








Statistica piețelor financiare

Capitolul 13: Învățare automată



Daniel Traian PELE

Academia de Studii Economice din București
IDA Institute Digital Assets
Blockchain Research Center
AI4EFin Artificial Intelligence for Energy Finance
Academia Română, Institutul de Prognoză Economică
MSCA Digital Finance

Acronime folosite în acest material (1/3)

- **AI**: Artificial Intelligence (inteligență artificială)
- **ARCH**: AutoRegressive Conditional Heteroskedasticity (heteroscedasticitate condiționată autoregresivă)
- **AUC**: Area Under the (ROC) Curve (aria de sub curba ROC)
- **BET**: Bucharest Exchange Trading (index) — indicele de referință al Bursei de Valori București
- **BVB**: Bursa de Valori București
- **CV**: Cross-Validation (validare încrucișată)
- **DM**: Diebold–Mariano (test) — testul Diebold–Mariano de comparare a prognozelor
- **DSR**: Deflated Sharpe Ratio (raportul Sharpe deflatat)
- **EODHD**: EOD Historical Data (market-data provider) — furnizor de date de piață
- **GARCH**: Generalised AutoRegressive Conditional Heteroskedasticity (heteroscedasticitate condiționată autoregresivă generalizată)
- **GB**: Gradient Boosting (boosting pe gradient)
- **GJR-GARCH**: Glosten–Jagannathan–Runkle GARCH (GARCH asimetric Glosten–Jagannathan–Runkle)
- **HAR**: Heterogeneous AutoRegressive (model) — model autoregresiv heterogen
- **HS**: Historical Simulation (simularea istorică)

Acronime folosite în acest material (2/3)

- **IBM**: International Business Machines (company) — compania International Business Machines
- **MA**: Moving Average (medie mobilă)
- **ML**: Machine Learning (învățare automată)
- **MLP**: MultiLayer Perceptron (perceptron multistrat)
- **MSE**: Mean Squared Error (eroarea pătratică medie)
- **NN**: Neural Network (NNk: k hidden layers) — rețea neuronală (NNk: k straturi ascunse)
- **NN-ARCH**: Neural-Network ARCH (model ARCH cu rețea neuronală)
- **NN1**: Neural Network with 1 hidden layer (rețea neuronală cu un strat ascuns)
- **NN3**: Neural Network with 3 hidden layers (rețea neuronală cu 3 straturi ascunse)
- **NN4**: Neural Network with 4 hidden layers (rețea neuronală cu 4 straturi ascunse)
- **OLS**: Ordinary Least Squares (metoda celor mai mici pătrate)
- **OLS-3**: OLS with 3 predictors (size, book-to-market, momentum) — OLS cu 3 predictori (mărime, raportul valoare contabilă/valoare de piață, momentum)
- **OMV**: Österreichische Mineralölverwaltung (Administrația austriacă a uleiurilor minerale, grupul OMV)

Acronime folosite în acest material (3/3)

- **PSR**: Probabilistic Sharpe Ratio (raportul Sharpe probabilistic)
- **QGB**: Quantile Gradient Boosting (gradient boosting cu pierdere cuantilică)
- **QLIKE**: Quasi-LIKElihood (loss function) — funcția de pierdere de tip cvasi-verosimilitate
- **QR**: Quantile Regression (regresie cuantilică)
- **RBF**: Radial Basis Function (network) — rețea cu funcții de bază radiale
- **RF**: Random Forest (pădure aleatoare)
- **RMSE**: Root Mean Squared Error (rădăcina erorii pătratice medii)
- **ROC**: Receiver Operating Characteristic (curve) — curba caracteristică de operare
- **RV**: Realised Variance (varianța realizată)
- **SFE**: Statistics of Financial Markets, Exercises (Quantlet collection of the textbook) — colecția de Quantlet-uri a manualului Statistics of Financial Markets
- **SHAP**: SHapley Additive exPlanations (explicații aditive Shapley)
- **SSE**: Sum of Squared Errors (suma pătratelor erorilor)

Întrebarea de azi și traseul

- **Întrebarea**: pot metodele flexibile de predicție să prognozeze riscul și randamentele financiare mai bine decît modelele din capitolele 7–10 și cum verificăm onest acest lucru?
  - **machine learning** (ML, învățare automată): algoritmi care învață din date o regulă de predicție, judecați după acuratețea pe date noi
- **Traseul** capitolului
  - predicție și inferență; overfitting, deplasare și varianță; validarea pentru serii de timp fără look-ahead bias
  - ridge și lasso; arbori de decizie, random forest, gradient boosting; rețele neuronale și modelul ARCH cu rețea neuronală din manual
  - trei aplicații: volatilitatea realizată, semnul randamentelor, VaR 1%; data snooping; o lucrare de referință

Rezultatele învățării

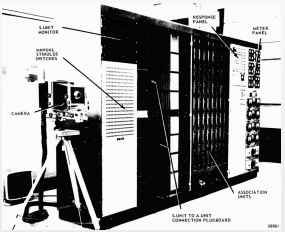
- Explicați diferența dintre predicție și inferență, precum și compromisul deplasare–variață
- Construiți o evaluare walk-forward cu purging și recunoașteți look-ahead bias și leakage-ul
- Estimați ridge, lasso, un arbore de regresie, un random forest, gradient boosting și o rețea neuronală mică
- Comparați prognozele prin R^2 în afara eșantionului, QLIKE și testul Diebold–Mariano, față de un reper adecvat
- Prognozați VaR 1% prin regresie cuantilică și corectați un raport Sharpe pentru numărul de strategii încercate

Bibliografie și instrumente

- Manual: Franke, Härdle & Hafner (2019), cap. 19 (rețele neuronale)
  - învățare statistică în Python: James et al. (2023); aprofundare: Hastie, Tibshirani & Friedman (2009)
  - machine learning în finanțe: Kelly & Xiu (2023); López de Prado (2018)
- Quantlet-urile Python ale capitolului: Quantlets/Ch_13
  - portate din Quantlet-urile SFE SFEnnarch și SFEnnjpyusd; modele din scikit-learn (mici, cu semințe fixe, cîteva minute în Colab)
- Notebook-ul cursului: deschideți în Google Colab
- Cursuri video: Machine learning in Financial Risk; Random Forests

Perceptronul, 1958

- Rosenblatt (1958): o mașină care învață să clasifice imagini ajustîndu-și ponderile după fiecare eroare
  - **perceptronul**: ieșirea este 1 dacă $w_0 + w^\top x > 0$ și 0 în caz contrar
  - x: vectorul intrărilor (semnalele fotocelulelor); w: ponderile lor, învățate din exemple; w_0: termenul de prag; $w^\top x = \sum_j w_j x_j$
  - Mark I Perceptron (Cornell, 1960): 400 de fotocelule, ponderi reglate de motoare electrice
- Trei repere ulterioare
  - 1986: retropropagarea erorii (back-propagation) antrenează rețele cu straturi ascunse Rumelhart, Hinton & Williams (1986)
  - 2001: random forest Breiman (2001b) și gradient boosting Friedman (2001)
  - 2024: Premiul Nobel pentru fizică acordat lui Hopfield și Hinton pentru rețelele neuronale



Mark I Perceptron, din manualul de utilizare

Figură: J.C. Hay, A.E. Murray (1960); domeniu public; Wikimedia Commons

Două culturi ale modelării statistice

- Breiman (2001a), statistician la Berkeley, descrie două culturi
  - **modelarea datelor**: presupunem un model stochastic (regresia liniară $y = x^\top \beta + \varepsilon$, GARCH), îl estimăm și îi testăm parametrii
  - **modelarea algoritmică**: tratăm mecanismul ca necunoscut; căutăm o regulă $\hat{f}$ care prezice bine y pe date noi
- Cursul a folosit prima cultură în capitolele 2–11
  - acest capitol o adaugă pe a doua și analizează în ce condiții este utilă în finanțe



Evans Hall, Universitatea California din Berkeley, sediul Departamentului de Statistică

Foto: Gabriel Classon (2021); CC BY 2.0; Wikimedia Commons

Predicție și inferență

- **Inferența**: este $\beta_j \neq 0$? cît de mare este efectul? (erori standard, teste, Capitolul 6)
  - exemplu: există efectul de levier (GJR-GARCH, Capitolul 9)?
- **Predicția**: cît de aproape este $\hat{y}$ de y pe date pe care modelul nu le-a văzut niciodată?
  - exemplu: volatilitatea săptămînii viitoare; VaR 1% de mîine
  - coeficienții pot fi greu de interpretat; contează eroarea pe date noi
- Cele două obiective pot intra în conflict
  - un estimator deplasat (ridge, lasso) poate prezice mai bine decît estimatorul nedeplasat OLS
  - un model care prezice bine nu demonstrează un mecanism cauzal

Învățarea supervizată: notații și funcția de pierdere (1/2)

- Datele: perechile (x_t, y_t), $t = 1, \ldots, n$
  - t: indicele observației (o zi, o săptămînă); n: numărul de observații
  - $x_t \in \mathbb{R}^p$: **variabilele explicative** (features), p numere cunoscute la momentul t (volatilități și randamente trecute)
  - y_t: **ținta** (target), mărimea pe care vrem să o prognozăm (volatilitatea săptămînii viitoare)
- Două tipuri de probleme
  - **regresie**: y este o variabilă continuă (volatilitate, randament)
  - **clasificare**: y este o clasă (creștere/scădere, neplată; Capitolul 12)

Învățarea supervizată: notații și funcția de pierdere (2/2)

- Învățarea constă în alegerea regulii $\hat{f}$ din familia $\mathcal{F}$ care minimizează **pierderea** medie $\frac{1}{n}\sum_t L(y_t, f(x_t))$
  - $f(x_t)$: prognoza dată de regula f pentru observația t; $L(y, \hat{y})$: costul prognozei $\hat{y}$ cînd se observă y
  - $\mathcal{F}$: regulile candidate (toate regresiile liniare, toți arborii de adîncime 3); $\hat{f}$: regula aleasă pe baza datelor
- Funcțiile de pierdere folosite în curs
  - eroarea pătratică $(y - \hat{y})^2$ (MSE, mean squared error); QLIKE pentru variațe (Capitolul 8)
  - pierderea cuantilică (pinball loss) pentru VaR (Capitolul 10); log loss pentru probabilități (Capitolul 12)
- Un **hiperparametru** se alege înainte de estimare
  - exemple: adîncimea arborelui, penalizarea λ, numărul de neuroni
  - parametrii (coeficienți, ponderi) se învață prin estimare

Avantajele și limitele învățării automate în finanțe

- **Ajută**: ținte cu structură și cu suficiente date
  - volatilitatea este persistentă (capitolele 8–9): este mult de prezis, iar efectele neliniare sînt posibile
  - secțiuni transversale cu mii de acțiuni și multe caracteristici (studiul de caz)
- **Dificultăți**: un **raport semnal–zgomot** mic
  - randamentele zilnice sînt aproape imprevizibile (Capitolul 7): un model flexibil învață zgomotul
  - piețele se schimbă (regimuri, crize): trecutul este un set de antrenare imperfect
  - o singură istorie a prețurilor: experimentul nu poate fi repetat, iar observațiile sînt dependente

Eroarea de antrenare și eroarea de test

- **Eroarea de antrenare**: pierderea medie pe datele folosite la estimare; **eroarea de test**: pe date noi din același proces
  - eroarea de antrenare scade întotdeauna cînd modelul devine mai flexibil
- **Overfitting**: modelul învață zgomotul din eșantionul de antrenare
  - simptom: o eroare de antrenare mică și o eroare de test mare
  - un arbore cu cîte o frunză pentru fiecare observație are eroare de antrenare zero, dar prognoze slabe
- **Underfitting**: modelul este prea rigid pentru a surprinde semnalul
  - exemplu: o prognoză constantă a volatilității într-o lume cu volatility clustering

Descompunerea deplasare–varianță (1/2)

- Modelul: $y = f(x) + \varepsilon$, $E[\varepsilon] = 0$, $\mathrm{Var}(\varepsilon) = \sigma^2$
  - f: relația adevărată, necunoscută, dintre x și y; ε: zgomotul, imprevizibil pe baza lui x, cu varianța σ^2
  - $\hat{f}$: estimația lui f obținută pe un eșantion de antrenare aleator; alt eșantion ar da alt $\hat{f}$
- Eroarea de test așteptată într-un punct nou x_0, cu $y_0 = f(x_0) + \varepsilon_0$:

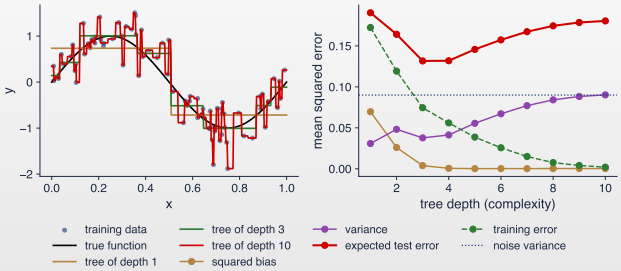
$$E\left[(y_0 - \hat{f}(x_0))^2\right] = \underbrace{\left(E[\hat{f}(x_0)] - f(x_0)\right)^2}_{\text{deplasare}^2} + \underbrace{\mathrm{Var}(\hat{f}(x_0))}_{\text{varianță}} + \underbrace{\sigma^2}_{\text{zgomot}}$$

  - $E[\cdot]$: media peste toate eșantioanele de antrenare posibile și peste zgomotul ε_0
  - deplasarea: distanța dintre estimația medie și valoarea adevărată; varianța: cît se modifică $\hat{f}(x_0)$ de la un eșantion la altul

Descompunerea deplasare–varianță (2/2)

- Derivarea: scriem $y_0 - \hat{f} = (f - E\hat{f}) + (E\hat{f} - \hat{f}) + \varepsilon_0$, toate în punctul x_0
  - ridicăm la pătrat și luăm media: cele trei pătrate dau deplasarea^2, varianța și σ^2
  - termenii micști au media 0: $E[E\hat{f} - \hat{f}] = 0$, iar ε_0 este independent de eșantionul de antrenare
- Interpretarea descompunerii
  - un model rigid: deplasare mare, varianță mică
  - un model flexibil: deplasare mică, varianță mare
  - σ^2 este pragul minim: niciun model nu coboară sub el; la randamentele zilnice, σ^2 reprezintă aproape toată eroarea

Deplasarea și varianța arborilor de regresie



- Simulare: $y = \sin(2\pi x) + \varepsilon$, $\sigma^2 = 0{,}09$, $n = 80$, 300 de eșantioane de antrenare; stînga: un eșantion cu arbori de adîncime 1, 3 și 10
- Adîncimea 1: deplasare^2 0,070, varianță 0,031; adîncimea 10: deplasare^2 0,000, varianță 0,090; eroarea de test este minimă la adîncimea 3 (0,132)

SFM_ch13_validation

Interpretarea erorii de test în formă de U

- Eroarea de antrenare scade de la 0,172 la 0,002; eroarea de test scade, apoi crește
  - adîncimea 10 are eroare de antrenare aproape zero și o eroare de test de 0,180, mai mare decît la adîncimea 3
- Consecințe practice
  - nu alegeți niciodată un model după eroarea de antrenare
  - complexitatea este un hiperparametru: se alege pe date care nu au fost folosite la estimare
  - în finanțe zgomotul este mare, deci cel mai bun model este de obicei mai simplu decît în această simulare

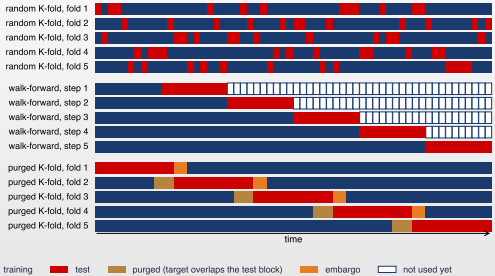
Eșantioanele de antrenare, validare și test

- Trei roluri pentru date
  - **antrenare**: estimăm parametrii fiecărui model candidat
  - **validare**: alegem hiperparametrii și modelul
  - **test**: măsurăm o singură dată eroarea alegerii finale
- Orice decizie luată pe baza eșantionului de test îl transformă într-un eșantion de validare
  - a încerca zece modele pe datele de test și a raporta cel mai bun înseamnă selecție de model, nu testare
- **Validarea încrucișată K-fold** (cross-validation): împărțim datele în K părți; fiecare parte este o dată setul de validare, iar celelalte antrenează modelul
  - validă pentru date i.i.d.; pentru o serie de timp doar în condiții stricte (fără dependență rămasă în erori) Bergmeir, Hyndman & Koo (2018)

Look-ahead bias și leakage

- **Look-ahead bias**: prognoza pentru ziua t folosește informație care nu era disponibilă la închiderea zilei $t-1$
  - variabile calculate cu prețuri viitoare (o medie mobilă centrată, o medie pe tot eșantionul)
  - scalare sau standardizare cu statistici ale întregului eșantion; date revizuite; componența de azi a indicelui (survivorship)
- **Leakage**: informația despre ținta de test ajunge în eșantionul de antrenare
  - ținte suprapuse: randamentul pe 21 de zile de la ziua t și cel de la ziua $t+1$ au 20 de zile comune
  - o partiție aleatoare pune ziua $t+1$ în antrenare și ziua t în test: modelul a văzut răspunsul
- Ambele fac un model să pară mai bun decît poate fi în timp real

Trei scheme de validare



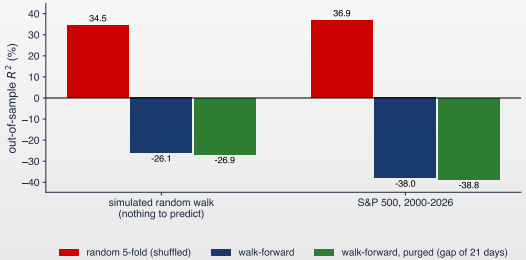
- **Walk-forward**: antrenăm pe trecut, testăm pe blocul următor și avansăm; fereastra extinsă păstrează tot trecutul
- **K-fold cu purging și embargo** López de Prado (2018): eliminăm rîndurile de antrenare ale căror ținte se suprapun cu blocul de test (purging) și cîteva rînduri de după el (embargo)

SFM_ch13_validation

Validarea walk-forward, pas cu pas

- Schema folosită în acest capitol pentru S&P 500 (date din 8 aprilie 2008)
  - cîte o partiție pentru fiecare an calendaristic din 2013: 14 partiții; modelul se reestimează în fiecare ianuarie
  - fereastra de antrenare: toate zilele anterioare (extinsă); test: fiecare zi a anului
- Purging: ținta este varianța medie din următoarele 5 zile
  - pentru 2013 se elimină ultimele 5 rînduri de antrenare (de la 24 decembrie 2012): țintele lor ajung în 2013
  - partiția pentru 2013 antrenează pe 1 188 de zile și testează pe 252; partiția pentru 2026 antrenează pe 4 458 de zile
- În interiorul fiecărei ferestre de antrenare, hiperparametrii se aleg printr-o împărțire walk-forward doar a datelor de antrenare
  - nu este nevoie de embargo: blocul de test vine întotdeauna după cel de antrenare

Leakage în practică: partiții aleatoare pe ținte suprapuse



- Random forest pentru randamentul din următoarele 21 de zile, pe baza randamentelor și volatilităților trecute; R^2 în afara eșantionului față de media de antrenare
- Mers aleator simulat: 5-fold aleator 34,5%, walk-forward −26,1%; S&P 500 (6 631 de zile): 36,9% și −38,0%

SFM_ch13_validation

Interpretare: leakage în prognoza unui mers aleator

- Mersul aleator nu are nimic de prezis, totuși partițiile aleatoare raportează $R^2 = 34{,}5\%$
  - random forest memorează zilele vecine, ale căror ținte au 20 din 21 de randamente comune cu ziua de test
- Walk-forward dă un R^2 negativ: random forest învață zgomotul și este inferior mediei de antrenare
  - purging-ul (eliminarea a 21 de zile) schimbă puțin rezultatul, deoarece în walk-forward doar rîndurile de la graniță se suprapun
- Regulă: cu ținte suprapuse sau date dependente, nu amestecați niciodată ordinea temporală

OLS cu multe variabile explicative corelate

- OLS (ordinary least squares, metoda celor mai mici pătrate): $\hat{\beta} = \arg\min_\beta \sum_t (y_t - x_t^\top \beta)^2 = (X^\top X)^{-1} X^\top y$
  - X: matricea $n \times p$ cu rîndul t egal cu $x_t^\top$; y: vectorul țintelor; $\arg\min_\beta$: valoarea lui β care minimizează suma
  - nedeplasat, dar varianța lui, $\sigma^2 (X^\top X)^{-1}$ (σ^2: varianța erorii), crește foarte mult cînd variabilele sînt multe sau puternic corelate
- Variabilele de volatilitate sînt puternic corelate: varianța realizată de azi, din această săptămînă și din această lună
  - OLS poate da coeficienți mari, de semne opuse, care se compensează în eșantion și eșuează în afara lui
- **Regresia penalizată**: minimizăm eroarea pătratică plus o penalizare pentru mărimea lui β
  - acceptăm o deplasare mică pentru o reducere mare a variației (compromisul din secțiunea 2)

Regresia ridge (1/2)

- Hoerl & Kennard (1970): minimizăm eroarea pătratică plus o penalizare a pătratelor coeficienților

$$\hat{\beta}^{ridge} = \arg\min_{\beta} \sum_t (y_t - x_t^\top \beta)^2 + \lambda \sum_j \beta_j^2 = (X^\top X + \lambda I)^{-1} X^\top y$$

- Notațiile
  - β_j: coeficientul variabilei j; $\sum_j \beta_j^2$: termenul de penalizare, mare cînd coeficienții sînt mari
  - $\lambda \geq 0$: intensitatea penalizării; $\lambda = 0$ dă OLS, $\lambda \to \infty$ dă $\hat{\beta} \to 0$
  - I: matricea identitate $p \times p$
- Consecința
  - $X^\top X + \lambda I$ este întotdeauna inversabilă pentru $\lambda > 0$: ridge funcționează chiar și cu mai multe variabile decît observații

Regresia ridge (2/2)

- **O singură variabilă**, centrată (cu media 0): $S_{xx} = \sum_t x_t^2$, $S_{xy} = \sum_t x_t y_t$

$$\hat{\beta}^{\,ridge} = \frac{S_{xy}}{S_{xx} + \lambda} = \frac{S_{xx}}{S_{xx} + \lambda}\,\hat{\beta}^{\,OLS}, \qquad \hat{\beta}^{\,OLS} = \frac{S_{xy}}{S_{xx}}$$

  - factorul $S_{xx}/(S_{xx} + \lambda)$ este în $(0, 1]$: fracțiunea din coeficientul OLS păstrată de ridge
- Exemplu: $S_{xx} = 100$, $S_{xy} = 30$
  - OLS: $30/100 = 0{,}30$
  - ridge cu $\lambda = 50$: $30/150 = 0{,}20$ (factor de contracție 0,67)
- Ridge contractă toți coeficienții spre zero, dar nu anulează niciunul exact

Lasso

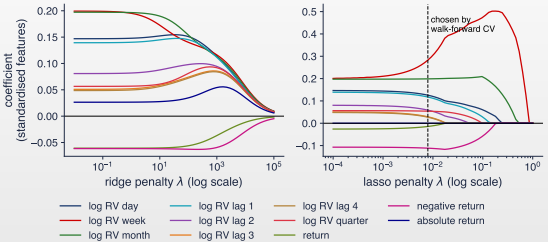
- Tibshirani (1996): $\hat{\beta}^{lasso} = \arg\min_\beta \sum_t (y_t - x_t^\top \beta)^2 + \lambda \sum_j |\beta_j|$
  - valoarea absolută are un punct unghiular în 0: unii coeficienți devin **exact zero** (selecția variabilelor)
- O singură variabilă: **soft thresholding** (prag de contracție)
  - $\hat{\beta}^{lasso} = \text{sign}(S_{xy})\ \max(|S_{xy}| - \lambda/2,\ 0)/S_{xx}$
  - $\text{sign}(S_{xy})$: semnul, $+1$ sau -1; dacă $|S_{xy}| \leq \lambda/2$, coeficientul este exact 0
  - aceleași date, $\lambda = 20$: $(30 - 10)/100 = 0{,}20$; zero pentru $\lambda \geq 60$
- În practică
  - standardizăm întîi variabilele (penalizarea depinde de unitățile lor), doar cu media și abaterea standard din antrenare
  - alegem λ prin validare încrucișată walk-forward; elastic net combină cele două penalizări



Robert Tibshirani, Universitatea Stanford

Foto: R. Tibshirani (2012); CC BY-SA 3.0; Wikimedia Commons

Traiectoriile ridge și lasso pentru variabilele de volatilitate



- Ținta: logaritmul mediei variabilei proxy a variației din următoarele 5 zile, S&P 500, 2008–2012 (1 193 de zile); 11 variabile standardizate (secțiunea 7)
- Lasso cu $\lambda = 0{,}0079$ (CV walk-forward): rămîn 10 din 11 variabile; coeficientul săptămînal crește de la $+0{,}199$ (OLS) la $+0{,}283$, iar randamentul absolut iese din model
- Matricea variabilelor este exact coliniară, $|r| = r - 2\min(r, 0)$ (rang 10): coeficienții OLS pentru r, $\min(r, 0)$, $|r|$ nu sînt unici (afișată: soluția de normă minimă); ceilalți sînt unici

SFM_ch13_shrinkage_trees

Interpretarea traiectoriilor

- Ridge (stînga): toți coeficienții se contractă lin și împreună; variabilele corelate își împart ponderea
  - randamentul negativ își păstrează semnul negativ pînă la capăt: piețele în scădere cresc volatilitatea viitoare (efectul de levier, Capitolul 9)
- Lasso (dreapta): variabilele intră pe rînd pe măsură ce λ scade
  - varianța săptămînală intră prima: cea mai informativă variabilă luată singură
  - la λ ales, lasso păstrează structura HAR (zi, săptămînă, lună) plus randamentul negativ, $-0{,}112$
- Lasso selectează, nu testează: un coeficient zero nu este un p-value

Arborii de regresie

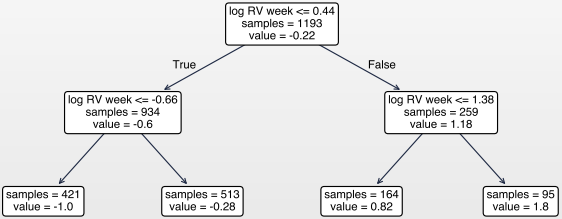
- Un **arbore de regresie** împarte spațiul variabilelor în dreptunghiuri și prezice media lui y în fiecare **frunză**
  - regula de împărțire: o variabilă j și un prag s: stînga $x_j \leq s$, dreapta $x_j > s$
- **Estimarea greedy**: în fiecare nod alegem (j, s) care minimizează $\mathrm{SSE} = \sum_{\textbf{stînga}}(y - \bar{y}_L)^2 + \sum_{\textbf{dreapta}}(y - \bar{y}_R)^2$
  - SSE: suma pătratelor erorilor; $\bar{y}_L$, $\bar{y}_R$: mediile lui y în grupul din stînga și în cel din dreapta (cele două prognoze)
  - repetăm în fiecare jumătate pînă la o regulă de oprire: adîncimea maximă, numărul minim de observații dintr-o frunză
- Puncte tari și puncte slabe
  - efecte neliniare și interacțiuni fără a le specifica; insensibil la transformările monotone ale lui x
  - variață mare: o mică schimbare a datelor poate schimba tot arborele (secțiunea 2)

Exemplu rezolvat: o împărțire, pas cu pas

s	$\bar{y}_L$	$\bar{y}_R$	SSE
0,70	0,70	1,52	1,708
0,90	0,80	1,68	1,248
1,25	0,80	1,97	0,227
1,75	1,00	2,15	0,505
2,20	1,22	2,20	1,468

- Șase săptămîni: x = volatilitatea din această săptămînă (%), y = cea din săptămîna următoare
  - x: 0,6; 0,8; 1,0; 1,5; 2,0; 2,4; y: 0,7; 0,9; 0,8; 1,6; 2,1; 2,2
  - fără împărțire: $\bar{y} = 1{,}38$, SSE $= 2{,}268$
- Candidați: mijloacele intervalelor dintre valorile consecutive ale lui x
  - cea mai bună împărțire $s = 1{,}25$: frunzele 0,80 și 1,97, SSE $= 0{,}227$
  - o singură împărțire explică $1 - 0{,}227/2{,}268 = 0{,}90$ din variație

Un arbore de adîncime 2 pentru volatilitatea săptămînii viitoare



- S&P 500, 2008–2012, ținta și variabilele în logaritmul variației (%^{2} pe zi); *value*: media țintei în nod
- Interpretare: prima împărțire, log RV săptămînal $\leq 0{,}44$, corespunde unei volatilități anuale de circa 19,8%; toate împărțirile folosesc varianța săptămînală, variabila aleasă prima de lasso

SFM_ch13_shrinkage_trees

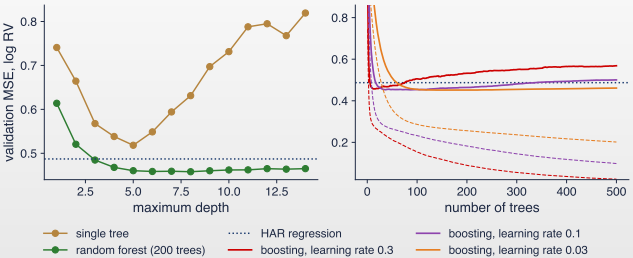
Random forest

- Breiman (2001b): media a B arbori adînci, fiecare crescut pe un eșantion bootstrap (Capitolul 2) din datele de antrenare (**bagging**)
  - la fiecare împărțire se încearcă doar o submulțime aleatoare de variabile: arborii devin mai puțin corelați
- Efectul mediei: B arbori, fiecare cu varianța σ^2 și corelația ρ între oricare doi
  - $\mathrm{Var}(\text{media}) = \rho\sigma^2 + (1-\rho)\sigma^2/B$: partea $\rho\sigma^2$, comună tuturor arborilor, nu scade cu B
  - cu $\rho = 0{,}3$, $\sigma^2 = 1$: $B = 1$: 1,000; $B = 10$: 0,370; $B = 300$: 0,302
  - mai mulți arbori nu produc overfitting; arborii mai puțin corelați (mai puține variabile la o împărțire) coboară pragul $\rho\sigma^2$
- Hiperparametrii folosiți aici: 300 de arbori, cel puțin 20 de observații într-o frunză, jumătate din variabile încercate la fiecare împărțire

Gradient boosting

- Friedman (2001): construim modelul pas cu pas, fiecare arbore mic nou învățînd ce au ratat cei anteriori
  - pornim de la $F_0 = \bar{y}$; la pasul m estimăm un arbore mic h_m pe reziduurile $y_t - F_{m-1}(x_t)$
  - actualizăm $F_m = F_{m-1} + \nu\, h_m$, cu **rata de învățare** $0 < \nu \leq 1$
  - pentru pierderea pătratică, reziduurile sînt minus gradientul pierderii: de aici numele
- Random forest comparat cu boosting
  - random forest: arbori adînci în paralel; media reduce varianța
  - boosting: arbori mici în secvență; fiecare pas reduce deplasarea; prea mulți pași produc overfitting
- Funcționează orice pierdere derivabilă: pierderea cuantilică dă quantile boosting (secțiunea 9)

Un arbore, un random forest și boosting pe date de volatilitate



- S&P 500, antrenare 2008–2018 (2 698 de zile), validare 2019–2022 (1 008 zile); linii întrerupte în dreapta: MSE de antrenare; punctat: regresia HAR (0,487)
- Un singur arbore: minimul 0,518 la adîncimea 5, 0,819 la adîncimea 14; random forest: 0,458 (adîncimea 8) și tot 0,465 la adîncimea 14

SFM_ch13_shrinkage_trees

Interpretare: media și oprirea timpurie

- Arborele singur intră repede în overfitting; random forest este aproape insensibil la adîncime
  - media elimină cea mai mare parte din varianța arborilor adînci; random forest depășește HAR în această perioadă de validare
- Boosting: eroarea de validare atinge un minim, apoi crește, în timp ce eroarea de antrenare continuă să scadă
  - rata de învățare 0,3: minimul 0,457 după 13 arbori, 0,568 după 500 de arbori
  - rata de învățare 0,03: minimul 0,451 după 154 de arbori, 0,461 după 500 de arbori: pașii mici sînt mai siguri
- Numărul de arbori din boosting este un hiperparametru; numărul de arbori dintr-un random forest nu este

Perceptronul și perceptronul multistrat (1/2)

- Un **neuron**: $h = g(b + w^\top x)$, o sumă ponderată a intrărilor trecută printr-o **funcție de activare** g
  - x: intrările; w: ponderile lor; b: o constantă (termenul liber al neuronului); h: ieșirea neuronului
  - g: o funcție neliniară fixată; fără ea, o rețea de neuroni ar fi doar o regresie liniară
- Funcțiile de activare
  - perceptronul folosește o funcție treaptă (0 sau 1)
  - rețelele moderne folosesc funcții g netede sau liniare pe porțiuni: logistică $1/(1 + e^{-z})$, tanh, ReLU $\max(0, z)$

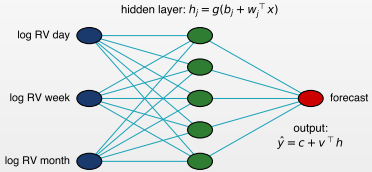
Perceptronul și perceptronul multistrat (2/2)

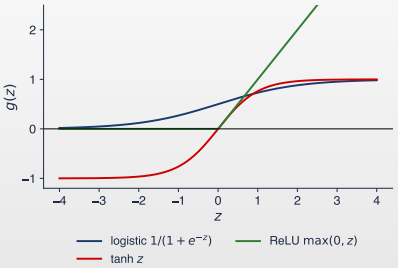
- **MLP** (multilayer perceptron, perceptron multistrat) cu un strat ascuns de K neuroni

$$\hat{y} = c + \sum_{k=1}^{K} v_k \, g(b_k + w_k^\top x)$$

  - w_k, b_k: ponderile și constanta neuronului ascuns k; v_k: ponderea neuronului k în ieșire; c: constanta ieșirii
  - interpretarea: un model liniar în K variabile neliniare $g(b_k + w_k^\top x)$, învățate și ele din date
- Numărul de parametri pentru 11 intrări și 16 neuroni
  - stratul ascuns: 11×16 ponderi $+16$ constante $= 192$; ieșirea: $16 + 1 = 17$; în total 209
- **Aproximarea universală** Hornik, Stinchcombe & White (1989): cu suficienți neuroni, un singur strat ascuns aproximează orice funcție continuă pe o mulțime compactă

O rețea mică și funcțiile ei de activare





- Stînga: trei intrări HAR, cinci neuroni ascunși, o prognoză; fiecare linie este o pondere
- Dreapta: funcțiile logistică și tanh se saturează pentru $|z|$ mare; ReLU $= \max(0, z)$ este liniară pentru $z > 0$ și se antrenează mai repede

SFM_ch13_neural_networks

Antrenarea unei rețele

- Minimizăm pierderea prin **coborîre pe gradient**: $\theta \leftarrow \theta - \eta \nabla_\theta L$
  - θ: toate ponderile și constantele; $\nabla_\theta L$: gradientul, vectorul derivatelor parțiale ale pierderii; $\eta > 0$: rata de învățare (mărimea pasului)
  - fiecare pas deplasează θ în direcția în care pierderea scade cel mai repede
  - **retropropagarea** calculează $\nabla_\theta L$ strat cu strat, cu regula de derivare a funcțiilor compuse Rumelhart, Hinton & Williams (1986)
  - o **epocă** = o trecere prin datele de antrenare; Adam este o metodă de gradient cu pași adaptivi
- Pierderea nu este convexă: ponderi inițiale diferite dau rețele diferite
  - fixăm semințele aleatoare și facem media a cinci rețele (un **ansamblu**)
- Împotriva overfitting-ului
  - standardizăm intrările și ținta cu statisticile de antrenare; o penalizare L_2 a ponderilor (ca la ridge); un număr fix de epoci
  - date puține, rețea mică: aici este suficient un strat ascuns cu 16 neuroni

Deep Blue și deep learning



IBM Deep Blue, Computer History Museum

Foto: Anton Chiang (2011); CC BY 2.0; Wikimedia Commons



Geoffrey Hinton, Premiul Nobel pentru fizică 2024

Foto: Arthur Petron (2024); CC BY-SA 4.0; Wikimedia Commons

- 1997: Deep Blue l-a învins pe campionul mondial la șah mai ales prin căutare și reguli de evaluare scrise de oameni, nu prin învățare din date
- Deep learning (multe straturi, volume uriașe de date) domină azi imaginile și textul; la prețurile financiare datele sînt puține și zgomotoase, deci rețelele mici sînt regula

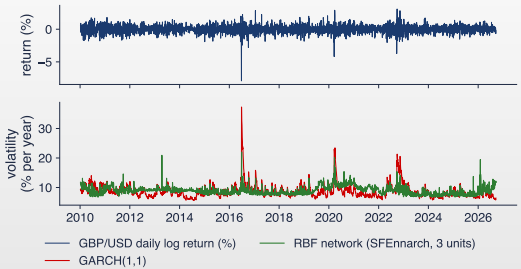
ARCH cu rețea neuronală: modelul din manual (1/2)

- Franke, Härdle & Hafner (2019), cap. 19: $r_t = f(x_{t-1}) + s(x_{t-1})\,\varepsilon_t$, cu f și s^2 estimate prin rețele neuronale
  - r_t: randamentul zilei t; x_{t-1}: intrările cunoscute la închiderea zilei $t-1$
  - $f(x_{t-1})$: media condiționată; $s^2(x_{t-1})$: varianța condiționată; ε_t: un șoc cu media 0 și varianța 1
- ARCH (Engle (1982)) este cazul particular $s^2(x) = \omega + \sum_i \alpha_i r_{t-i}^2$
  - $\omega > 0$, $\alpha_i \geq 0$: varianța este o funcție liniară de pătratele randamentelor trecute
  - rețeaua nu impune nicio formă funcției s^2

ARCH cu rețea neuronală: modelul din manual (2/2)

- Quantlet-ul SFEnnarch: GBP/USD, intrările x = ultimele 3 randamente, randamentul obligațiunilor germane pe 10 ani, randamentul aurului
  - pasul 1: o rețea $\hat{f}$ pentru medie; reziduurile $e_t = r_t - \hat{f}(x_{t-1})$
  - pasul 2: o rețea pentru e_t^2 dă varianța condiționată $\hat{s}^2(x_{t-1})$
- Rețea **RBF** (radial basis function, cu funcții de bază radiale) cu 3 unități, ca în Quantlet
  - unitatea j: $\exp(-\|x - c_j\|^2/s_j^2)$; c_j: centrul ei (obținut prin clustering); s_j: lățimea ei; $\|x - c_j\|$: distanța de la x la c_j
  - unitatea este aproape de 1 cînd x este lîngă c_j și aproape de 0 departe de el; ieșirea: o combinație liniară a celor 3 unități
  - nimic nu impune $\hat{s}^2 > 0$: o rețea pentru varianță are nevoie de un prag minim, pe care GARCH îl garantează prin restricțiile sale

ARCH cu rețea neuronală pentru GBP/USD



- Date zilnice din 7 ianuarie 2010 (4 048 de zile); volatilitatea anualizată a rețelei RBF și a modelului GARCH(1,1) ($\hat{\alpha} = 0{,}077$, $\hat{\beta} = 0{,}892$)
- Volatilitatea medie: rețeaua 8,9%, GARCH 8,8%; corelația celor două traiectorii 0,45 (un MLP mic: 0,43)

SFM_ch13_neural_networks

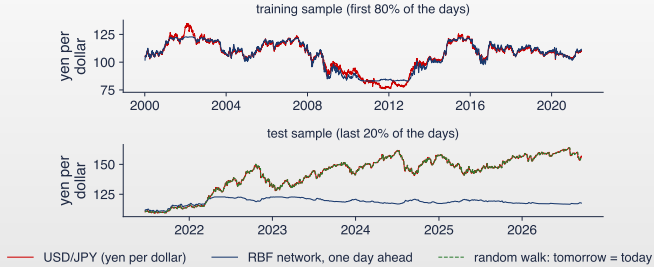
Interpretare: ARCH cu rețea neuronală și GARCH

- După votul pentru Brexit, GARCH atinge maximul de 37,3% (28 iunie 2016); rețeaua doar 22,0%
  - rețeaua vede doar ultimele 3 randamente: un randament mare îi crește varianța timp de 3 zile, apoi efectul dispare
  - GARCH(1,1) (Capitolul 9): $\sigma_t^2 = \omega + \alpha\, r_{t-1}^2 + \beta\, \sigma_{t-1}^2$, cu σ_{t-1}^2 varianța condiționată de ieri
  - termenul $\beta\, \sigma_{t-1}^2$ transmite tot trecutul: volatility clustering are nevoie de memorie, nu doar de flexibilitate
- Rețeaua pentru medie explică 0,08% din randamente în eșantion: cursurile de schimb sînt aproape un mers aleator
  - partea utilă a modelului NN-ARCH este varianța, ca la GARCH
- Lecția: variabilele bune (mediile HAR, secțiunea 7) contează mai mult decît tipul de rețea

Prognoza nivelului unui curs de schimb: SFEnnjpyusd

- Quantlet-ul SFEnnjpyusd prognozează cursul yen–dolar din ultimele 3 valori, cu o rețea RBF cu 3 unități
  - primele 80% din zile pentru antrenare, ultimele 20% pentru test; aici USD/JPY din 2000, cu separarea la 21 iunie 2021
- Două capcane ale prognozei nivelului
  - o serie de niveluri pare previzibilă deoarece $P_t \approx P_{t-1}$: reperul corect este mersul aleator $\hat{P}_t = P_{t-1}$ (Capitolul 7)
  - o rețea cu unități mărginite nu poate extrapola: nivelurile de test urcă pînă la 163,9, iar maximul din antrenare a fost 134,9
- Codul original rescalează prognozele de test cu intervalul seriei de test, necunoscut în timp real; aici se folosește intervalul din antrenare

Rețeaua RBF comparată cu mersul aleator



- RMSE (root mean squared error, rădăcina erorii pătratice medii) în yeni: antrenare: rețeaua 2,64, mersul aleator 0,66; test: rețeaua 27,56, mersul aleator 0,89
- Interpretare: în afara eșantionului rețeaua este de 31 de ori mai slabă decît „mîine = azi"; pe randamente, aceeași rețea are un R^2 în afara eșantionului de $-0{,}14\%$

SFM_ch13_neural_networks

Ținta și variabilele explicative (1/2)

- Variabila proxy zilnică a variației din Capitolul 8 (fără date intrazilnice): o estimație a variației zilei t din prețurile de deschidere, maxim, minim și închidere

$$v_t = o_t^2 + \tfrac{1}{2}(u_t - d_t)^2 - (2\ln 2 - 1)c_t^2$$

  - o_t: randamentul logaritmic peste noapte (de la închiderea precedentă la deschidere)
  - u_t, d_t, c_t: logaritmii maximului, minimului și închiderii față de deschidere (termenul Garman–Klass, Garman & Klass (1980))

- **Ținta**: $y_t = \ln\left(\frac{1}{5}\sum_{j=1}^{5} v_{t+j}\right)$, logaritmul variației realizate din săptămîna următoare
  - $j = 1, \ldots, 5$: următoarele cinci zile de tranzacționare; ținta este cunoscută abia la închiderea zilei $t + 5$
  - logaritmul apropie ținta de distribuția Normală și face prognozele pozitive după exp

Ținta și variabilele explicative (2/2)

- **Variabilele explicative** la închiderea zilei t
  - HAR: $\ln v_t$, logaritmul mediei pe 5 zile (săptămîna) și pe 22 de zile (luna)
  - extinse (11): în plus 4 laguri ale lui $\ln v_t$ ($\ln v_{t-1}, \ldots, \ln v_{t-4}$), media pe 66 de zile, r_t, $\min(r_t, 0)$, $|r_t|$
- Notațiile
  - r_t: randamentul logaritmic al zilei t; $|r_t|$: valoarea lui absolută
  - $\min(r_t, 0)$: randamentul, dacă este negativ, și 0 în rest; permite scăderilor și creșterilor să aibă efecte diferite (efectul de levier)

Reperul HAR (1/2)

- Modelul **HAR** (heterogeneous autoregressive, autoregresiv heterogen) Corsi (2009)

$$y_t = \beta_0 + \beta_d \ln v_t + \beta_w \ln v_t^{(w)} + \beta_m \ln v_t^{(m)} + u_t$$

  - $v_t^{(w)}$, $v_t^{(m)}$: media lui v pe ultimele 5 zile (săptămîna) și pe ultimele 22 de zile (luna)
  - β_0: constanta; $\beta_d, \beta_w, \beta_m$: ponderile celor trei orizonturi; u_t: eroarea

- Interpretarea
  - trei orizonturi ale participanților la piață (zilnic, săptămînal, lunar); estimat prin OLS
  - cu ponderi pozitive pe cele trei orizonturi, un șoc al volatilității se atenuează lent
  - un model liniar simplu care imită memoria lungă (Capitolul 11) și este greu de depășit

Reperul HAR (2/2)

- Întrebarea acestei secțiuni
  - adaugă lasso, random forest, gradient boosting sau un MLP ceva peste HAR în afara eșantionului?
  - un studiu amplu cu date intrazilnice găsește cîștiguri modeste, mai ales din variabile suplimentare Christensen, Siggaard & Veliyev (2023)
- Schema de evaluare: walk-forward din secțiunea 3, reestimare anuală; S&P 500 din 2013, Bitcoin din 2018
  - prognoza variației $\hat{h}_t = \exp(\hat{y}_t + \hat{s}^2/2)$, cu $\hat{s}^2$ varianța reziduală din antrenare (media log-Normală)
  - $\hat{y}_t$: prognoza logaritmului variației; $\exp(\hat{y}_t)$ singur ar subestima varianța medie, de aici termenul $\hat{s}^2/2$

Evaluarea (1/3): R^2 în afara eșantionului

- R^2 **în afara eșantionului** față de o prognoză de referință $\tilde{y}_t$ Campbell & Thompson (2008)

$$R^2_{OOS} = 1 - \frac{\sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \tilde{y}_t)^2}$$

  - $\hat{y}_t$: prognoza modelului; sumele se calculează doar pe zilele de test
  - reperul $\tilde{y}_t$: media din antrenare sau HAR
- Interpretarea
  - $R^2_{OOS} \leq 1$; $R^2_{OOS} > 0$: eroare pătratică mai mică decît a reperului; $R^2_{OOS} < 0$: mai slab decît reperul
  - spre deosebire de R^2 din eșantion, poate fi negativ
- Pas cu pas, S&P 500, MLP față de HAR
  - $1 - 0{,}397/0{,}433 = 1 - 0{,}917$, adică $+8{,}3\%$

Evaluarea (2/3): QLIKE

- **QLIKE** (Capitolul 8): o funcție de pierdere pentru prognozele variației, robustă la o variabilă proxy zgomotoasă Patton (2011)

$$L_t = \frac{\mathrm{RV}_t}{\hat{h}_t} + \ln \hat{h}_t$$

  - RV_t: varianța realizată din următoarele 5 zile (ținta, în niveluri); $\hat{h}_t$: prognoza ei
  - pentru un RV_t dat, pierderea este minimă la $\hat{h}_t = \mathrm{RV}_t$; un QLIKE mediu mai mic este mai bun
- Asimetria
  - pentru $\mathrm{RV} = 1$: $\hat{h} = 0{,}5$ costă 1,307, $\hat{h} = 2$ costă 1,193
  - subestimarea este penalizată mai mult: în managementul riscului, subestimarea volatilității este eroarea costisitoare

Evaluarea (3/3): testul Diebold–Mariano

- Testul **Diebold–Mariano** Diebold & Mariano (1995): au două modele A și B aceeași pierdere așteptată?

$$d_t = L_t^A - L_t^B, \qquad \mathrm{DM} = \frac{\bar{d}}{\sqrt{\widehat{\mathrm{LRV}}/n}} \sim N(0,1) \text{ la acuratețe egală}$$

  - L_t^A, L_t^B: pierderile celor două modele în ziua t; $\bar{d}$: media lui d_t; n: numărul zilelor de test
  - $\widehat{\mathrm{LRV}}$: varianța pe termen lung Newey–West a lui d_t (5 laguri), deoarece țintele suprapuse fac d_t autocorelat

- Interpretarea
  - $\mathrm{DM} < 0$: A are pierderea mai mică; $\mathrm{DM} > 0$: B este mai bun
  - $|\mathrm{DM}| > 1{,}96$: diferența este semnificativă la 5%

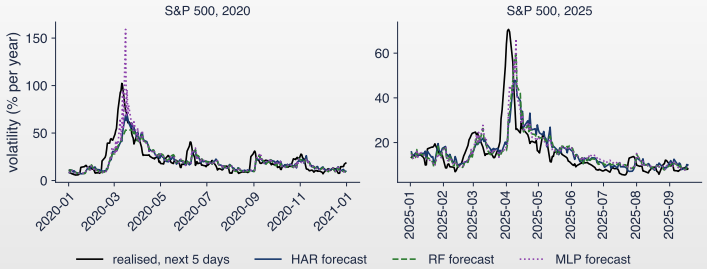
Rezultatele: S&P 500 și Bitcoin

	Model	R^2_{OOS} față de medie	R^2_{OOS} față de HAR	QLIKE	DM față de HAR (p)
S&P 500	HAR	61,5	–	0,414	–
	Lasso	64,2	+7,1	0,400	−5,12 (< 0,001)
	RF	61,9	+1,0	0,416	+0,32 (0,747)
	GB	60,4	−2,7	0,421	+1,02 (0,308)
	MLP	64,7	+8,3	0,398	−3,96 (< 0,001)
Bitcoin	HAR	44,0	–	3,111	–
	Lasso	45,1	+2,0	3,108	−0,64 (0,520)
	RF	43,4	−1,2	3,119	+0,81 (0,418)
	GB	41,7	−4,2	3,148	+2,12 (0,034)
	MLP	43,8	−0,3	3,123	+1,10 (0,272)

- Prognoze pentru 3 444 de zile (S&P 500, de la 2 ianuarie 2013) și 3 178 de zile (Bitcoin, de la 1 ianuarie 2018); R^2_{OOS} în %, pe log RV; GB: gradient boosting
- S&P 500: lasso și MLP depășesc semnificativ HAR (+7,1%; +8,3%); random forest și boosting nu
- Bitcoin: niciun model nu depășește HAR; boosting este semnificativ mai slab (DM +2,12)

SFM_ch13_volatility_forecast

Prognozele în perioadele turbulente din 2020 și 2025



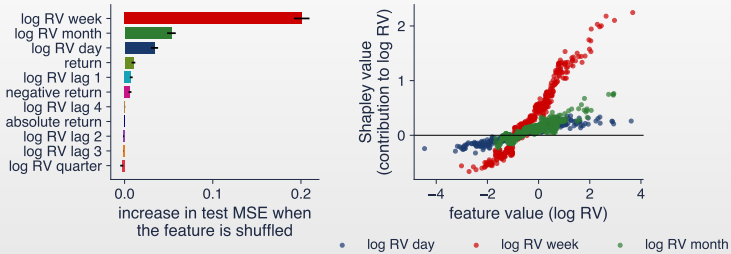
- S&P 500: volatilitatea realizată anualizată din următoarele 5 zile, $\sqrt{252\,\mathrm{RV}_t}$, și prognozele HAR, random forest și MLP făcute la închiderea zilei t
- Interpretare: toate modelele reacționează abia după primul șoc (martie 2020, aprilie 2025); random forest este limitat de cele mai mari valori văzute în antrenare, iar MLP poate depăși ținta

SFM_ch13_volatility_forecast

Importanța variabilelor: permutare și valori Shapley

- **Importanța prin permutare**: amestecăm o variabilă în datele de test și măsurăm cu cît crește MSE de test
  - creștere mare: modelul se bazează pe variabilă; variabilele corelate își împart (și își ascund) importanța
- **Valorile Shapley** (teoria jocurilor): $\hat{f}(x) = \phi_0 + \sum_j \phi_j(x)$, ϕ_j este contribuția echitabilă a variabilei j la prognoză
  - ϕ_0: prognoza medie; $\phi_j(x) > 0$: variabila j ridică această prognoză peste medie, $\phi_j(x) < 0$: o coboară
  - ϕ_j este media modificării lui $\hat{f}$ cînd j se adaugă la fiecare submulțime a celorlalte variabile
  - SHAP (Shapley additive explanations) Lundberg & Lee (2017): versiuni rapide pentru arbori; pentru un model liniar $\phi_j = \beta_j(x_j - \bar{x}_j)$
- Ambele descriu modelul, nu economia: importanța nu înseamnă cauzalitate

Variabilele folosite de random forest



- Stînga: random forest pe 11 variabile, antrenat pînă în 2019, testat în 2020–2026 (1 682 de zile): amestecarea variației săptămînale crește MSE cu 0,201, a celei lunare cu 0,053, a celei zilnice cu 0,034
- Dreapta: valorile Shapley exacte ale unui random forest pe cele 3 variabile HAR; media $|\phi|$: săptămîna 0,52, luna 0,13, ziua 0,13; efectul este neliniar și crește cu nivelul volatilității

SFM_ch13_volatility_forecast

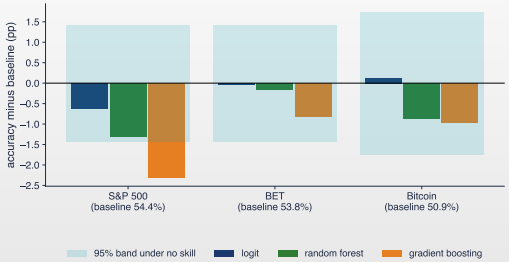
O problemă de clasificare

- Ținta: $y_t = 1\{r_{t+1} > 0\}$; variabilele la închiderea zilei t: ultimele 5 randamente, sumele pe 5, 21 și 63 de zile, volatilitatea pe 21 și 63 de zile
  - $1\{\cdot\}$: indicatorul, egal cu 1 dacă afirmația este adevărată și cu 0 în caz contrar; $y_t = 1$: randamentul de mîine este pozitiv
  - modele: logit (Capitolul 12), clasificatori random forest și gradient boosting; walk-forward, reestimare anuală
  - S&P 500 și BET din 2008, Bitcoin din 2018
- Indicatori (definiți în Capitolul 12)
  - acuratețea: ponderea zilelor cu semnul corect; AUC (aria de sub curba ROC): 0,5 înseamnă nicio capacitate de ordonare
- Motivul unor așteptări modeste: eficiența în formă slabă (Capitolul 7)
  - orice tipar sigur din randamentele trecute ar dispărea prin tranzacționare

Reperul clasei majoritare

- **Reperul clasei majoritare**: prognozăm mereu clasa mai frecventă din eșantionul de antrenare (aici: „creștere")
  - S&P 500 din 2008: 54,4% din zile sînt în creștere, deci „mereu creștere" are dreptate în 54,4% din zile
  - o acuratețe de 53% pare o abilitate; față de acest reper este o pierdere
- Testul față de reper: $z = (\widehat{\mathrm{acc}} - \mathrm{acc}_0)/\sqrt{\mathrm{acc}_0(1-\mathrm{acc}_0)/n}$
  - $\widehat{\mathrm{acc}}$: acuratețea modelului; acc_0: acuratețea reperului; n: zilele de test; $z \approx N(0,1)$ dacă modelul nu are un avantaj real
  - S&P 500, $n = 4\,705$: eroarea standard 0,73 puncte; o bandă de 95% de $\pm 1{,}4$ puncte în jurul reperului
  - random forest: 53,1%, $z = -1{,}79$ (p 0,074); boosting: 52,1%, $z = -3{,}19$ (p 0,001)
- Pentru randamente, reperul corespunzător este prognoza zero sau media istorică Welch & Goyal (2008)

Acuratețea față de reper



- Acuratețea minus acuratețea reperului (puncte procentuale); banda: $\pm 1{,}96$ erori standard în jurul lui zero
- AUC pentru logit și pentru random forest: S&P 500 0,511 și 0,507, BET 0,533 și 0,533, Bitcoin 0,515 și 0,519; logit prognozează „creștere” în 86% din zilele S&P 500

SFM_ch13_sign_prediction

Interpretare: limitele prognozei semnului

- Pe nicio piață, niciun model nu este semnificativ superior regulii „mereu creștere"; modelele flexibile sînt mai slabe decît logit
  - S&P 500: logit $-0{,}6$, random forest $-1{,}3$, boosting $-2{,}3$ puncte; Bitcoin: logit $+0{,}1$, random forest $-0{,}9$
- Raportul semnal–zgomot al randamentelor zilnice este foarte mic
  - modelele flexibile găsesc tipare în zgomotul anilor de antrenare (secțiunea 2); logit este aproape de reper deoarece aproape nu se mișcă
- Chiar și un avantaj real de un punct procentual ar rămîne rareori profitabil după costurile de tranzacționare: o strategie zilnică bazată pe semn tranzacționează foarte des
  - cîștigurile aduse de machine learning în prognoza randamentelor provin din secțiuni transversale mari, pe orizonturi lunare, nu dintr-un singur indice pentru ziua următoare (secțiunea 11)

Regresia cuantilică (1/2)

- Koenker & Bassett (1978): estimăm cuantila condiționată de ordin α, $q_\alpha(x) = x^\top \beta_\alpha$, minimizînd **pierderea cuantilică** (pinball loss)
  - $q_\alpha(x)$: nivelul sub care y se află cu probabilitatea α, condiționat de x; β_α: coeficienți specifici nivelului α
- Pierderea pentru o eroare $u = y - q$:

$$\rho_\alpha(u) = u\,(\alpha - 1\{u < 0\})$$

  - sub cuantilă ($u < 0$) eroarea costă $(1-\alpha)|u|$, deasupra ei $\alpha|u|$
  - cu $\alpha = 0{,}01$, minimul se atinge cînd aproximativ 1% din observații se află sub q

Regresia cuantilică (2/2)

- Cu $\alpha = 0{,}01$ și $q = -\mathrm{VaR}$: pierderea cuantilică medie din Capitolul 10
  - $(\alpha - I_t)(r_t + \mathrm{VaR}_t)$, cu $I_t = 1$ dacă $r_t < -\mathrm{VaR}_t$ (o depășire) și 0 în rest
- VaR 1% pentru ziua $t+1$: $\widehat{\mathrm{VaR}}_t = -\hat{q}_{0{,}01}(x_t)$, cu x_t = volatilitatea zilnică, săptămînală și lunară (rădăcinile variabilei proxy) și r_t
  - nu presupunem nicio distribuție: datele aleg cum se transformă volatilitatea în coadă; CAViaR Engle & Manganelli (2004) este o versiune dinamică
- **Quantile gradient boosting**: aceeași pierdere, cu arbori în boosting (secțiunea 5)
  - flexibil, dar la 1% fiecare frunză vede puține observații din coadă: variață mare

Backtesting pentru VaR 1%: metode cuantilice, HS și GARCH-t (1/2)

	Metoda	x	rata (%)	p Kupiec	p ind.	pierdere $\times 100$	VaR mediu
S&P 500	QR	28	0,81	0,252	0,020	3,38	2,70
	QGB	42	1,22	0,213	0,015	3,51	2,46
	HS	46	1,33	0,061	$<0,001$	4,78	2,97
	GARCH-t	52	1,51	0,005	0,008	3,50	2,42
Bitcoin	QR	23	0,72	0,098	0,563	13,03	10,46
	QGB	47	1,48	0,012	0,235	13,19	8,86
	HS	32	1,01	0,974	0,332	13,01	9,30
	GARCH-t	41	1,29	0,117	0,558	13,15	8,52

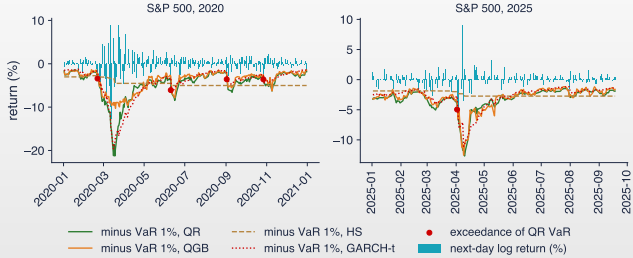
- Coloanele
  - x: numărul depășirilor ($r_{t+1} < -\widehat{\mathrm{VaR}}_t$) în cele n zile de test; rata $= x/n$, ținta 1%
  - p Kupiec: p-value-ul testului Kupiec al ratei; p ind.: p-value-ul testului de independență Christoffersen (Capitolul 10)
  - pierdere $\times 100$: pierderea cuantilică medie (mai mică este mai bună); VaR mediu: media VaR 1%, în %
- Metodele
  - QR: regresie cuantilică liniară; QGB: quantile gradient boosting
  - HS: simulare istorică pe 500 de zile; GARCH-t: reestimat în fiecare ianuarie (Capitolul 10)

SFM_ch13_quantile_var

Backtesting pentru VaR 1%: metode cuantilice, HS și GARCH-t (2/2)

- S&P 500 (3 448 de zile de la 2 ianuarie 2013): QR are rata corectă (0,81%) și cea mai mică pierdere
  - depășirile lui nu sînt complet independente (p ind. 0,020)
  - GARCH-t este depășit prea des (1,51%)
- Bitcoin (3 182 de zile): HS are cea mai bună rată (1,01%)
  - QGB este respins de testul Kupiec (1,48%, p 0,012)
- Interpretarea p-value-urilor
  - p Kupiec sub 0,05: rata depășirilor diferă semnificativ de 1%
  - p ind. sub 0,05: depășirile apar grupat, deci VaR reacționează prea lent

Prognozele VaR 1% în 2020 și 2025



- S&P 500: randamentele zilei următoare și minus VaR 1% pentru cele patru metode; punctele: depășirile VaR obținut prin regresie cuantilică
- Interpretare: QR și GARCH-t urmează volatilitatea; HS reacționează lent; în 2020, HS a fost depășit de 10 ori, QR de 4 ori, iar depășirile HS apar grupat (p ind. $< 0{,}001$)

SFM_ch13_quantile_var

Cea mai bună dintre N strategii (1/2)

- **Data snooping**: încercarea multor modele sau reguli pe aceleași date și raportarea celei mai bune White (2000)
  - cu suficiente încercări, o regulă fără nicio abilitate pare excelentă în eșantion
  - sute de predictori ai randamentelor publicați nu ar trece de o corecție pentru testare multiplă Harvey, Liu & Zhu (2016)
- Raportul Sharpe anual al unei strategii fără abilitate pe Y ani este aproximativ $N(0, 1/Y)$
  - raportul Sharpe: randamentul mediu în exces împărțit la abaterea lui standard (Capitolul 1); Y: numărul de ani de date
  - media 0: nicio abilitate; varianța $1/Y$: eroarea de estimare, care scade odată cu lungimea eșantionului

Cea mai bună dintre N strategii (2/2)

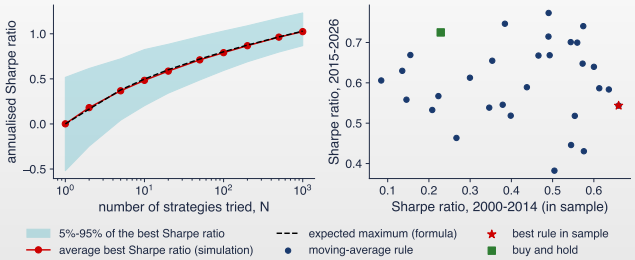
- Maximul așteptat al celor N încercări independente, fără abilitate Bailey & López de Prado (2014)

$$\mathrm{E}[\max] \approx \sigma\left[(1-\gamma)\,\Phi^{-1}\left(1-\tfrac{1}{N}\right)+\gamma\,\Phi^{-1}\left(1-\tfrac{1}{Ne}\right)\right]$$

  - σ: abaterea standard a rapoartelor Sharpe între încercări ($1/\sqrt{Y}$ mai sus); N: numărul de încercări
  - Φ^{-1}: funcția cuantilă a distribuției Normale standard; $e \approx 2{,}718$; $\gamma = 0{,}5772$: constanta Euler–Mascheroni
  - $\Phi^{-1}(1-1/N)$ crește aproximativ ca $\sqrt{2\ln N}$: avantajul dat de noroc crește lent, dar fără limită

- Exemplu: 10 ani, $\sigma = 1/\sqrt{10} = 0{,}316$, $N = 100$
  - $0{,}316\,[(1-\gamma)\,2{,}326+\gamma\,2{,}680] = 0{,}80$: un asemenea raport Sharpe apare, în medie, doar prin noroc
  - simulare: cea mai bună dintre 10, 100 și 1000 de strategii fără abilitate are un raport Sharpe mediu de 0,48; 0,79; 1,02

Distorsiunea de selecție în practică



- Stînga: cel mai bun raport Sharpe anual dintre N strategii fără abilitate (10 ani); dreapta: 30 de reguli cu medii mobile pe S&P 500 (investit sau numerar), 2000–2014 față de 2015–2026
- Interpretare: cea mai bună regulă în eșantion (MA 50/200, Sharpe 0,66) obține apoi 0,54, locul 22 din 30, sub buy and hold (0,72); rapoartele Sharpe din eșantion și din afara lui au corelația 0,04

SFM_ch13_data_snooping

Raportul Sharpe deflatat (1/2)

- Bailey & López de Prado (2014): probabilitatea ca adevăratul raport Sharpe să depășească ceea ce ar arăta din noroc cea mai bună dintre N încercări

$$\mathrm{DSR} = \Phi\Big(\frac{(\widehat{SR} - SR_0)\sqrt{T-1}}{\sqrt{1 - \hat{\gamma}_3 \widehat{SR} + \frac{\hat{\gamma}_4 - 1}{4}\widehat{SR}^2}}\Big)$$

- Notațiile
  - $\widehat{SR}$: raportul Sharpe zilnic al strategiei alese; T: numărul de zile
  - SR_0: maximul așteptat al celor N încercări fără abilitate (mai sus), cu σ = abaterea standard a celor N rapoarte Sharpe încercate
  - $\hat{\gamma}_3$: asimetria, $\hat{\gamma}_4$: boltirea randamentelor zilnice (3 pentru distribuția Normală); Φ: funcția de repartiție Normală standard
- Interpretarea
  - DSR este o probabilitate în $(0, 1)$; peste 0,95: raportul Sharpe este semnificativ după corecția pentru N încercări
  - asimetria negativă și cozile groase măresc numitorul și reduc DSR

Raportul Sharpe deflatat (2/2)

- **Pas cu pas**: cea mai bună regulă MA, $N = 30$, $T = 3\,769$ de zile
  - $\widehat{SR} = 0{,}0416$ pe zi ($\times\sqrt{252} = 0{,}66$); $\hat{\gamma}_3 = -0{,}44$, $\hat{\gamma}_4 = 12{,}0$
  - $SR_0 = 0{,}0105\,[(1-\gamma)\,1{,}834 + \gamma\,2{,}249] = 0{,}0219$ (anual 0,35)
  - z, argumentul lui Φ din formula DSR: $z = 1{,}20$, deci DSR $= \Phi(z) = 0{,}88$
  - fără corecția pentru N (PSR, probabilistic Sharpe ratio, cu $SR_0 = 0$): 0,994
- Interpretare: luată singură, regula pare semnificativă; după 30 de încercări nu mai este (DSR sub 0,95), așa cum confirmă și rezultatele ulterioare

Empirical Asset Pricing via Machine Learning

- Gu, Kelly & Xiu (2020): care metode prezic cel mai bine randamentul lunar al acțiunilor americane individuale?
  - aproape 30 000 de acțiuni, 1957–2016; 94 de caracteristici ale firmelor, în interacțiune cu 8 variabile macroeconomice, plus 74 de variabile indicatoare de industrie: 920 de variabile
- Schema de estimare: 18 ani de antrenare (1957–1974), 12 de validare (1975–1986), 30 de test (1987–2016)
  - reestimare o dată pe an; fereastra de antrenare crește, cea de validare avansează: o schemă walk-forward
  - modele: OLS, regresii penalizate și cu reducerea dimensiunii, random forest, arbori în boosting
  - rețelele NN1–NN5: NNk are k straturi ascunse



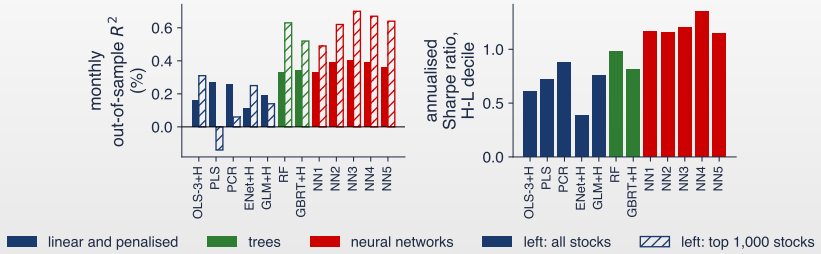
Charles M. Harper Center, Chicago Booth, unde au lucrat Gu și Xiu

Foto: Michael Barera (2013); CC BY-SA 4.0; Wikimedia Commons

Evaluarea din lucrare

- $R^2_{OOS} = 1 - \sum_{(i,t)}(r_{i,t+1} - \hat{r}_{i,t+1})^2 / \sum_{(i,t)} r^2_{i,t+1}$: reperul este o prognoză **zero**, nu media istorică
  - $r_{i,t+1}$: randamentul acțiunii i în luna $t+1$; $\hat{r}_{i,t+1}$: prognoza lui; sumele parcurg toate perechile acțiune–lună din perioada de test
  - autorii arată că media istorică a unei acțiuni este atît de zgomotoasă încît este ușor de depășit: reperul zero este mai exigent
- Teste Diebold–Mariano între toate perechile de modele, cu o corecție Bonferroni pentru cele 12 comparații (pragul de semnificație se împarte la numărul comparațiilor)
  - același instrument ca în secțiunea 7, aplicat unui panel de acțiuni
- Valoarea economică: în fiecare lună, sortăm acțiunile în decile după prognoză; cumpărăm decila de sus și vindem decila de jos
  - portofolii ponderate cu valoarea de piață; raportul Sharpe, drawdown și rulajul (ponderea portofoliului tranzacționată lunar)

Rezultatele: R^2 predictiv și valoarea economică



- Stînga: R^2_{OOS} lunar (tabelul 1): OLS cu toate cele 920 de variabile −3,46%; OLS-3 (mărime, valoare, momentum) 0,16%; random forest 0,33%; NN3 0,40% (0,70% pentru cele mai mari 1 000 de acțiuni)
- Dreapta: raportul Sharpe anual al diferenței dintre decile, ponderată cu valoarea de piață (tabelul 7): OLS-3 0,61, random forest 0,98, NN4 1,35

SFM_ch13_case_gkx

Interpretare și limite

- Rezultatele lucrării
  - modelele neliniare (arbori, rețele) sînt superioare celor liniare deoarece surprind interacțiunile dintre caracteristici
  - toate metodele sînt de acord asupra semnalelor principale: tendințele prețului (momentum, inversarea pe termen scurt), lichiditatea, volatilitatea
  - rețelele puțin adînci sînt suficiente: 3–4 straturi dau rezultate mai bune decît 5; deep learning nu ajută la randamentele lunare
- De reținut
  - un R^2_{OOS} sub 1% pe lună este un rezultat important în acest context: comparați cu prognozele semnului din secțiunea 8
  - portofoliile pe baza rețelelor își schimbă lunar peste 100% din valoare: costurile de tranzacționare reduc cîștigurile
  - dovezile provin dintr-o secțiune transversală de mii de acțiuni; ele spun puțin despre alegerea momentului pentru un singur indice

O întrebare de cercetare deschisă

- **Ajută prognozele de volatilitate prin machine learning la Bursa de Valori București, unde lichiditatea este redusă și datele intrazilnice sînt rare?**
  - azi: la S&P 500, lasso cîștigă +7,1% față de HAR; la Bitcoin niciun cîștig semnificativ (+2,0%, p 0,520)
  - BET nu are prețuri maxime și minime în datele noastre, deci variabila proxy trebuie construită din randamente la pătrat, mai zgomotoase
- Întrebarea rămîne deschisă: tranzacționare rară, un calendar diferit de cel de la New York, puține crize în eșantion
- Lucrări înrudite, ca lectură suplimentară: Pele et al. (2023) (clasificarea statistică a activelor cripto); Pele & Mazurencu-Marinescu-Pele (2026) (cît de precise pot fi comparațiile de prognoze)
- Instrumentele AI pot accelera un astfel de studiu; nu înlocuiesc verificarea lui Wang et al. (2023)

Utilizări posibile ale instrumentelor AI

- **Literatura**: o listă a studiilor despre HAR și machine learning pentru pieţele emergente și de frontieră
- **Cod**: o primă versiune a comparației walk-forward dintre HAR, lasso și random forest pentru BET și trei acțiuni BVB
- **Robustețe**: alte variabile proxy (randamente absolute, randamente pe două zile), alte orizonturi (1, 5, 22 de zile), variabile S&P 500 de la închiderea precedentă
- Exemplu de prompt
  - `Write Python code that compares HAR, lasso and a random forest for the weekly log variance of the BET index (proxy: squared daily returns), with yearly walk-forward refits from 2012, purging the last 5 training rows, and reports the out-of-sample R2, QLIKE and Diebold-Mariano tests.`

Verificări necesare

- Ordinea temporală: fără partiții amestecate; variabilele folosesc doar date pînă la închiderea zilei t; ultimele ținte de antrenare nu ajung în perioada de test
- Scalarea și alegerea hiperparametrilor doar în fereastra de antrenare (standardizarea, λ, numărul de arbori)
- Calendarele: BVB se închide înaintea bursei din New York; variabilele S&P 500 trebuie să provină de la închiderea precedentă din New York
- Un reper în fiecare tabel (HAR, clasa majoritară, simularea istorică) și un test al diferenței
- Numărul de modele încercate, raportat; referințele: fiecare lucrare citată trebuie să existe; verificați DOI-ul

Idee de proiect

- **Întrebarea**: prognozează modelele de machine learning volatilitatea acțiunilor BVB mai bine decît HAR?
  - date: BET, Banca Transilvania, OMV Petrom, BRD (EODHD); S&P 500 și DAX ca variabile externe
- Pași
  - o variabilă proxy a variației pentru fiecare acțiune (amplitudinea, unde există, altfel randamentele la pătrat), țintă săptămînală
  - HAR, lasso, random forest și un MLP mic, walk-forward din 2014, reestimare anuală cu purging
  - R^2 în afara eșantionului, QLIKE și Diebold–Mariano; valori Shapley pentru variabilele externe
- Livrabil: un tabel, un grafic și un paragraf despre ce pot și ce nu pot arăta datele
- Declarați orice utilizare a instrumentelor AI și enumerați erorile acestora pe care le-ați corectat

Idei de reținut

- Machine learning se judecă după eroarea pe date noi; compromisul deplasare–variață stabilește cît de flexibil trebuie să fie un model
- Seriile de timp au nevoie de validare walk-forward cu purging; partițiile aleatoare pe ținte suprapuse simulează o predictibilitate falsă
- Ridge și lasso contractă coeficienții; random forest face media arborilor; boosting adaugă arbori; rețelele învață variabile neliniare
- Volatilitatea: HAR este greu de depășit, cîștigurile sînt de cîteva procente; semnul randamentelor: niciun cîștig față de „mereu creștere"
- VaR 1%: regresia cuantilică liniară funcționează; modelelor cuantilice flexibile le lipsesc datele din coadă
- Raportați numărul de încercări și deflatați raportul Sharpe; comparați fiecare model cu un reper simplu

Formule de reținut

Mărimea	Formula
Deplasare–varianță	$E[(y_0 - \hat{f})^2] = (E\hat{f} - f)^2 + \mathrm{Var}(\hat{f}) + \sigma^2$
Ridge	$(X^\top X + \lambda I)^{-1} X^\top y$; 1D: $S_{xy}/(S_{xx} + \lambda)$
Lasso (1D)	$\mathrm{sign}(S_{xy})\max(\lvert S_{xy}\rvert - \lambda/2, 0)/S_{xx}$
Random forest	$\mathrm{Var} = \rho\sigma^2 + (1-\rho)\sigma^2/B$
Boosting	$F_m = F_{m-1} + \nu h_m$
MLP	$\hat{y} = c + \sum_k v_k g(b_k + w_k^\top x)$
HAR	$y_t = \beta_0 + \beta_d \ln v_t + \beta_w \ln v_t^{(w)} + \beta_m \ln v_t^{(m)} + u_t$
R^2_{OOS}	$1 - \sum(y - \hat{y})^2 / \sum(y - \bar{y})^2$
Pierderea cuantilică	$\rho_\alpha(u) = u(\alpha - 1\{u < 0\})$
DSR	$\Phi\big((\widehat{SR} - SR_0)\sqrt{T-1}/\sqrt{1 - \hat{\gamma}_3\widehat{SR} + (\hat{\gamma}_4 - 1)\widehat{SR}^2/4}\big)$

Autoevaluare

- **Întrebare**: un model are un R^2 de antrenare de 80% și un R^2 în afara eșantionului de -5%. Ce s-a întîmplat?
  - **Răspuns**: overfitting: modelul a învățat zgomotul eșantionului de antrenare și pierde în fața reperului pe date noi
- **Întrebare**: de ce este greșit K-fold aleator pentru o țintă de tip randament pe 21 de zile?
  - **Răspuns**: țintele vecine se suprapun, deci răspunsurile de test ajung în antrenare; folosiți walk-forward cu purging
- **Întrebare**: un clasificator are dreptate în 53% din zile, iar piața a crescut în 54% dintre ele. Este util?
  - **Răspuns**: nu: „mereu creștere” are dreptate în 54% din zile
- Urmează: Capitolul 14, activele cripto

Bibliografie (1/3)

- Bailey, D.H., López de Prado, M. (2014). The deflated Sharpe ratio: correcting for selection bias, backtest overfitting, and non-normality. *Journal of Portfolio Management*, 40(5), 94–107.
- Bergmeir, C., Hyndman, R.J., Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120, 70–83.
- Breiman, L. (2001a). Statistical modeling: the two cultures. *Statistical Science*, 16(3), 199–231.
- Breiman, L. (2001b). Random forests. *Machine Learning*, 45(1), 5–32.
- Campbell, J.Y., Thompson, S.B. (2008). Predicting excess stock returns out of sample: can anything beat the historical average? *Review of Financial Studies*, 21(4), 1509–1531.
- Christensen, K., Siggaard, M., Veliyev, B. (2023). A machine learning approach to volatility forecasting. *Journal of Financial Econometrics*, 21(5), 1680–1727.
- Christoffersen, P.F. (1998). Evaluating interval forecasts. *International Economic Review*, 39(4), 841–862.
- Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2), 174–196.
- Diebold, F.X., Mariano, R.S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263.
- Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1007.
- Engle, R.F., Manganelli, S. (2004). CAViaR: conditional autoregressive value at risk by regression quantiles. *Journal of Business & Economic Statistics*, 22(4), 367–381.
- Franke, J., Härdle, W.K., Hafner, C.M. (2019). *Statistics of Financial Markets: An Introduction*, 5th ed. Springer.

Bibliografie (2/3)

- Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.
- Garman, M.B., Klass, M.J. (1980). On the estimation of security price volatilities from historical data. *Journal of Business*, 53(1), 67–78.
- Gu, S., Kelly, B., Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5), 2223–2273.
- Harvey, C.R., Liu, Y., Zhu, H. (2016). . . . and the cross-section of expected returns. *Review of Financial Studies*, 29(1), 5–68.
- Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning*, 2nd ed. Springer.
- Hoerl, A.E., Kennard, R.W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67.
- Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.
- James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J. (2023). *An Introduction to Statistical Learning with Applications in Python*. Springer.
- Kelly, B., Xiu, D. (2023). Financial machine learning. *Foundations and Trends in Finance*, 13(3–4), 205–363.
- Koenker, R., Bassett, G. (1978). Regression quantiles. *Econometrica*, 46(1), 33–50.
- Kupiec, P.H. (1995). Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*, 3(2), 73–84.

Bibliografie (3/3)

- Lundberg, S.M., Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- López de Prado, M. (2018). *Advances in Financial Machine Learning*. Wiley.
- Patton, A.J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1), 246–256.
- Pele, D.T., Mazurencu-Marinescu-Pele, M. (2026). Finite-sample precision limits for expected shortfall forecast comparisons. *Mathematics*, 14(13), 2316.
- Pele, D.T., Wesselhöfft, N., Härdle, W.K., Kolossiatis, M., Yatracos, Y.G. (2023). Are cryptos becoming alternative assets? *European Journal of Finance*, 29(10), 1064–1105.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1), 267–288.
- Wang, H., Fu, T., Du, Y., Gao, W., et al. (2023). Scientific discovery in the age of artificial intelligence. *Nature*, 620, 47–60.
- Welch, I., Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, 21(4), 1455–1508.
- White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5), 1097–1126.