Algorithmic market efficiency? Machine Learning, Outperformance and Arbitrage Activity along the Cross-Section of Stock Returns

Master Thesis - MSc in International Finance - HEC Paris

Constantin Schesch

Supervisor : Augustin Landier

September 19, 2020

Table of contents

Introduction

Quick literature overview

1) Literature on the Efficient Markets Hypothesis

⇒ **Timmermann et al., "Efficient market hypothesis and forecasting" (2004, International Journal of Forecasting)** : argues that EMH is fundamentally a statistical forecasting question.

2) Literature on the Cross-Section Predictability of Stock Returns

⇒ **Green et al., "The Characteristics that Provide Independent Information about Average U.S. Monthly Stock Returns" (RFS, 2017)** : build a database of 94 factors that partially explain the cross-section of returns.

3) Literature on Machine Learning in Finance

⇒ **Gu et al., "Empirical Asset Pricing via Machine Learning" (RFS, 2020)** : show that basic ML methods achieve greater out-of-sample explanatory power and that ML-portfolios substantially outperform the market.

⇒ Usually takes a cross-sectional view, sees ML as 'learning' rational but latent risk factors.

4) Literature on Arbitrage and Short Interest

⇒ **Hanson et al., "The Growth and Limits of Arbitrage: Evidence from Short Interest" (RFS, 2014)** : measure arbitrage capital allocated to a strategy by cross-sectional regressions of short interest on the signal.

'Informational' vs 'algorithmic' market efficiency

A market is efficient with respect to information set I_t if it is impossible to make economic profits by trading on the basis of information set I_t. [Jensen, 1968]

$$\mathbb{E}_t(r_{t+1}|I_t) = 0 \tag{1}$$

where $r_{t+1} = R_{t+1}Q_{t+1}$ is the raw return R_{t+1} discounted by the stochastic factor Q_{t+1}. But $\mathbb{E}_t(r_{t+1}|I_t)$ is only defined as some $\mathcal{I}$-measurable function $g \in \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$ such that

$$\forall f \in \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R}) \qquad \mathbb{E}\big(g(I_t)f(I_t)\big) = \mathbb{E}\big(r_{t+1}f(I_t)\big) \tag{2}$$

<u>Not all functions f are available to arbitrageurs:</u> they are only as good as the algorithms they can use to analyse information and predict returns. We call $\mathcal{A}_t \subset \mathcal{M}(\mathfrak{I}, \mathcal{I}, \mathbb{R})$ the set of algorithms available at time t and say that a market is <u>algorithmically efficient</u> with respect to $\mathcal{A}_t$ if:

$$\forall a \in \mathcal{A}_t \qquad \mathbb{E}_t\big(r_{t+1}\; a(I_t)\big) = 0 \tag{3}$$

'Informational' vs 'algorithmic' market efficiency

Market Efficiency Matrix		Algorithmic Market Efficiency		
		Semi-strong: public algorithms	**Strong: public and private algorithms**	**Super-strong: discovered and undiscovered algorithms**
Informational Market Efficiency	**Weak: price information**	Prices reflect all information from past prices, as analyzed by all publicly available algorithms.	Prices reflect all information from past prices, as analyzed by all public and privately discovered algorithms.	Prices reflect all information from past prices, as analyzed by all discovered and undiscovered algorithms.
	Semi-strong: public information	Prices reflect all public information, as analyzed by all publicly available algorithms.	Prices reflect all public information, as analyzed by all public and privately discovered algorithms.	Prices reflect all public information, as analyzed by all discovered and undiscovered algorithms.
	Strong: public and private information	Prices reflect all public and private information, as analyzed by all publicly available algorithms.	Prices reflect all public and private information, as analyzed by all public and privately discovered algorithms.	Prices reflect all public and private information, as analyzed by all discovered and undiscovered algorithms.

Structure of the Machine Learning Algorithm

Data Sources

Figure 1: Data Sources

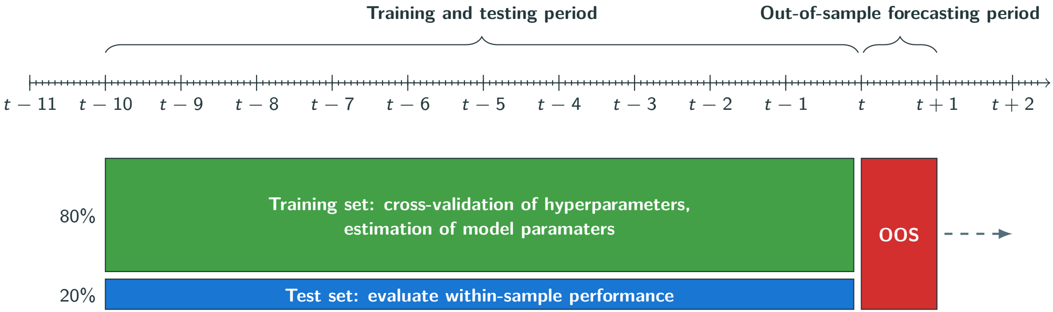
Data Source	Retrieved on	Observations	Variables of interest
Factors from Green et al. (2017)	Sept. 2019	3,760,208	94 factors (see Annex)
CRSP Database	Oct. 2019	4,514,430	*ret, mktcap, shrout*
Fama-French factors	June 2019	1,119	*smb, hml, rmw, cma, mom*
Amit Goyal's macro predictors	Nov. 2019	1,776	16 macro factors
Compustat Supplemental Short Interest File	Feb. 2020	2,221,113	*shortintadj*
CRSP / Comp. Merged Security Monthly	Feb. 2020	6,130,813	*cshtrm*

$\Rightarrow$ Drop observations with missing returns

$\Rightarrow$ Standardize factor database

$\Rightarrow$ Merging main database (indexed by *year* and *Permno*) with Compustat Supplemental Short Interest File (indexed by *gvkey* but also *iid*) is far from perfect, in part because of the increasing number of observations (see later discussion)

Algorithm: Machine Learning Component, Loop Component



Code example : the 'machine learning' part is very simple!

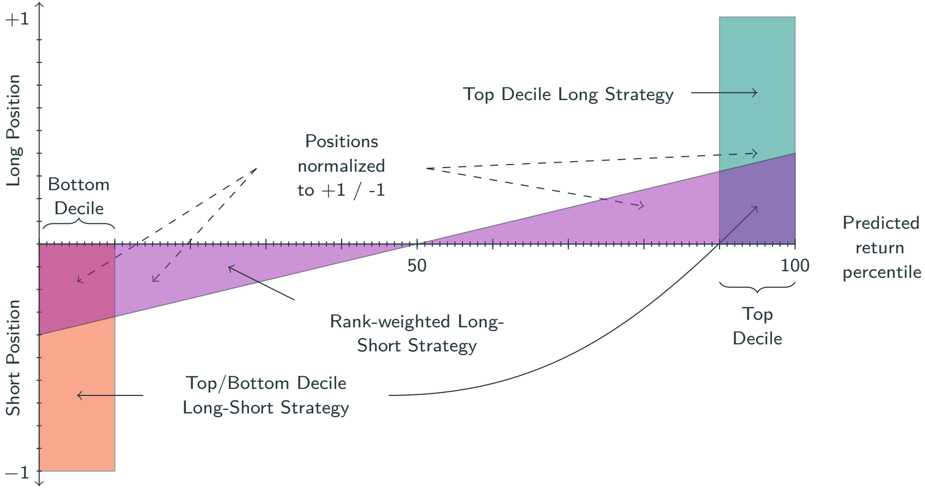
```
nn3_parameters = {'hidden_layer_sizes': [(16,8,4), (32,16,8), (64,32,16)],
          'activation': ['logistic'],
          'learning_rate': ['invscaling'],
          'power_t': [0.1, 0.5, 0.9],
          'learning_rate_init': np.logspace(-4,0,5),
          'max_iter': [250],
          'warm_start': [True]}

nn3_estimator = GridSearchCV(MLPRegressor(), nn3_parameters, cv=2, return_train_score=True, n_jobs=-1).fit(prevfacs_train, prevrets_train)

testscores = nn3_estimator.cv_results_['mean_test_score']
trainscores = nn3_estimator.cv_results_['mean_train_score']
```

Portfolio Construction Component

We test three strategies: (i) a top-decile long strategy (ii) a top/bottom decile long/short strategy (iii) a rank-weighted long/short strategy (with $w_i^t = \left[rank_i - \frac{n_t+1}{2}\right] \times W$).

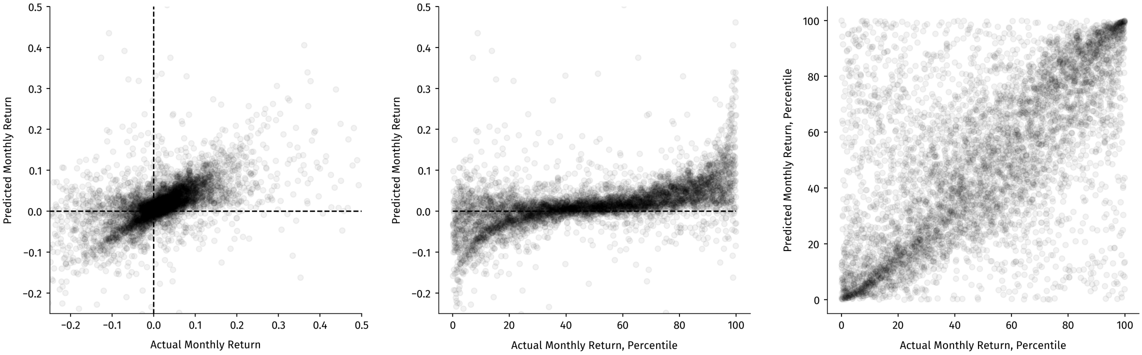


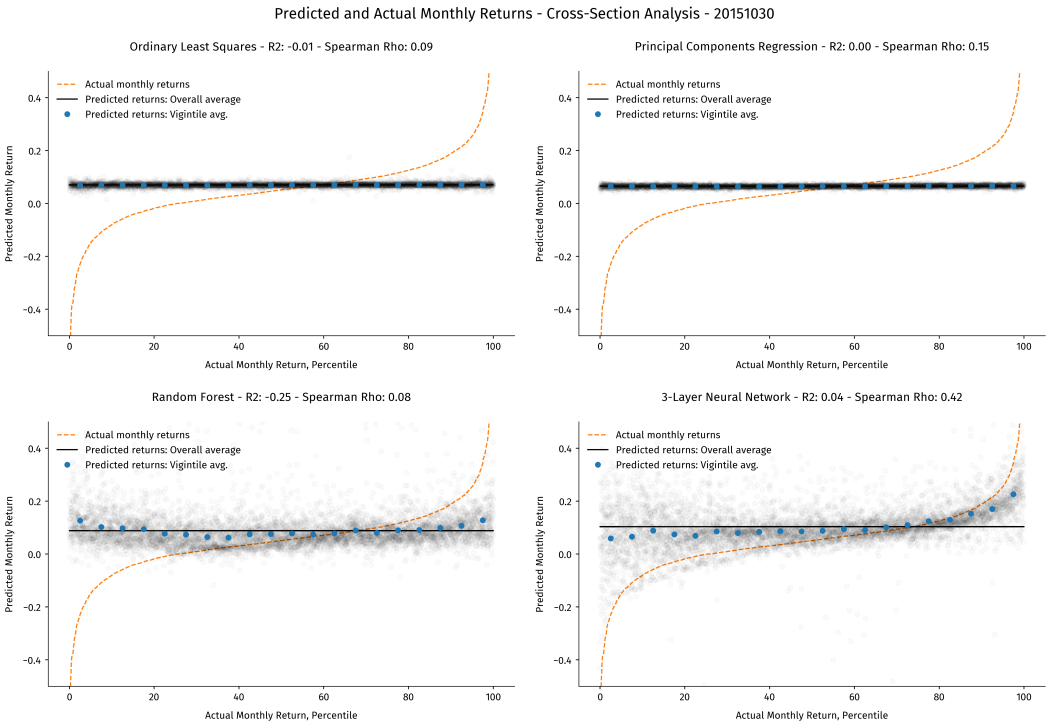
Predictive Performance of Machine Learning Models

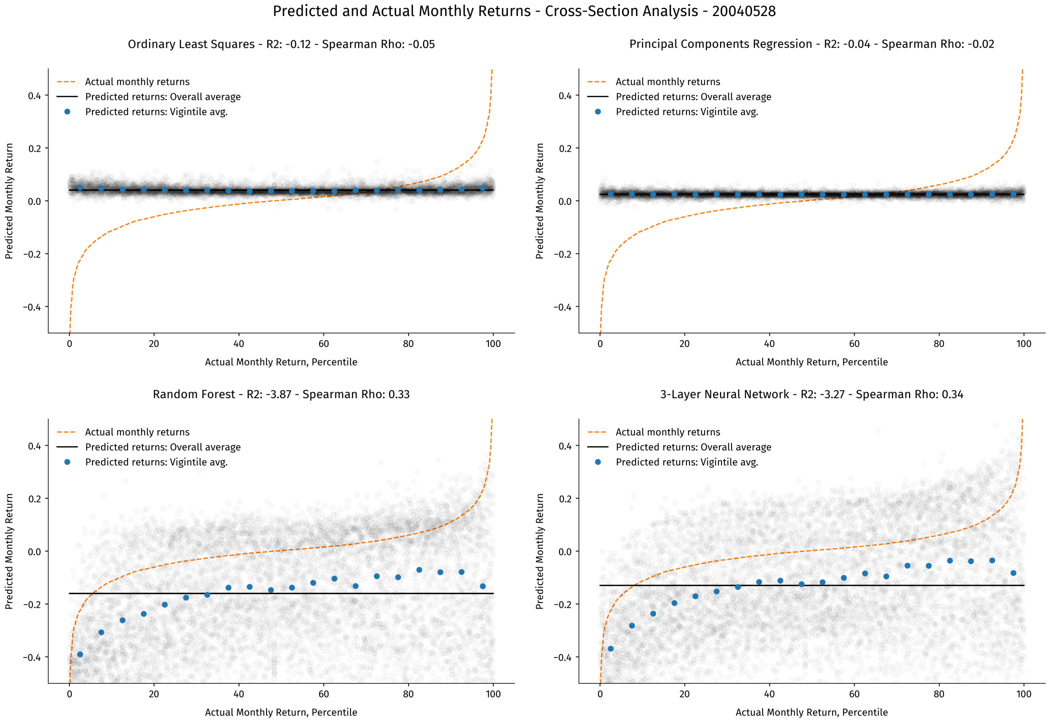
Predictive Performance of Machine Learning Models

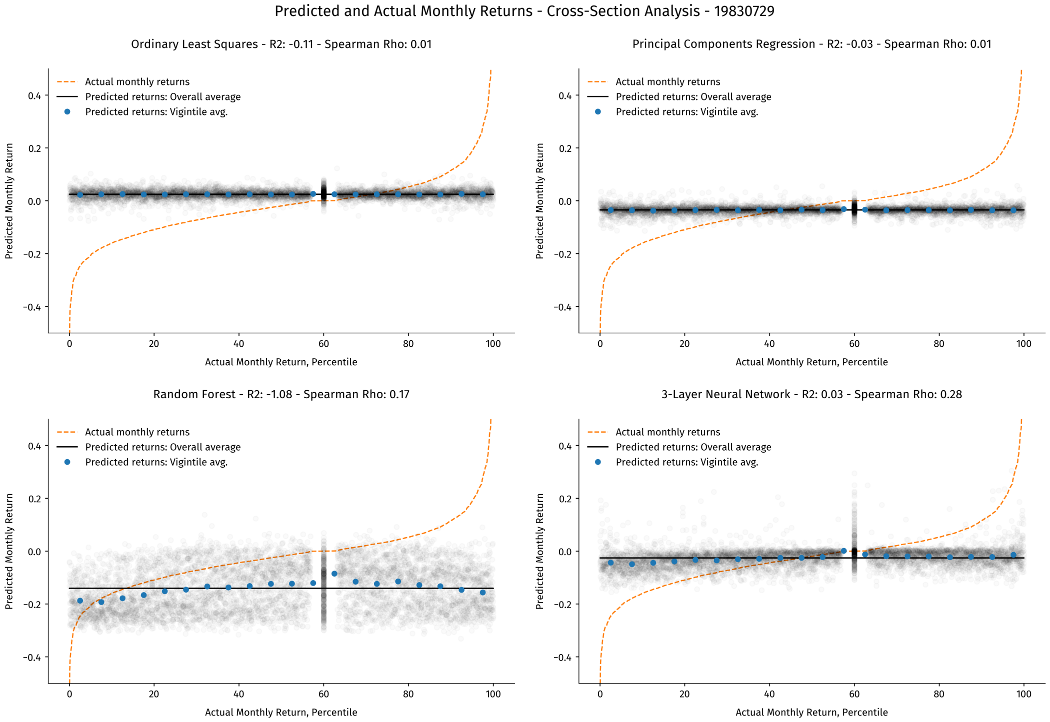
Figure 2: Predictive Performance of Machine Learning Models - Cross Section Analysis

Predicted and Actual Monthly Returns - 3-Layer Neural Network - December 2016




Predicted and Actual Monthly Returns - Cross-Section Analysis - 20151030
Ordinary Least Squares - R2: -0.01 - Spearman Rho: 0.09
Principal Components Regression - R2: 0.00 - Spearman Rho: 0.15
Random Forest - R2: -0.25 - Spearman Rho: 0.08
3-Layer Neural Network - R2: 0.04 - Spearman Rho: 0.42
Actual monthly returns
Predicted returns: Overall average
Predicted returns: Vigintile avg.
Predicted Monthly Return
Actual Monthly Return, Percentile

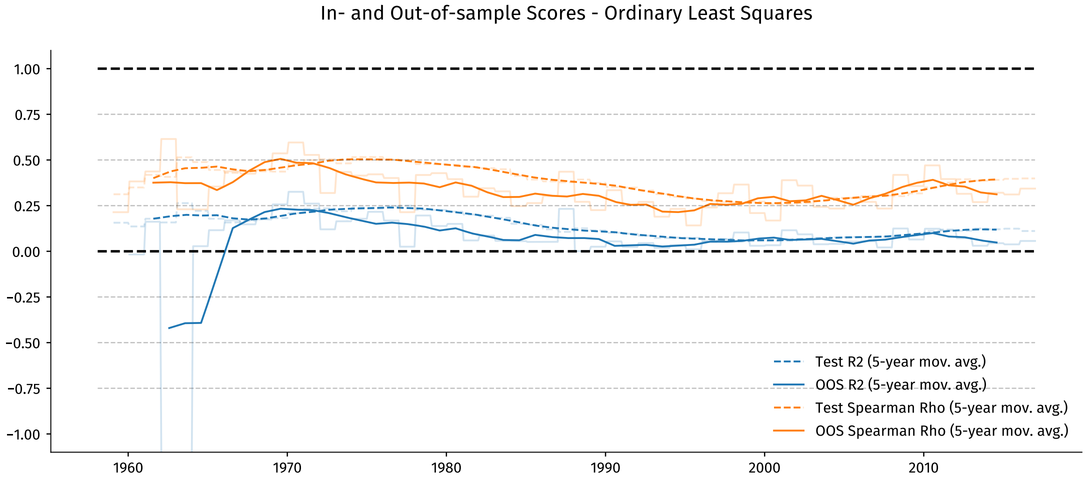

Predicted and Actual Monthly Returns - Cross-Section Analysis - 20040528
Ordinary Least Squares - R2: -0.12 - Spearman Rho: -0.05
Principal Components Regression - R2: -0.04 - Spearman Rho: -0.02
Random Forest - R2: -3.87 - Spearman Rho: 0.33
3-Layer Neural Network - R2: -3.27 - Spearman Rho: 0.34
Actual monthly returns
Predicted returns: Overall average
Predicted returns: Vigintile avg.
Predicted Monthly Return
Actual Monthly Return, Percentile
0.4
0.2
0.0
−0.2
−0.4
0
20
40
60
80
100



Predicted and Actual Monthly Returns - Cross-Section Analysis - 19830729
Ordinary Least Squares - R2: -0.11 - Spearman Rho: 0.01
Principal Components Regression - R2: -0.03 - Spearman Rho: 0.01
Random Forest - R2: -1.08 - Spearman Rho: 0.17
3-Layer Neural Network - R2: 0.03 - Spearman Rho: 0.28
Actual monthly returns
Predicted returns: Overall average
Predicted returns: Vigintile avg.
Predicted Monthly Return
Actual Monthly Return, Percentile
0.4
0.2
0.0
−0.2
−0.4
0
20
40
60
80
100

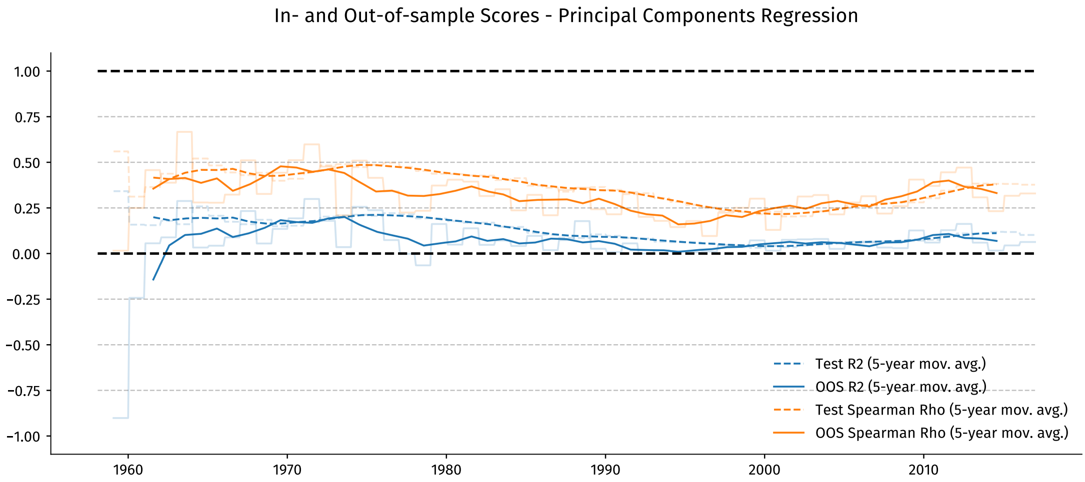
Predictive Performance of Machine Learning Models

Figure 3: Predictive Performance of Machine Learning Models - Time Series Analysis



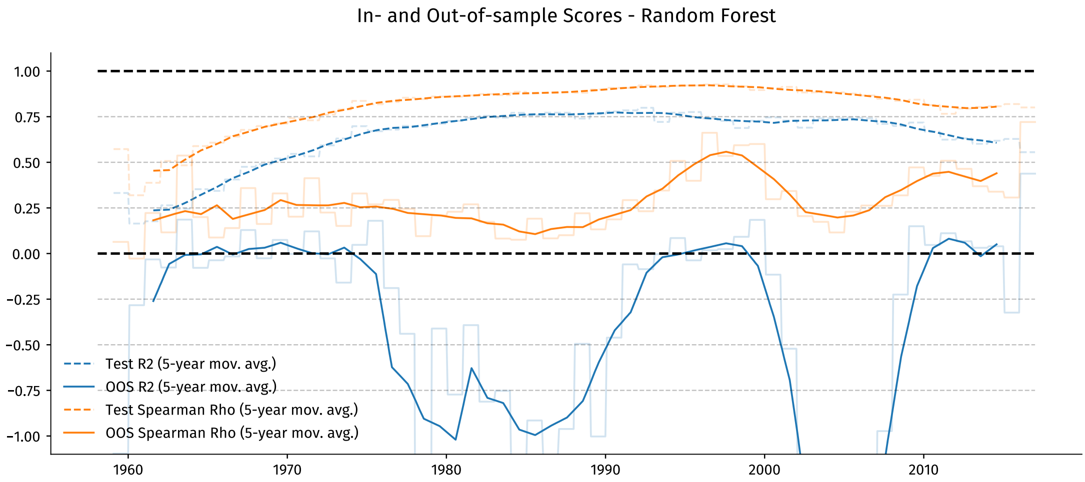
Predictive Performance of Machine Learning Models

Figure 4: Predictive Performance of Machine Learning Models - Time Series Analysis



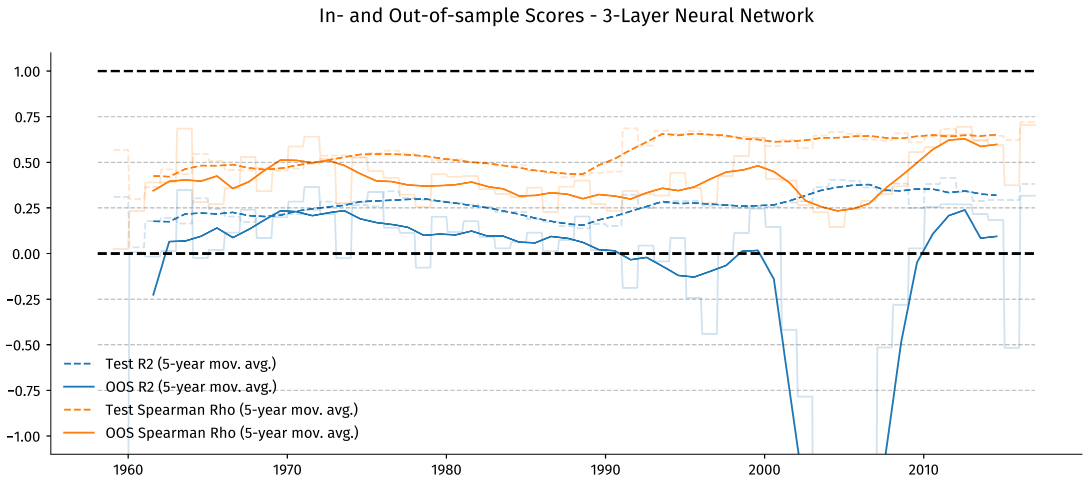
Predictive Performance of Machine Learning Models

Figure 5: Predictive Performance of Machine Learning Models - Time Series Analysis



Predictive Performance of Machine Learning Models

Figure 6: Predictive Performance of Machine Learning Models - Time Series Analysis



Machine Learning Portfolio Performances

Machine Learning Portfolio Performances

Figure 7: Top Decile Long Strategies - Performance of Machine Learning Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.56	1.03	45	-19.0	-32.0	0.80	(9.15)	0.88	(7.89)	0.84	(8.41)	0.88	(6.73)
Lasso	2.39	1.21	104	-20.2	-33.0	1.56	(4.20)	1.51	(5.77)	1.42	(5.65)	1.47	(5.36)
Ridge	2.36	1.20	104	-20.2	-32.8	1.54	(4.14)	1.49	(5.71)	1.40	(5.62)	1.44	(5.30)
Enet	2.38	1.21	104	-20.2	-33.1	1.56	(4.23)	1.51	(5.86)	1.42	(5.73)	1.46	(5.44)
PCR	2.02	1.06	89	-22.0	-35.1	1.18	(6.02)	1.11	(7.10)	1.02	(6.15)	1.07	(5.69)
PLS	3.00	1.26	89	-30.4	-46.4	2.18	(4.78)	2.16	(4.23)	2.24	(3.44)	2.50	(4.18)
Tree	3.33	1.78	162	-21.8	-32.4	2.49	(3.92)	2.45	(3.75)	2.54	(3.91)	2.56	(3.84)
Forest	4.81	2.15	151	-18.0	-27.3	4.02	(3.73)	3.94	(3.55)	4.03	(3.60)	4.11	(3.60)
GBRT	4.24	1.89	143	-11.9	-27.9	3.44	(6.66)	3.27	(7.14)	3.60	(6.96)	3.36	(6.96)
NN1	4.15	2.05	79	-16.7	-30.1	3.36	(2.82)	3.35	(2.80)	3.52	(2.90)	3.45	(2.85)
NN2	5.43	2.40	101	-16.2	-32.1	4.69	(3.28)	4.73	(3.30)	4.83	(3.34)	4.85	(3.33)
NN3	5.80	2.26	109	-16.1	-29.1	5.04	(3.39)	4.94	(3.17)	5.20	(3.36)	5.02	(3.14)
NN5	3.47	1.61	108	-14.7	-26.0	2.64	(10.78)	2.51	(9.81)	2.74	(9.81)	2.54	(8.64)
NN10	3.99	1.79	127	-21.2	-33.3	3.16	(11.15)	3.09	(10.49)	3.25	(10.89)	3.17	(9.92)

Machine Learning Portfolio Performances

Figure 8: Top/bottom Decile Long/short Strategies - Performance of ML Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	1.46	0.84	62	-21.8	-36.2	1.08	(3.32)	1.30	(3.47)	1.11	(3.87)	1.33	(3.19)
Lasso	0.67	0.26	106	-25.2	-98.2	0.68	(1.19)	0.86	(1.29)	0.80	(1.38)	1.26	(1.81)
Ridge	0.71	0.29	107	-25.3	-98.3	0.72	(1.24)	0.89	(1.33)	0.83	(1.44)	1.29	(1.84)
Enet	0.66	0.26	106	-25.1	-98.2	0.68	(1.19)	0.85	(1.29)	0.78	(1.36)	1.26	(1.81)
PCR	-0.30	-0.21	84	-40.1	-99.7	-0.18	(-0.37)	-0.03	(-0.06)	-0.08	(-0.16)	0.37	(0.61)
PLS	-1.30	-0.57	65	-32.5	-100.0	-0.63	(-1.87)	-0.63	(-1.73)	-0.91	(-2.58)	-0.65	(-1.87)
Tree	4.67	1.88	164	-21.1	-36.2	4.20	(3.71)	4.49	(3.90)	4.59	(4.04)	4.81	(4.25)
Forest	6.36	2.11	150	-30.3	-44.9	5.87	(3.17)	6.19	(3.31)	6.24	(3.38)	6.51	(3.48)
GBRT	5.07	1.01	168	-30.0	-77.0	5.28	(2.49)	5.32	(2.56)	5.75	(2.45)	5.76	(2.66)
NN1	6.27	2.17	85	-22.0	-41.2	5.66	(2.97)	6.02	(3.18)	6.18	(3.22)	6.22	(3.20)
NN2	6.24	1.97	101	-24.7	-87.3	5.84	(2.42)	6.16	(2.57)	6.07	(2.55)	6.41	(2.66)
NN3	6.18	1.84	108	-25.0	-93.8	5.81	(2.30)	6.07	(2.45)	6.39	(2.53)	6.33	(2.49)
NN5	2.80	1.09	126	-29.4	-37.0	2.24	(4.52)	2.43	(4.32)	2.62	(5.40)	2.58	(4.71)
NN10	2.74	1.13	131	-32.8	-41.4	2.20	(4.09)	2.42	(4.06)	2.49	(4.39)	2.52	(4.60)

Figure 9: Reference from Gu et al. (2019) ⇒ qualitatively similar results, but much higher alpha!

Table 8: Drawdowns, Turnover, and Risk-adjusted Performance of Machine Learning Portfolios

	OLS-3 +H	PLS	PCR	ENet +H	GLM +H	RF	GBRT +H	NN1	NN2	NN3	NN4	NN5
						Drawdowns and Turnover (Value Weighted)						
Max DD(%)	69.60	41.13	42.17	60.71	37.09	52.27	48.75	61.60	55.29	30.84	51.78	57.52
Max 1M Loss(%)	24.72	27.40	18.38	27.40	15.61	26.21	21.83	18.59	37.02	30.84	33.03	38.95
Turnover(%)	58.20	110.87	125.86	151.59	145.26	133.87	143.53	121.02	122.46	123.50	126.81	125.37
						Drawdowns and Turnover (Equally Weighted)						
Max DD(%)	84.74	32.35	31.39	33.70	21.01	46.42	37.19	18.25	25.81	17.34	14.72	21.78
Max 1M Loss(%)	37.94	32.35	22.33	32.35	15.74	34.63	22.34	12.79	25.81	12.50	9.01	21.78
Turnover(%)	57.24	104.47	118.07	142.78	137.97	120.29	134.24	112.35	112.43	113.76	114.17	114.34
						Risk-adjusted Performance (Value Weighted)						
Mean Ret.	0.94	1.02	1.22	0.60	1.06	1.62	0.99	1.81	1.92	1.97	2.26	2.12
FF5+Mom α	0.39	0.24	0.62	-0.23	0.38	1.20	0.66	1.20	1.33	1.52	1.76	1.43
t-stats	2.76	1.09	2.89	-0.89	1.68	3.95	3.11	4.68	4.74	4.92	6.00	4.71
R^2	78.60	34.95	39.11	28.04	30.78	13.43	20.68	27.67	25.81	20.84	20.47	18.23
IR	0.54	0.21	0.57	-0.17	0.33	0.77	0.61	0.92	0.93	0.96	1.18	0.92
						Risk-adjusted Performance (Equally Weighted)						
Mean Ret.	1.34	2.08	2.45	2.11	2.31	2.38	2.14	2.91	3.31	3.27	3.33	3.09
FF5+Mom α	0.83	1.40	1.95	1.32	1.79	1.88	1.87	2.60	3.07	3.02	3.08	2.78
$t(\alpha)$	6.64	5.90	9.92	4.77	8.09	6.66	8.19	10.51	11.66	11.70	12.28	10.68
R^2	84.26	26.27	40.50	20.89	21.25	19.91	11.19	13.98	10.60	9.63	11.57	14.54
IR	1.30	1.15	1.94	0.93	1.58	1.30	1.60	2.06	2.28	2.29	2.40	2.09

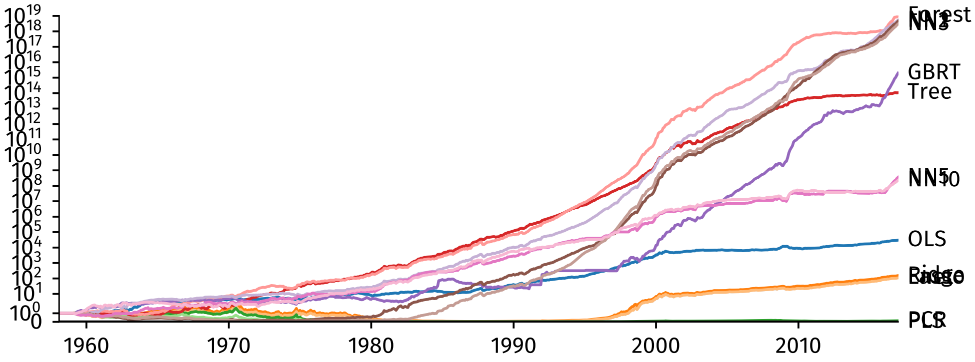
Machine Learning Portfolio Performances

Figure 10: Rank-weighted Long/short Strategies - Performance of ML Portfolios over 1958-2016

	1958-2016												
	Avg. Ret.	Sharpe Ratio	Avg. Turn.	Max. Loss	Max. DD	CAPM-alpha	(t-stat)	FF3-alpha	(t-stat)	CH4-alpha	(t-stat)	FF5-alpha	(t-stat)
OLS	0.84	0.47	52	-11.4	-44.6	0.56	(1.82)	0.76	(2.18)	0.60	(2.27)	0.83	(2.16)
Lasso	0.42	0.11	67	-20.0	-91.7	0.31	(0.81)	0.51	(1.13)	0.41	(1.05)	0.79	(1.60)
Ridge	0.43	0.12	67	-20.1	-91.6	0.32	(0.82)	0.52	(1.14)	0.41	(1.07)	0.79	(1.60)
Enet	0.41	0.11	67	-20.0	-92.1	0.31	(0.78)	0.50	(1.11)	0.40	(1.01)	0.78	(1.57)
PCR	0.01	-0.16	52	-32.4	-97.3	-0.11	(-0.34)	0.01	(0.04)	-0.01	(-0.03)	0.35	(0.82)
PLS	-0.90	-0.63	37	-27.0	-99.9	-0.62	(-2.20)	-0.52	(-1.65)	-0.76	(-2.55)	-0.46	(-1.53)
Tree	3.27	1.86	100	-10.9	-23.1	2.63	(3.86)	2.80	(4.05)	2.86	(4.26)	2.89	(4.24)
Forest	3.60	1.79	85	-27.0	-35.4	2.93	(3.51)	3.15	(3.73)	3.13	(3.84)	3.27	(3.92)
GBRT	0.37	0.10	95	-19.0	-94.8	-0.04	(-0.08)	0.11	(0.22)	0.15	(0.30)	0.43	(0.74)
NN1	3.64	1.79	63	-21.0	-37.1	2.93	(3.39)	3.19	(3.67)	3.19	(3.79)	3.36	(3.74)
NN2	3.35	1.59	65	-18.9	-87.2	2.74	(2.52)	2.95	(2.76)	2.96	(2.78)	3.11	(2.89)
NN3	3.24	1.52	70	-19.8	-86.7	2.67	(2.38)	2.89	(2.65)	2.93	(2.66)	3.09	(2.78)
NN5	2.12	1.17	83	-18.1	-28.6	1.56	(3.52)	1.75	(3.51)	1.80	(3.93)	1.84	(3.97)
NN10	2.30	1.24	90	-20.4	-44.4	1.77	(3.52)	1.94	(3.60)	2.00	(3.95)	2.06	(3.93)

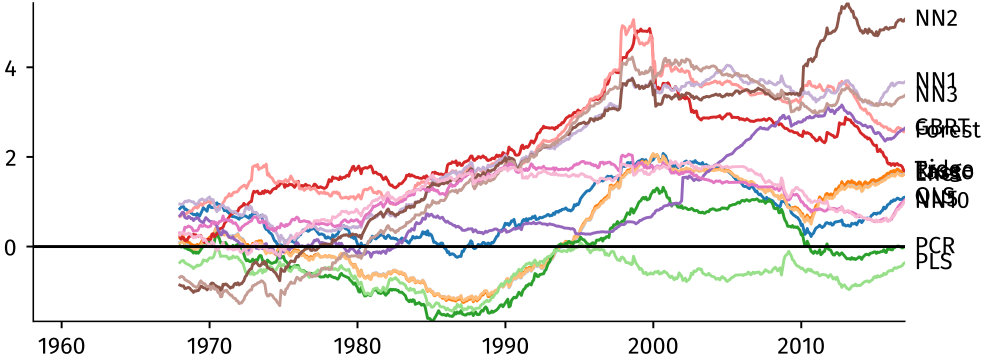
Machine Learning Portfolio Performances

Value-weighted Compound Returns - Top/Bottom Decile Long/Short Strategies

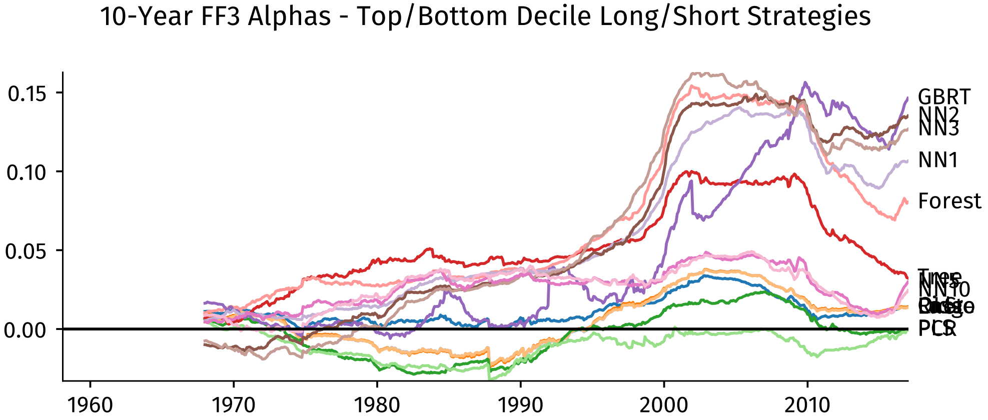


Machine Learning Portfolio Performances

10-Year Sharpe Ratios - Top/Bottom Decile Long/Short Strategies



Machine Learning Portfolio Performances

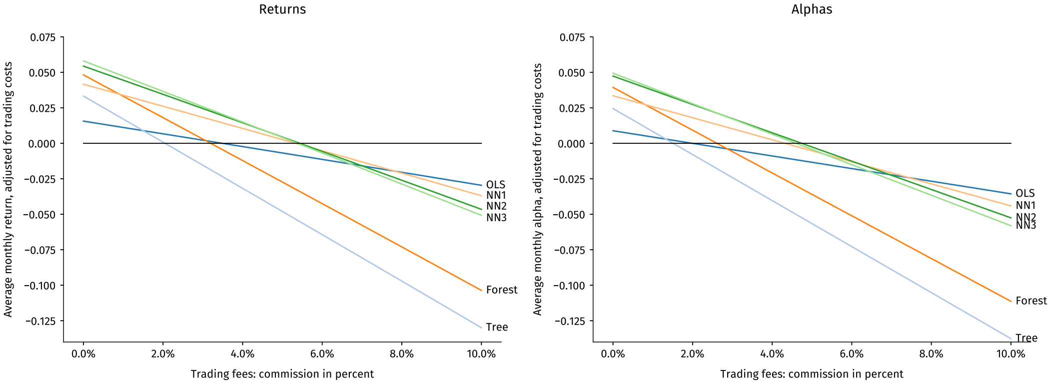


Robustness Checks on Machine Learning Portfolio Performances

⇒ **Small companies** : we restrict our investment universe to the **top 100 and top 1000 companies** by market cap, performance stays broadly the same and alphas remain significant.

⇒ **Transaction costs** : we compute returns after proportional **trading fees**; our method is crude and not very conservative; however, our strategies were not optimized to reduce turnover, which could yield much higher post-transaction-cost returns.

Figure 11: Robustness Check : Performance of ML Portfolios after Transaction Costs



Machine Learning and Short Interest Activity

Partial equilibrium approach: one stock, uninformed agents and informed agents (price-takers)

$$r_{t+1} = \mu + a(\mathbf{z}_t) + \varepsilon_{t+1} \tag{4}$$

Informed agents know a and markets are algorithmically efficient w.r.t. $\mathcal{A}_t^{uninf.} \subset \mathcal{A}_t^{inf.} \setminus \{a\}$.

$$q_t^u = \frac{1}{\gamma_u} \times \frac{\mu}{\sigma^2_{a(\mathbf{z}_t)} + \sigma^2_\varepsilon} \qquad q_t^i = \frac{1}{\gamma_u} \times \frac{\mu + a(\mathbf{z}_t)}{\sigma^2_\varepsilon} \tag{5}$$

General equilibrium approach: many stocks, share $k \in [0, 1]$ of informed agents

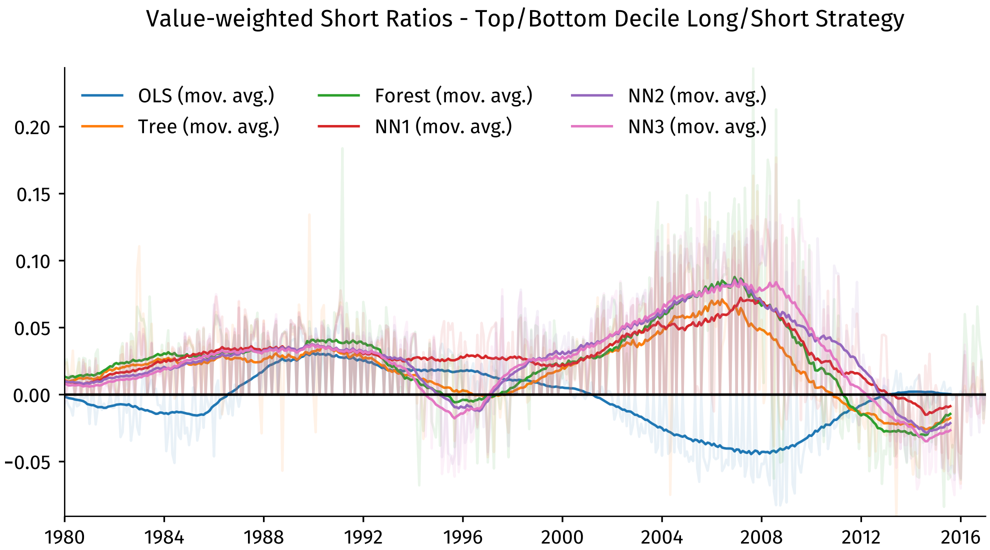
$$E^i(\mathbf{r}_{t+1}) = E^u(\mathbf{r}_{t+1}) + a(\mathbf{z}_t) \qquad \mathbb{V}_t^i(\mathbf{r}_{t+1}) = \mathbb{V}_t^u(\mathbf{r}_{t+1}) \qquad \mathbf{w}' a(\mathbf{z}_t) = 0 \tag{6}$$

$$\Rightarrow \qquad \mathbb{E}^i(\mathbf{r}_{t+1}) = (1-k) \times a(\mathbf{z}_t) + \frac{\text{Cov}(\mathbf{r}_{t+1}, r_m)}{\mathbb{V}(r_m)} \times \mathbb{E}(r_m) \tag{7}$$

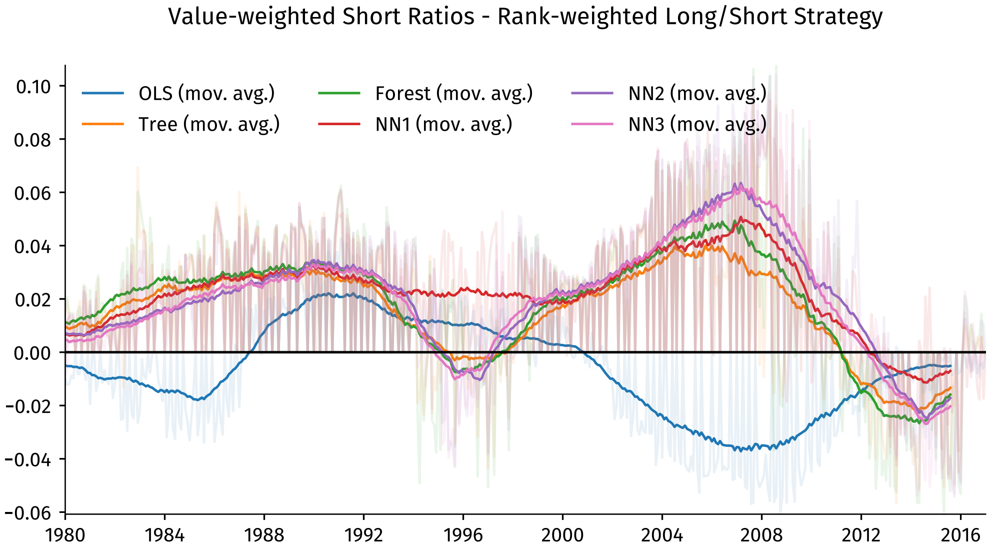
$$\Rightarrow \qquad SR^i = -k \times \mathbf{w}^{-1} - k(1-k) \times \frac{1}{\gamma} \times \mathbb{V}(r_{t+1})^{-1} \times a(\mathbf{z}_t) \times \mathbf{w}^{-1} \tag{8}$$

$\Longrightarrow$ In both, we expect a negative correlation between Short Ratios and the signal!

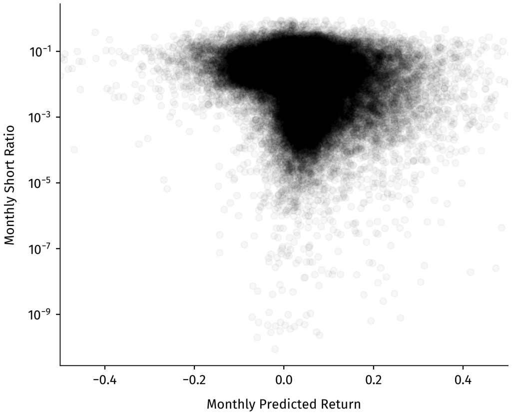
Short Interest Activity around Machine Learning Portfolios

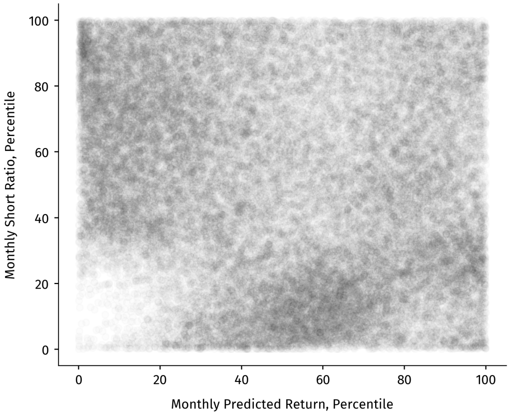


Short Interest Activity around Machine Learning Portfolios

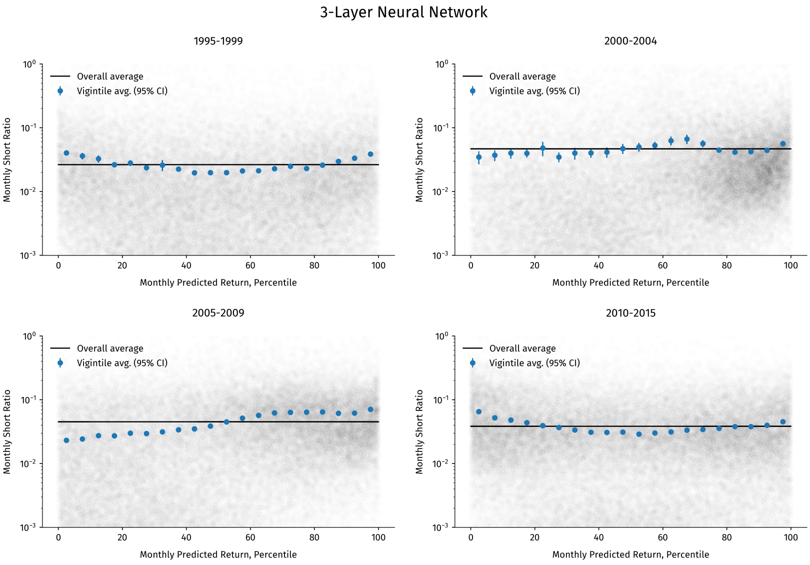


Short Interest Activity and Machine Learning Predictions

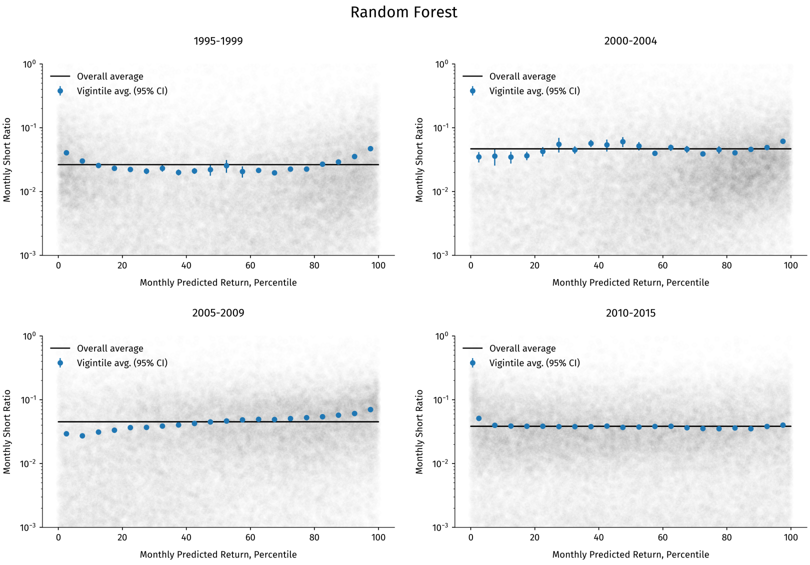




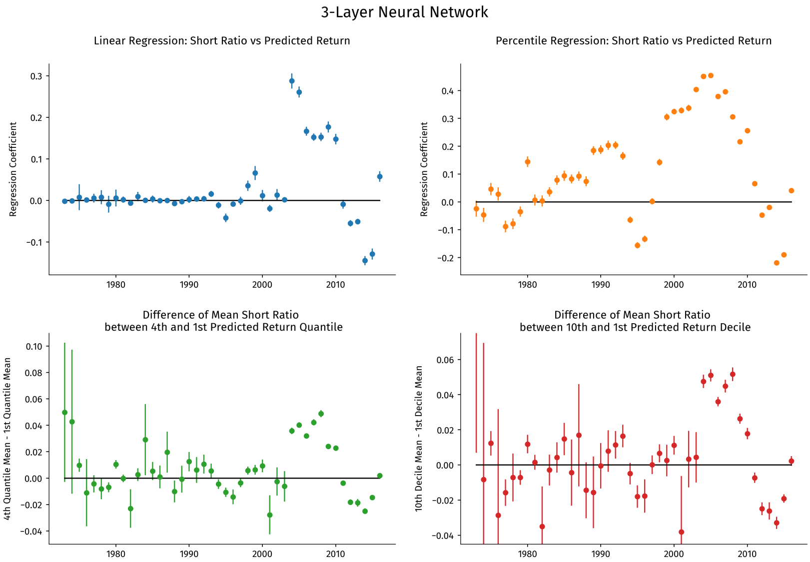
Short Interest Activity and Machine Learning Predictions



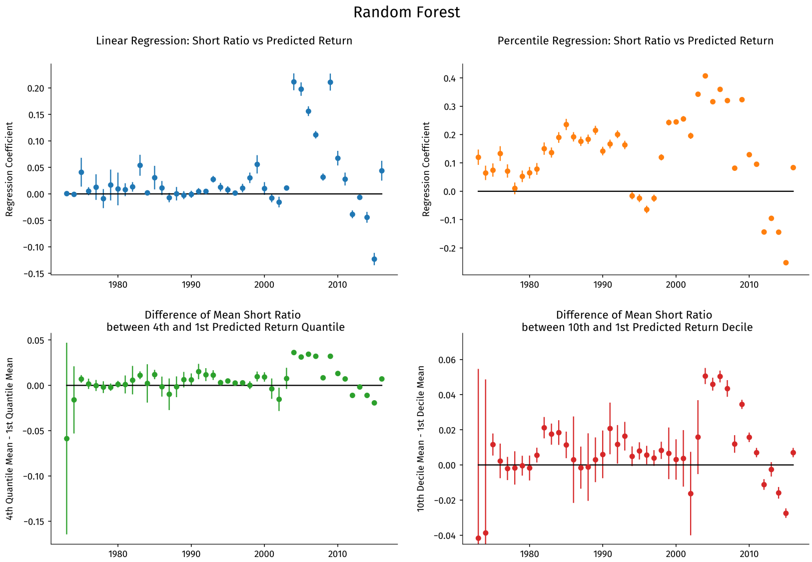
Short Interest Activity and Machine Learning Predictions



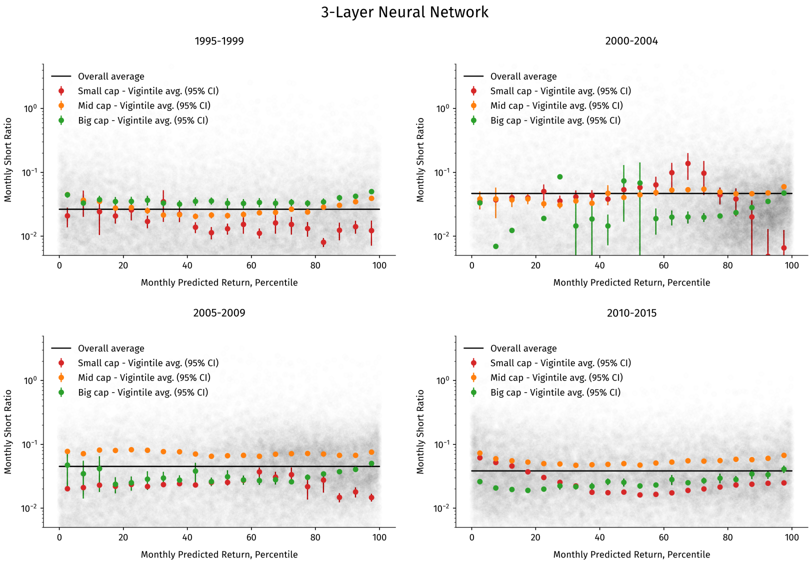
Short Interest Activity and Machine Learning Predictions



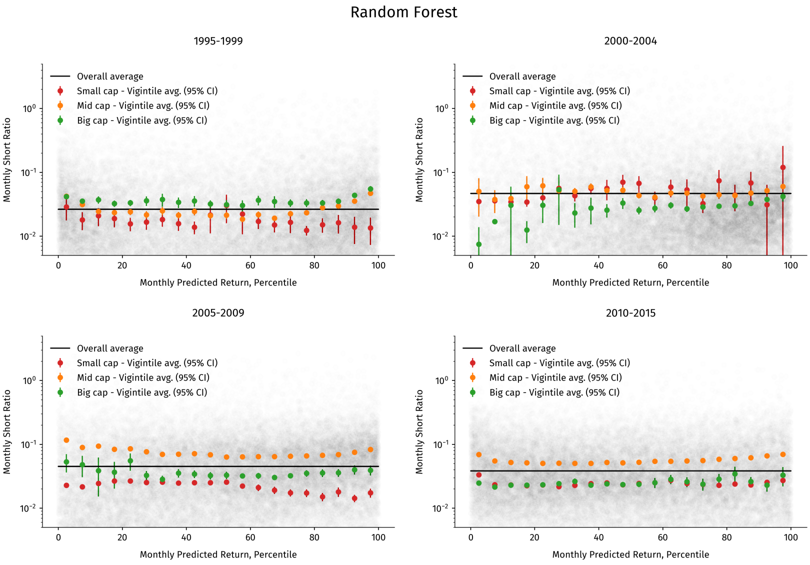
Short Interest Activity and Machine Learning Predictions



Sorting by Market Cap does not Produce a Clear Pattern Either



Sorting by Market Cap does not Produce a Clear Pattern Either



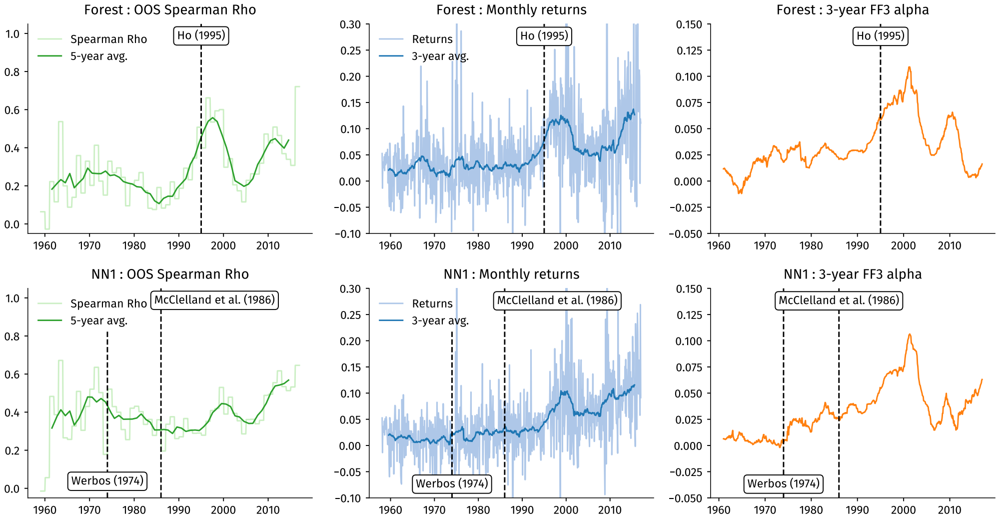
Is there Post-publication Alpha Decay in Machine Learning?

Is there Post-publication Alpha Decay in Machine Learning?

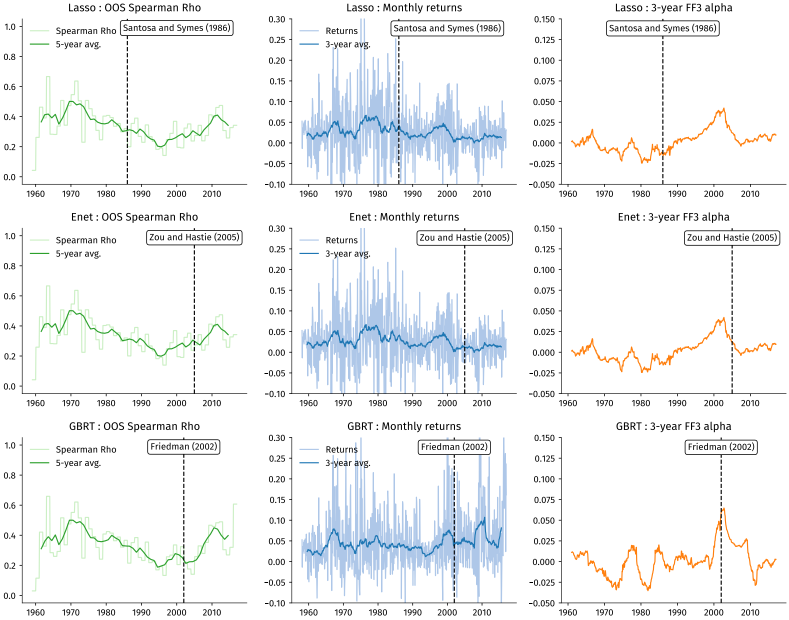
Figure 12: Publication Dates of Machine Learning Methods

Method	First Publication	Notes
OLS	Legendre (1805)	
Lasso	Santosa and Symes (1986), Tibshirani (1996)	Developed independently
Ridge	Tikhonov (1943), Hoerl (1962)	Invention, dissemination into statistics
Enet	Zou and Hastie (2005)	
PCR	Pearson (1901)	First publication for PCA
PLS	Wold (1966)	
Tree	Belson (1959)	See Loh (2014) for a detailed history
Forest	Ho (1995)	
GBRT	Friedman (2002)	
NNs	McCulloch and Pitts (1943)	Initial idea for ANNs
	Werbos (1974)	First suggested to use backpropagation
	McClelland et al. (1986)	Popularization of backpropagation for NNs
	LeCun et al. (1989)	Popularization of backpropagation for CNNs

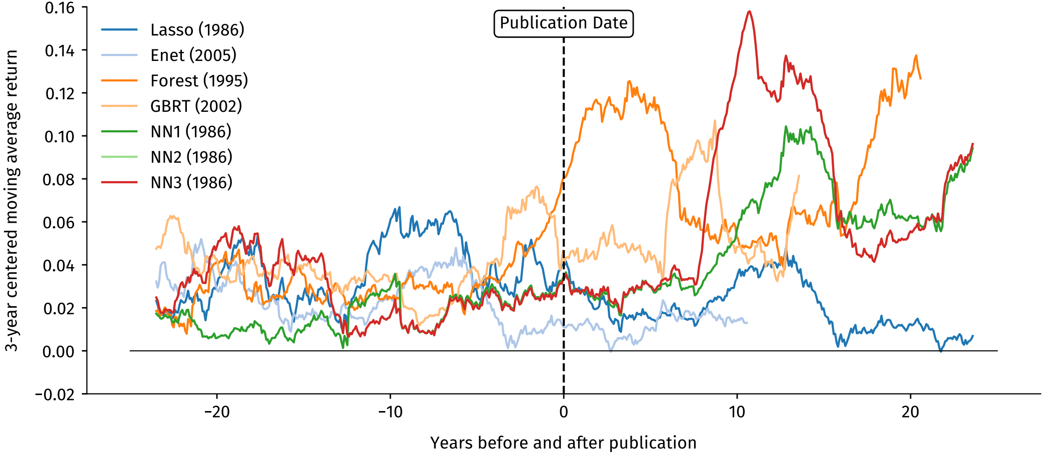
Is there Post-publication Alpha Decay in Machine Learning? (No.)



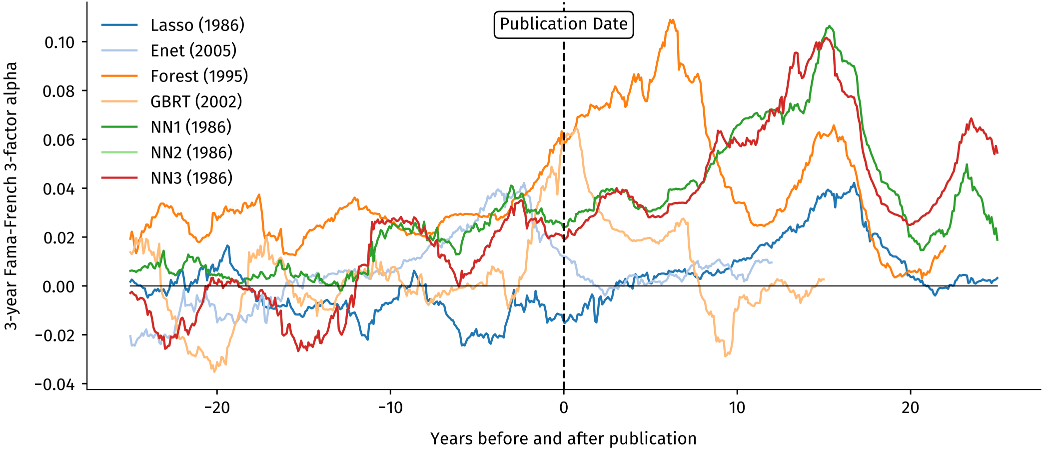
Is there Post-publication Alpha Decay in Machine Learning? (No.)



Is there Post-publication Alpha Decay in Machine Learning? (No.)



Is there Post-publication Alpha Decay in Machine Learning? (No.)



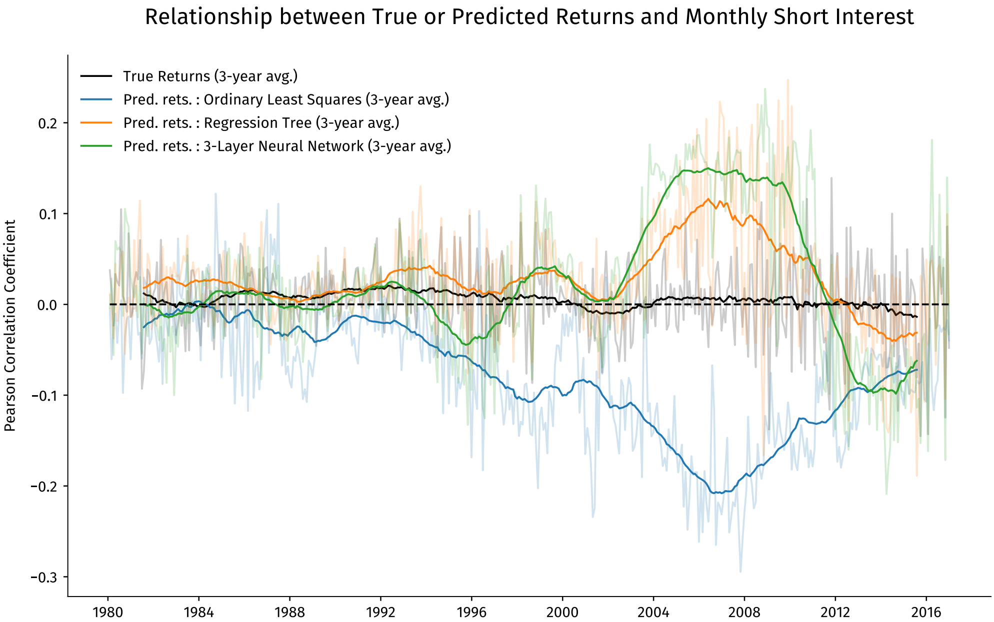
Post-Publication Decline of 3-Year FF3 Alpha for Top/Bottom Decile Strategies

	Lasso - 3Y-FF3-Alpha			Enet - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	-2.14**	-2.14***	-2.60*	-0.34	-1.09*	-0.38
	(0.92)	(0.64)	(1.36)	(0.57)	(0.57)	(0.83)
Obs	600	600	600	444	444	444
Adj. R^2	0.01	0.02	0.01	0.01	0.01	0.01

	Forest - 3Y-FF3-Alpha			GBRT - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Publication Dummy	4.04**	3.99*	6.84***	0.47	-1.57	-2.88*
	(1.75)	(2.07)	(1.59)	(1.85)	(1.43)	(1.69)
Obs	564	564	564	480	480	480
Adj. R^2	0.13	0.13	0.18	0.02	0.03	0.05

	NN1 - 3Y-FF3-Alpha			NN2 - 3Y-FF3-Alpha			NN3 - 3Y-FF3-Alpha		
	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic	Linear	Quadratic	Quintic
Post-Pub. Dummy	-0.16	-0.16	-2.18	2.65	2.65	-4.95*	2.65	2.65	-4.95*
	(1.41)	(0.84)	(1.48)	(2.85)	(2.46)	(2.79)	(2.85)	(2.46)	(2.79)
Obs	600	600	600	600	600	600	600	600	600
Adj. R^2	0.14	0.15	0.15	0.09	0.10	0.16	0.09	0.10	0.16

Machine Learning and Market Prescience



So what's actually going on?

Which factors contribute explanatory power? Models disagree

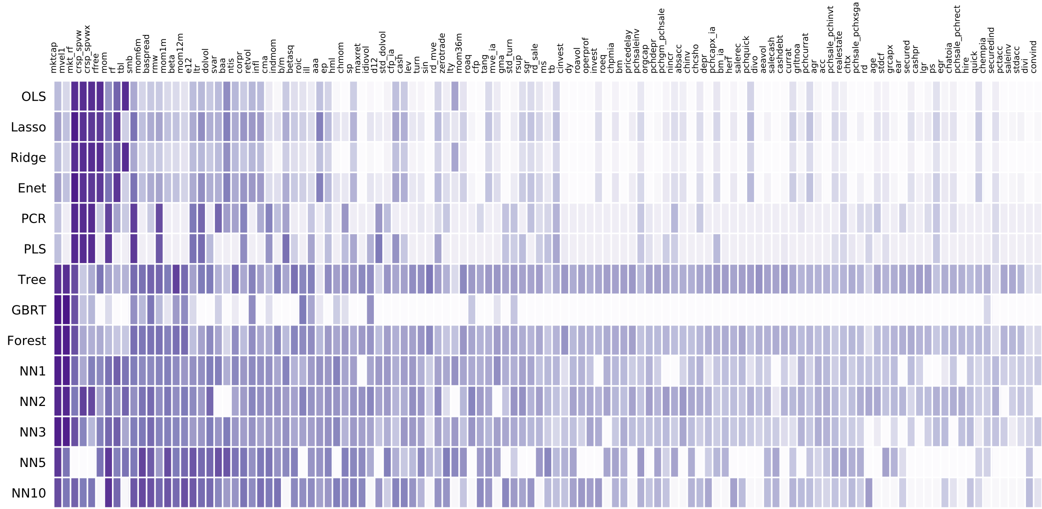


Figure 13: Percentage Decrease in Train R^2 Induced by the Masking of each Factor

Which factors contribute explanatory power? Models disagree

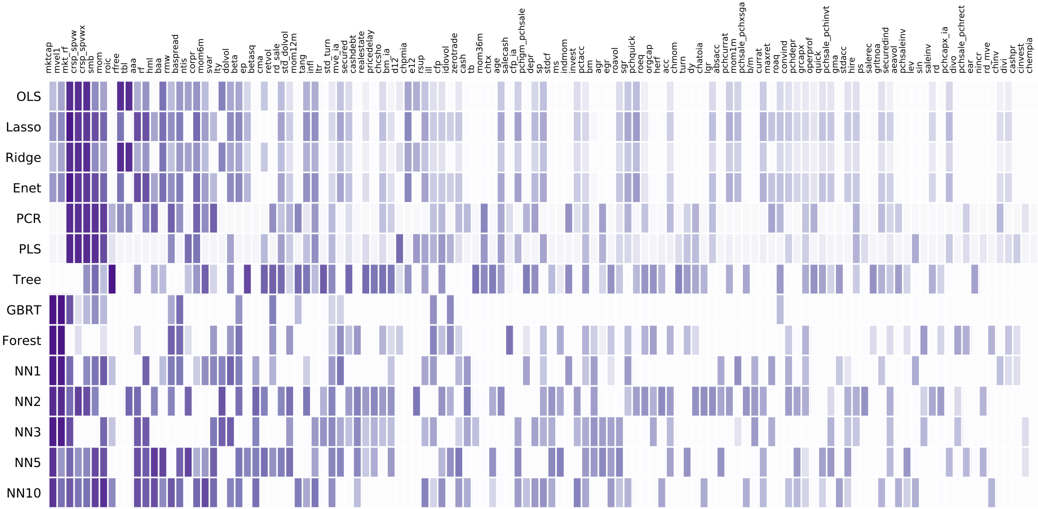
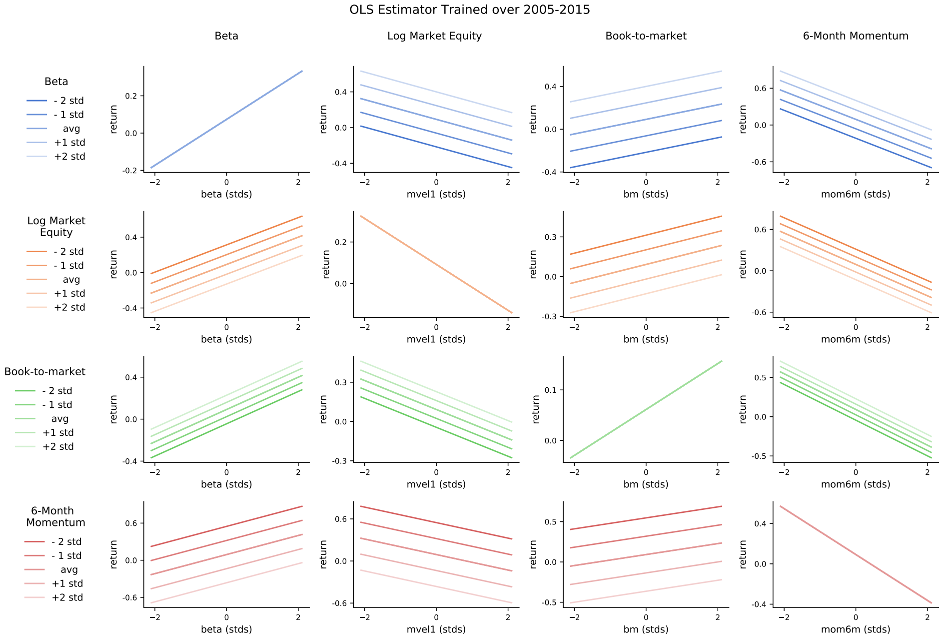
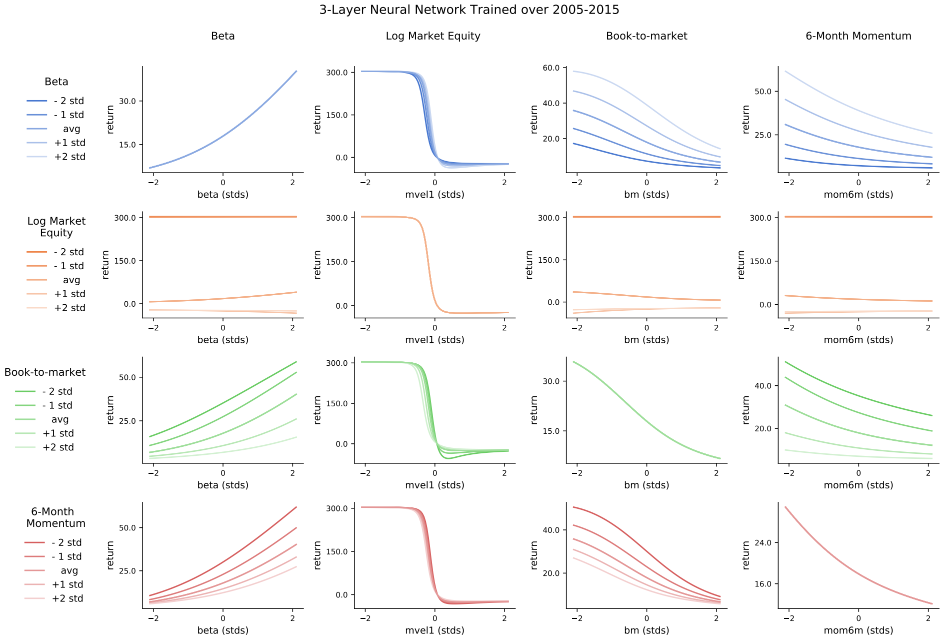


Figure 14: Percentage Decrease in Out-of-sample R^2 Induced by the Masking of each Factor

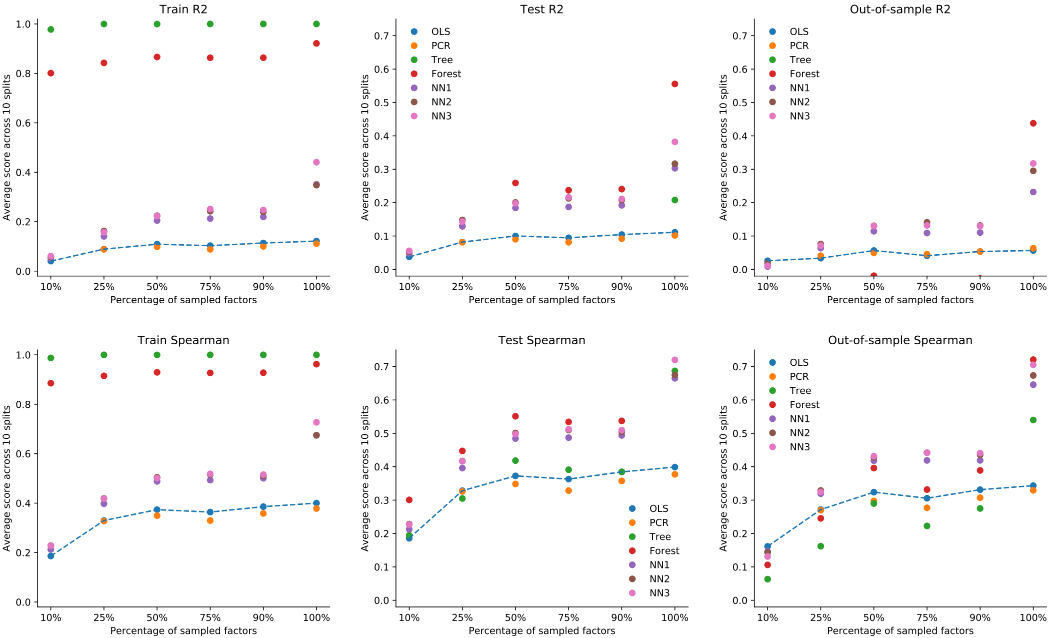
Non-linearities, not factors, seem to be the main source of explanatory power



Non-linearities, not factors, seem to be the main source of explanatory power



Marginal utility of factors vs marginal utility of better methods



Conclusion

In short:

$\Rightarrow$ Our analysis on arbitrage and post-publication decline failed to find evidence of arbitrageurs or of decaying alpha

$\Rightarrow$ It seems that markets are either profoundly algorithmically inefficient or that they are algorithmically efficient; the latter seems much more plausible...

$\Rightarrow$ Absence of evidence is not evidence of absence: better data, other settings, more refined identification etc. could find evidence of arbitrage or post-publication decline

Ideas for future research:

$\rightarrow$ see whether mutual funds with higher ML investment have higher returns

$\rightarrow$ study the development of secret, private ML algorithms

$\rightarrow$ think about regulatory and macro-financial implications