Abstract

This thesis explores the potential use of machine learning in the financial world of stocks. By implementing 12 machine learning models on the methods described in "Empirical Asset Pricing via Machine Learning", we analyze each model's predictive capabilities and the variables they draw upon. Based on each model's prediction, we create a corresponding zero-net-investment portfolio, where we further analyze the performance of each machine learning portfolio. The prediction-analysis showed obvious flaws, which lead to the indication of the OLS-model being the best performing method. This indication got firmly denied in the creation of the zero-net-investment portfolio, which in accordance with the article showed Neural Network models to be the best performing machine learning portfolio for equally weighted portfolio returns.

Contents

1 Metode

Denne sektion vil gennemgå de machine learning modeller, som benyttes i analyserne. Hver undersektion starter med en metodebeskrivelse af de modeller, der benyttes i artiklen. Dette efterfølges af objektfunktionen, der benyttes til at estimere modellens parametre. Til sidst vil der være en beskrivelse af, hvordan fremgangsmåden i dette projekt afviger fra artiklens.

1.1 Inddeling af datasæt

I analysen af aktier er tid et vigtigt element, og derfor skal der tages højde for dataens rækkefølge. Artiklen benytter en kombination af metoderne recursive og rolling, der har til formål at analysere to forskellige aspekter af aktiens udvikling. Rolling deler datasættet ind i identiske intervaller, hvor hvert interval rykkes med en periode. Et interval starter med et træningssæt, der benyttes til at træne modellens parametre. Dette efterfølges af et valideringssæt, der bruges til at fintune modellens hyperparametre. Til sidst er der testsættet, som benyttes til at evaluere modellens estimater. Selvom intervallerne ikke er disjunkte, er det vigtigt at de tre sæt i hvert interval er. Denne metode analyserer aktiens udvikling over korte perioder og baseret på hver periode former den en hypotese, som bruges til at forudsige udviklingen i næste periode. Recursive modellen er en iterationsproces, der analyserer aktiens udviklingen over en periode. For hver iteration vokser træningssættet med en tidsperiode, hvilket betyder det er aktiens overordnede udvikling, der evalueres. I artiklen har de 60 års data, hvor de første 18 er træningssættet, de næste 12 er valideringssættet og de sidste 30 er testsættet. For hver iteration vokser træningssættet med et år, som tages fra testsættet, og valideringssættet bliver dermed rykket et år frem. Kombinationen ligger i at recursive bruges på træningssættet og rolling bruges på valideringssættet, så i stedet for at rykke et interval frem for hver iteration, er det et valideringssæt der rykkes.[1] I dette projekt bruges rolling-metoden, som den er beskrevet ovenfor, fordi dataen går mindre end 30 år tilbage.

1.2 Lineær regression

Den første model der benyttes i analysen, er den lineære regression, hvor *target*-variablen estimieres ved *ordinary least squares* metoden (OLS). Den lineære regressions hypotese dannes ved:[2]

$$h(x_i) = \sum_{i=0}^{N} \theta_i x_i = \sum_{i=1}^{N} \theta^\top x_i,$$

hvor $\theta \in \mathbb{R}^{N+1}$, $\mathbf{x} \in \mathbb{R}^N$ og $x_0 = 1$. Dette er kun gældende for en tidsperiode, så for vores datasæt der indeholder N forskellige aktier over T tidsperioder, vil hypotesen være en $NT \times 1$ vektor, givet ved:[3]

$$h(x_{i,t}) = \sum_{t=1}^{T} \sum_{i=0}^{N} \theta_i x_{i,t} = \sum_{t=1}^{T} \sum_{i=1}^{N} \theta^\top x_{i,t}$$

Bemærk, at den i'te parameter, og dermed *target*-værdien, er uafhængig af historisk data og kun baseres på dataen af i'te aktie. I OLS vælges θ til at være de parametre, der minimerer *standard least squares* objektfunktionen (l_2)[4]:

[1] [1] S. 12
[2] [2] S. 84
[3] [2] S. 84
[4] [2] S. 84

$$\mathcal{L}(\theta) = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\theta^\top x_{i,t} - y_{i,t}\right)^2.$$

I analysen benyttes metoderne, OLS og OLS-3 . OLS-3 trænes kun på variablerne 'momentum', 'size' og 'book-to-market'.

1.2.1 Huber robust objektfunktion

Data kan have enkelte værdier, der er markant større end de andre variabel værdier. Sådanne værdier kaldes outliers, og kan resultere i misvisende R^2-værdier, da forskellen mellem disse værdier og estimaterne, kan være relativt store. Huber loss er en videreudvikling af (l_2) tabsfunktionen, og har til formål at mindske outliernes indflydelse. Funktionen er givet ved[5]:

$$\mathcal{L}_H(\theta) = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} H\left(\theta^\top x_{i,t} - y_{i,t}\right),$$

hvor

$$H(x) = \begin{cases} x^2 & for \quad |x| \leq \xi \\ 2\xi|x| - \xi^2 & for \quad |x| > \xi \end{cases},$$

hvor ξ vælges til at være 99.9%-fraktilen. I dette projekt benyttes Sklearn's lineære regressionsmodel med indbygget Huber objektfunktion. Sklearn's Huber funktion har bibetingelser defineret ved:

$$H(x) = \begin{cases} \frac{1}{2}x^2 & for \quad |x| \leq \xi \\ \xi|x| - \frac{1}{2}\xi^2 & for \quad |x| > \xi \end{cases},$$

hvor $\xi = 1.35$.

1.3 Elastic Net

Elastic net er en regulariseringsmodel, der benyttes til at bekæmpe overfitting af den lineære regression. Dette gøres ved at tilføje et strafled til tabsfunktionen:

$$\mathcal{L}(\theta, \lambda, \rho) = \mathcal{L}(\theta) + \phi(\theta, \lambda, \rho),$$

hvor strafleddet er givet ved[6]:

$$\phi(\theta, \lambda, \rho) = \lambda(1-\rho)\sum_{i=1}^{P}|\theta_i| + \frac{1}{2}\lambda\rho\sum_{i=1}^{P}\theta_i^2, \quad \lambda, \rho \geq 0.$$

I artiklens appendiks viser de, at ρ holdes konstant på 0.5 og dermed er det kun λ, parametrene tilpasses efter. Til dette benytter de i artiklen accelerated proximal gradient og Huber loss, hvorimod der i projektet benyttes Sklearn's stochastic gradient descent (SGD) og Huber loss. SGD er givet ved:

$$\theta_{j+1} = \theta_j - \eta\left(\frac{\partial\mathcal{L}(\theta_j, \lambda, \rho)}{\partial\theta_j}\right), \quad \eta(t) = \frac{1}{\lambda(t_0 + t)},$$

[5][13] S. 2234 (PDF S. 12)
[6][13] S. 2235 (PDF S. 13)

hvor j er en indeksering af iterationer og η er læringsraten, der sikrer at SGD konvergerer mod parameterløsninger for tabsfunktionen.[7] I artiklen sætter de $\rho = 0.5$ og finder en optimal λ-værdi i intervallet $\left(10^{-4}, 10^{-1}\right)$. Dette gøres også i projektet.

1.4 Principal Component Regression & Partial Least Squares

Principal component regression (PCR) er en lineær regressionsmodel baseret på et underrum af datasættets vektorrum. Modellen starter med at implementere principal component analysis (PCA), som finder den første lineære kombination (principal component), der ved at minimere den kvadrede afstand til alle punkter, maksimerer variationen af dataen. Dernæst findes de næste principle components, der er ortogonale med den første, sådan at der løses for[8]:

$$w_j = \arg\max_{w} Var(Zw), \quad s.t. \quad \|w\|^2 = 1, \quad Cov(Zw, Zw_l) = 0, \quad l = 1, 2, \ldots, j-1.$$

Hvert w er en principal component, samt er en søjle i matricen Ω, som angiver dimension reduceringen. Ω er en $P \times K$ matrix, hvor P er antal af variable og K er antal af principal components. Efter PCA processen kan opstilles den lineære model[9]:

$$R = (X\Omega)\theta + \varepsilon,$$

og bruge den som beskrevet ovenfor, men hvor vægtene nu er principal components. Den essentielle forskel mellem PCR og PLS (partial least squares) er ved dannelsen af principal components, bruger PLS de sande target-værdier:

$$w_j = \arg\max_{w} Cov^2(R, Zw), \quad s.t. \quad \|w\|^2 = 1, \quad Cov(Zw, Zw_l) = 0, \quad l = 1, 2, \ldots, j-1.$$

Dette kategoriserer PLS som en supervised-learning metode, hvorimod PCR, grundet PCA metoden, er en kombination af supervised- og unsupervised-learning.[10] I artiklen benyttes valideringssættet til at finde det optimale antal af principal components, K. Dette gøres også i dette projekt.

1.5 Random Forest

Random forest (RF) er en ensemble metode baseret på regressionstræer. Et regressionstræ er en binær opdelingsmetode af observationerne i et datasæt. For hver node i træet er målet at finde den kombination af variabler og split-punkt, som minimerer følgende residualsum:

$$\min_{j,s} \left(\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right),$$

hvor j er en variabel, s er et gennemsnitspunkt mellem to sideliggende punkter for variablen j. Punktet s inddeler observationerne for variabel j ind i to intervaller:

$$R_1(j,s) = \{X | X_j \leq s\} \quad \& \quad R_2(j,s) = \{X | X_j > s\},$$

[7] [3]
[8] [13] S. 2235 (PDF. S. 13)
[9] [13] S. 2235 (PDF. S. 13)
[10] [13] S. 2235 (PDF. S. 13)

hvor $c_v = avg(y_i|x_i \in R_v(j,s))$ for $v \in \{1,2\}$. Bemærk at residualsummen udregnes for ethvert sideliggende punkter for alle variable, hvilket gør det til en regne-tung metode. Denne proces fortsættes for alle noder indtil en forudbestemt grænse nås. Dette kan fx være en nedre grænse for antal observationer i en node, øvre grænse for trædybde eller øvre grænse for antal blade mm.[11] Random forest er en samling af disse regressionstræer, hvor hvert træ baseres på et bootstrap sample. Dette sample består af rækker fra datasættet, hvor hver række vælges tilfældigt med tilbagelægning. Efter hvert træ er trænet på et individuelt bootstrap sæt, som beskrevet ovenfor, baserer random forest algoritmen sit estimat på gennemsnittet af alle træers estimationer:

$$f_B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x),$$

hvor $\{T_b\}_1^B$ er indsamlingen af alle regressiontræer og $b = 1,\ldots,B$ er en indeksering af bootstrap samples'ne.[12] I artiklen benytter de valideringssættet til at finde optimum for trædybden, antal variabler i hver iteration af inddelingsprocessen og antallet af bootstrap samples. I dette projekt benyttes cost-complexity pruning til at sætte en øvre grænse for træ dybde. Der tunes for hyperparameteren α, som minimerer udtrykket[13]:

$$\sum_{m=1}^{|T|}\left(\sum_{x_i\in R_m}(y_i - c_m)^2\right) + \alpha|T|, \quad c_m = \frac{1}{N_m}\sum_{x_i\in R_m} y_i,$$

hvor $|T|$ er antallet af blade.

1.6 Gradient Boosted Regression Trees

Gradient boosted regression trees (GBRT) er også en ensemble metode bygget på regressionstræer. Modellen fungerer ved at lave disse træer, hvor det næste træ baseret på det forriges estimation. Modellen initieres i bladet $f_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i,\gamma)$, hvor γ er estimationerne. Det næste skridt er en iterationsproces for hvert træ, $m \in [1,M]$, som starter med at finde residualet for hver target variabel[14]:

$$r_{i,m} = -\left[\frac{\partial L\left(y_i, f(x_i)\right)}{\partial f(x_i)}\right]_{f=f_{m-1}}, \quad i = 1,2,\ldots,N.$$

Næste skridt er at tilpasse et decision tree til disse residualer, hvor de tilsvarende blade vil angive terminal regionerne, $R_{j,m}, j = 1,2,\ldots,J_m$. Disse regioner er altså de blade, angivet ved j, som indeholder et eller flere residualer. Hernæst findes output-værdien for hver region. Denne estimation findes ved[15]:

$$\gamma_{j,m} = \arg\min_\gamma \sum_{x_i\in R_{j,m}} L(y_i, f_{m-1}(x_i) + \gamma).$$

Nu er der tilbage at finde den nye estimation baseret på træet m:

$$f_m(x) = f_{m-1}(x) + \nu\sum_{j=1}^{J_m}\gamma_{j,m}\cdot 1_{\left(x\in R_{j,m}\right)}, \quad \nu \in (0,1),$$

[11] [4] S. 307
[12] [4] S. 588
[13] [4] S. 307
[14] [4] S. 361
[15] [4] S. 361

hvor ν er læringsraten. Ligningen angiver, at det nye estimat er baseret på det forrige træs estimat, samt summen af estimationerne fra det nye træ.[16] I artiklen tuner de hyperparametrene; antal træer, læringsraten og trædybde.

1.7 Neural Network

Neural network (NN) er en model bygget på lag og neuroner. Modellen starter i første lag, kaldet input-laget, hvor prædiktor variablerne deles ind i deres respektive neuroner. Hernæst dannes de næste lag, kaldet gemte lag (hidden layers), hvor man på forhånd har besluttet antallet af hidden layers, samt antallet af neuroner i disse lag. I hver neuron udregnes outputtet ved en aktiveringsfunktion. Artiklen benytter ReLU-aktiveringsfunktionen, $f(x) = \max\{0, x\}$, hvoraf outputtet er givet ved:

$$x_i^l = f\left(\theta_{i,0}^l + \sum_{j=1}^{n} x_j^{l-1}\theta_{i,j}^l\right) = f\left(z_i^l\right).$$

Her er $i = 1, \ldots, m$ en indeksering af lag l's neuroner og $j = 1, \ldots, n$ er en indeksering af det forrige lags neuroner.[17] Det sidste lag i en neural network regressionmodel indeholder en neuron, hvor outputtet findes som en lineær kombination[18]:

$$g\left(x^l\right) = \theta_0^l + \sum_{j=1}^{n} x_j^{l-1}\theta_j^l = g\left(z^l\right).$$

For at optimere hver vægt og bias benyttes "back propagation". Denne metode starter med at måle forudsigelsesfejlen, givet ved tabsfunktionen[19]:

$$R(\theta) = \left(y - g\left(x^l\right)\right).$$

Dernæst går processen lagene igennem den modsatte vej, deraf navnet back propagation. For hvert lag i hver neuron angiver metoden en bedre vægt og bias, hvor den, gennem gradient descend, løser følgende:

$$\theta_{i,0}^{r+1} = \theta_{i,0}^r - \gamma\frac{\partial R}{\partial \theta_{i,0}^r},$$

$$\theta_{i,j}^{r+1} = \theta_{i,j}^r - \gamma\frac{\partial R}{\partial \theta_{i,j}^r},$$

hvor r er en indeksering af antal gange back propagation benyttes og γ er læringsraten.[20] I artiklen har de på forhånd bestemt følgende NN-modeller: NN1 har et gemt lag med 32 neuroner, NN2 har 2 gemte lag med 32 og 16 neuroner, respektivt, NN3 har 3 gemte lag med 32, 16 og 8 neuroner, respektivt, NN4 har 4 gemte lag med 32, 16, 8 og 4 neuroner respektivt, og NN5 har 5 gemte lag med 32, 16, 8, 4 og 2 neuroner, respektivt. Derudover tuner de læringsraten til stokastic gradient descend, samt bruger de early stopping sammen med lasso-regularisering, batch normalization og ensemble. Lasso regulariseringsmodellen er ENet, hvor ρ sættes til 0. Early stopping er en metode, der stopper optimeringen af modellens parametre, hvis fejlen i valideringssættet ikke forbedres over 5 epochs. En epoch er en gennemgang af dataen efter den er

[16] [4] S. 361
[17] [13] S. 2242 (PDF S. 20)
[18] [4] S. 396
[19] [4] S. 396
[20] [4] S. 396

inddelt i batches. Her bliver NN-modellen trænet i hver batch, og for hver epoch er dataen inddelt forskelligt, men batch-størrelsen er konstant. I dette projekt benyttes early stopping og normalisering af sættene mht. træningssættets forventningsværdi og varians.

1.8 Evalueringsmetoder

For at evaluere modellernes forudsigelsesevner benyttes følgende out-of-sample R^2-model[21]:

$$R^2 = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3} \left(y_{i,t+1} - \hat{y}_{i,t+1}\right)^2}{\sum_{(i,t)\in\mathcal{T}_3} y_{i,t+1}^2},$$

hvor $\mathcal{T}_3$ angiver testsættet og i,t er en indeksering af aktie og tid, respektivt. Artiklen forklarer, at de har fjernet den historiske forventede afkastrate i nævneren, fordi den støjer for meget.
For at kunne sammenligne de forskellige modellers forudsigelsesevner, benyttes Diebold og Mariano testen[22]:

$$DM_{i,j} = \frac{\bar{d}_{i,j}}{\sqrt{\hat{\sigma}^2_{d_{i,j}}}}, \quad i,j \in 1,\ldots,n,$$

hvor i,j er en indeksering af modellerne, n er antallet af modeller og $d_{(i,j),t}$ er gennemsnittet af to modellers estimationsafvigelser i periode t[23]:

$$d_{(i,j),t} = \frac{1}{n_{3,t}} \sum_{a=1}^{n_{3,t}} \left(\left(e^i_{a,t}\right)^2 - \left(e^j_{a,t}\right)^2 \right).$$

Her angiver $e^i_{a,t}$ og $e^j_{a,t}$ den enkelte estimations afvigelse fra den sande værdi for den respektivte model, og $n_{3,t}$ er antallet af aktier for testsættet i periode t. P-værdien udregnes for en to-siddet t-test, hvor resultatet angives med *, hvis differensen er 5%-signifikant mht. Bonferri korrektionen[24]:

$$\alpha_{bk} = \frac{0.05}{\frac{12\cdot 11}{2}} \approx 0.0007576,$$

og med **, hvis differensen er 5%-signifikant.
Med følgende iterationsproces analyseres modellerne for den enkelte variabels indflydelse på forudsigelserne. Processen går over hver variabel, hvor den respektive søjle sættes til 0 og derfra udregnes modellens out-of-sample R^2. Variablens indflydelse findes ved at trække denne værdi fra modellens originale R^2-værdi. Til forskel for artiklen, som illustrerer variablernes indflydelse for alle inddelinger, gøres det i projektet kun for sidste iteration.[25]

2 Resultater

I dette projekt er der blevet indsamlet månedlig data for 3844 aktier af firmaer, som findes på NYSE, AMEX eller NASDAQ. Dataen rækker over 28 år, hvor den starter i februar 1997 og slutter i august 2024. Variabelsamlingen er hentet fra to kilder: variablerne, der angiver aktiepriser, bliver opdateret månedligt og er hentet

[21] [13] S. 2246 (PDF S. 24)
[22] [13] S. 2246 (PDF S. 24)
[23] [13] S. 2246 (PDF S. 24)
[24] [16]
[25] [13] S. 2246 (PDF S. 24)

fra yfinance og de resterende variabler, som opdateres årligt, er webscrapet fra companiesmarketcap[26]. I denne sektion vil dataen kun præsenteres og sammenlignes med artiklens resultater. I den næste sektion vil mulige årsager for resultaternes afvigelser diskuteres.

2.1 Evaluering af estimationer

Modellerne trænes på et datasæt med 27 variabler for at estimere 'adj close' prisen for periode $t+1$. I tabel 1 vises modellernes estimationsevner, hvor interessante værdier for NN-modellerne og de fleste top 500 firmaer fremvises. NN-modellerne har ekstremt lave R^2_{oos}-værdier for alle firmaer og bot 500 firmaer, men har derimod høje værdier for top 500 firmaer. Bemærk især, at for top 500 estimationerne er det kun modellerne, der direkte bruger OLS-metoden, hvilket inkluderer ENet, som har R^2_{oos}-værdier nær 0. Dette indikerer, at variabelværdierne følger en trend, som ikke-OLS modeller kan opfange, hvilket diskuteres nærmere i næste afsnit. Observeres R^2_{oos}-værdierne for alle firmaer alene, så er udfaldet næsten det modsatte af observationerne fra artiklen.

Table 1: Evaluering af de årlige aktiepris-estimationer (R^2_{oos}-værdier)

	OLS +H	OLS-3 +H	PLS	PCR	ENet +H	RF	GBRT +H	NN1	NN2	NN3	NN4	NN5
Alle	0.01	-0.48	-0.02	-1.16	0.12	0.01	0.09	-29.60	-123.59	-191.86	-309.08	-116.39
Top	0.08	0.01	0.99	0.99	0.02	0.89	0.71	0.86	0.76	0.94	0.93	0.78
Bot	0.08	-0.01	-0.56	-18.53	0.00	0.01	0.26	-256.36	-87.71	-572.74	-5310	-6884

Den første kolonne angiver de benyttede modeller og den anden kolonne angiver deres årlige R^2-værdi for hver iterations testsæt. De årlige R^2-værdier er baseret på alle firmaer, de 500 firmaer med højst markedsværdi og de 500 med lavest markedsværdi, respektivt. Til forskel for artiklen, som træner deres model på alle aktier og derefter tester den på et testsæt med 500 firmaer, trænes modellerne i dette projektet også på et datasæt med 500 firmaer.

Til forskel for artiklen, har NN-modellerne de største estimationsafvigelser med faktorer, der er mange gange større end alle andre R^2_{oos}-værdier. Derudover er det ENet, der har den højeste R^2_{oos}-værdi, hvorimod artiklen angiver, at den har den næst laveste værdi.
Figur 1 illustrerer kompleksiteten af 4 modeller med en fælles faktor. Alle grafer falder i kompleksitet i perioderne omkring år 2021, hvilket indikerer at pandemien har skabt et shock, der påvirker modellerne. Dette ses især på graferne for PCR og ENet, der op til perioden er relativt stabile med deres optimale hyperparametre liggende i intervallerne $[23, 26]$ og $[25, 26]$, respektivt.

Da der kun er et overlap på 6 år mellem dette projekts testsæt og artiklens, er det ikke idealt at sammenligne dem, men det er værd at bemærke, at forholdene mellem de forskellige modeller går igen. Som eksempel har PCR-modellen været bedre til at danne principal components i begge analyser. Selvom forskellen ikke er nært så stor i dette projekt, illustrerer det stadig effekten ved at PLS benytter supervised learning ved dannelsen af de lineære kombinationer. Et andet fælles træk for begge analyser er volatiliteten af RF-modellen, der op til flere gange har ekstreme udsving ved valg af optimal trælængde.

Til forskel for tabel 1, angiver værdierne i tabel 2 direkte sammenligninger af modellernes forudsigelsesevner. Diebold-Mariano testværdierne er asymptotisk $\mathcal{N}(0,1)$-fordelte med nulhypotesen: at der ikke er

[26] [15]

nogen forskel modellernes forudsigelsesevner.[27] I tabellen er første række yderst bemærkelsesværdigt, da den indikerer at OLS-metoden præsterer bedre end alle andre modeller, og RF-metoden præsterer næst bedst. Disse observationer kan virke overraskende, da tabel 1 angiver at ENet- og GBRT-modellerne har højere R^2_{oos}-værdier. Denne observation er et eksempel på, hvordan det kan være forgæves at sammenligne modeller baseret på R^2_{oos}-værdier alene. Et yderligere eksempel er, at selvom GBRT har den næsthøjeste R^2_{oos}-værdi, klarer den sig værre mod alle modellerne i DM-testene, hvilket specielt er gældende for NN4-modellen, der har en R^2_{oos}-værdi på -309. Observer de to værdier med ** i første række. Dette betyder at OLS-metoden er 5%-signifikant bedre til at estimere end OLS-3 og RF-metoden. Bemærk især at DM-testen er lavere for RF end for OLS-3, hvilket kunne indikere, at OLS-3 metoden klarer sig mindre dårligt end RF-metoden, men det er forkert. Det ses på anden række, at RF-metoden er 5%-signifikant bedre til at estimere end OLS-3 metoden, hvilket indikerer, at $d_{i,j}$ for OLS og RF varierer mindre ift. OLS og OLS-3. Derfor gælder transitivitet ikke ift. DM-testværdierne, da værdierne kun illustrerer, hvilket af to modeller, der afviger mindst og ikke graden af afvigelserne. Sammenlignet med artiklens tabel, så er fortegnene for næsten alle værdier i dette projekts DM-test modsat dem i artiklens.

Table 2: Diebold-Mariano testene

	OLS-3 +H	ENet +H	PLS	PCR	GBRT +H	RF	NN1	NN2	NN3	NN4	NN5
OLS+H	-2.769**	-1.698	-1.788	-1.761	-1.205	-3.125**	-1.753	-1.746	-1.740	-1.742	-1.744
OLS-3+H		-1.010	-1.073	-1.434	-1.205	2.451**	-1.731	-1.740	-1.736	-1.740	-1.738
ENet+H			-0.365	-1.077	-1.205	1.594	-1.709	-1.735	-1.733	-1.738	-1.732
PLS				-1.737	-1.205	1.707	-1.752	-1.745	-1.740	-1.742	-1.743
PCR					-1.205	1.723	-1.752	-1.745	-1.740	-1.742	-1.743
GBRT+H						1.205	1.204	1.200	1.198	1.193	1.201
RF							-1.750	-1.745	-1.739	-1.742	-1.743
NN1								-1.743	-1.737	-1.741	-1.740
NN2									-1.729	-1.739	1.734
NN3										-1.745	1.734
NN4											1.741

Diebold-Marino testene bliver udført som beskrevet i metodeafsnittet. Her indikerer en positiv testværdi at søjle-modellen har bedre estimationer for alle periode and den tilhørende række-metode og vice versa. Som beskrevet i metodeafsnittet, angiver * at testværdien er 5%-signifikant mht. Bonferri korrektionen og ** at testværdien er 5%-signifikant.

[27] [13] S. 2247 (PDF S. 25)

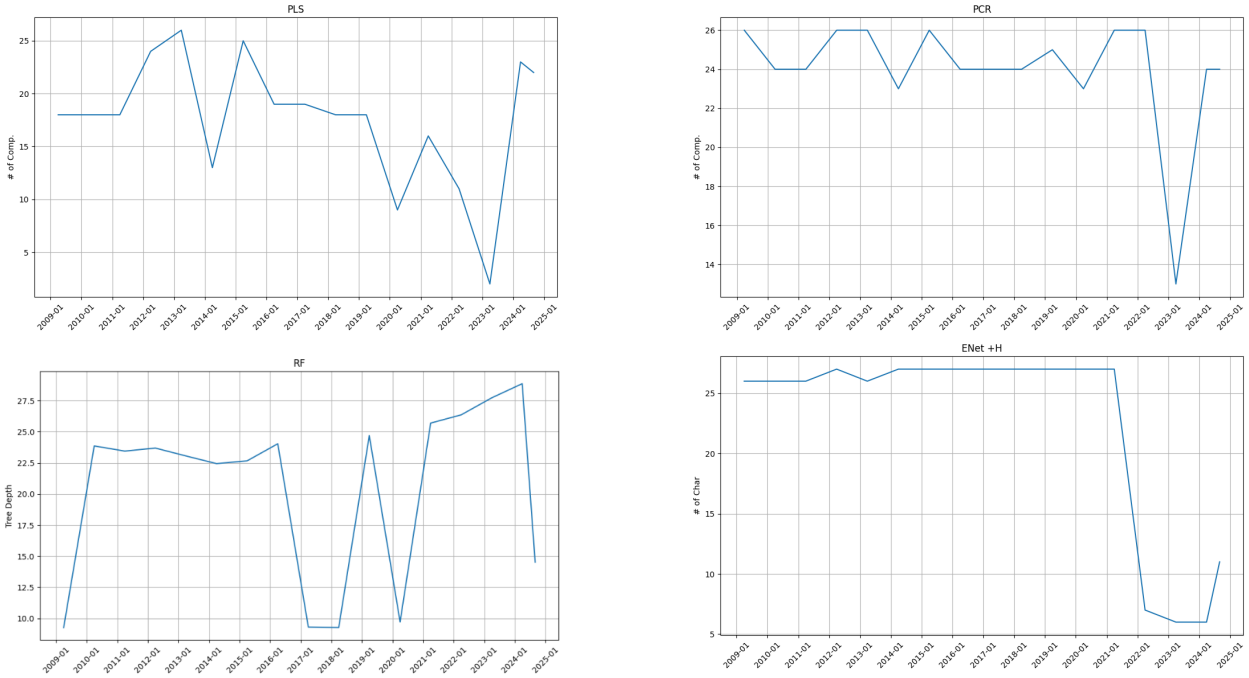
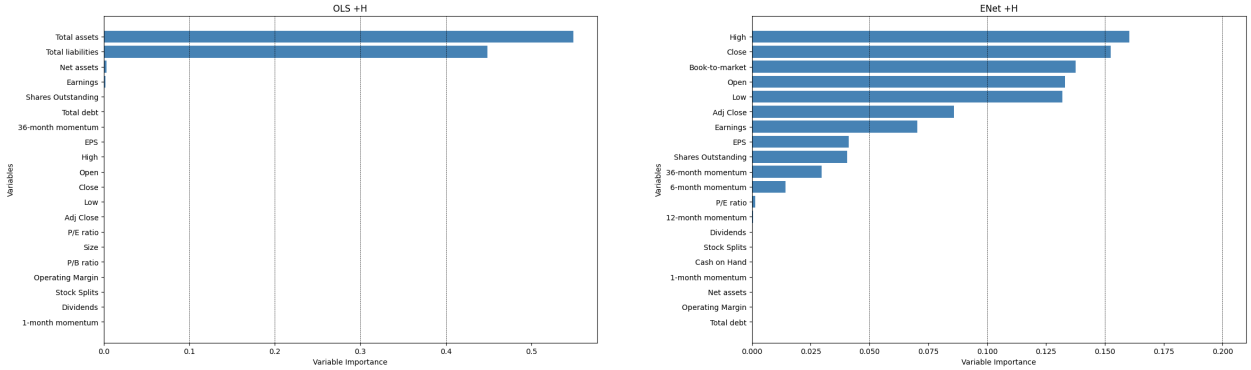


Figure 1: Disse figurer illustrerer den enkelte models kompleksitet. For PCR og PLS illustrer figurerne antallet af principal components for hver iterations testsæt. For RF vises den gennemsnitlige trælængde for hvert testsæt, og for ENet antallet af variabler, der ikke er presset ned til 0.

I figur 3 er variablerne for hver model rangeret, baseret på den indflydelse de har på modellens R^2-værdi, hvor rangen går fra mindst indflydelse op til højst indflydelse. For hver variabel tages den samlede sum over alle modeller, hvorved de inddeles i en orden med højst rangsum øverst. Ud fra hver variabel kan man anskue den enkelte variables indflydelse på den respektive model, baseret på hvor mørk den tilhørende farve er. Det observeres på figur 3, at de fleste mørke farver er grupperet i rækkerne for yfinance's månedlige aktiepriser. Bemærk, at farverne er lyse i nogle af NN-modellerne for 'Adj Close', hvilket skyldes, at modellerne slet ikke benytter de fleste variabler. Dette medfører, at de har en effekt på 0, så selvom 'Adj Close' er den mest indflydelsesrige variabel, rangerer den ikke højere end 10, og bliver dermed tildelt en hvidere farve. Det kommer heller ikke som en overraskelse, at aktiepris-variablerne er i den høje ende, hvis man observerer figur 1 i appendikset. Her ses disse variabler til at være stærkt korrelerede med target variablen 'Adj Close t+1'.



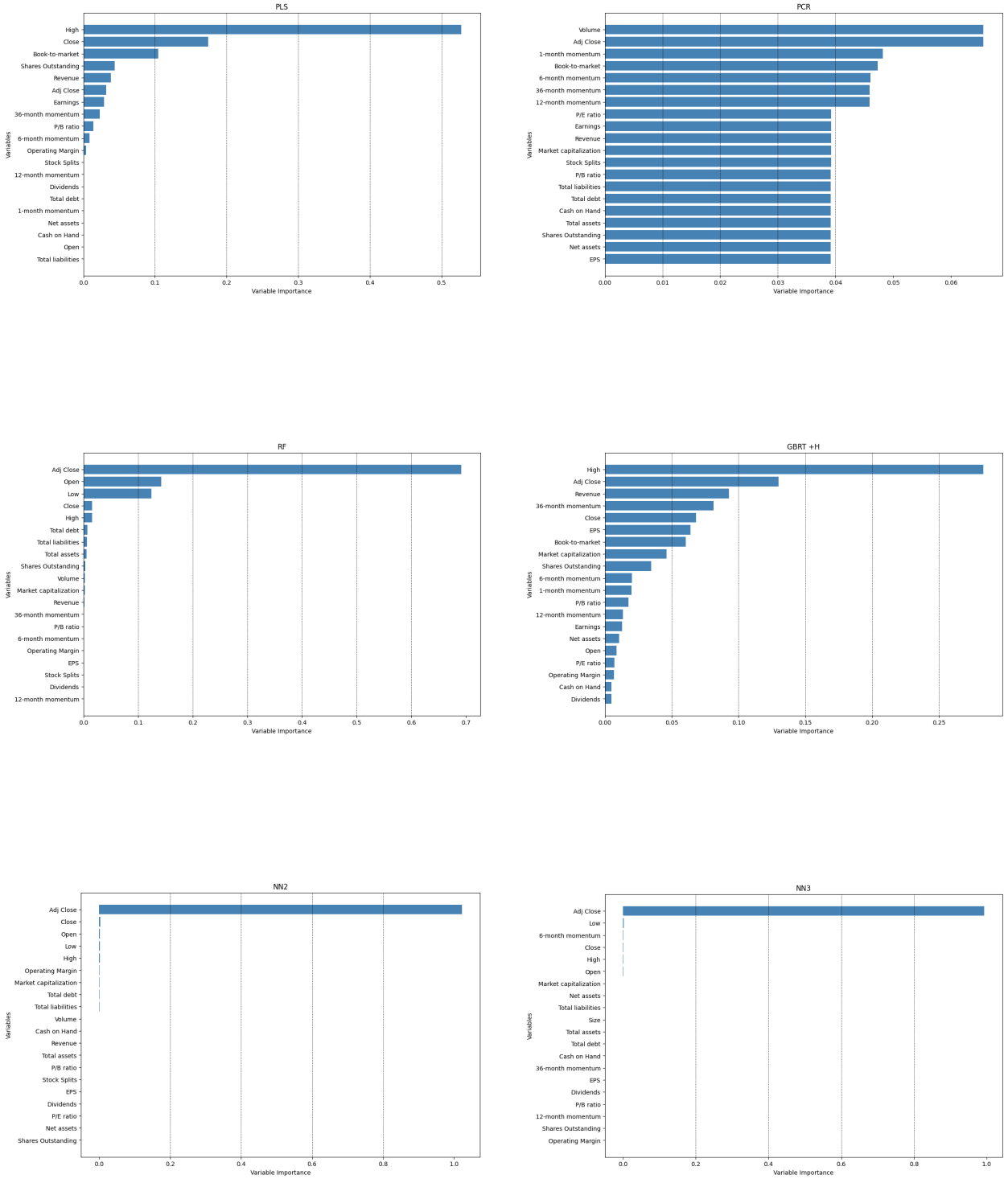


Figure 2: Disse figurer viser den enkelte variabels indflydelse på R^2-værdien for hver model, som beskrevet i metodebeskrivelsen.

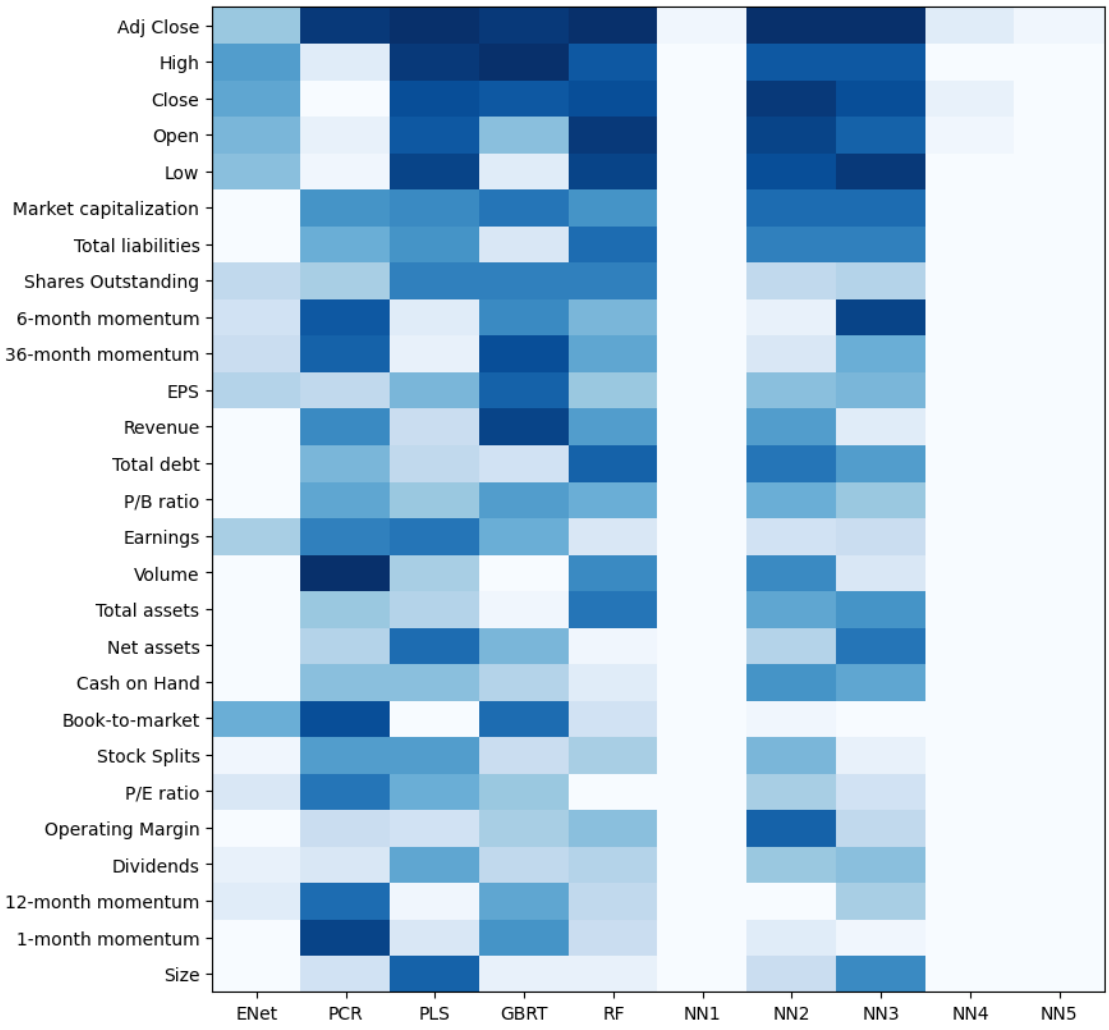


Figure 3: Oversigt af den enkelte variabels indflydelse for alle modeller.

Figur 2 angiver også at det hovedsageligt er de 5 aktiepris-variabler, der har størst indflydelse på modellernes estimationer. Når det så er sagt, er der OLS-metoden, som slet ikke gør brug af disse variable, og er den model, der dominerer i tabel 1 og tabel 2. Derfor kunne det tænkes, at OLS-metoden har opfanget trends i sine variabler, som gør dens estimationer bedre end de andre modellers. Observer til sidst, at PCR-modellen, der virker til at bruge de fleste variabler lige meget, observeres i tabel 2 til at have bedre estimationsevner end GBRT og NN-modellerne. Dette, sammen med OLS-modellen, kan sætte spørgsmålstegn ved, hvorvidt aktiepris-variabler burde have sådan en stor indflydelse modellerne. Derfor mindes der til sidst om, at RF-modellen, som kraftigt påvirkes af de tiltalte variabler, præsterede næst bedst i DM-testene.

2.2 Machine learning porteføljer

For at kunne fortsætte vores analyse, finder vi den tilsvarende merafkastsrate for vores target variabel. Lad Adj Close være betegnet ved P, sådan at afkastraten findes ved:

$$r_{t+1} = \frac{P_{t+1}}{P_t} - 1.$$

Givet at $\hat{P}_{t+1}$ er næste periodes estimerede 'adj close' pris, findes den estimerede afkastrate ved:

$$\hat{r}_{t+1} = \frac{\hat{P}_{t+1}}{P_t} - 1.$$

Hvorved den estimerede merafkastraten findes ved:

$$\hat{er}_{t+1} = \hat{r}_{t+1} - r_{f,t+1}.$$

Her er $r_{f,t}$ notation for den risikofrie rate i periode t. Fordi vores data hovedsageligt består af amerikanske aktier, har vi fundet data for obligationsraten i USA, og benytter dette som den risikofrie rate. For hver måned rangeres aktierne mht. decil, hvorefter de tildeles to vægte. Den første vægt er en ligevægtskonstant, hvilket betyder at hver aktie vægter lige meget. Denne konstant findes ved:

$$w^{eq} = \frac{1}{N},$$

hvor N er antallet af aktier for den respektive måned. Den anden vægt vægter hver aktie mht. firmaets markedsværdi, som findes ved:

$$w_i^{mc} = \frac{mc_i}{\sum_{j=1}^{N} mc_j},$$

hvor i og j er en indexering af aktierne for den respektive måned. Med disse vægte føjet til hver aktie, deler vi nu datasættet med estimerede merafkastrater ind i 10 datasæt mht. aktiens decile rang. Vi benytter disse decilrang datasæt til at forme en zero-net-investment(zni) portefølje over perioden 01-2014 til 12-2023, hvor vi for hver periode/måned shorter aktierne i decilrang 1 datasættet, og longer aktierne i decilrang 9 datasættet. I artiklen defineres den enkelte porteføljes merafkastrate ved:

$$\hat{er}_{t+1}^{P} = \sum_{i=1}^{N} w_{i,t}^{P} \cdot \hat{er}_{i,t+1},$$

hvor P angiver den enkelte portefølje.[28] Til sidst findes den sande Sharp ratio for hver portefølje:[29]

[28] [13] S.2261 (PDF S. 39)
[29] [5] S. 57

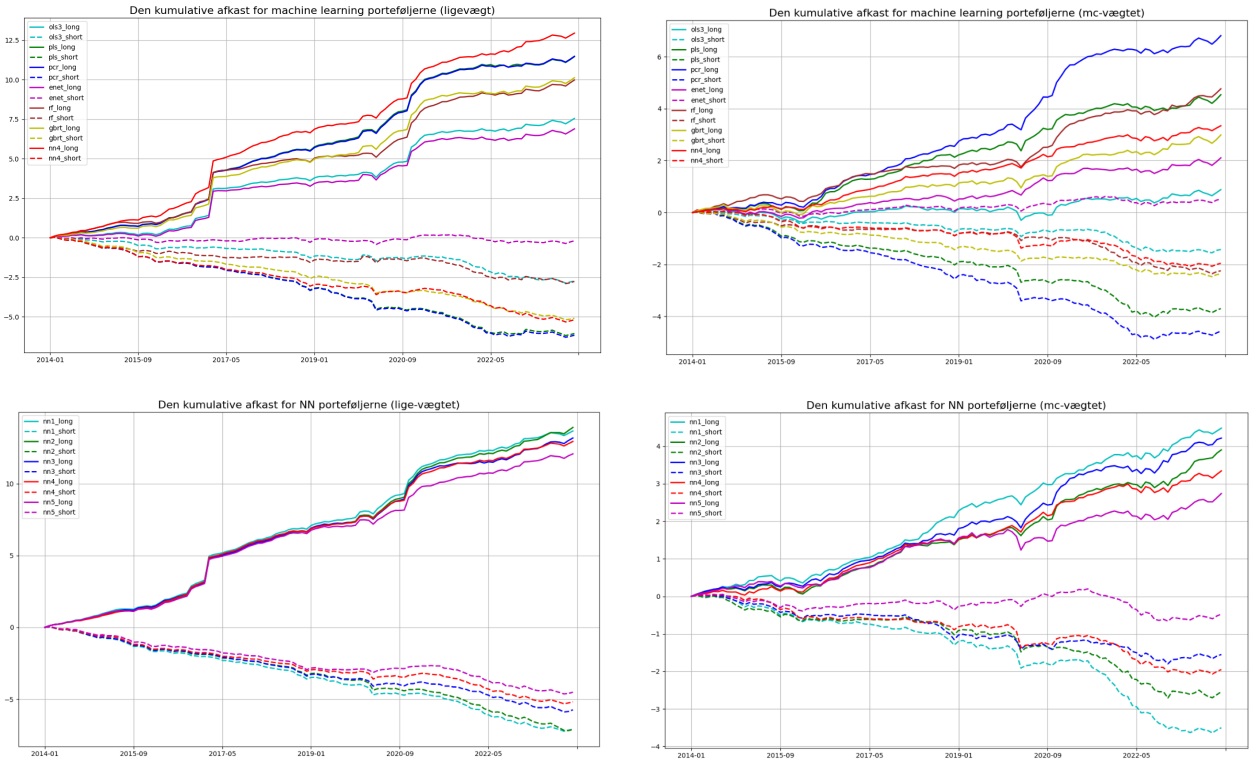


Figure 4: Disse figurer viser den akkumulerede merafkastsrate for hver zni-portefølje baseret på diverse gennemgåede modeller og på NN-modeller, respektivt.

$$SR^P = \frac{er^P_{t+1}}{\sigma^P_R}.$$

I appendikset S. 3-6 findes 2 tabeller, ligevægtet og markedsværdi-vægtet, der viser, hvordan machine learning porteføljerne præsterer. En overraskende observation er præstationen af OLS- og NN-modellerne, især ift. udfaldet af tabel 1 og 2. Observeres værdierne for den OLS-baserede zni-portefølje, resulterer modellen i de laveste sharp-ratio værdier for begge vægttyper, hvilket er det strengt modsatte af hvad tabellerne indikerer. Derudover præsterer NN-modellerne markant bedre end forventet, hvor NN1- og NN2-modellerne resulterer i de bedste sharp-ratio værdier for begge vægttyper. En forklaring på dette kan være, at selvom NN-modellerne har de mest upræcise estimationer, er de bedst til at fange de underliggende trends i udviklingen af target variablen. For at uddybe dette, overvej at NN-modellen har fejlestimeret den sande værdi med en afvigelse på x. Hvis denne model fortsætter med at fejlestimere i samme grad, vil estimationerne i sig selv være dårlige, og dermed resultere i lave værdier for tabellerne 1 og 2, men ranginddelingen af estimationerne vil ligne de samme 10 datasæt, som hvis man havde rang-indelt de sande værdier, fordi man i teorien har adderet/fratrukket den samme værdi, x, fra de sande værdier. Dette overraskende udfald passer også overens med artiklens observationer, der bedømmer NN-modellerne til at være de bedste strategier, for at skabe zni-porteføljer ud fra.

På figur 4 har vi plottet de akkumulerede merafkastrater for de forskellige modellers short og long porteføljer, dvs. de porteføljer der har rang 0 og 9, respektivt. En interessant observation er, at ligevægtede long-porteføljer præsterer omtrent dobbelt så godt, sammenlignet med de markedsværdi-vægtede porteføljer. For ifølge sharpe-ratio, indikerer den, at man burde benytte markedsværdi-vægtede rater frem for de

ligevægtede. Et eksempel er NN1-modellen, der for mc-vægtet porteføljer, har en god sharpe-ratio værdi på 4.44, hvorimod den ligevægtede NN1-model har 1.73. Figur 4 illustrerer derimod, at den ligevægtede NN1-model præsterer markant bedre, hvor den akkumulerede merafkastrate for long-porteføljen nærmest er tre gange større. Der er lille tvivl om, at det er variansen, der får sharpe-ratio til at "vælge" de markedsværdi-vægtede modeller, hvilket skyldes, at den ser lave varianser som sikrere muligheder. Ulempen er til gengæld, at den ikke kan behandle varianser på andre måder end størrelse, hvilket resulterer i, at den kan gå glip af større afkastrater som dem, der er beskrevet i vores eksempel. Det observeres til sidst, at for mc-vægtede porteføljer er PCR den bedst præsterende model, hvilket kan virke noget overraskende, da der hverken i vores eller artiklens tabeller har været observationer, der ville indikere dette.

Table 3: Oversigt over machine learning zni-porteføljernes adfærd i procent

	OLS +H	OLS-3 +H	PLS	PCR	ENet +H	RF	GBRT +H	NN1	NN2	NN3	NN4	NN5
MC-vægtet:												
Max DD	63.78	18.68	28.15	13.82	29.24	6.70	10.67	2.86	3.12	4.07	4.97	40.06
Max 1M loss	16.83	11.55	10.56	6.20	10.71	3.30	7.74	2.82	3.07	3.99	4.85	19.93
Turnover	21.31	31.23	71.70	81.25	30.22	84.49	46.48	74.58	67.95	63.16	57.46	57.46
Ligevægtet:												
Max DD	440.61	22.56	29.24	10.13	7.98	30.27	1.54	1.13	0.00	0.00	1.47	8.99
Max 1M loss	97.96	20.20	24.91	6.78	7.67	26.12	1.53	1.12	-5.73	-1.92	1.46	8.60
Turnover	28.69	36.71	46.42	46.27	34.37	46.53	41.70	48.10	48.40	47.65	47.32	46.55

Tabellen angiver det største fald med afbrydelse (Max DD), det største fald uden afbrydelser (Max 1M loss) og ændringen i vægte, for at opretholde vægtfordelingen for hver periode (Turnover).

I tabel 3 analyseres udviklingen af zni-porteføljerne for begge vægttyper. Maximum drawdown er defineret ved:

$$MaxDD = \max_{0 \le t_1 \le t_2 \le T} \left(Y_{t_1} - Y_{t_2} \right),$$

hvor Y_t er den akkumulerede log afkastsrate.[30] Formålet med denne metode, er at undersøge det største samlede fald i log afkast. Grunden til, at Y_t er den akkumulerede log afkast, skyldes at denne metode tillader perioder, hvor der ikke sker nogen ændring, dvs. det eneste der stopper metoden, er enhver stigning i afkast. Max 1M loss er en metode, der undersøger det største fald i afkast på en måned. Til sidst er der Turnover, som er givet ved:

$$\frac{1}{T} \sum_{t=1}^{T} \left(\sum_{i=1}^{n} \left| w_{i,t+1} - \frac{w_{i,t}(1 + r_{i,t+1})}{1 + \sum_{j=1}^{n} w_{j,t} \cdot r_{j,t+1}} \right| \right).$$

Brøken indikerer ændringen i vægtforholdene for hver aktie til tiden t, og fordi vægtene er defineret til at være et af to muligheder, skal de for periode t+1 tilpasses aktiens vægt til definitionen, hvor denne tilpasning/ændring er turnover-værdien. Denne metode angiver altså de samlede ændringer, der foretages for hver periodes portefølje.

[30] [13] S. 2267-2268 (PDF S. 45-46)

Fordi artiklens machine learning portefølje og dette projekts portefølje rækker over to forskellige tidsperioder, er det besværgeligt at sammenligne Turnover værdierne, da værdien skalerer med tid. Hvad angår de første to metoder, er der to steder, hvor vores værdier adskiller sig fra artiklens tabel. Først er der vores OLS-model for ligevægtede porteføljer, som falder med hhv. 440% og 98% for Max DD og Max 1M loss, hvilket er ekstreme værdier, både mht. vores tabel, og artiklens tabel. Derudover har NN2- og NN3-modellerne også unikke værdier for ligevægtsporteføljerne. Først har de max DD værdier på 0%, hvilket betyder, at henover 10 år har zni-porteføljen været ikke-aftagende, og for at supplerer til dette, har det største månedlige fald, faktisk været en stigning på hhv. 5.7% og 1.9%. I vores tabel er disse unikke værdier, hvorimod det i artiklens tabel er ukendte.

3 Diskussion

Som nævnt i forrige sektion vil jeg nu gennemgå mulige grunde til, vores resultater afviger fra artiklens. Sættes modellernes forudsigelsesevner (tabel 1) op mod artiklens, ses det, at OLS, OLS-3, PLS, PCR og NN-modellerne resulterer i værdier, som er markant anderledes. En forklaring på NN-modellerne kan være tuningen af hyperparametrene. I artiklen sætter de batch size til 10000, hvorimod jeg benytter TensorFlows standardstørrelse på 32. Derudover bruger de også lasso-regularisering, der ligesom ENet, indfører et strafled. I min fremgangsmåde bruger jeg kun early stopping, hvorimod de bruger de early stopping og tuner mht. lasso-hyperparameteren og læringsraten. For PCR og PLS modellerne tuner jeg også mht. det optimale antal af principal components. Forskellen mellem mit og deres ligger måske i, at jeg har 27 variabler, hvorimod de i artiklen har 920, hvilket tillader at gøre modellen mere kompleks. Dette gælder også for OLS metoden, så jeg mistænker, at det er forskellen i input variable, der kan skabe denne forskel. For OLS-3 metoden mistænker vi, at forskellen kommer af, at artiklen træner deres model over en længere periode ift. os.
Jeg indså desværre for sent, at for top 500 og bot 500 firmaerne, bliver modellerne fortsat trænet på datasættet for alle firmaer. Derfor blev modellerne trænet på datasættene for top og bot 500, hvorefter deres forudsigelsesevner blev testet. En forklaring på hvorfor top 500 forudsigelserne resulterede i høje R^2-værdier, kan skyldes den enkelte variables fordeling. Ser man på appendikset S. 7-9, er top 500 datasættet det eneste, hvor de fem aktiepris-variable, samt target variablen er grupperet i 0 efter normalisering. Husk, at det tidligere blev nævnt, at disse variabler også er de mest indflydelsesrige variabler, samt at de er stærkt korrelerede med target-variablen, jf. appendiks S. 2. For at perspektivere dette, fjernes de fem variabler, hvilket resulterer i en $R2_{oos}$-værdi på -0.096 for NN2 top 500.

4 Konklusion

I den empiriske analyse af modellernes forudsigelsesevner, var OLS-modellen den dominerende metode efterfulgt af RF-modellen. Det regnes med, at de store fejlestimationer for NN-modellerne, skyldes at hyperparametrene ikke blev tunet nok. Det kom derfor som en overraskelse, at for ligevægtede zni-porteføljer, var det NN-modellerne der præsterede bedst, hvor NN2- og NN3-porteføljerne var ikke-aftagende over 10 år. Sammenligningen med artiklens NN-modeller viste, at projektets NN-modeller havde akkumuleret et større afkast over 10 år end deres havde over 30 år. For mc-vægtede porteføljer var det PCR-modellen, der formede den bedste portefølje, hvilket ikke blev indikeret af nogen tabelværdier. Selvom dette er den eneste model, hvor alle variabler har den relative samme indflydelse på estimeringsevnen, kan det ikke konkluderes dette som årsagen. Det kan endeligt konkluderes, at selvom machine learning modeller har vist svage

estimeringsevner, så har de væres succesfulde til at bygge porteføljer.

Litteraturliste

[1]: Gu, Shihao; Kelly, Brian & Xiu, Dacheng (2018): ”Artiklens appendix”.

[2]: Abu-Mostafa, Yaser S.; Magdon-Ismail, Malik & Lin, Hsuan-Tien (2012): ”Learning from data”.

[3]: Scikit Learn: (https://scikit-learn.org/1.5/modules/sgd.html#mathematical-formulation).

[4]: Hastie, Trevor; Tibshirani, Robert & Friedman Jerome (2017): ”The elements of statistical learning”.

[5]: Poulsen, Rolf & Lando, David (2023): ”Finance 1 and beyond”.

[6]: Scikit Learn: (https://scikit-learn.org/1.5/auto_examples/ensemble/plot_gradient_boosting_regression.html).

[7]: Github: (https://kirenz.github.io/regression/docs/randomforest.html).

[8]: kristina969. Github: (https://github.com/kristina969/Empirical-Asset-Pricing-via-Machine-Learning-Evidence-from-the-German-Stock-Market).

[9]: Scikit Learn: (https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestRegressor.html).

[10]: Geeksforgeeks: (https://www.geeksforgeeks.org/implementing-neural-networks-using-tensorflow/?ref=ml_lbp).

[11]: Reintech: (https://reintech.io/blog/how-to-create-a-neural-network-with-tensorflow).

[12]: Geeksforgeeks: (https://www.geeksforgeeks.org/using-early-stopping-to-reduce-overfitting-in-neural-networks/).

[13]: Gu, Shihao; Kelly, Brian & Xiu, Dacheng (2018): ”Empirical Asset Pricing via Machine Learning”.

[14]: Federal Reserve Bank of St. Louis: (https://fred.stlouisfed.org/series/DGS20)

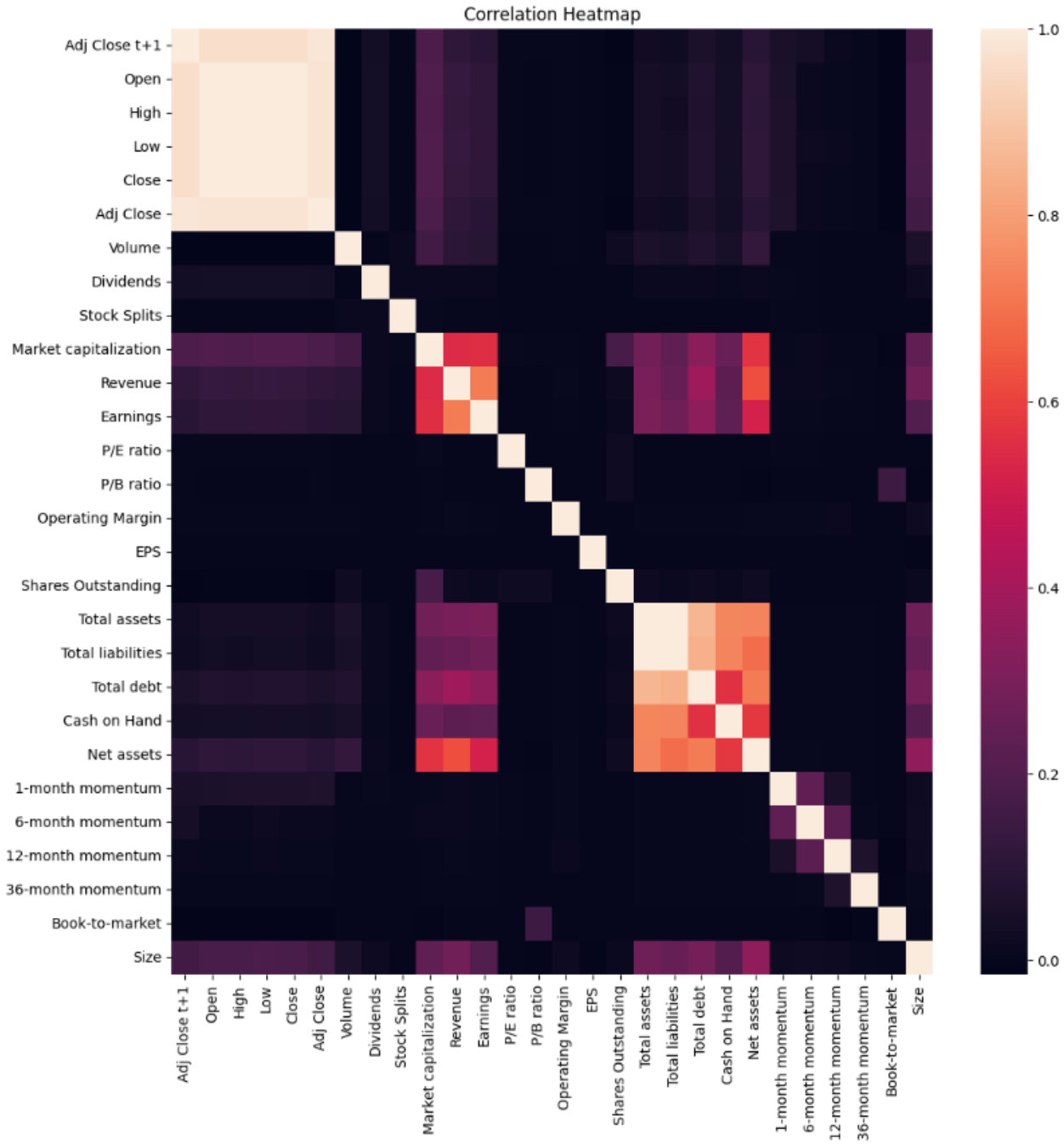
[15]: Companies marketcap: (https://companiesmarketcap.com/)

[16]: Amplitude: (https://amplitude.com/explore/experiment/what-is-bonferroni-correction)

Kilderne kan findes som pdf på link:
https://drive.google.com/drive/folders/1YBLpnfUg5TvDcnOOWxvZTFH4lxOL9TYi?usp=sharing

Appendix

Figur 1: Oversigt over korrelationen mellem alle variable



Machine learning porteføljе (ligevægt):

OLS-3

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-2.836	-0.020	0.071	-0.977	-0.196	-0.046	0.082	-1.954	-0.147	-0.047	0.081	-2.001
1	-0.714	0.001	0.050	0.089	-0.052	-0.023	0.061	-1.321	-0.043	-0.022	0.061	-1.262
2	-0.577	0.004	0.051	0.288	-0.030	-0.011	0.055	-0.692	-0.026	-0.010	0.055	-0.660
3	-0.430	0.007	0.053	0.437	-0.017	-0.002	0.054	-0.146	-0.015	-0.001	0.053	-0.060
4	-0.250	0.009	0.055	0.534	-0.006	0.006	0.052	0.413	-0.006	0.006	0.052	0.405
5	-0.028	0.010	0.057	0.607	0.003	0.013	0.051	0.883	0.003	0.012	0.052	0.807
6	0.290	0.011	0.059	0.655	0.014	0.019	0.055	1.187	0.014	0.019	0.056	1.198
7	0.804	0.014	0.066	0.746	0.029	0.026	0.060	1.516	0.028	0.025	0.059	1.495
8	1.842	0.018	0.072	0.888	0.057	0.036	0.066	1.888	0.057	0.035	0.066	1.865
High(H)	7.811	0.094	0.419	0.780	0.682	0.131	0.426	1.066	0.881	0.131	0.426	1.067
H-L	10.647	0.114	0.407	0.974	0.878	0.177	0.415	1.477	1.028	0.178	0.415	1.488

PLS

PCR

ENet

OLS

RF

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-1.295	-0.000	0.057	-0.005	-1.161	0.019	0.129	0.500	-0.076	-0.020	0.082	-0.855
1	-0.757	0.004	0.051	0.303	-0.993	0.007	0.057	0.418	-0.019	-0.010	0.056	-0.590
2	-0.641	0.006	0.052	0.407	-0.984	0.007	0.058	0.441	-0.009	-0.004	0.054	-0.255
3	-0.511	0.005	0.054	0.326	-0.974	0.010	0.061	0.556	-0.002	-0.001	0.055	-0.042
4	-0.357	0.005	0.057	0.283	-0.961	0.007	0.061	0.417	0.003	0.003	0.053	0.213
5	-0.160	0.006	0.058	0.366	-0.941	0.008	0.063	0.450	0.009	0.007	0.054	0.469
6	0.116	0.007	0.060	0.413	-0.908	0.010	0.065	0.531	0.015	0.010	0.056	0.625
7	0.556	0.009	0.066	0.447	-0.849	0.010	0.064	0.548	0.023	0.015	0.061	0.839
8	1.428	0.013	0.075	0.621	-0.703	0.012	0.061	0.705	0.037	0.026	0.072	1.246
High(H)	6.282	0.089	0.414	0.741	0.529	0.058	0.398	0.509	0.371	0.117	0.415	0.975
H-L	7.577	0.089	0.403	0.762	1.690	0.040	0.404	0.342	0.448	0.137	0.407	1.165

GBRT NN1 NN2

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-3.847	-0.040	0.065	-2.124	-0.113	-0.055	0.077	-2.470	-0.115	-0.055	0.077	-2.481
1	-0.050	-0.018	0.056	-1.087	-0.055	-0.028	0.063	-1.527	-0.048	-0.029	0.061	-1.637
2	-0.034	-0.007	0.052	-0.475	-0.036	-0.015	0.056	-0.933	-0.029	-0.017	0.054	-1.058
3	-0.021	0.001	0.051	0.069	-0.023	-0.006	0.052	-0.395	-0.016	-0.006	0.052	-0.411
4	-0.010	0.008	0.053	0.544	-0.012	0.002	0.052	0.112	-0.005	0.001	0.051	0.060
5	0.005	0.014	0.054	0.892	-0.002	0.010	0.051	0.700	0.005	0.010	0.052	0.641
6	0.025	0.017	0.058	1.034	0.009	0.018	0.051	1.199	0.016	0.019	0.052	1.249
7	0.058	0.020	0.063	1.108	0.023	0.030	0.058	1.799	0.031	0.029	0.058	1.719
8	0.129	0.029	0.071	1.425	0.061	0.041	0.066	2.171	0.056	0.044	0.067	2.264
High(H)	8.862	0.119	0.417	0.983	2.899	0.151	0.423	1.237	4.924	0.154	0.423	1.258
H-L	12.708	0.158	0.405	1.355	3.012	0.205	0.409	1.738	5.039	0.209	0.410	1.767

NN3 NN4 NN5

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-0.143	-0.045	0.076	-2.046	-0.151	-0.040	0.074	-1.854	-0.284	-0.035	0.069	-1.761
1	-0.065	-0.027	0.062	-1.509	-0.067	-0.027	0.063	-1.450	-0.181	-0.021	0.061	-1.215
2	-0.043	-0.016	0.057	-0.992	-0.044	-0.015	0.058	-0.900	-0.129	-0.012	0.056	-0.760
3	-0.027	-0.007	0.053	-0.452	-0.029	-0.006	0.054	-0.410	-0.076	-0.004	0.054	-0.259
4	-0.015	0.001	0.051	0.041	-0.016	0.001	0.051	0.043	-0.015	0.002	0.055	0.136
5	-0.003	0.009	0.051	0.635	-0.005	0.009	0.051	0.583	0.065	0.009	0.056	0.576
6	0.009	0.017	0.052	1.135	0.008	0.016	0.052	1.053	0.178	0.015	0.057	0.891
7	0.025	0.028	0.056	1.730	0.024	0.028	0.057	1.699	0.356	0.025	0.062	1.392
8	0.054	0.041	0.066	2.151	0.064	0.040	0.067	2.059	0.751	0.035	0.070	1.728
High(H)	6.092	0.147	0.424	1.205	8.190	0.144	0.421	1.182	8.019	0.135	0.419	1.120
H-L	6.235	0.192	0.409	1.627	8.341	0.184	0.408	1.557	8.303	0.170	0.409	1.442

Machine learning porteføljе (mc-vægtet):

OLS-3 PLS PCR

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-0.898	-0.010	0.053	-0.689	-0.226	-0.028	0.073	-1.323	-0.115	-0.035	0.073	-1.659
1	-0.717	0.007	0.045	0.549	-0.051	-0.018	0.057	-1.083	-0.043	-0.019	0.056	-1.188
2	-0.580	0.010	0.045	0.748	-0.030	-0.006	0.050	-0.433	-0.026	-0.006	0.050	-0.393
3	-0.435	0.014	0.047	1.038	-0.017	0.001	0.047	0.079	-0.015	0.001	0.047	0.106
4	-0.251	0.009	0.046	0.706	-0.006	0.010	0.046	0.739	-0.006	0.009	0.046	0.700
5	-0.037	0.013	0.053	0.837	0.003	0.015	0.044	1.210	0.003	0.016	0.044	1.265
6	0.280	0.012	0.053	0.759	0.014	0.024	0.051	1.613	0.013	0.025	0.050	1.754
7	0.788	0.015	0.063	0.839	0.029	0.028	0.052	1.852	0.028	0.030	0.056	1.844
8	1.826	0.013	0.069	0.631	0.055	0.034	0.062	1.904	0.054	0.040	0.063	2.210
High(H)	6.494	0.010	0.074	0.449	0.156	0.042	0.094	1.555	0.179	0.063	0.114	1.921
H-L	7.391	0.020	0.054	1.286	0.382	0.070	0.072	3.355	0.294	0.098	0.097	3.508

ENet OLS RF

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-2.046	0.006	0.049	0.399	-3.558	0.008	0.070	0.412	-0.052	-0.017	0.057	-1.056
1	-0.759	0.007	0.046	0.545	-0.992	0.004	0.053	0.278	-0.019	-0.006	0.050	-0.449
2	-0.641	0.006	0.045	0.487	-0.984	0.005	0.055	0.321	-0.009	0.001	0.046	0.077
3	-0.511	0.008	0.045	0.640	-0.974	0.007	0.053	0.464	-0.002	0.005	0.045	0.372
4	-0.355	0.005	0.048	0.376	-0.960	0.006	0.052	0.395	0.003	0.007	0.046	0.553
5	-0.163	0.005	0.051	0.326	-0.940	0.005	0.054	0.310	0.009	0.011	0.048	0.826
6	0.108	0.006	0.049	0.394	-0.907	0.006	0.050	0.448	0.015	0.012	0.048	0.859
7	0.550	0.008	0.061	0.430	-0.845	0.006	0.048	0.450	0.023	0.018	0.052	1.199
8	1.414	0.010	0.056	0.608	-0.694	0.007	0.047	0.516	0.037	0.026	0.053	1.701
High(H)	3.798	0.019	0.069	0.958	1.969	0.008	0.046	0.574	0.103	0.044	0.081	1.877
H-L	5.844	0.013	0.049	0.950	5.527	-0.001	0.047	-0.058	0.154	0.061	0.063	3.380

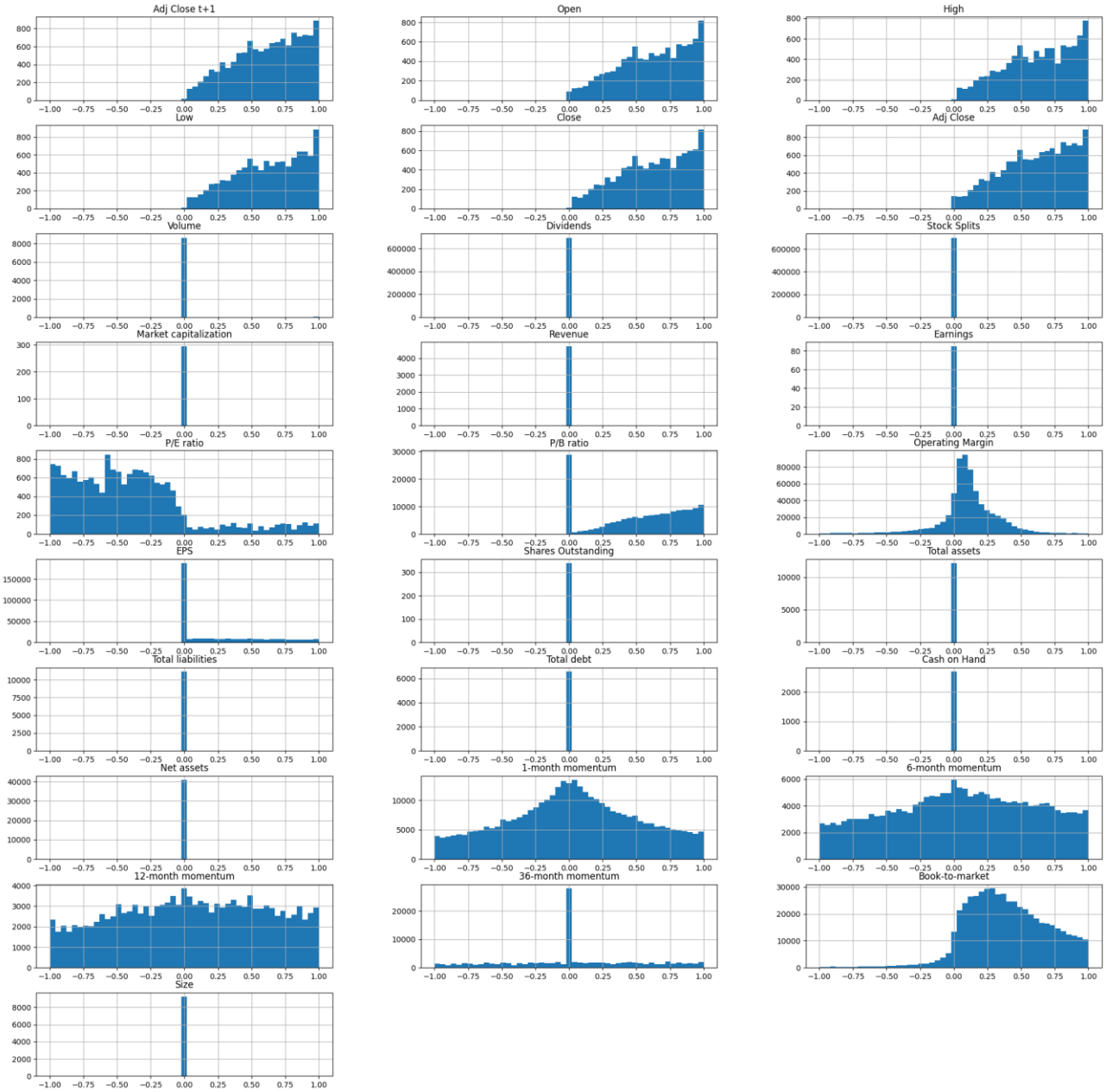
GBRT NN1 NN2

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-0.265	-0.018	0.053	-1.199	-0.113	-0.027	0.065	-1.434	-0.112	-0.019	0.063	-1.057
1	-0.050	-0.007	0.050	-0.457	-0.055	-0.017	0.054	-1.114	-0.048	-0.012	0.048	-0.866
2	-0.034	0.001	0.044	0.096	-0.036	-0.010	0.046	-0.726	-0.029	-0.006	0.044	-0.488
3	-0.022	0.009	0.045	0.679	-0.023	0.001	0.046	0.039	-0.016	0.000	0.043	0.035
4	-0.010	0.019	0.049	1.327	-0.012	0.006	0.045	0.456	-0.005	0.007	0.044	0.569
5	0.004	0.025	0.052	1.671	-0.002	0.013	0.046	0.952	0.005	0.014	0.045	1.077
6	0.024	0.030	0.057	1.783	0.009	0.018	0.046	1.396	0.016	0.022	0.049	1.563
7	0.057	0.030	0.065	1.585	0.022	0.028	0.048	2.009	0.030	0.032	0.052	2.107
8	0.127	0.026	0.068	1.301	0.056	0.038	0.057	2.291	0.055	0.041	0.061	2.363
High(H)	1.741	0.028	0.088	1.101	0.369	0.040	0.066	2.108	0.334	0.035	0.070	1.738
H-L	2.006	0.046	0.063	2.545	0.482	0.067	0.052	4.441	0.446	0.055	0.050	3.795

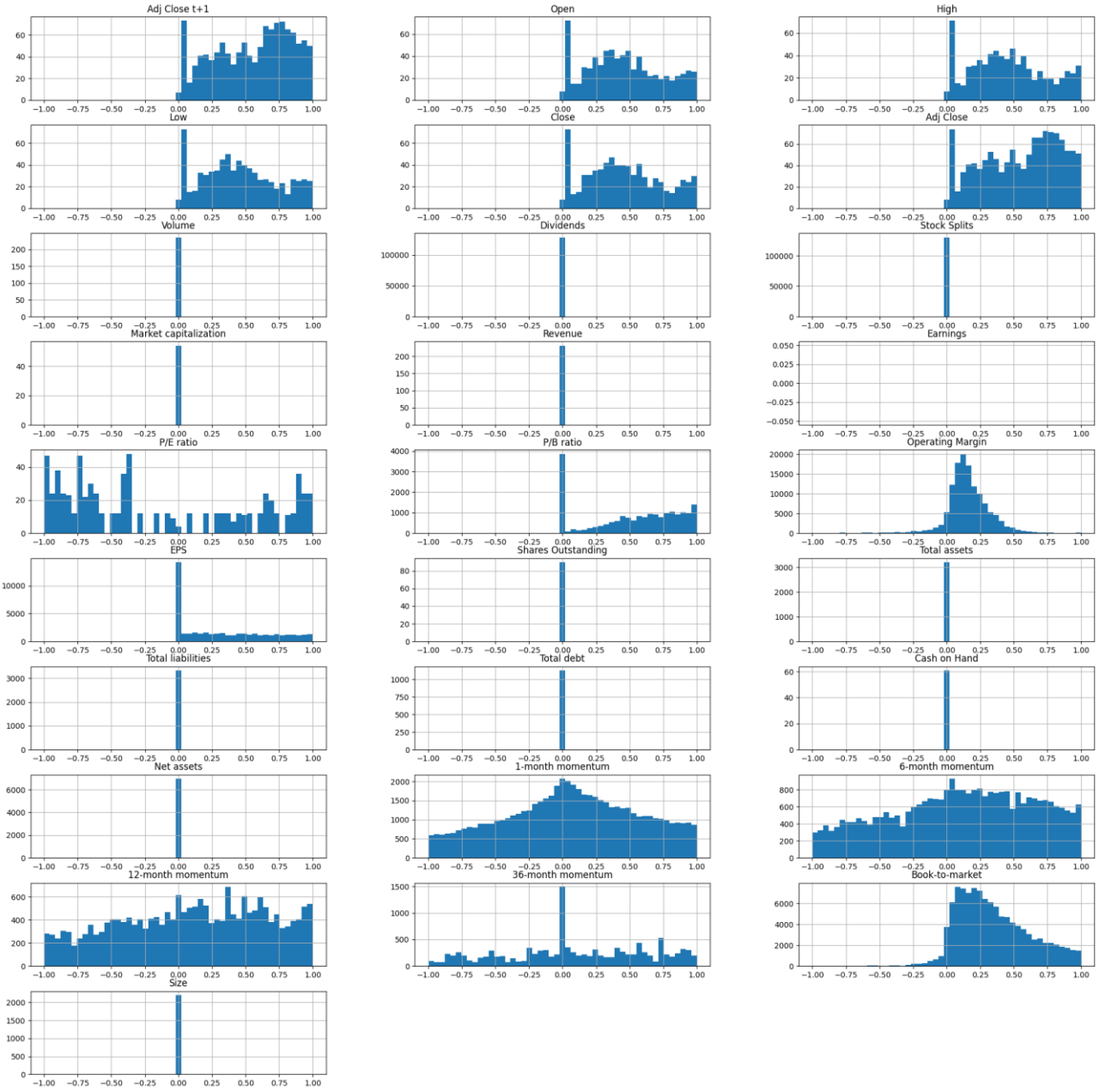
NN3 NN4 NN5

	Pred Avg_x	True Avg_x	SD_x	SR_x	Pred Avg_y	True Avg_y	SD_y	SR_y	Pred Avg	True Avg	SD	SR
Low(L)	-0.155	-0.011	0.054	-0.728	-0.168	-0.014	0.060	-0.836	-0.296	-0.003	0.049	-0.200
1	-0.065	-0.013	0.053	-0.871	-0.067	-0.009	0.055	-0.591	-0.183	-0.007	0.051	-0.494
2	-0.043	-0.009	0.048	-0.657	-0.044	-0.007	0.051	-0.475	-0.129	-0.007	0.048	-0.523
3	-0.027	-0.003	0.047	-0.219	-0.028	-0.000	0.046	-0.000	-0.078	-0.002	0.046	-0.114
4	-0.015	0.004	0.045	0.273	-0.016	0.004	0.045	0.323	-0.016	0.005	0.049	0.321
5	-0.003	0.013	0.046	0.944	-0.005	0.011	0.044	0.830	0.061	0.009	0.052	0.587
6	0.009	0.018	0.047	1.339	0.007	0.017	0.046	1.270	0.178	0.014	0.055	0.895
7	0.024	0.029	0.050	1.991	0.023	0.027	0.049	1.941	0.354	0.023	0.060	1.331
8	0.053	0.040	0.060	2.278	0.062	0.033	0.053	2.164	0.762	0.027	0.062	1.495
High(H)	0.406	0.038	0.076	1.740	0.490	0.030	0.066	1.587	2.111	0.025	0.076	1.158
H-L	0.561	0.050	0.047	3.687	0.658	0.044	0.043	3.581	2.408	0.028	0.051	1.931

Alle aktier


Adj Close t+1
Open
High
Low
Close
Adj Close
Volume
Dividends
Stock Splits
Market capitalization
Revenue
Earnings
P/E ratio
P/B ratio
Operating Margin
EPS
Shares Outstanding
Total assets
Total liabilities
Total debt
Cash on Hand
Net assets
1-month momentum
6-month momentum
12-month momentum
36-month momentum
Book-to-market
Size

Top 500 aktier



Bot 500 aktier