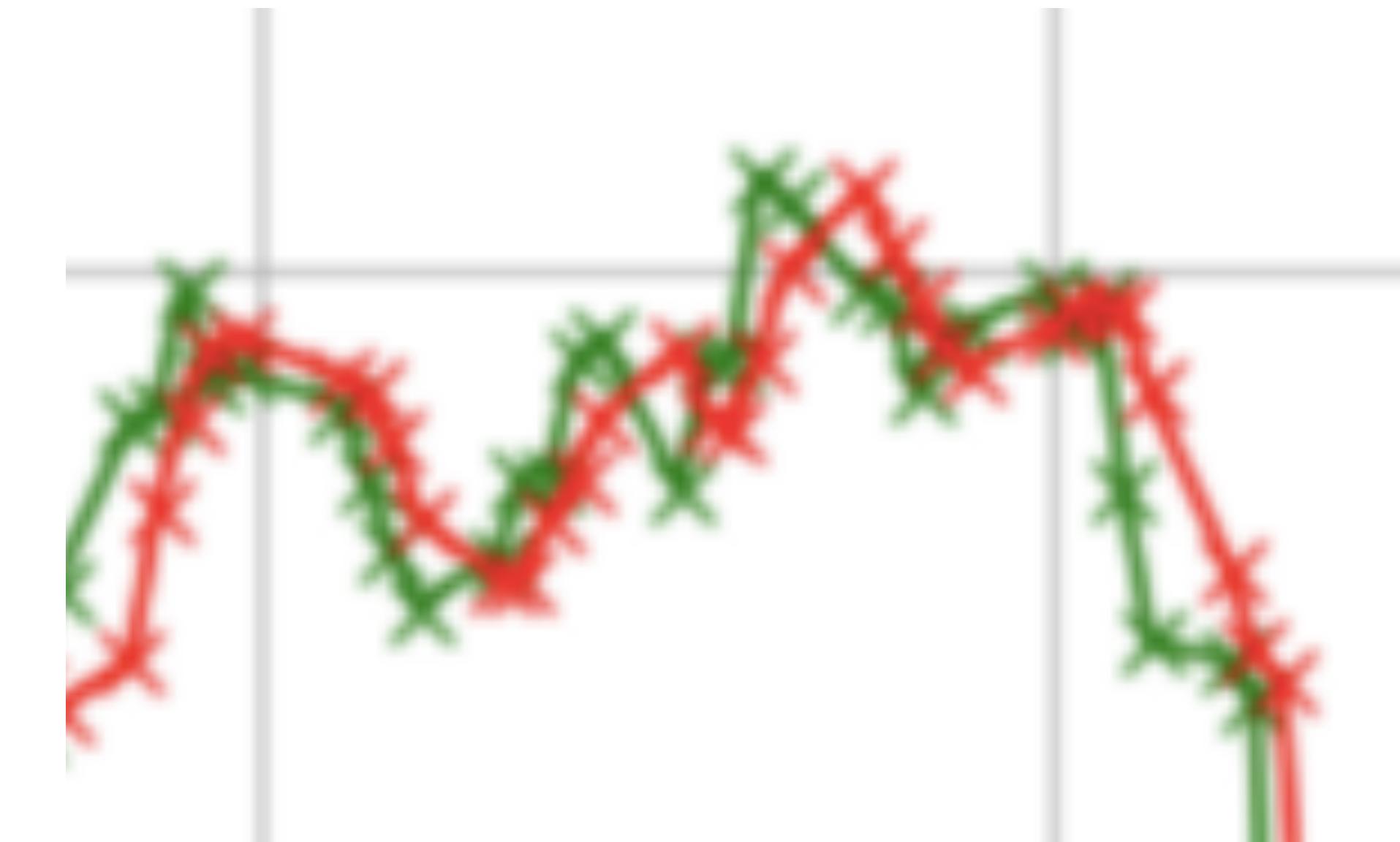
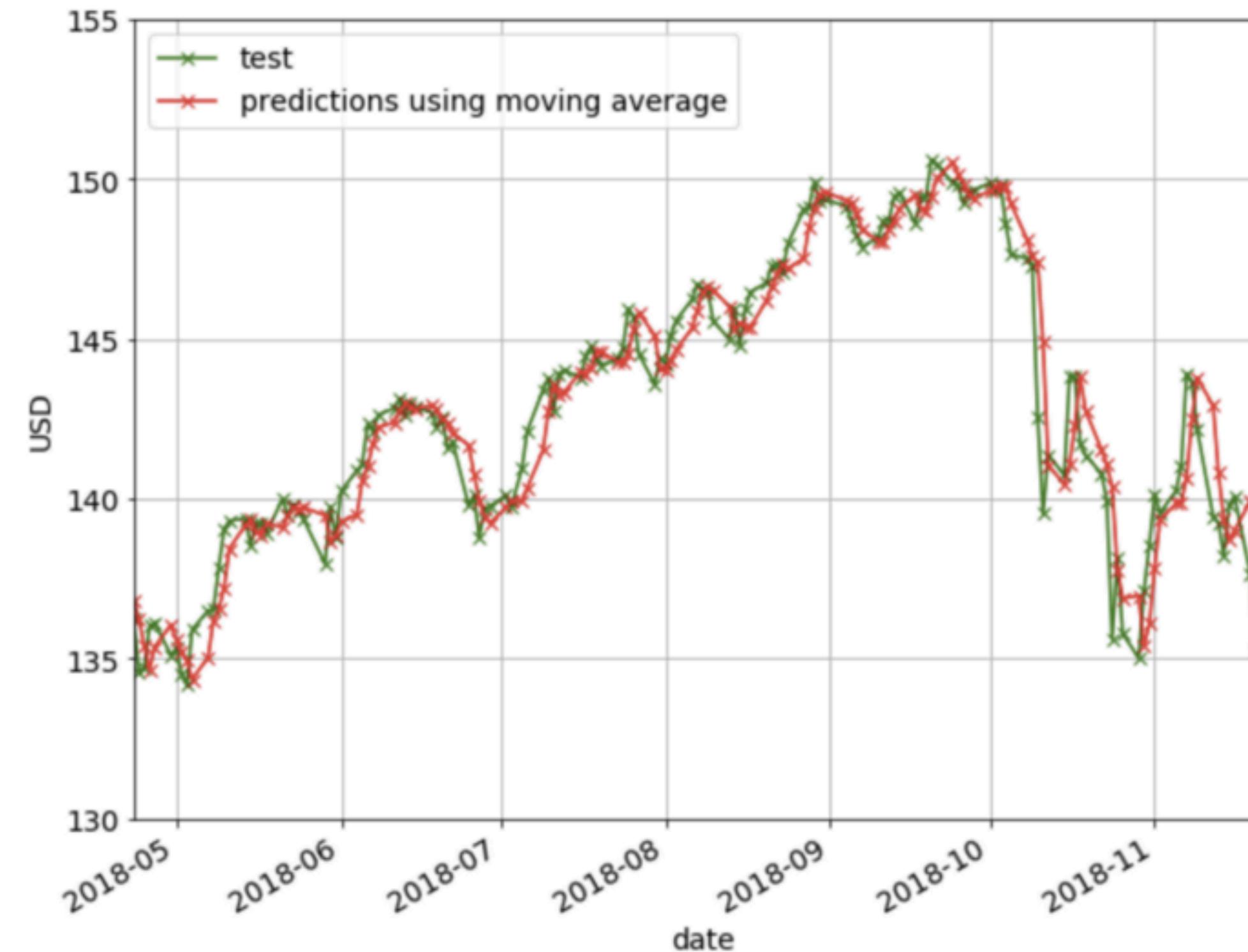
$$dr_t = a(b - r_t)dt + \sigma \sqrt{r_t} dW_t$$



# Forecasting the price in the future



# Forecasting the price in the future



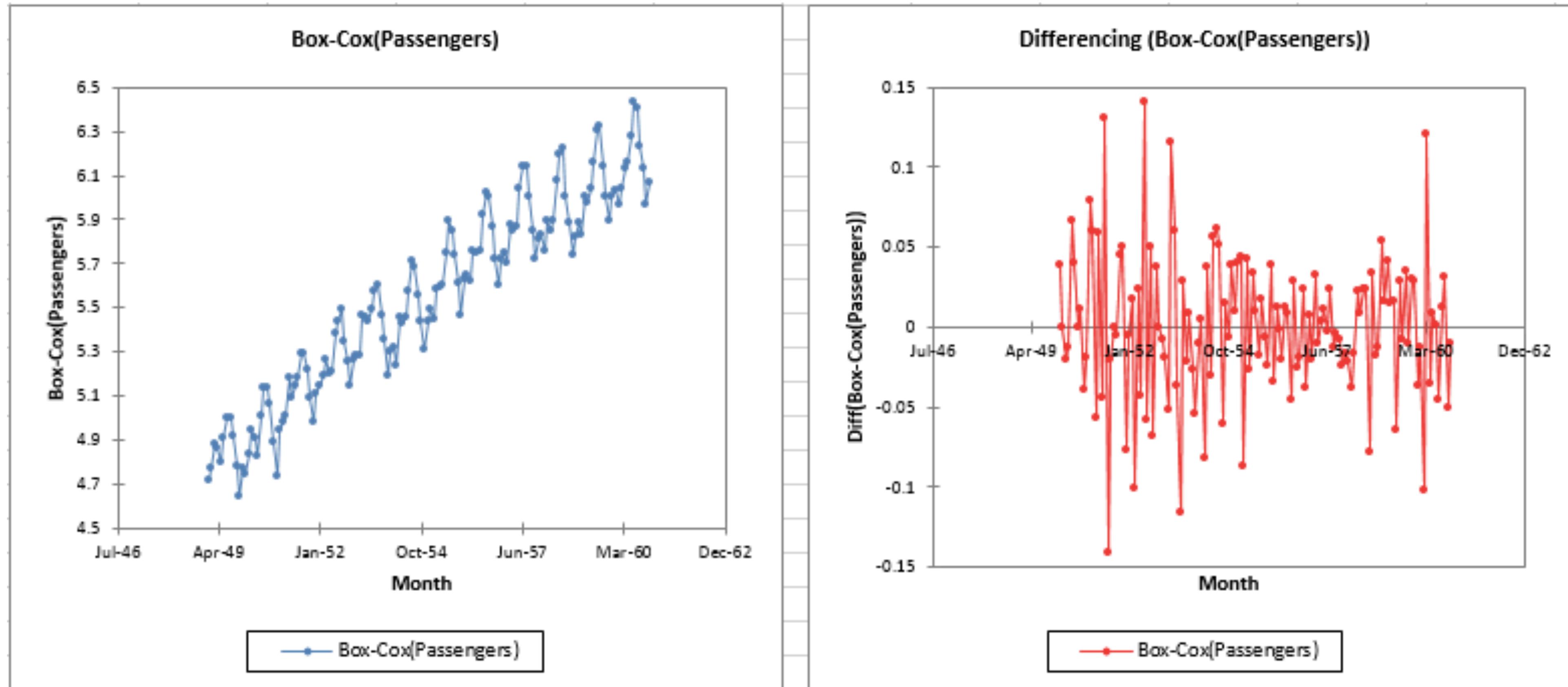
**Let's train the models!**

# **Fixing inputs and outputs 2.0**

# Why machine learning fails?

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3031282](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3031282)

# Treating it as a regular ML exercise



# Treating it as a regular ML exercise

## Difference operator [\[ edit \]](#)

---

*Main article: [Finite difference](#)*

In time series analysis, the first difference operator : $\nabla$

$$\nabla X_t = X_t - X_{t-1}$$

$$\nabla X_t = (1 - L)X_t .$$

Similarly, the second difference operator works as follows:

$$\nabla(\nabla X_t) = \nabla X_t - \nabla X_{t-1}$$

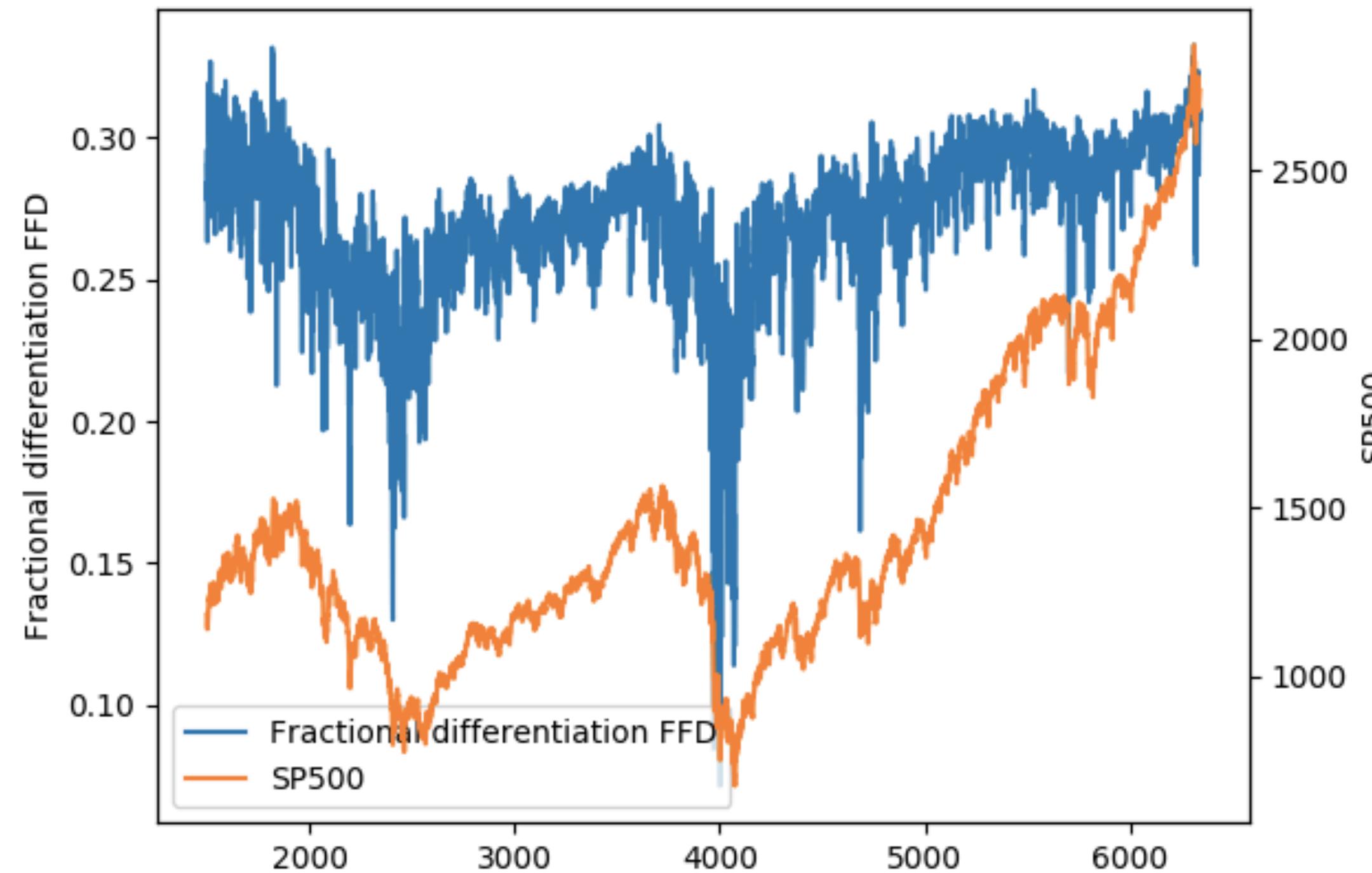
$$\nabla^2 X_t = (1 - L)\nabla X_t$$

$$\nabla^2 X_t = (1 - L)(1 - L)X_t$$

$$\nabla^2 X_t = (1 - L)^2 X_t .$$

The above approach generalises to the  $i$ -th difference operator  $\nabla^i X_t = (1 - L)^i X_t .$

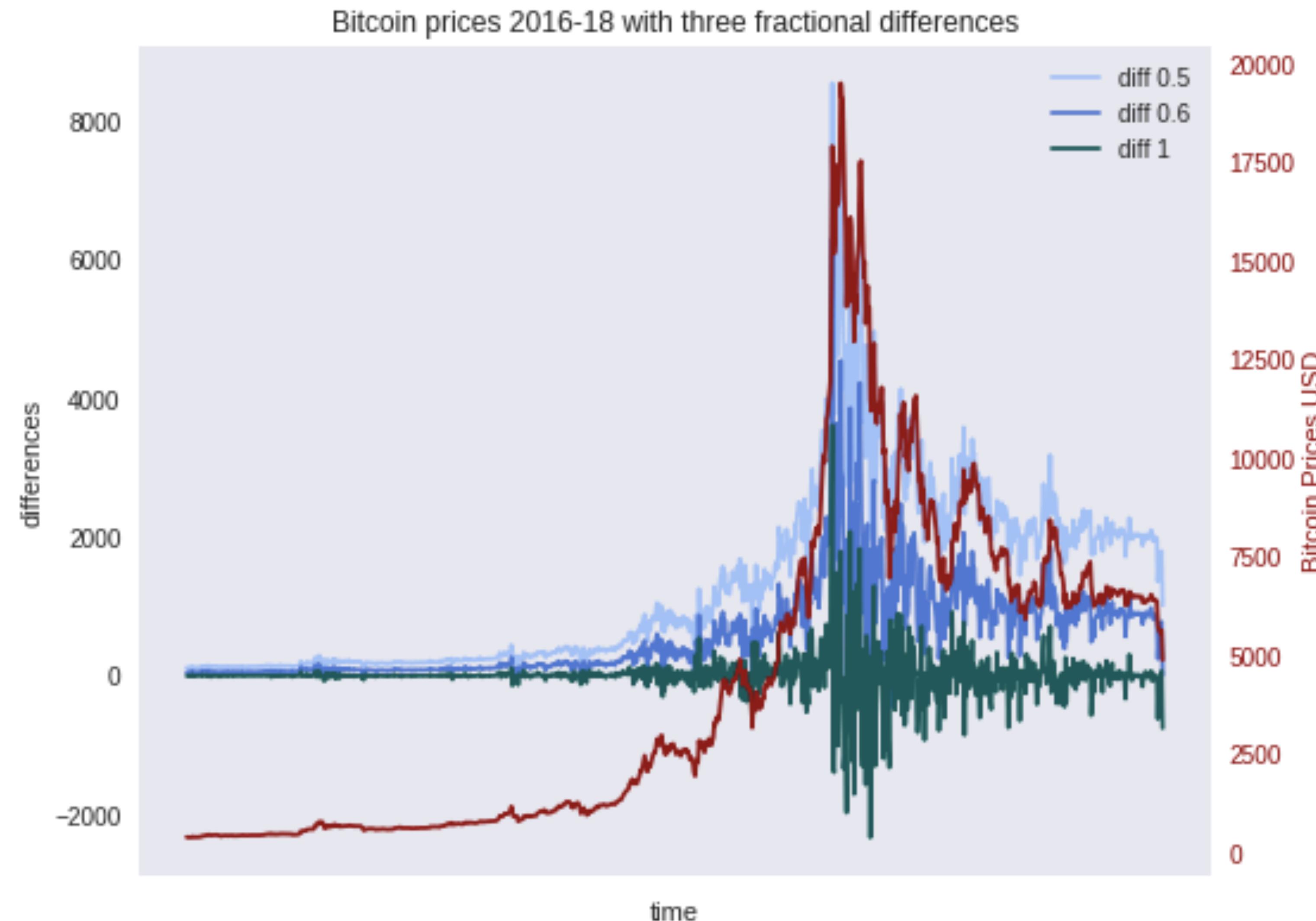
# Treating it as a regular ML exercise



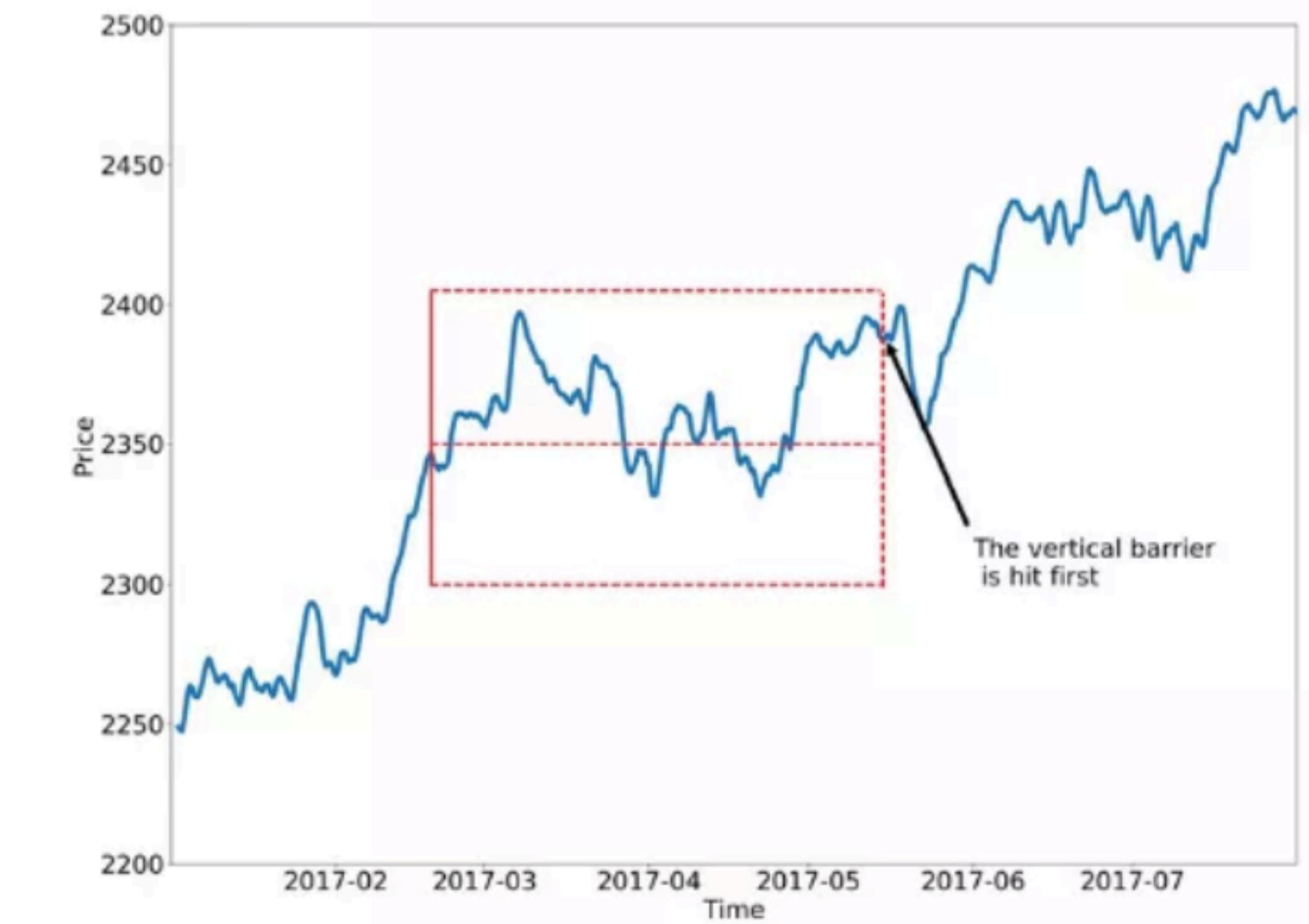
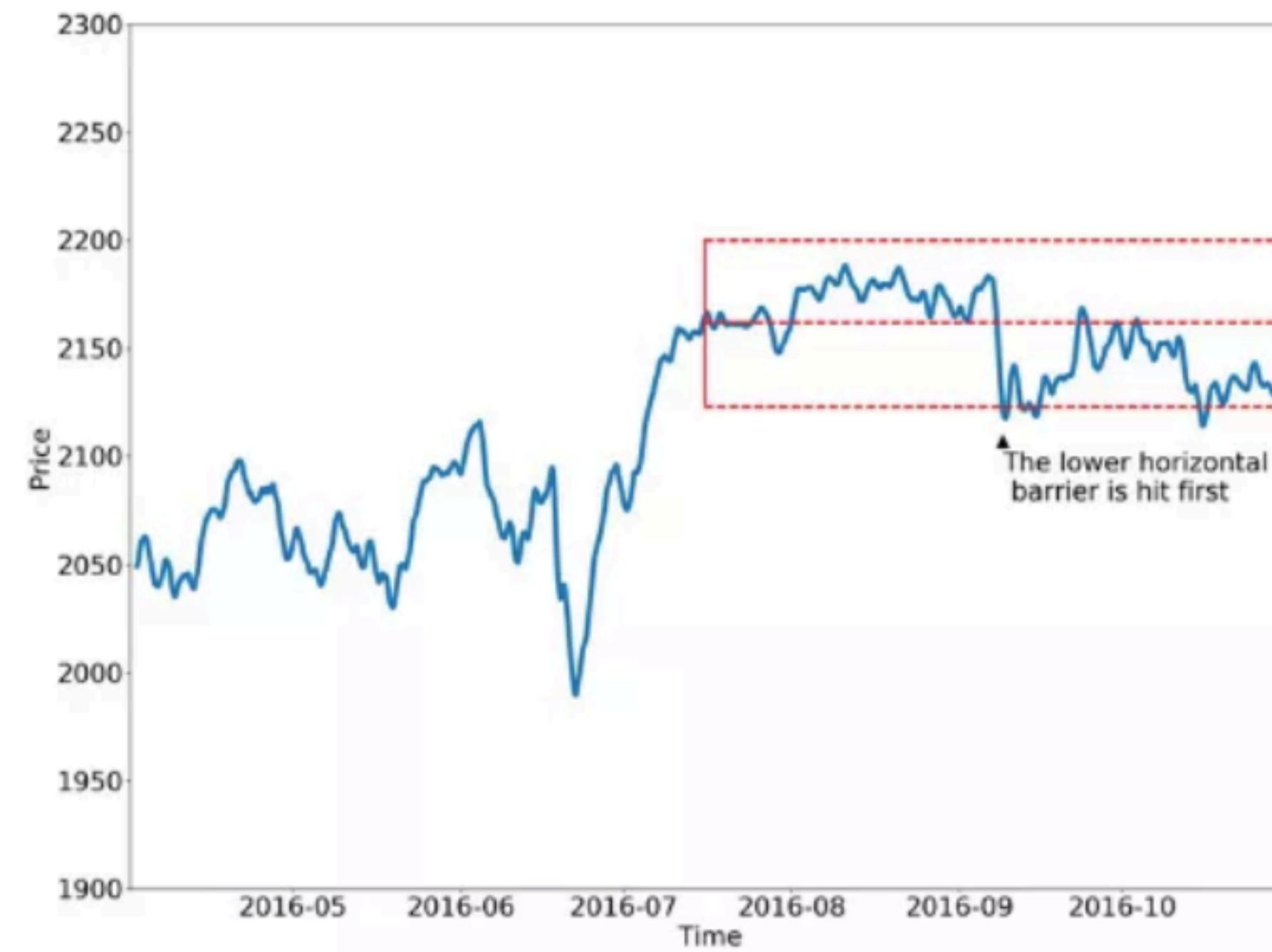
$$(\mathbf{I} - \mathbf{B})\mathbf{X}_t = \mathbf{X}_t - \mathbf{B}\mathbf{X}_t = \mathbf{X}_t - \mathbf{X}_{t-1}$$

$$\begin{aligned}(1 - B)^d &= \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k \\&= \sum_{k=0}^{\infty} \frac{\prod_{a=0}^{k-1} (d-a)}{k!} (-B)^k \\&= 1 - dB + \frac{d(d-1)}{2!} B^2 - \dots.\end{aligned}$$

# Treating it as a regular ML exercise



# Forecasting the price in the future



# Backtesting is not a research tool

## **SNIPPET 8.1 MARCOS' FIRST LAW OF BACKTESTING—IGNORE AT YOUR OWN PERIL**

“Backtesting is not a research tool. Feature importance is.”

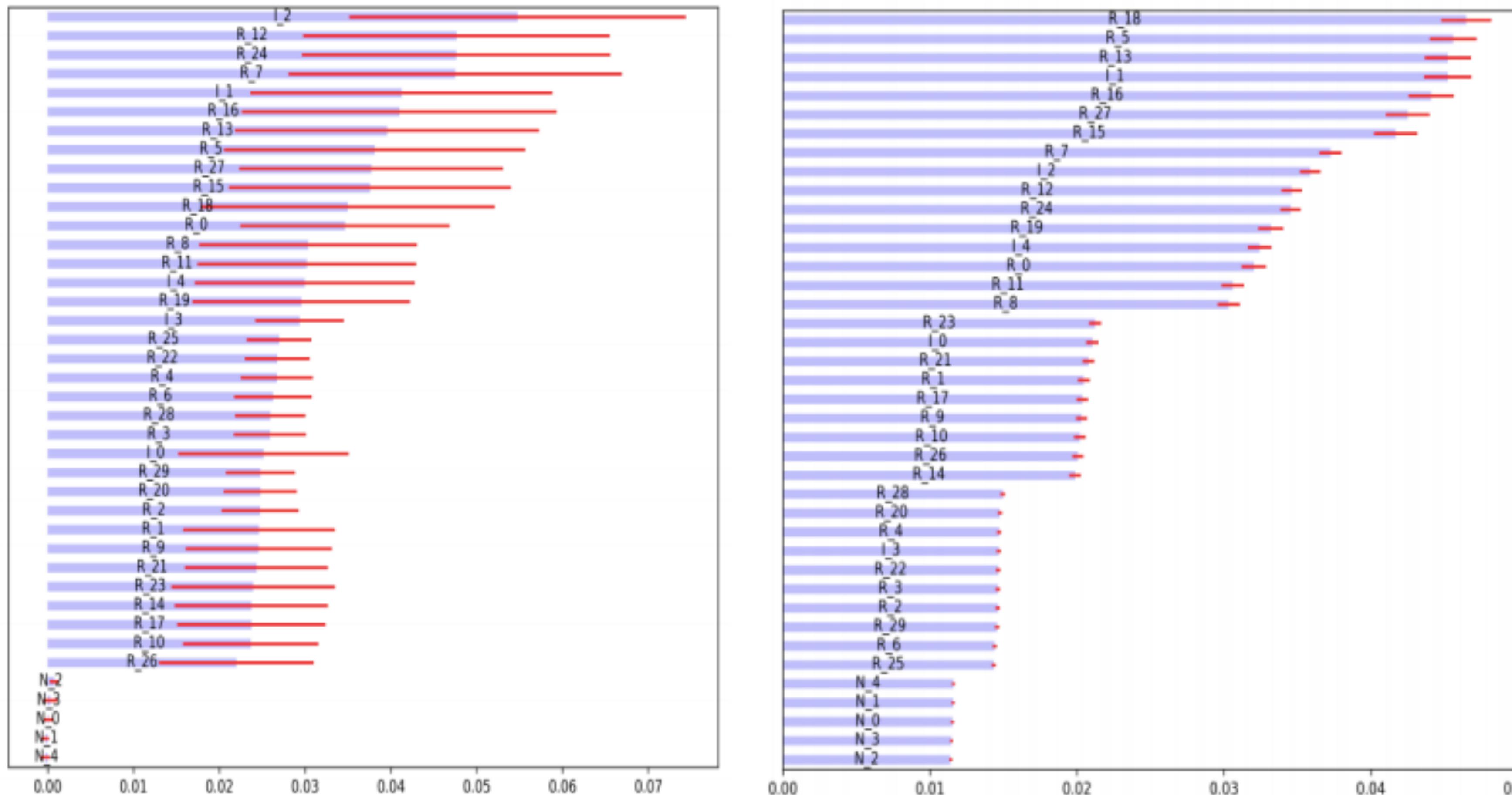
—Marcos López de Prado  
*Advances in Financial Machine Learning* (2018)

## **SNIPPET 11.1 MARCOS' SECOND LAW OF BACKTESTING**

“Backtesting while researching is like drinking and driving.  
Do not research under the influence of a backtest.”

—Marcos López de Prado  
*Advances in Financial Machine Learning* (2018)

# Feature Importance is a Research tool



**Let's experiment...**

**Let's train the models... again!**

# Future improvements

For active trading

- Going to the higher frequency, exploiting deep structure of trades rather aggregated bars. Create your own bars. Filter bars!
- Multitask learning: combine predictions of the floating horizon with meta-predictions of volatility and other market properties
- Rigorous CV and Backtesting

# Research directions

For active trading

- Instruments pricing (options, futures, etc)
- Trading as optimal execution (RL)
- Portfolio management as optimal control (RL)
- Style modeling with factors -> Policy modeling with IRL
- GANs for simulations of stochastic processes
- Tensor networks for hierarchical representations learning

# **Education roadmap**

From zero to hero

 Quantopian Lectures

# Lectures

Lecture 1	Introduction to Research	A simple tutorial.
Lecture 2	Introduction to Python	Some basic tools.
Lecture 3	Introduction to NumPy	How to use NumPy.
Lecture 4	Introduction to pandas	An introduction to pandas.
Lecture 5	Plotting Data	A brief primer.
Lecture 6	Means	Measures of centrality.
Lecture 7	Variance	Measures of dispersion.

 QuantStart

# Successful Algorithmic Trading

A step-by-step guide to developing systematic trading strategies using the Python programming language



Michael L. Halls-Moore, PhD.

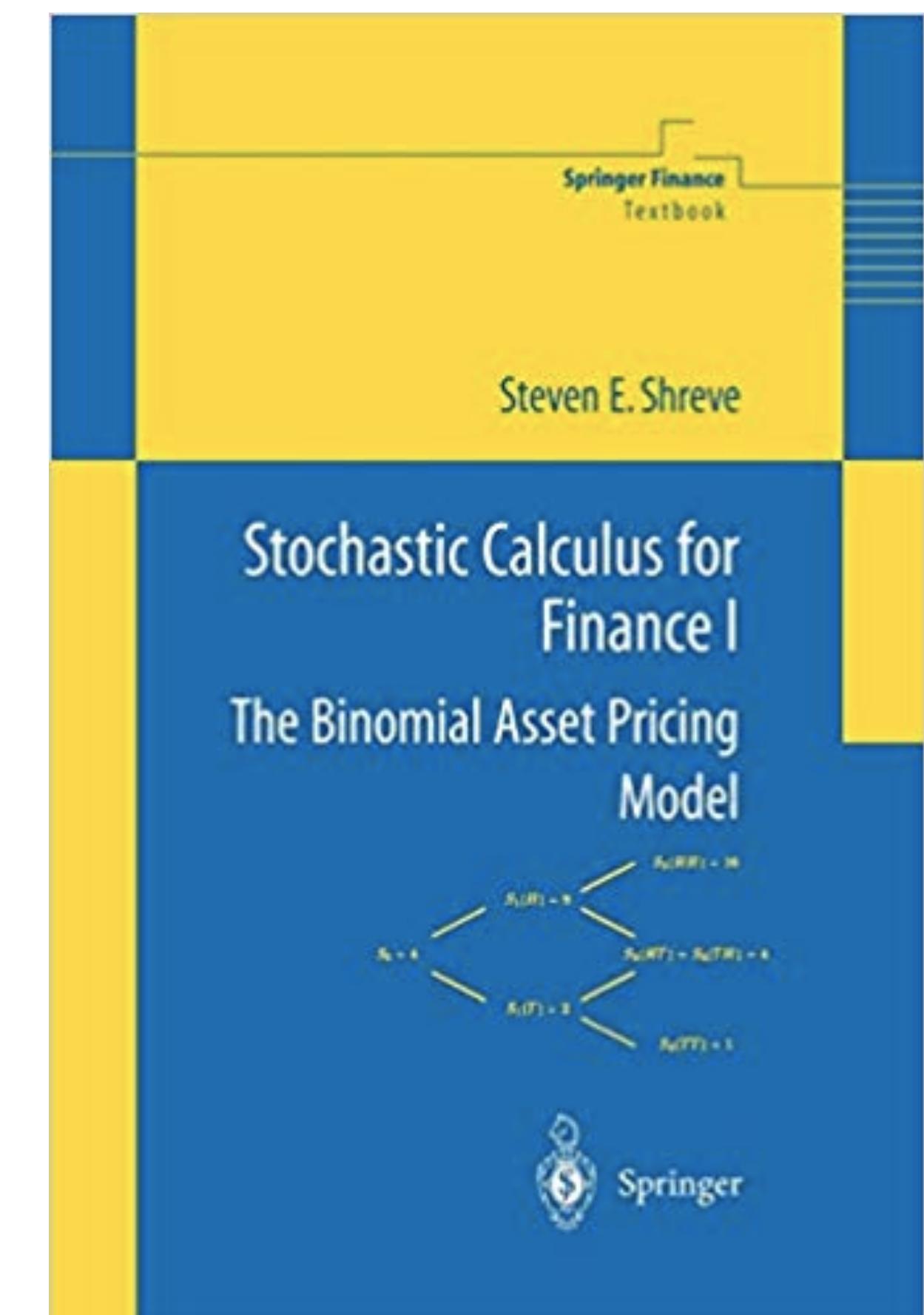
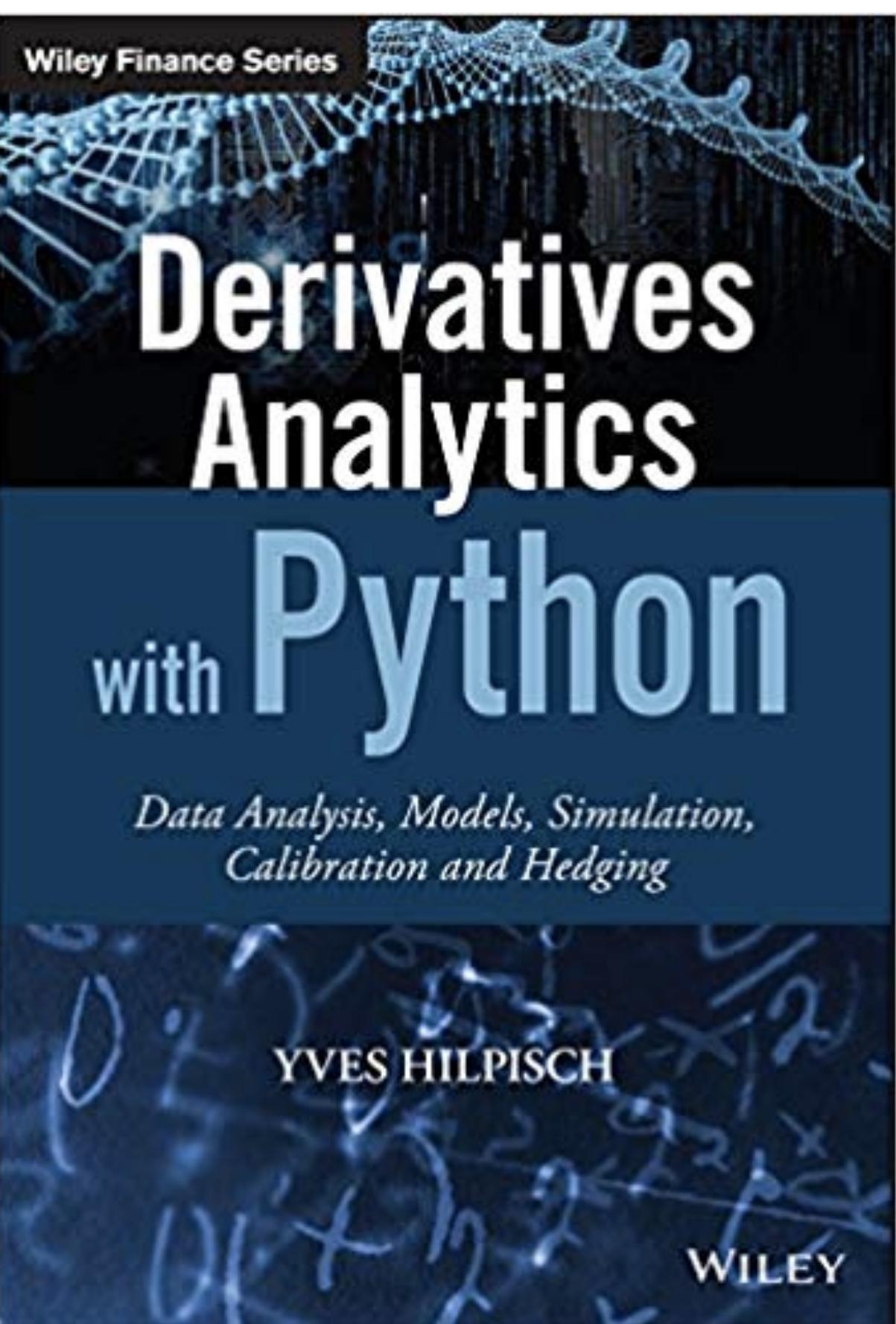
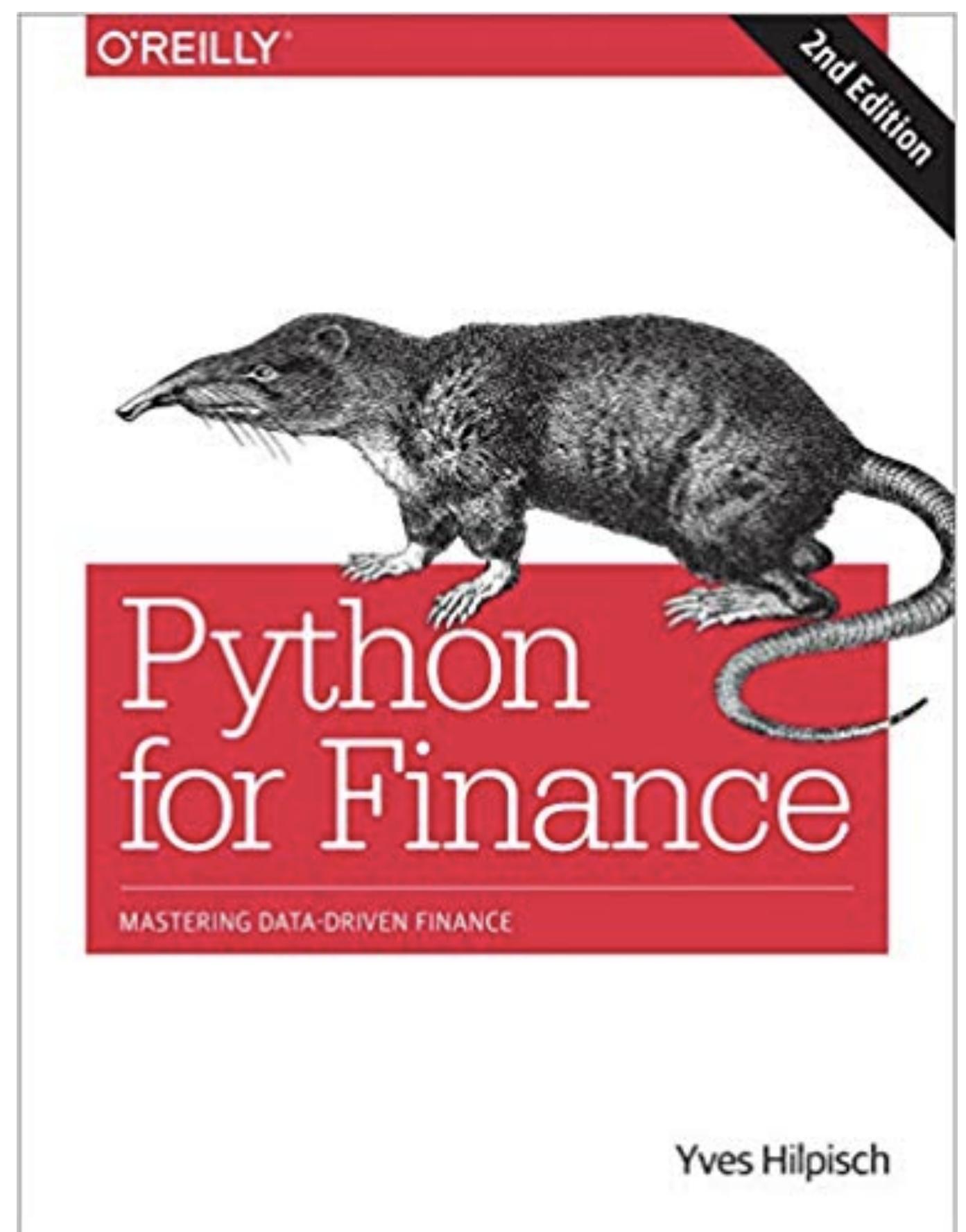
 QuantStart

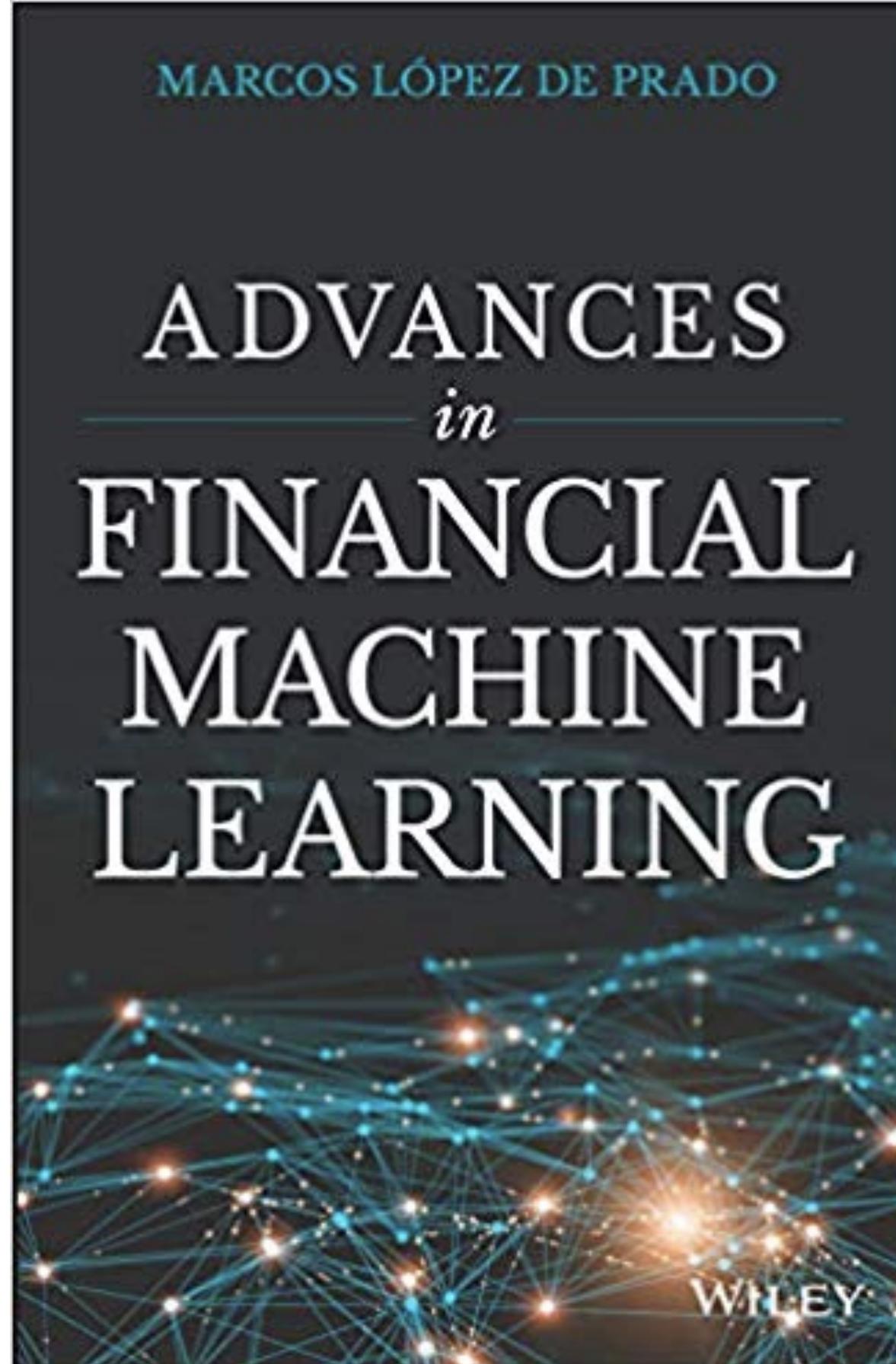
# Advanced Algorithmic Trading

Bayesian statistics, time series analysis and machine learning for profitable systematic trading strategies



Michael L. Halls-Moore, PhD.





Browse > Data Science > Machine Learning

## Machine Learning and Reinforcement Learning in Finance Specialization

Reinforce Your Career: Machine Learning in Finance. Extend your expertise of algorithms and tools needed to predict financial markets.

★★★★★ 3.8 (822 ratings)



Igor Halperin

Enroll

Starts Feb 29

Premium Access: €34/mo

Financial aid available

9,092 already enrolled

<b>id</b>	<b>era</b>	<b>feature1</b>	...	<b>feature310</b>	<b>target</b>
n2b2e3dd163cb422	era1	0.75	...	0.00	0.25
n177021a571c94c8	era1	1.00	...	0.25	0.75
n7830fa4c0cd8466	era1	0.25	...	1.00	0.00
nc584a184cee941b	era1	0.25	...	0.00	1.00
nc5ab8667901946a	era1	0.75	...	0.25	0.25
n84e624e4714a7ca	era1	0.00	...	0.75	1.00

<http://numer.ai/>

# ML Roadmap

- Coursera Andrew Ng Machine Learning
- Scikit-learn gallery examples
- Kaggle 1-2 tournaments
- Coursera Deep Learning specialization
- Tensorflow / PyTorch gallery examples
- Kaggle 1-2 tournaments
- Digging into theory (i.e. Bishop PRML book, Bayesian, causal, generative modeling)
- Digging into state of the art (<https://paperswithcode.com/> and conferences, papers)
- Learning business applications and direct AI benefits

# Let's connect :)

FB, IG: @rachnogstyle

Medium, Twitter: @alexrachnog

Linkedin: Alexandr Honchar