Procjena latentnih faktora u financijskim vremenskim nizovima zasnovana na strojnom učenju

Linardić, Ivan

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 548

PROCJENA LATENTNIH FAKTORA U FINANCIJSKIM VREMENSKIM NIZOVIMA ZASNOVANA NA STROJNOM UČENJU

Ivan Linardić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 548

PROCJENA LATENTNIH FAKTORA U FINANCIJSKIM VREMENSKIM NIZOVIMA ZASNOVANA NA STROJNOM UČENJU

Ivan Linardić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 548

Pristupnik:	**Ivan Linardić (0036523896)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	doc. dr. sc. Stjepan Begušić

Zadatak: **Procjena latentnih faktora u financijskim vremenskim nizovima zasnovana na strojnom učenju**

Opis zadatka:

U modeliranju zavisnosti velikog broja financijskih vremenskih nizova koriste se procjenitelji kovarijacijskih matrica zasnovani na faktorskim modelima kako bi pružili bolje uvjetovane procjene koje su manje osjetljive na šum. Pritom su najčešće korištene metode zasnovane na nenadziranom učenju, poput analize glavnih komponenti i procjenitelja najveće izglednosti, koristeći financijske vremenske nizove na određenom prozoru procjene. No, zbog dinamičke prirode financijskih sustava, nije poznato koliko dobro takve metode predviđaju buduće faktorske strukture i matrice kovarijance izvan uzorka procjene. U sklopu ovog rada potrebno je prvo implementirati osnovne modele nenadziranog učenja za procjenu latentnih faktora, te analizirati njihovu predikcijsku moć, koristeći klizeće prozore na velikom broju financijskih vremenskih nizova. Nadalje, potrebno je formulirati problem procjene latentnih faktora kao zadatak nadziranog učenja, te implementirati okvir za treniranje takvih modela na podatcima, uključujući definiranje ulaznih značajki, izlaznih varijabli i funkcija cilja. Potom je potrebno razviti i implementirati konkretne modele nadziranog učenja, počevši od jednostavnijih linearnih modela sve do složenijih nelinearnih modela visokog kapaciteta, te ih trenirati koristeći razvijeni okvir i podatke. Sve razmatrane modele potrebno je ispitati izvan uzorka procjene koristeći povijesne tržišne podatke, u kontekstu različitih mjera te rizika portfelja optimiranih koristeći različite procjene kovarijance.

Rok za predaju rada: 28. lipnja 2024.

Ovaj uspjeh stoji na remenima ljudi koji su uvijek bili uz mene. Posebno bih htio zahvaliti svojim roditeljima i zaručnici.

Sadržaj

1. Uvod **2**

2. Faktorski modeli financijskih vremenskih nizova **3**

3. Modeli strojnog učenja za procjenu latentnih faktora **14**

3.1. Osnovni modeli 14

3.2. Linearna regresija 14

3.3. Polinomijalna regresija 15

3.4. Regresor slučajnih šuma 15

3.5. Umjetne neuronske mreže 17

4. Podaci i implementacija procjene latentnih faktora **20**

4.1. Procjena pozitivno definitne kovarijacijske matrice 29

4.2. Mjere performanse 30

5. Rezultati **33**

6. Zaključak **40**

Literatura **41**

Sažetak **44**

Abstract **45**

1. Uvod

Kad bi se netko htio uključiti u svijet financija, steći imovinu i učinkovito upravljati njome, ubrzo bi se upoznao s pojmom rizika. Cilj je ulagača minimizirati rizičnost vlastitog portfelja kako bi tijekom vremena imao stabilnu zaradu. Rizik se može mjeriti varijabilnošću povrata imovine u portfelju. U tom slučaju se računa kovarijacijskom matricom povrata - statističkim alatom koji pruža informacije o individualnim rizicima i zajedničkom kretanju povrata različitih imovina. U financijama su ove matrice najčešće visokodimenzionalne i mijenjaju se tijekom vremena zato što opisuju međuodnos povrata velikog broja imovine koji se također konstantno mijenjaju. Poželjno je znati kakva će biti kovarijacijska matrica u budućnosti kako bi se znalo modelirati budući rizik. Za izračun budućih kovarijacijskih matrica potrebni su povijesni podaci što zna predstavljati problem jer su u pravilu vremenski prozori iz kojih se ti podaci dobivaju kratki. Tako se dobije uzoračka kovarijacijska matrica koja ima puno veći broj dimenzija od broja uzoraka iz kojih je izračunata. U takvim slučajevima standardni procjenitelji budućeg međuodnosa imovine, koji se temelje na opisanoj matrici, imaju puno šuma zbog čega može doći do grešaka u upravljanju rizikom i optimizaciji portfelja. Kako bi se zaobišao ovaj problem, potrebno je pronaći novi pristup procjeni buduće kovarijance. Među vremenskim nizovima povrata imovine postoje zajednički latentni (skriveni) faktori koji se mogu opisati faktorskim modelima. Njima se može rekonstruirati kovarijanca povrata imovine. Time se problem svodi na procjenu budućih faktora iz kojih se zatim izračuna procjena buduće kovarijacijske matrice. Cilj diplomskog rada je usporediti različite modele strojnog učenja u kvaliteti procjene budućih faktora.

2. Faktorski modeli financijskih vremenskih nizova

Tržišna cijena je trenutna cijena po kojoj se imovina ili usluga može kupiti ili prodati. Vrijednost povrata r_d u danu d se računa kao omjer razlike u cijenama imovine tog i prijašnjeg dana te cijene prijašnjeg dana:

$$r_d = \frac{p_d - p_{d-1}}{p_{d-1}} . \tag{2.1}$$

Ovako izračunati povrati predstavljaju stopu povrata na investiciju te mogu biti pozitivni (dobitak) ili negativni (gubitak). Odnos povrata i cijene prikazan je na slici 2.1.

Srednja ili očekivana vrijednost slučajne varijable X ima posebnu važnost u statistici zato što opisuje gdje je distribucija zadane varijable centrirana [1]. Računa se kao aritmetička sredina realiziranih vrijednosti slučajne varijable. Varijancom slučajne varijable X opisuje se njena raspršenost oko srednje vrijednosti i karakterizira se varijabilnost u distribuciji. Računa se kao očekivanje kvadriranog odstupanja slučajne varijable od njene srednje vrijednosti:

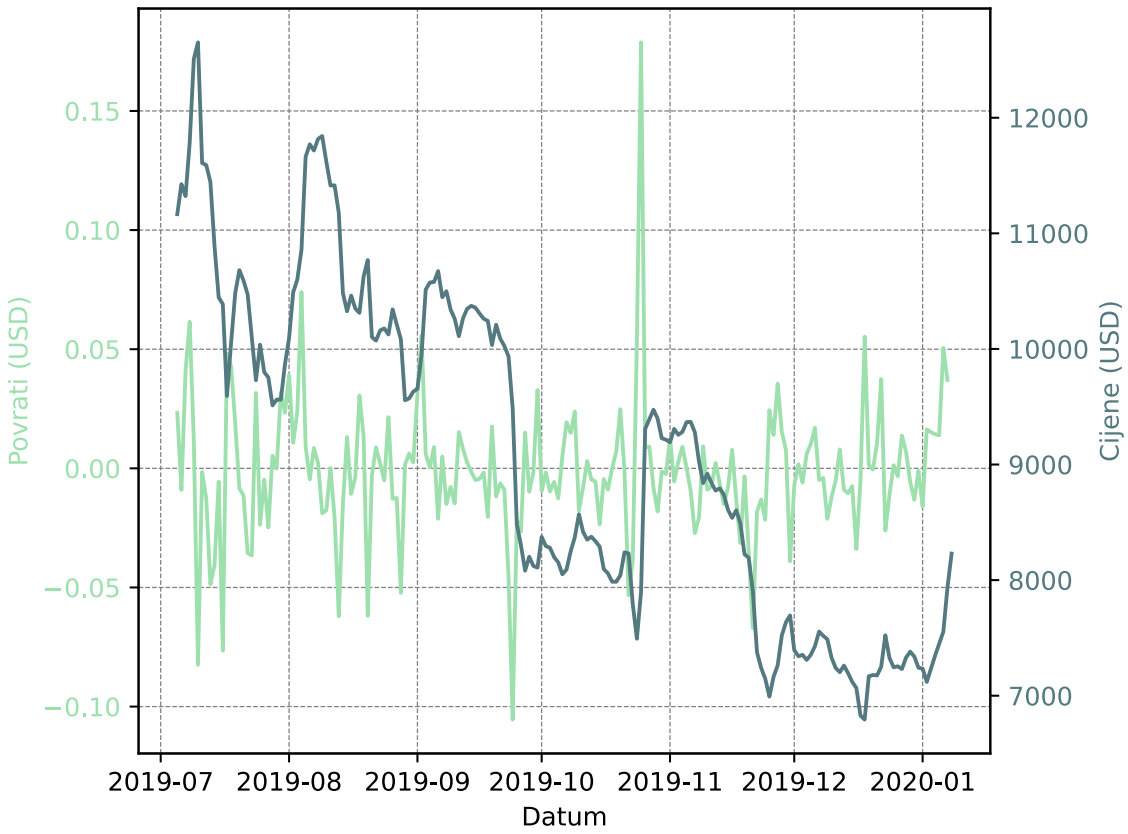
$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (r_i - \mu)^2 . \tag{2.2}$$

N predstavlja ukupan broj realizacija slučajne varijable.

Međusobna ovisnost dvije slučajne varijable se mjeri njihovom kovarijancom:

$$\hat{\sigma}_{ik} = \frac{1}{N-1} \sum_{j=1}^{N} (r_{ji} - \mu_i)(r_{jk} - \mu_k) . \tag{2.3}$$

Pojednostavljeno, ako se uz velike vrijednosti X očitavaju i velike vrijednosti Y, odnosno ako se uz male vrijednosti X očitavaju i male vrijednosti Y, može se pretpostaviti da će te dvije varijable imati pozitivnu kovarijancu [1]. Predznak indicira je li njihov odnos pozi-



Slika 2.1. Odnos cijene i povrata financijske imovine.

tivan - ako jedna raste i druga raste, ili negativan - ako jedna raste druga pada. Magnituda varijance ništa ne govori o snazi međuodnosa zato što ovisi o korištenim mjernim jedinicama [1]. Kada su X i Y statistički nezavisne slučajne varijable njihova kovarijanca je nula. Obratno generalno ne vrijedi - dvije slučajne varijable mogu imati kovarijancu nula i još uvijek biti statistički zavisne. Kovarijanca opisuje samo linearni odnos između dvije slučajne varijable te ako je jednaka nuli, varijable X i Y još uvijek mogu imati nelinearan odnos što znači da nisu nezavisne [1].

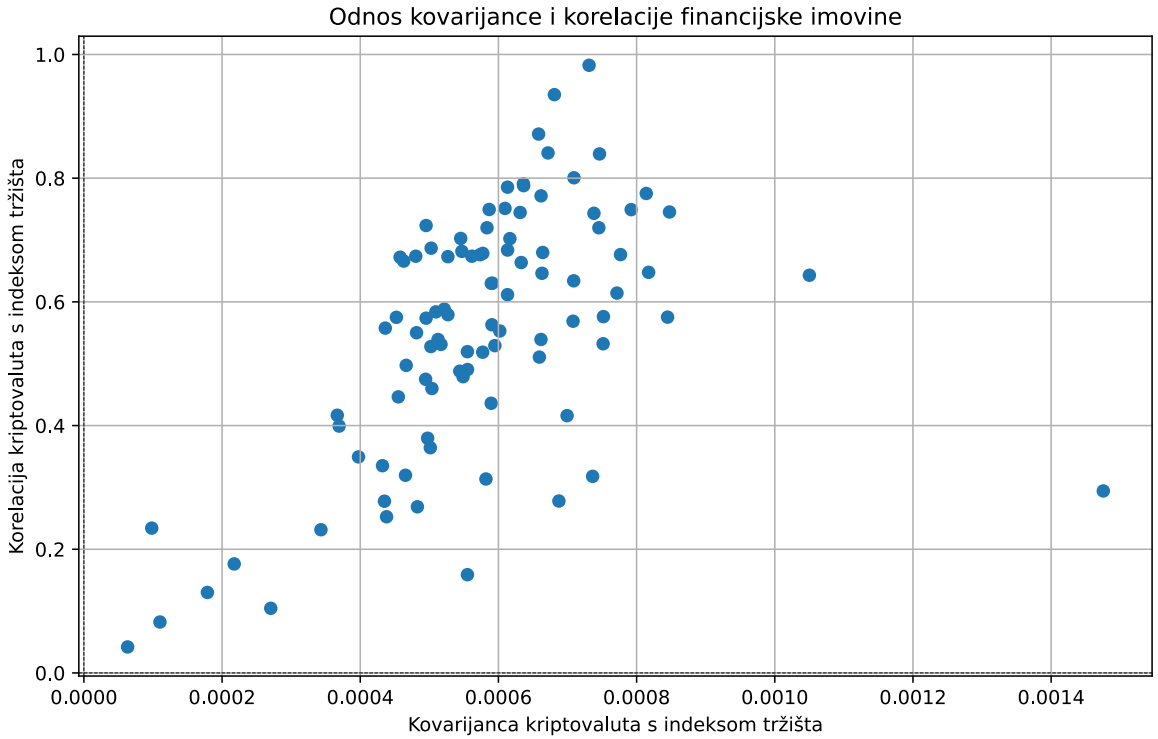
Mjera međuovisnosti koja ne ovisi o mjernim jedinicama i magnitudom izriče snagu odnosa dvije slučajne varijable zove se korelacijski koeficijent [1]. Računa se formulom

$$\hat{\rho}_{ik} = \frac{\hat{\sigma}_{ik}}{\hat{\sigma}_i \hat{\sigma}_k}, \tag{2.4}$$

gdje je $\hat{\sigma}_{ik}$ kovarijanca slučajnih varijabli, a $\hat{\sigma}_i$ i $\hat{\sigma}_k$ njihove standardne devijacije koje se računaju formulom

$$\hat{\sigma} = \sqrt{\hat{\sigma}^2}. \tag{2.5}$$

Vrijednost korelacije zadovoljava nejednakost $-1 \leq \hat{\rho} \leq 1$ te iznosi 0 onda kada je i kovarijanca jednaka 0, a 1 ako su slučajne varijable savršeno linearno međuovisne. Odnos kovarijance i korelacijskog koeficijenta prikazan je slikom 2.2.



Slika 2.2. Prikaz odnosa kovarijance i korelacijskog koeficijenta povrata 100 kriptovaluta u odnosu na povrate indeksa tržišta. Moguće je primijetiti kako su vrijednosti korelacijskog koeficijenta skalirane u odnosu na kovarijancu.

Veći broj slučajnih varijabli prikazuje se slučajnim vektorom. Ako je Ω prostor uzoraka, slučajni vektor x je moguće prikazati kao funkciju iz prostora uzoraka u N-dimenzionalni realni vektor $\mathbb{R}^N$ [2]:

$$x : \Omega \to \mathbb{R}^N . \tag{2.6}$$

Kovarijacijska matrica slučajnog vektora je kvadratna matrica koja sadrži sve kovarijance i varijance ulaznih varijabli tog vektora [3]. Ona je multivarijatna generalizacija varijance jedne slučajne varijable. Računa se formulom

$$\Sigma = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)'(x_i - \mu) . \tag{2.7}$$

Ako se s $x_1, ..., x_N$ označi N elemenata slučajnog vektora x, slijedi da je Σ kvadratna ma-

trica dimenzionalnosti $N \times N$ i strukture [3]

$$\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & \dots & \sigma_{x_1 x_N} \\ \vdots & \ddots & \vdots \\ \sigma_{x_N x_1} & \dots & \sigma_{x_N}^2 \end{bmatrix}, \tag{2.8}$$

gdje su elementi na dijagonali varijance slučajnih varijabli, a oni izvan dijagonale njihove međusobne kovarijance.

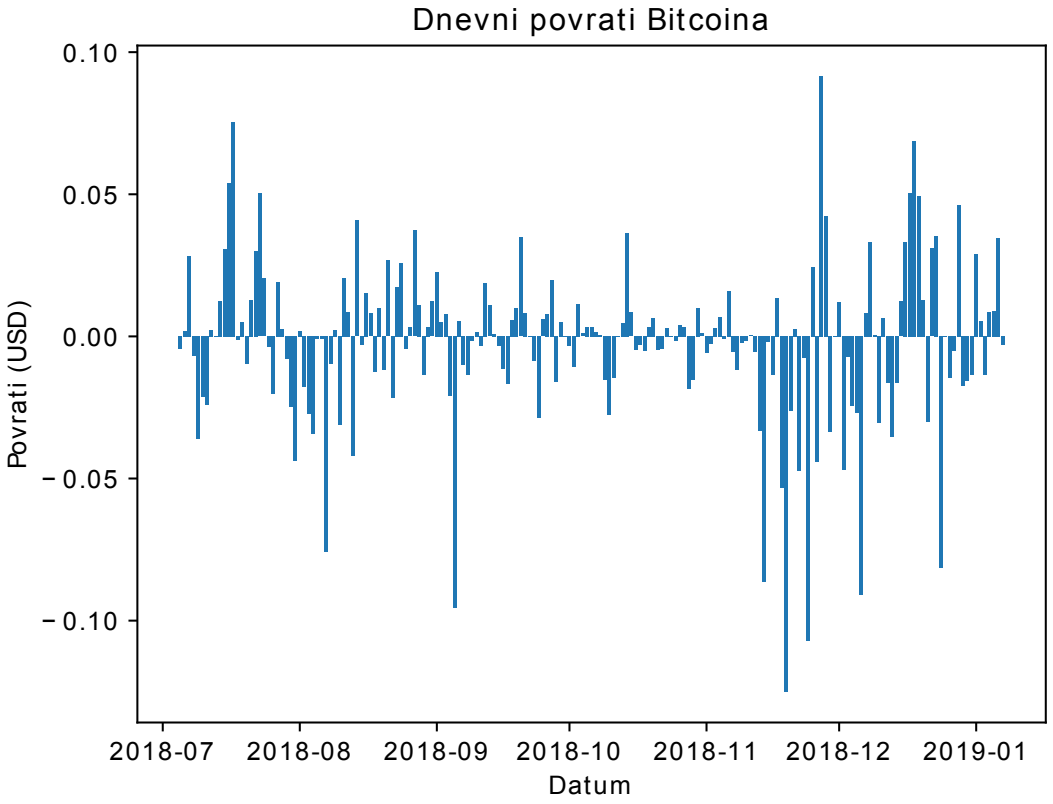
Osnovni zahtjev svake statističke analize financijskih podataka je da podaci koji se proučavaju imaju neka statistička svojstva koja ostaju stabilna tijekom vremena, u suprotnom nema smisla pokušati ih identificirati [4].

Nepromjenjivost statističkih svojstava povrata tijekom vremena odgovara hipotezi o stacionarnosti kojom se tvrdi da je za svaki skup trenutaka u vremenu $t_1, \dots, t_i$ i bilo koji vremenski interval τ združena distribucija povrata $r(t_1, N), \dots, r(t_i, N)$ ista kao združena distribucija povrata $r(t_1 + \tau, N), \dots, r(t_i + \tau, N)$ [4]. Odnosno, stacionarnost osigurava da distribucija ne ovisi o vremenu t, što omogućava da se opažanja iz različitih trenutaka u vremenu mogu iskoristiti za izračun jednog zajedničkog statističkog momenata.

Unatoč tomu što se koristi u postojećim modelima, pretpostavka o stacionarnosti povrata nema empirijsku podlogu zato što povrati u prošlosti ne moraju na ikoji način odražavati buduće povrate. Dobro je poznata činjenica da zarada od proizvoda/imovine u različitim trenucima na slobodnom tržištu nema nikakve značajne autokorelacije. Funkcijom

$$C(\tau) = \rho(r(t, N), r(t + \tau, N)) \tag{2.9}$$

je opisana autokorelacija povrata koja prolaskom vremena brzo konvergira u nulu [4]. Ova činjenica se često koristi kao jedan od dokaza učinkovitog tržišta [4]. Nepostojanje autokorelacije je intuitivno lako za razumjeti - ako povrati pokazuju značajnu korelaciju, ona se može iskoristiti za kreiranje jednostavne strategije s pozitivnom očekivanom zaradom; takve strategije statističke arbitraže će posljedično reducirati korelaciju osim za vrlo kratke periode koji predstavljaju vrijeme potrebno da tržište reagira na nove informacije [4].



Slika 2.3. Fenomen grupiranja volatilnosti na primjeru povrata Bitcoina.

Nepostojanje korelacije među povratima ne implicira i međusobnu neovisnost njihove promjene tijekom vremena. Jednostavne nelinearne funkcije povrata, poput apsolutnih ili kvadriranih povrata, pokazuju pozitivnu autokorelaciju i dosljednost. Razlog tomu je poznati fenomen grupiranja volatilnosti: izglednije je da će velike promjene u cijenama biti praćene velikim promjenama, nego malim [4]. Graf na slici 2.3. prikazuje ovaj fenomen na dnevnim povratima Bitcoina.

U jedno-faktorskim modelima procjene povrata portfelja, poput *Capital Asset Pricing Modela (CAPM)*, očekivani povrati imovine su određeni isključivo njenom osjetljivošću na rizik tržišta. Iako više-faktorski modeli bolje objašnjavaju različite druge aspekte rizika očekivanih povrata, CAPM jako dobro objašnjava volatilnost cijena tijekom vremena za što koristi tržišne β koje nisu direktno osmotrive i mijenjaju se tijekom vremena [5]. Ove β je moguće izračunati iz podataka i iskoristiti ih za rekonstrukciju kovarijacijske matrice povrata. One su zapravo koeficijenti latentnih faktora na kojima počiva rješenje problema procjene.

Svrha faktorske analize je opisati kovarijacijski odnos između velikog broja varijabli uz pomoć nekolicine latentnih slučajnih vrijednosti koje se zovu faktori [6]. Faktorski model se temelji na pretpostavci da se varijable mogu grupirati na temelju njihove korelacije. Odnosno, pretpostavlja da su sve varijable unutar jedne grupe međusobno jako korelirane i da su slabo korelirane s varijablama u ostalim grupama. Iz toga se može zaključiti da svaka grupa predstavlja jedan skriveni konstrukt, ili faktor, koji je odgovoran za njihovu korelaciju. Zato se na faktorsku analizu može gledati kao na pokušaj procjene kovarijacijske matrice [6], za što se koristi i u ovom radu.

Primjer rastava slučajnog vektora x na faktore izgleda ovako:

$$\begin{aligned} x_1 - \mu_1 &= \beta_{11}F_1 + \beta_{12}F_2 + \cdots + \beta_{1p}F_P + \epsilon_1 \\ x_2 - \mu_2 &= \beta_{21}F_1 + \beta_{22}F_2 + \cdots + \beta_{2p}F_P + \epsilon_2 \\ &\vdots \\ x_N - \mu_N &= \beta_{n1}F_1 + \beta_{n2}F_2 + \cdots + \beta_{np}F_P + \epsilon_N \,. \end{aligned}$$

Što se matrično može zapisati kao:

$$x - \mu = BF + \epsilon \,. \tag{2.10}$$

U danim jednadžbama μ je vektor srednjih vrijednosti slučajnih varijabli dimenzionalnosti $N \times 1$, F je vektor dimenzionalnosti $P \times 1$ koji čine zajednički latentni slučajni faktori modela, a vrijednosti vektora ϵ, dimenzionalnosti $N \times 1$, predstavljaju dodatne izvore varijacije u podacima. Koeficijenti β_{ij} se zovu *koeficijentima* i-te varijable na j-tom faktoru, zato se matrica B naziva matrica faktorskih koeficijenata.

Navedeni opis faktorskog modela sa sobom vuče određene implikacije u vezi kovari-

jacije slučajnih varijabli. Stoga se za slučajne vektore F i ϵ pretpostavlja da vrijedi [6]:

$$E(F) = \underset{(P\times 1)}{\mathbf{0}}, \qquad \sigma_{ij}(F) = E[FF'] = \underset{(P\times P)}{I},$$

$$E(\epsilon) = \underset{(N\times 1)}{\mathbf{0}}, \qquad \sigma_{ij}(\epsilon) = E[\epsilon\epsilon'] = \underset{(N\times N)}{\Psi} = \begin{bmatrix} \psi_1 & 0 & \dots & 0 \\ 0 & \psi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \psi_N \end{bmatrix}.$$

Na temelju ovih pretpostavki moguće je dobiti strukturu kovarijance faktorskog modela, koja u financijama dodatno uključuje skaliranje faktorskih koeficijenata varijancom indeksa tržišta:

$$\hat{\Sigma} = \hat{\beta}\hat{\beta}'\sigma^2_{ind} + \hat{\Psi}. \tag{2.11}$$

$\hat{\Sigma}$ je procijenjena kovarijacijska matrica, a $\hat{\Psi}$ idiosinkratska komponenta. Iz navedenog rastava kovarijance moguće je dobiti izraze za varijancu i kovarijancu pojedinačnih varijabli [6]:

$$\hat{\sigma}^2_i = \hat{\beta}^2_{i1} + \dots + \hat{\beta}^2_{iP} + \hat{\psi}_i$$
$$\hat{\sigma}_{ij} = \hat{\beta}_{i1}\hat{\beta}_{j1} + \dots + \hat{\beta}_{iP}\hat{\beta}_{jP}.$$

Koeficijenti faktora (β) se računaju izravno iz podataka te predstavljaju odnos pojedine imovine i cjelokupnog tržišta. Idiosinkratsku komponentu nije moguće izračunati izravno iz podataka. Iz tog razloga koristi se njena procjena:

$$\hat{\Psi} = S - \hat{\beta}\hat{\beta}'\sigma^2_{ind}, \tag{2.12}$$

gdje je S uzoračka kovarijacijska matrica.

Tržišni indeks je portfelj kojim se predstavlja tržište. Povrati ovog indeksa se koriste pri usporedbi imovine i tržišta. Pri kreiranju tržišnog indeksa, povrati imovine skalirani su proporcionalno njihovoj tržišnoj kapitalizaciji.

Težine kojima se povrati množe u indeksu ponderiranom prema tržišnoj kapitaliza-

ciji računaju se formulom

$$w_i = \frac{mcap_i}{\sum_{j=0}^{N} mcap_j}\,, \tag{2.13}$$

gdje je $mcap_i$ iznos tržišne kapitalizacije pojedine imovine, a N ukupan broj odabranih imovina. Iz toga proizlazi da je ukupan povrat tržišnog indeksa zbroj umnožaka pojedinačnih povrata imovine i njihovog udjela u ukupnoj tržišnoj kapitalizaciji:

$$r_{ind} = \sum_{i=0}^{N} w_i r_i\,. \tag{2.14}$$

U financijama se s β označava volatilnost financijske imovine ili portfelja u odnosu na tržište. Tržišna β pruža investitoru procjenu koliko će dodatnog rizika preuzeti ako u svoj portfelj uvrsti neku imovinu te mu pomaže razumjeti kreće li se ta imovina u skladu s ostatkom tržišta [7]. Računa se tako da se kovarijancu povrata imovine i tržišta podijeli s varijancom povrata tržišta:

$$\hat{\beta} = \frac{\hat{\sigma}_{i,ind}}{\sigma^2_{ind}}\,. \tag{2.15}$$

Konačni iznos tržišne bete ovisi o vremenskom prozoru nad kojim se računa. U kontekstu korištenja faktorskog modela za procjenu rizičnosti portfelja, ove će β poslužiti kao koeficijenti za izračun kovarijacijske matrice.

Što znače različite vrijednosti tržišnih beta?

- $\beta > 1$: Promjena u povratima imovine je tijekom vremena volatilnija od povrata tržišta.[7]
- $\beta = 1$: Promjena povrata imovine prati promjene u povratima tržišta. Dodavanje ove imovine u portfelj ne povećava rizik portfelja, ali ni ne povećava izglednost da će portfelj ostvariti veću zaradu od predviđene.[7]
- $0 < \beta < 1$: Povrati imovine su manje volatilni od povrata tržišta. Uključivanjem ove imovine u portfelj čini ga manje rizičnim od istog portfelja bez te imovine.[7]
- $\beta < 0$: Negativna tržišna beta znači da je imovina negativno korelirana s povratima tržišta. [7]

Prvi pokušaji procjene kovarijacijske matrice faktorskog modela temeljili su se na

izravnoj predikciji tržišnih β. Taj se pristup nije pokazao dobrim iz dva razloga:

1. korišteni modeli bi redovito davali kao procjenu β koje objašnjavaju više od 100% varijance radi čega bi neki elementi na tragu idiosinkratske komponente bili negativni:
$$\hat{\sigma}_i^2 - \hat{\beta}^2 \sigma_{ind}^2 < 0$$
što na kraju rezultira kovarijacijskom matricom koja nije pozitivno definitna,

2. nije moguće očuvati trag uzoračke kovarijacijske matrice (S) što može biti poželjno svojstvo pri procijeni buduće varijance.

Pozitivna definitnost je nužno svojstvo kovarijacijske matrice radi kojeg se prešlo na novi pristup procjeni. Ono osigurava da su svi elementi na tragu procijenjene idiosinkratske komponente pozitivni. Taj pristup se temelji na procjeni korelacijskog koeficijenta između financijske imovine i indeksa tržišta iz kojeg se kasnije izračunaju koeficijenti faktorskog modela.

Kako iz korelacijskog koeficijenta izračunati tržišne bete? Najlakši način za odgovoriti na to pitanje je krenuti obrnutim smjerom - kako od tržišnih β-a doći do korelacijskog koeficijenta. Iz formule 2.11 za rastav kovarijance faktorskog modela može se izvesti formula za varijancu pojedine imovine:

$$\hat{\sigma}_i^2 = \hat{\beta}_i^2 \sigma_{ind}^2 + \hat{\psi}_i^2 \,. \tag{2.16}$$

Pod pretpostavkom da ne postoje dodatni izvori varijacije u podacima (idiosinkratska komponenta je jednaka 0) vrijedi:

$$\hat{\sigma}_i^2 = \hat{\beta}_i^2 \sigma_{ind}^2 \,. \tag{2.17}$$

Na dobiveni izraz primjeni se ograničenje da β ne smiju objašnjavati više od 100% varijance:

$$\hat{\sigma}_i^2 - \hat{\beta}_i^2 \sigma_{ind}^2 \geq 0 \,. \tag{2.18}$$

Nejednadžba se podijeli s varijancom imovine:

$$\hat{\beta}_i^2 \sigma_{ind}^2 \leq \hat{\sigma}_i^2$$

$$\hat{\beta}_i^2 \frac{\sigma_{ind}^2}{\hat{\sigma}_i^2} \leq 1 \,.$$

Zatim se β zamijeni prema formuli 2.15 i nakon korjenovanja dobije:

$$\frac{\hat{\sigma}_{i,ind}^2 \sigma_{ind}^2}{\sigma_{ind}^4 \hat{\sigma}_i^2} \leq 1$$

$$\sqrt{\frac{\hat{\sigma}_{i,ind}^2}{\sigma_{ind}^2 \hat{\sigma}_i^2}} \leq 1$$

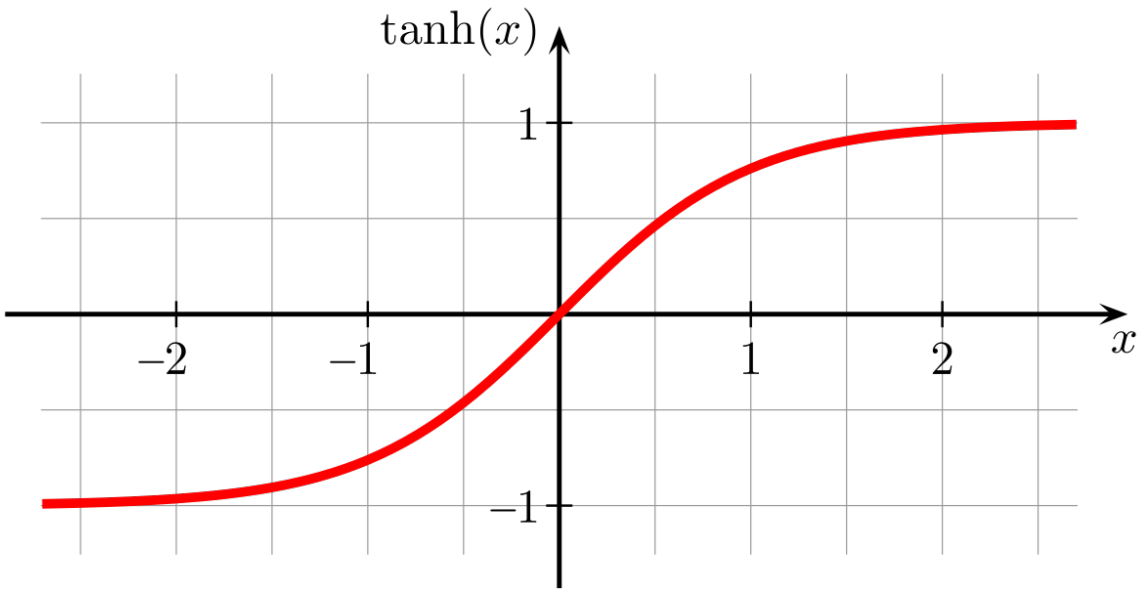
$$-1 \leq \frac{\hat{\sigma}_{i,ind}}{\hat{\sigma}_i \sigma_{ind}} \leq 1$$

što je upravo izraz za korelacijski koeficijent (2.4). Tržišnu β dobijemo iz korelacijskog koeficijenta tako da ga pomnožimo omjerom standardnih devijacija povrata imovine i tržišta:

$$\hat{\rho}_{i,ind} \frac{\hat{\sigma}_i}{\sigma_{ind}} = \frac{\hat{\sigma}_{i,ind}}{\hat{\sigma}_i \sigma_{ind}} \frac{\hat{\sigma}_i}{\sigma_{ind}} = \frac{\hat{\sigma}_{i,ind}}{\sigma_{ind}^2} = \hat{\beta}_i \,. \tag{2.19}$$

Ovako izračunate β imaju garanciju da objašnjavaju manje od 100% varijance zato što su podaci financijskih vremenskih nizova nad kojima su izračunate standardizirani - vrijednost korelacijskog koeficijenta nikada neće izaći iz ograničenja $|\rho| \leq 1$.

Različiti modeli na različit način modeliraju povezanost ulaznih parametara i ciljne varijable (korelacijskog koeficijenta). Kod modela koji se temelje na neuronskim mrežama u izlaznom sloju se kao aktivacijska funkcija koristi tangens hiperbolni (slika 2.4.) koji ima horizontalne asimptote u točkama -1 i 1 te odlično može poslužiti za predikciju vrijednosti korelacije. Više o samim modelima se može pronaći u poglavlju 3.



Slika 2.4. Tangens hiperbolni [8].

3. Modeli strojnog učenja za procjenu latentnih faktora

U ovom poglavlju opisane su teorijske osnove i funkcioniranje modela strojnog učenja koji su korišteni u procjeni latentnih faktora.

3.1. Osnovni modeli

U istraživanjima u kojima se isprobavaju različiti modeli i uspoređuje njihova efikasnost potrebno je postaviti osnovne modele koji služe kao referentna točka u usporedbi. Ako napredniji (kompliciraniji) modeli imaju lošije rezultate od ovih osnovnih, moguće je doći do zaključka da uvođenjem naprednijih metoda i algoritama nije postignut željeni napredak i zato nema potrebe za njima.

U ovome radu osnovni modeli zapravo nisu modeli već povijesne tržišne β koje se dobiju iz podataka formulom 2.15 Ideja je da će tržišne bete u budućnosti ostati iste kao i u prošlosti, odnosno da je koeficijente faktorskog modela moguće procijeniti njihovim trenutnim vrijednostima. Izračun tržišne β ovisi o vremenskom prozoru nad kojim se računa iz čega proizlaze četiri osnovna "modela", a radi se o β izračunatim nad vremenskim prozorima u prošlost od 7, 30, 90 i 180 dana.

3.2. Linearna regresija

Regresijom nad skupom D dobiva se funkcija h kao aproksimacija stvarne funkcije preslikavanja ulaznih parametara u ciljnu varijablu [9]. Empirijska pogreška funkcije h na skupu za učenje D najjednostavnije se može definirati kao [9]

$$E(h|D) = \sum_{i=1}^{N} (y_i - h(x_i))^2 \,. \tag{3.1}$$

Pogreška se mjeri kao zbroj kvadratnih odstupanja predviđene i stvarne vrijednosti [9]. Regresija je linearna ako vrijednost $h(x)$ linearno ovisi o ulaznim vrijednostima x:

$$h(\vec{x}) = v_1 x_1 + v_2 x_2 + \cdots + v_n x_n + v_0 = \sum_{i=1}^{n} v_i x_i + v_0 = v'x + v_0 \,, \tag{3.2}$$

gdje su v_i parametri težina koje treba naučiti na temelju skupa primjera D [9]. Cilj je pronaći regresijsku funkciju h koja minimizira empirijsku pogrešku [9]. Istrenirani model vrši predikciju uvrštavanjem neviđenih uzoraka u regresijsku funkciju. Linearna regresija spada u algoritme nadziranog strojnog učenja.

3.3. Polinomijalna regresija

Ako je linearan model prejednostavan, empirijska pogreška bit će i nakon optimizacije prevelika [9]. U tom slučaju moguće je odabrati složeniji regresijski model - polinomijalni. Polinomijalna regresija drugog reda s dvije ulazne vrijednosti x_1 i x_2 ima oblik

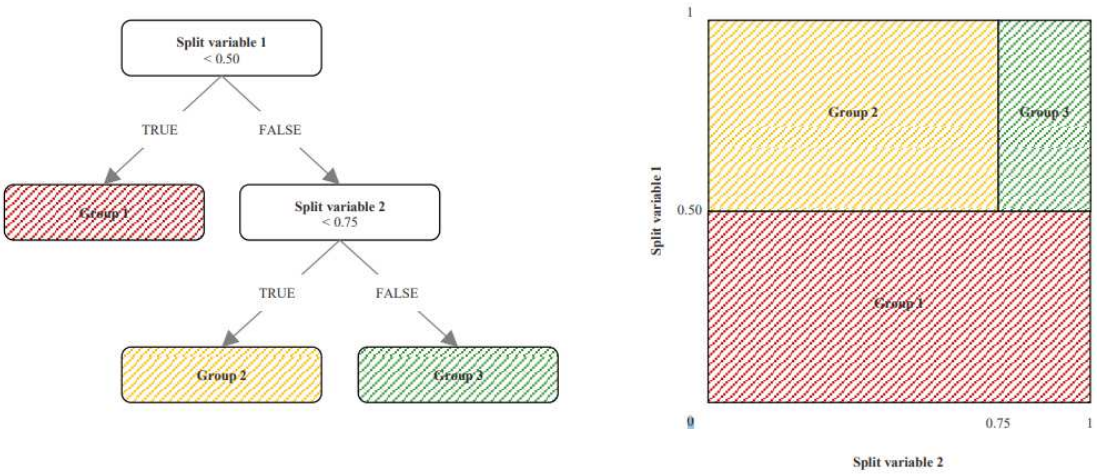
$$h(x; v) = v_0 + v_1 x_1 + v_2 x_2 + v_3 x_1 x_2 + v_4 x_1^2 + v_5 x_2^2 \,. \tag{3.3}$$

Značajka $x_1 x_2$ se naziva interakcijska značajka (eng. *cross-term*) i modelira interakciju između tih vrijednosti. U slučaju većeg broja ulaza formulu čine svi parovi interakcijskih značajki.

3.4. Regresor slučajnih šuma

Slučajne šume (eng. *Random Forest*) su algoritam nadziranog strojnog učenja koji kombinira više stabala odluke u jedinstveni model koji predviđa kontinuirane vrijednosti [10], a radi na temelju procesa *bagginga* (skraćeno od *Bootstrap Aggregating*, slika 3.2.) - nasumično se izabere podskupove podataka za treniranje i nad njima izgradi manja stabla odluke [10]. Slika 3.1. prikazuje stiliziranu vizualizaciju strukture i funkcioniranja jednog stabla odluke.

Svako stablo odluke rekurzivno particionira prostor ulaznih parametara tako da su uzorci sa sličnim ciljnim vrijednostima zajedno grupirani. Za svaku moguću podjelu $\theta = (j, t_m)$ na temelju parametra j i praga t_m, podskup podataka Q_m s n_m uzoraka u



Slika 3.1. Stilizirana vizualizacija strukture i funkcioniranja regresijskog stabla [5].

čvoru m se dijeli na podskupove $Q_m^{lijevo}(\theta)$ i $Q_m^{desno}(\theta)$:

$$Q_m^{lijevo}(\theta) = (x, y)|x_j \leq t_m Q_m^{desno}(\theta) = Q_m \, Q_m^{lijevo}(\theta) \, .$$

Kvaliteta podjele parametara u čvoru m se računa s pomoću funkcije gubitka H koja je u ovom slučaju srednja kvadratna pogreška (formula 3.1):

$$G(Q_m, \theta) = \frac{n_m^{lijevo}}{n_m} H(Q_m^{lijevo}(\theta)) + \frac{n_m^{desno}}{n_m} H(Q_m^{desno}(\theta)) \, . \tag{3.4}$$

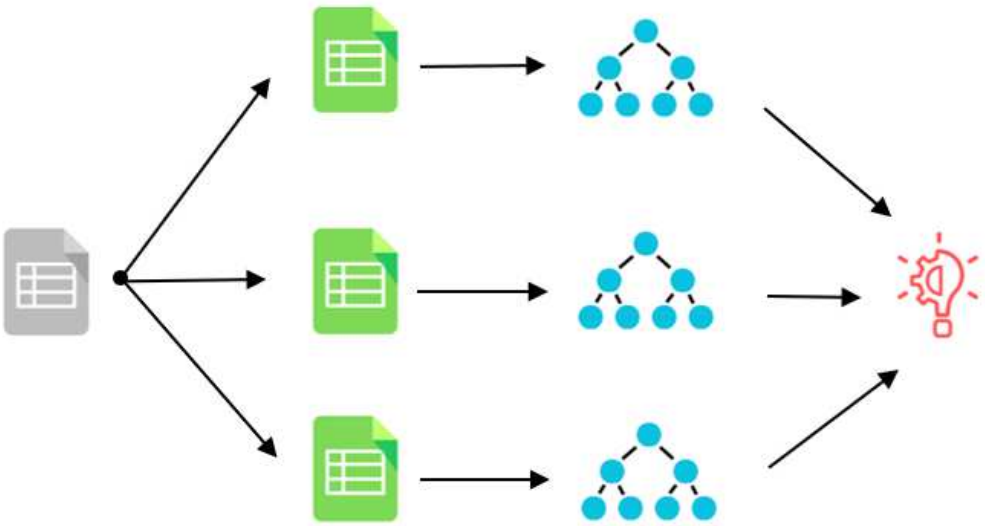
Odabir parametara koji će se koristiti u idućoj iteraciji rekurzije vrši se minimizacijom funkcije kvalitete:

$$\theta = argmin_\theta G(Q_m, \theta) \, . \tag{3.5}$$

Rekurzija traje dok se ne stigne do maksimalne dopuštene dubine ili $n_m < min_{uzoraka}$ ili $n_m = 1$ [11]. Konačna predikcija se dobije kao aritmetička sredina svih vrijednosti na listovima stabla odluke.

Neke prednosti regresora slučajnih šuma u odnosu na druge modele su:

- ne čini pretpostavke o distribuciji podataka,
- nema problema s predikcijom varijabli koje imaju nelinearan odnos,
- izbjegava prenaučenost kreiranjem slučajnih podskupova podataka na kojima tre-



Slika 3.2. Proces bagginga.

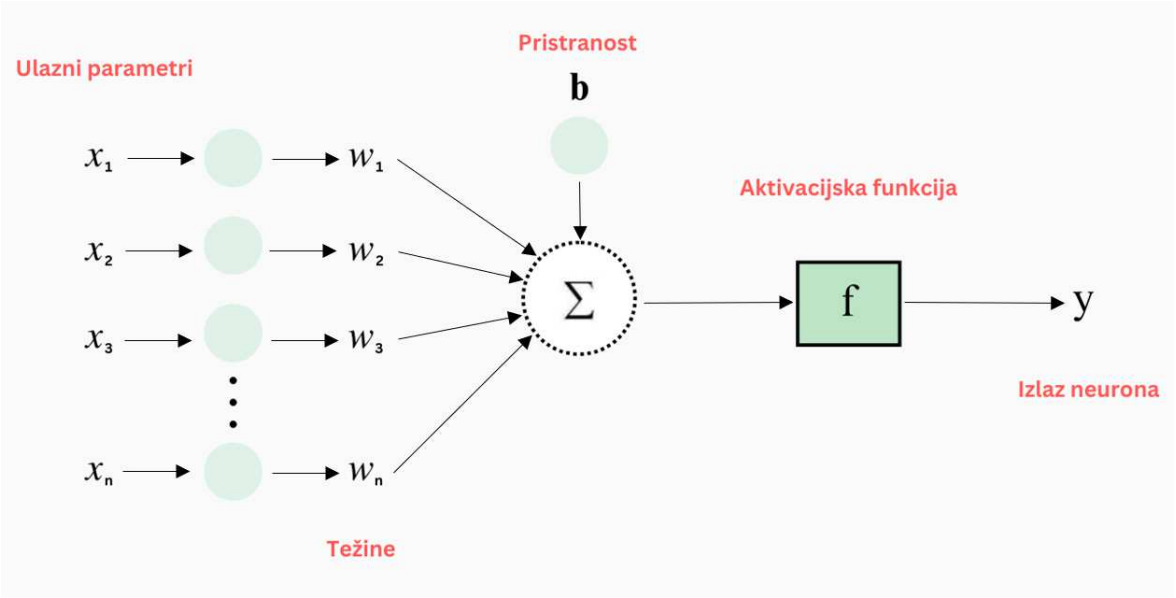
nira stabla odluke,

- koristi model bijele kutije - objašnjenje uvjeta podjele je lako objasniti korištenjem Booleove logike [11],
- pruža informaciju o korisnosti pojedinih ulaznih parametara u dobivanju konačne predviđene vrijednosti izraženu u postocima,

3.5. Umjetne neuronske mreže

Umjetne neuronske mreže su tip programske arhitekture koji se koristi u dubokom učenju i koji oponaša rad neurona ljudskog živčanog sustava. Primjenom algoritama dubokog učenja moguće je naučiti mrežu kako riješiti različite tipove problema nadziranog strojnog učenja. Funkcioniranje jednog neurona je prikazano slikom 3.3. - neuron prima brojčane ulaze (mogu biti izlazi njegovih prethodnika ili vrijednosti ulaznih parametara) koji se množe s odgovarajućim težinama, zbrajaju i zatim predaju kao argument aktivacijskoj funkciji kojom se određuje izlazna vrijednost.

Proces treniranja neuronskih mreža jest traženje optimalnih vrijednosti težina i pristranosti (eng. *bias*), a funkcionira ovako:



Slika 3.3. Neuron umjetne neuronske mreže.

1. težine svih neurona se inicijaliziraju nasumičnim vrijednostima,
2. odredi se jesu li vrijednosti težina veće ili manje od optimalnih,
3. optimizatorom se izračunaju nove vrijednosti težina,
4. koraci 2. i 3. se ponavljaju sve dok se ne dostigne zadani broj iteracija (epoha) ili se ispuni neki od uvjeta zaustavljanja (npr. greška predikcije se ne smanjuje određeni broj iteracija).

Optimizatori su algoritmi dubokog učenja kojima se mijenjaju težine neuronskih mreža kako bi se minimizirala greška predikcije. Bitan argument optimizatora je stopa učenja - proizvoljno odabran decimalan broj kojim se podešava magnituda promjene. Optimizatori isprobani u istraživanju su stohastički gradijentni spust (eng. *stohastic gradient descent - SGD*) i adaptivna estimacija momenta (eng. *adaptive moment estimation* - Adam)

SGD je inačica algoritma gradijentnog spusta. Njime se iterativno osvježavaju težine neuronske mreže u smjeru negativnog gradijenta funkcije gubitka:

$$\theta_{n+1} = \theta_n - \alpha * \nabla J(\theta_n + 1)\,, \tag{3.6}$$

gdje θ_{n+1} predstavlja nove vrijednosti težina, θ_n trenutne, a α je fiksna stopa učenja. Ono što je posebno u vezi SGD-a je što se nove vrijednosti težina računaju za svaki uzorak.

Adam optimizator proširiva način rada SGD-a tako da prilagođava stopu učenja kod računanja svake težine u mreži pri čemu koristi procjene prvog (srednja vrijednost) i drugog (necentrirana varijanca) momenta gradijenta:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla J(\theta_n)$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla J(\theta_n))^2 \,,$$

gdje su m_t/v_t nova vrijednost prvog/drugog momenta, β_1/β_2 hiperparametar propadanja momenta i $\nabla J(\theta_n)$ iznos gradijenta. Formula po kojoj se računaju nove vrijednosti težina glasi:

$$\theta_{n+1} = \theta_n - \frac{\alpha m_t}{\sqrt{v_t} + \epsilon} \,. \tag{3.7}$$

ϵ je ovdje maleni broj (npr. 10^{-8}) koji se dodaje kako se ne bi dogodilo dijeljenje s nulom čime se zadržava numerička stabilnost [12].

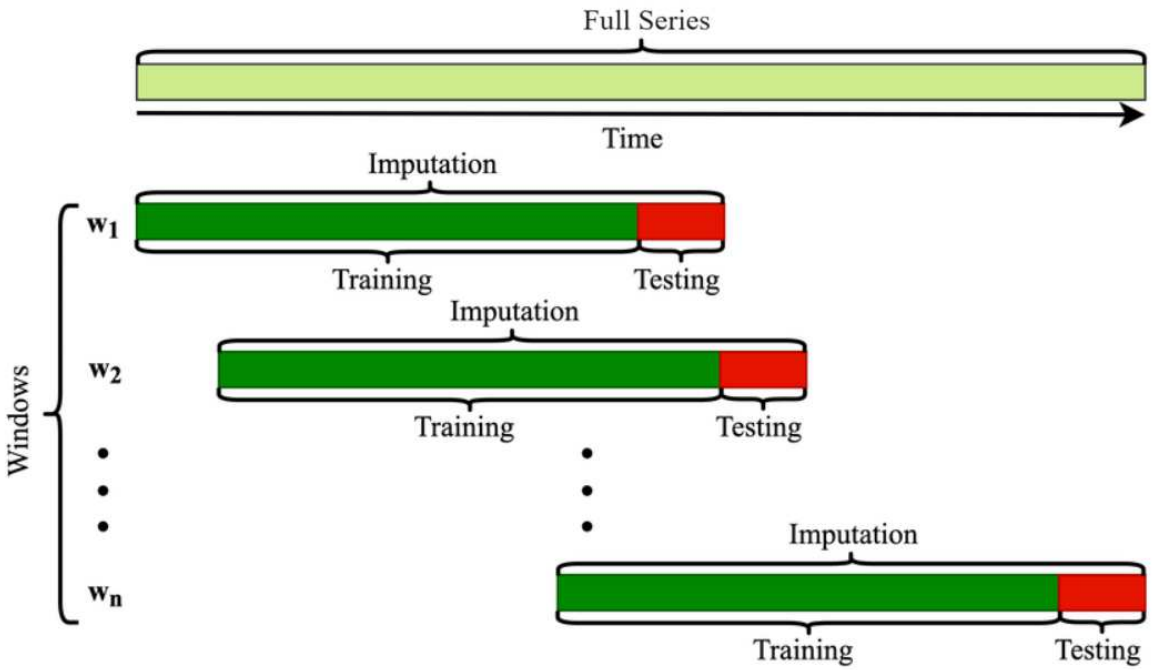
Umjetne neuronske mreže se često koriste u rješavanju problema klasifikacije gdje uspješno modeliraju preslikavanje ulaznih parametara u neku kategorijsku zavisnu varijablu (npr. prepoznavanje objekata na slici). U ovom radu se njima predviđa kontinuirana vrijednost korelacijskog koeficijenta, odnosno problem procjene je regresijski. Neuronske mreže jako dobro rade s velikim skupovima podataka i mogu prepoznati kompliciranе uzorke u podacima koji su možda izvan dohvata jednostavnijih modela.

4. Podaci i implementacija procjene latentnih faktora

U ovom poglavlju opisano je kako se u istraživanju procjenjuju latentni faktori u financijskim vremenskim nizovima. Objašnjava se na koji način i u kojem trenutku se koriste prethodno opisane matematičke formule i računarski modeli. Daje se pregled ulaznih parametara, implementacijskih detalja i tehnologija. Opisana je cjelovita ideja istraživanja, od prikupljanja podataka do procjene latentnih faktora i kovarijacijske matrice. U pisanju koda korištena je interaktivna platforma Jupyter Notebook, programski jezik Python i njegove knjižnice za provođenje matematičkih operacija (Numpy, Scipy), upravljanje podacima (Pandas) i implementaciju modela strojnog učenja (Statsmodels, Tensorflow, Scikit Learn).

Financijska imovina s kojom je provedeno istraživanje su kriptovalute. Podaci su uzeti s istraživačke platforme *Coinpaprika* te se sastoje od dnevne tržišne kapitalizacije, volumena trgovanja i cijena nešto više od dvije tisuće kriptovaluta u razdoblju od 2013. do 2021. godine. U istraživanju se koriste podaci od 2017. godine nadalje radi velikog broja nedostajućih vrijednosti među starijim podacima (poveći broj kriptovaluta je nastao koju godinu nakon 2013.). Pri dohvaćanju se kriptovalute filtriraju prema uvjetima popisanima u tablici 4.1.

Prvi korak u procjeni latentnih faktora je prikupljanje podataka za treniranje i testiranje modela. Temelji se na mehanizmu objedinjenog (eng. *pooled*) klizećeg prozora (slika 4.1.). On radi tako da se za neko razdoblje, čije trajanje je određeno duljinom prozora, dohvate informacije o financijskoj imovini i izračunaju ulazni parametri. Prozor se zatim pomiče za fiksni broj dana unaprijed i cijeli proces se ponavlja dok se ne dođe do krajnje datumske granice. Zove se objedinjeni, zato što se podaci, koji se dobiju u svakom pomaku, spremaju u zajednički (objedinjeni) skup podataka iz kojeg će nastati podaci za



Slika 4.1. Princip rada klizećeg prozora [13].

treniranje/testiranje tako da se taj skup izravna (eng. *flatten*). Izravnanjem se dobije novi skup podataka čiji stupci su ulazni parametri [14], a redci uzorci financijske imovine kroz vrijeme što je prikazano slikom 4.2. Jedna posebnost u primjeni klizećeg prozora u ovom

KRIPTOVALUTA	ZNAČAJKE								
	avg_mcap	curr_mcap	avg_vol	ret_var	avg_return	beta7	beta30	beta90	beta180
btc_bitcoin	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.
eth_ethereum	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.
usdt_tether	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.	1.1.2018., ..., 1.5.2021.
...	...	...	...	...	...	...	...	...	...

KRIPTOVALUTA	ZNAČAJKE									
	date	avg_mcap	curr_mcap	avg_vol	ret_var	avg_return	beta7	beta30	beta90	beta180
btc_bitcoin	1.1.2018.	22.938.709.551.200.000	22.938.709.551.200.000	22.938.709.551.200.000	0.001850114	-3,1831E+16	0.813917941	0.813917941	0.813917941	0.813917
btc_bitcoin	2.1.2018.	23.588.104.032.400.000	23.588.104.032.400.000	23.588.104.032.400.000	0.001850114	-3,1831E+16	0.813917941	0.813917941	0.813917941	0.813917
...	...	...	...	...	...	...	...	...	...	...
eth_ethereum	1.1.2018.	7.388.065.406.600.000	7.388.065.406.600.000	7.388.065.406.600.000	0.002813495	0.003183138	0.969118189	0.969118189	0.969118189	0.969118
eth_ethereum	2.1.2018.	8.462.686.705.100.000	8.462.686.705.100.000	8.462.686.705.100.000	0.002813495	0.003183138	0.969118189	0.969118189	0.969118189	0.969118
...	...	...	...	...	...	...	...	...	...	...

Slika 4.2. Izravnavanje podataka s obzirom na ulazne parametre modela.

istraživanju je što se on dijeli na dva manja vremenska prozora iz kojih se dobiju "podaci u uzorku" (eng. *in-sample*) i "podaci izvan uzorka" (eng. *out-of-sample*). Podaci izvan uzorka se koriste za izračun korelacijskog koeficijenta koji se predaje kao ciljna varijabla modelima. Oni se nalaze u budućnosti od odabranog referentnog trenutka koji dijeli dva

manja vremenska prozora. Duljina njihovog prozora je sedam dana. Podaci u uzorku se koriste za izračun vrijednosti ulaznih parametara koji se predaju modelima. Njihov vremenski prozor se nalazi u prošlosti u odnosu na odabrani referentni trenutak i traje, ovisno o modelu, sedam, trideset, devedeset ili sto osamdeset dana.

U svrhu implementacije mehanizma klizećeg prozora napisano je nekoliko pomoćnih klasa, a sam algoritam ide ovako:

1. instancira se podatkovna struktura koja predstavlja objedinjeni skup podataka,
2. s pomoću klase za rukovanje podacima dohvate se, unutar zadanih datumskih ograničenja klizećeg prozora, vrijednosti tržišne kapitalizacije, volumena trgovanja i povrata kriptovaluta,
3. izračunaju se težine indeksa tržišta,
4. izračunaju se povrati indeksa tržišta,
5. za svaku dohvaćenu kriptovalutu izračunaju se ulazni parametri i spreme u objedinjeni skup podataka,
6. klizeći prozor se pomakne za fiksni broj dana,
7. ponavljaju se koraci 2.-6. dok se vremenskim prozorom ne dođe do završnog datuma vremenskog niza.

`CryptoDataHandler` je pomoćna klasa za rukovanje podacima koja pri inicijalizaciji učitava podatke u Pandas okvire podataka (eng. *data frame*) i u kojoj se pri dohvaćanju provodi filtriranje podataka iz tablice 4.1. Ulazni parametri koji se računaju i spremaju u objedinjeni skup podataka opisani su u tablici 4.2. Modeli su trenirani sa svim ili podskupom parametara što će u poglavlju s rezultatima biti istaknuto.

Tržišna kapitalizacija predstavlja ukupnu vrijednost imovine na tržištu izraženu u američkim dolarima (USD). Računa se kao umnožak vrijednosti jedinice imovine u USD i ukupnog broja te imovine na tržištu [15]. Volumen trgovanja je količina imovine koja promijeni vlasnika (koristi se u trgovanju) tijekom određenog razdoblja, najčešće jednog dana. Generalno, imovina s većim dnevnim volumenom je likvidnija te njene promjene

Svi podaci
* Filtriranje prema datumskim ograničenjima zadanima širinom klizećeg prozora.
Tržišna kapitalizacija
* Ne smije sadržavati nepoznate vrijednosti. * Mora svaki dan biti veća od zadanog praga od milijun USD. * Dohvaća se samo prvih N kriptovaluta poredanih silazno prema tržišnoj kapitalizaciji. Broj N se proizvoljno određuje.
Volumen trgovanja
* Mora u svakom trenutku biti veći od nula. * Ne smije imati nepoznate vrijednosti.
Cijena
* Ne postoje uvjeti filtriranja po cijeni kriptovalute.
Povrati
* Ne smiju drastično odskakati od ostalih povrata (eng. *outlier*), u suprotnom se smatraju šumom. Gornja i donja granica ekstremnih vrijednosti su empirijski određene. * Varijanca im mora biti različita od nula. Služi za izbacivanje *stablecoinova* iz skupa podataka koji smetaju predikciji.

Tablica 4.1. Uvjeti filtriranja podataka.

u cijeni imaju veći značaj [16]. Svi parametri, osim povijesnih β, su transformirani prije treniranja. Primjenjuju se dva pristupa transformaciji:

1. standardizacija korištenjem Z vrijednosti,

2. primjena prirodnog logaritma (trenutna i prosječna tržišna kapitalizacija, prosječni volumen), kvadratnog korijena (varijanca povrata) ili anualizacije (trenutni i varijanca povrata).

Transformacije se provode zato što postoje velike razlike u redovima veličina parametara što modelima otežava modeliranja njihovog odnosa s ciljnom varijablom i povećava pogrešku procjene.

Idući korak procjene latentnih faktora je treniranje modela strojnog učenja za što se koriste njihove ugrađene metode. Bitno je osigurati da su dimenzionalnosti podataka za treniranje ispravne kako se ne bi dogodile greške kasnije u predikciji. Glavna razlika je u dimenzionalnosti između skupa podataka kojeg čine samo povijesne β i ostalih koji

Parametar	Definicija
beta7	uzoračka povijesna tržišna β na vremenskom prozoru od 7 dana
beta30	uzoračka povijesna tržišna β na vremenskom prozoru od 30 dana
beta90	uzoračka povijesna tržišna β na vremenskom prozoru od 90 dana
beta180	uzoračka povijesna tržišna β na vremenskom prozoru od 180 dana
avg_mcap	prosječna tržišna kapitalizacija u prethodnih 180 dana
current_mcap	tržišta kapitalizacija u referentnom trenutku promatranja
avg_vol	prosječni volumen trgovanja u prethodnih 180 dana
avg_ret	prosječni povrati u prethodnih 180 dana
var_ret	varijanca povrata u prethodnih 180 dana

Tablica 4.2. Opis ulaznih parametara modela strojnog učenja.

sadrže vrijednosti svih ulaznih parametara.

Model linearne regresije je uzet iz knjižnice statsmodels. Točnije, radi se o klasi *Ordinary Least Squares* (OLS). Podacima se kod korištenja ovog modela prije treniranja i testiranja metodom `add_constant` dodaje stupac jedinica radi množenja s težinom koja određuje pristranost regresije. Za treniranje se koristi metoda `fit`, a za predikciju metoda `predict`. Najveća prednost ovog modela je detaljni ispis rezultata nakon treniranja. Primjer ispisa modela koji se trenirao nad skupom podataka sastavljenog od povijesnih tržišnih β prikazan je na slici 4.3.

Polinomijalna regresija je ostvarena s pomoću nekoliko klasa i funkcija iz knjižnice Scikit Learn. Funkcijom `make_pipeline` kreira se instanca klase `Pipeline` koja omogućava izvođenje operacija nad podacima korak po korak. Predaju joj se instance klasa `PolynomialFeatures` i `Ridge`. Klasa `PolynomialFeatures` kreira novi skup podataka koji se sastoji od svih polinomijalnih kombinacija ulaznih parametara stupnja manjeg ili jednakog od zadanog. U istraživanju se koristi polinom drugog stupnja koji na primjeru podataka s dva ulazna parametra a i b ima kombinacije: $1, a, b, a^2, ab, b^2$. `Ridge` klasom je implementirana OLS regresija s L2 regularizacijom. Regularizacija se koristi u svrhu sprječavanja prenaučenosti modela polinomijalne regresije.

Kao što je opisano u odjeljku 3.4., *Random Forest* regresor trenira nad podskupovima podataka zadani broj stabala odluke iz čijih izlaza usrednjavanjem daje konačnu predikciju. Takvim načinom predikcije se kontrolira prenaučenost modela. Implementacija i ovog modela je uzeta iz knjižnice Scikit Learn. Pri instanciranju se objektu predaju idući hiperparametri:

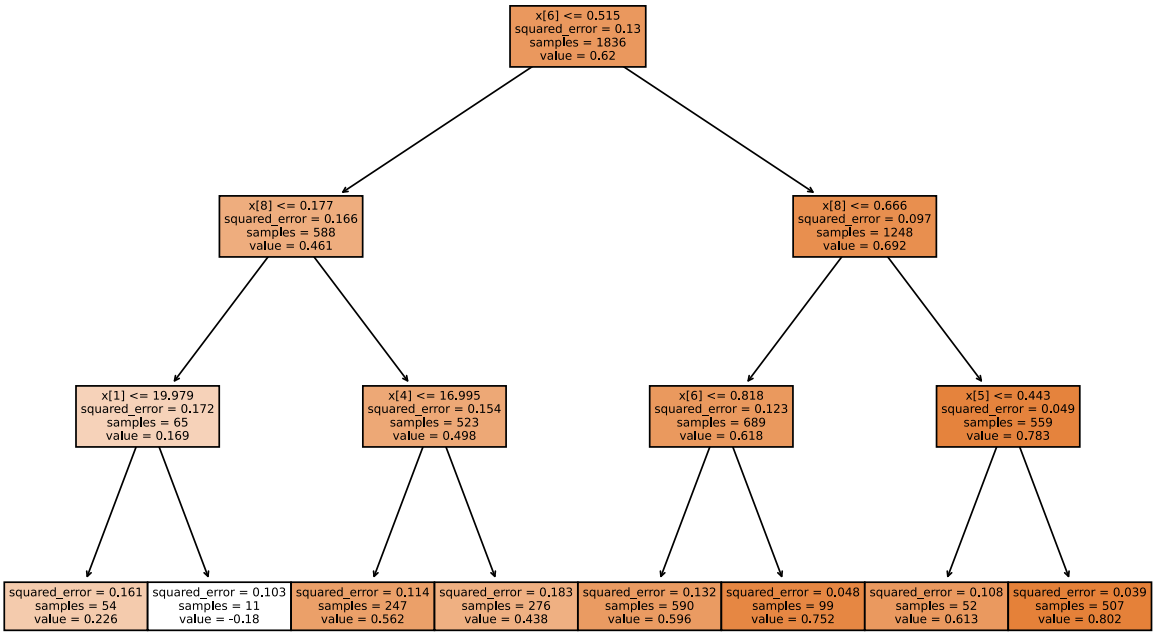
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     10   R-squared (uncentered):                   0.861
Model:                            OLS   Adj. R-squared (uncentered):              0.858
Method:                 Least Squares   F-statistic:                              322.3
Date:                Wed, 08 May 2024   Prob (F-statistic):                    6.34e-88
Time:                        22:09:12   Log-Likelihood:                         -134.24
No. Observations:                 212   AIC:                                      276.5
Df Residuals:                     208   BIC:                                      289.9
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
7              0.3411      0.081      4.211      0.000       0.181       0.501
30             0.1733      0.171      1.013      0.312      -0.164       0.511
90             0.1717      0.289      0.594      0.553      -0.398       0.742
180            0.2980      0.227      1.314      0.190      -0.149       0.745
==============================================================================
Omnibus:                       66.783   Durbin-Watson:                   1.766
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              282.788
Skew:                           1.178   Prob(JB):                     3.92e-62
Kurtosis:                       8.144   Cond. No.                         26.4
==============================================================================
```

Slika 4.3. Ispis rezultata linearne regresije nakon treniranja.

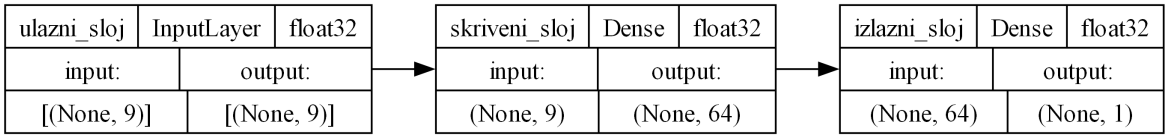
- **n_estimators** broj stabala odluke u modelu; koristi se 50 stabala odluke,
- **max_depth** maksimalna dopuštena dubina pojedinog stabla odluke; zadana vrijednost dubine je 3,
- **random_state** broj koji kontrolira nasumičnost procesa *bagginga* i nasumičnost uzorkovanja ulaznih parametara pri razdvajanju čvora; koristi se empirijski odabran broj 37.

Za sve ostale hiperparametre koriste se njihove zadane (eng. *default*) vrijednosti. Na slici 4.4. se nalazi primjer stabla odluke nakon treniranja na kojem je moguće pratiti slijed odluka do listova.

Neuronske mreže nude puno opcija u dizajniranju njihove arhitekture. Za potrebe ovog istraživanja u izradi modela je korištena knjižnica TensorFlow. Koriste se Keras sekvencijski modeli unaprijedne neuronske mreže s tri sloja - ulazni, skriveni i izlazni. Primjer arhitekture jedne od korištenih neuronskih mreža prikazan je na slici 4.5. Ulazni



Slika 4.4. Primjer stabla odluke od kojih je napravljen *Random Forest* regresor.



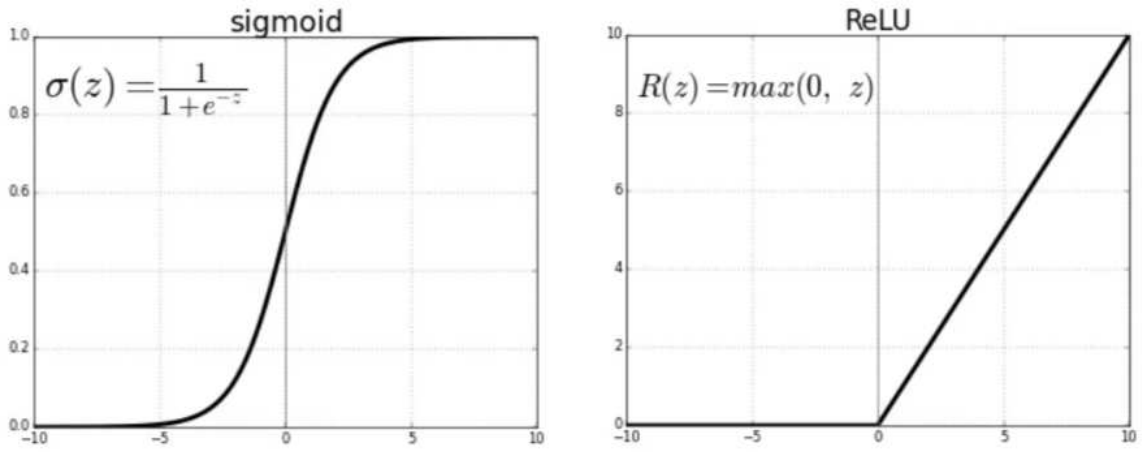
Slika 4.5. Primjer arhitekture modela umjetne neuronske mreže.

sloj *InputShape* je komponenta koja definira oblik i tip podataka. Nužna je za određivanje ulazne dimenzije tenzora na čemu se gradi ostatak arhitekture. Ovaj sloj ne provodi nikakve izračune već samo služi kao ulazna točka modela. Skriveni sloj čini potpuno povezani (gusti) unaprijedni sloj koji implementira operaciju

$$output = activation(dot(input, kernel) + bias)$$

gdje je *activation* aktivacijska funkcija koju se primjenjuje na svaki neuron i predaje ju se u argumentima funkcije sloja, *kernel* je matrica težina koju kreira sam sloj i *bias* je vektor pristranosti kojeg isto interno kreira sloj [17] (kako pristranost utječe na pojedini neuron prikazano je slikom 3.3.). U istraživanju su kao aktivacijske funkcije isprobane *Rectified Linear Unit* (ReLU) i sigmoidalna funkcija (Slika 4.6.). Ovaj sloj, u svim implementiranim modelima, čine šezdeset i četiri neurona. Izlazni sloj je također potpuno

povezani unaprijedni sloj, ali sastoji se od samo jednog neurona čiji izlaz predstavlja procjenu korelacijskog koeficijenta. Kako je ovaj koeficijent ograničen uvjetom $-1 < \rho < 1$ kao aktivacijska funkcija izabran je tangens hiperbolni (Slika 2.4.) koji s asimptotama u kodomeni na -1 i 1 zadovoljava zadano ograničenje. Metodom *compile* sekvencijalnog



Slika 4.6. Sigmoidalna i ReLU aktivacijska funkcija [18].

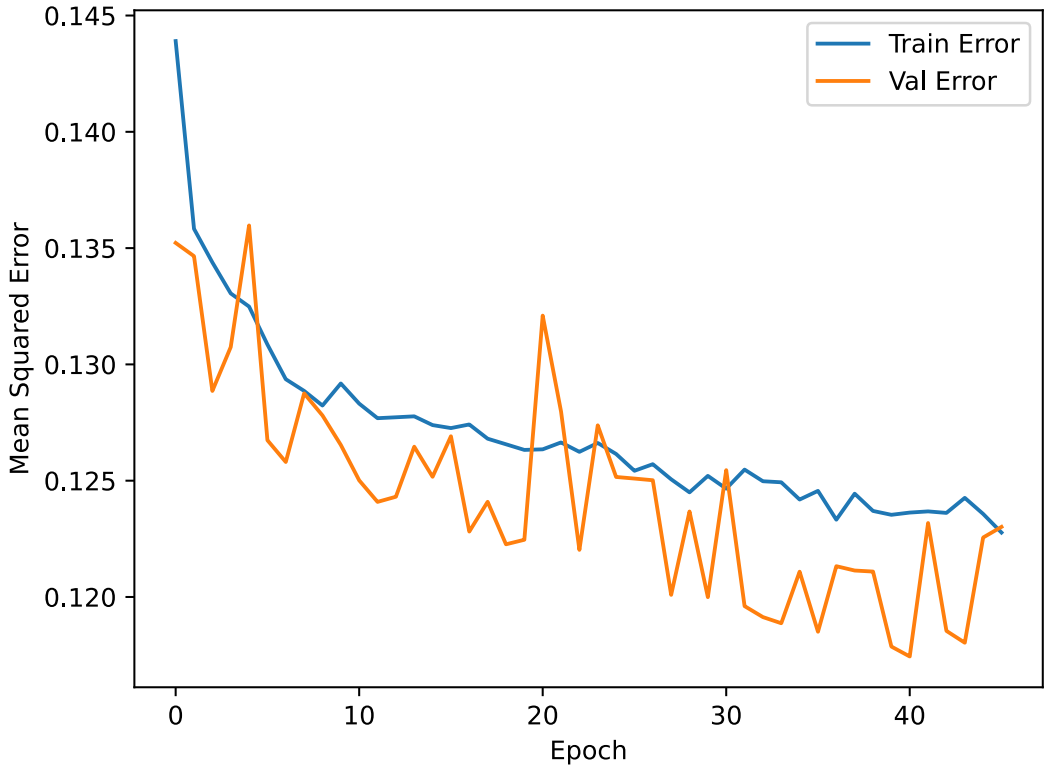
modela konfigurira se njegovo treniranje. Korišteni argumenti metode su *optimizer, loss* i *metrics*. Kao optimizatori korišteni su stohastički gradijentni spust (eng. *stohastic gradient descent - SGD*) i Adam optimizator. Funkcija gubitka i mjera po kojoj se evaluira modele je srednja kvadratna pogreška (eng. *mean squared error*).

Metoda *fit*, kojom se trenira neuronska mreža, prima argumente (navedeni su samo oni koji se koriste):

- **x** - ulazni parametri,
- **y** - ciljna varijabla; vrijednosti korelacijskog koeficijenta za dane ulazne parametre,
- **batch size** - korišteni broj uzoraka po iteraciji treniranja; u istraživanju je podešen na 32 uzorka po iteraciji,
- **epochs** - broj iteracija treniranja modela; koristi se 100 epoha,
- **validation split** - postotak podataka koji se koristi u validaciji tijekom treniranja; 10% podataka za treniranje je korišteno u svrhu validacije,
- **callbacks** - lista implementacija klase `keras.callbacks.Callback` koji se primjenjuju tijekom treniranja; iskorišteni su za rano stopiranje treniranja modela u slu-

čaju da se vrijednost funkcije gubitka ne poboljšava 5 epoha za redom. To je standardna metoda regularizacije u strojnom učenju kojom se izbjegava prenaučenost kod korištenja iterativne metode učenja.

Metoda za treniranje vraća `History` objekt koji sadrži zapise o vrijednostima gubitka i metrika tijekom treniranja. On daje bolji uvid u proces treniranja i što je sve pošlo po krivu. Primjer grafa promjene srednje kvadratne pogreške po epohi treniranja za neuronsku mrežu koja u srednjem sloju koristi sigmoidalnu aktivacijsku funkciju dan je slikom 4.7.



Slika 4.7. Promjena srednje kvadratne pogreške po epohama treniranja neuronske mreže.

Treći korak procjene latentnih faktora je kreiranje procjene kovarijacijske matrice. U ovom koraku se također koristi objedinjeni klizeći prozor, ali nad razdobljem određenim za testiranje. Procjena se radi u svakom pomaku prozora idućim redoslijedom:

1. ulazni parametri se predaju modelu koji vraća predikciju korelacijskog koeficijenta,

2. formulom 2.19 se izračuna vektor procjena faktorskih β (koeficijenti latentnih faktora; tržišne β)

3. izračuna se procjena idiosinkratske komponente faktorskog modela formulom 2.12

4. matričnim umnoškom vektora faktorskih β kreira se matrica koeficijenata koja se zatim pomnoži varijancom indeksa tržišta radi ispravnog skaliranja vrijednosti njenih elemenata,

5. prema izrazu 2.11 kreira se procijenjena kovarijacijska matrica.

4.1. Procjena pozitivno definitne kovarijacijske matrice

Unatoč računanju faktorskih β nad standardiziranim podacima i prelasku na procjenu korelacijskog koeficijenta, neki od regresijskih modela su i dalje imali problem da su za procjenu davali singularnu kovarijacijsku matricu (matrica koja nema inverz i nije pozitivno definitna). Razlog tomu su bili negativni elementi na dijagonali procijenjene idiosinkratske komponente. Kako bi se izbjegli negativni elementi, u izračunu se, umjesto predviđenih, koriste povijesne β vremenskog prozora od sto osamdeset dana. Po tome se razlikuje procjena kovarijacijske matrice regresijskih modela od ostalih. Koraci implementacije su:

1. dohvate se predikcije modela,

2. kreira se matrica koeficijenata faktora iz procijenjenih β,

3. kreira se matrica koeficijenata faktora iz povijesnih β,

4. izračuna se uzoračka kovarijacijska matrica S,

5. prema formuli 2.12 se dobije vrijednost idiosinkratske komponente,

6. iskoristi se formula 2.11 kako bi se dobila procjena kovarijacijske matrice.

4.2. Mjere performanse

Kako bi se mogla dati ocjena koliko je dobra dobivena procjena međuodnosa financijske imovine koriste se dvije mjere performanse: log-izglednost i Frobeniusova norma reziduala procijenjene i uzoračke kovarijacijske matrice.

Neka je skup D slučajan uzorak uzorkovan iz skupa svih mogućih primjera X. Vrijedi pretpostavka da se primjeri $x \in X$ ravnaju po nekoj distribuciji

$$x^{(i)} \sim P(x|\theta) \tag{4.1}$$

gdje je θ vektor parametara te distribucije [19]. Budući da je D slučajan uzorak iz X, on je ujedno i reprezentativan uzorak te se onda smije pretpostaviti i da se primjeri $x^{(i)} \in D$ također ravnaju po istoj distribuciji kao i primjeri iz populacije X [19]. Vjerojatnost da je iz dane distribucije izvučen baš uzorak D je

$$p(D|\theta) = p(x^{(1)}, \dots, x^{(N)}|\theta) = \prod_{i=1}^{N} p(x^{(i)}|\theta)\,. \tag{4.2}$$

Izraz vrijedi samo u slučaju ako su primjeri nezavisni i identično distribuirani (i.i.d.), zato što onda vrijedi $p(x^{(i)}, x^{(j)}) = p(x^{(i)})p(x^{(j)})$ te se vjerojatnost slučajnog vektora može napisati kao produkt vjerojatnosti pojedinačnih slučajnih varijabli [19]. U stvarnosti, vjerojatnost $p(D|\theta)$ ovisi samo o parametru θ zato što je uzorak D fiksiran - to je dobiveni skup podataka. Tu činjenicu se može naglasiti izmjenom zapisa formule 4.2:

$$p(D|\theta) = p(x^{(1)}, \dots, x^{(N)}|\theta) = \prod_{i=1}^{N} p(x^{(i)}|\theta) \equiv L(\theta|D)\,. \tag{4.3}$$

Funkcija L zove se funkcija izglednosti (engl. *likelihood function*). Ona parametrima θ pridjeljuje vjerojatnost da se iz populacije s parametrima θ izvuče uzorak D [19]:

$$L : \theta \to p(D|\theta)\,. \tag{4.4}$$

U radu se koristi prirodni logaritam funkcije izglednosti koji produkt vjerojatnosti pre-

tvara u sumu:

$$\ln L(\theta|D) = \ln \prod_{i=1}^{N} p(x^{(i)}|\theta) = \sum_{i=1}^{N} \ln p(x^{(i)}|\theta)\,. \tag{4.5}$$

To je poželjno iz dva razloga:

1. asimptotička svojstva sume su lakša za analizirati [20],

2. produkti nisu numerički stabilni, imaju tendenciju brzo konvergirati u nulu ili beskonačnost ovisno o tome jesu li vrijednosti pojedinog uzorka veće ili manje od 1. Sume su dakle stabilnije iz numeričke perspektive što je korisno ako se proračuni izvode na računalu koje ima ograničenu preciznost i ne može razlikovati jako mali broj od nule, odnosno jako veliki broj od beskonačnosti [20].

Distribucija koja se koristi za izračun log-izglednosti je multivarijatna normalna (Gaussova) distribucija čija je funkcija log-izglednosti:

$$\ln L(\mu, \hat{\Sigma}; D) = -\frac{mN}{2} \ln(2\pi) - \frac{m}{2} \ln det(\hat{\Sigma}) - \frac{1}{2} \sum_{j=1}^{m} (x_j - \mu)' \hat{\Sigma}^{-1} (x_j - \mu)\,. \tag{4.6}$$

Njeni su parametri srednja vrijednost i kovarijacijska matrica. U implementaciji se srednja vrijednost dobije iz podataka izvan uzorka, a matrica je procijenjena nekim od modela. Bitno je primijetiti da je funkcija izglednosti dobro definirana samo ako je determinanta kovarijacijske matrice strogo pozitivna što implicira da je Σ ograničena na skup pozitivno definitnih matrica [21].

Jedna od najstarijih i najjednostavnijih matričnih normi je Frobeniusova norma. Definira se kao kvadratni korijen zbroja svih kvadriranih elemenata matrice [22]:

$$\left\| A \right\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}\,. \tag{4.7}$$

Ova norma na neki način izriče "veličinu" matrice - matrica s malim/velikim vrijednostima elemenata ima mali/veliki iznos Frobeniusove norme [22]. U radu se koristi kako

bi se izmjerilo koliko je procijenjena kovarijacijska matrica različita od uzoračke (4.8):

$$\left\| \hat{\Sigma} - S \right\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{\sigma}_{ij} - s_{ij})^2} \,. \tag{4.8}$$

Što je procjena bolja, vrijednosti reziduala su bliže nuli te je iznos norme manji.

5. Rezultati

U standardnoj Markowitzevoj teoriji portfelja, portfelj minimalne varijance ima najmanji rizik od svih mogućih izbora ulaganja u imovinu[23]. Izračun vektora optimalnih težina w_{MV}, koje predstavlja udio pojedine imovine u portfelju, ovisi jedino o međuodnosu (kovarijacijskoj matrici), a ne i o očekivanim povratima imovine [23]. Radi se o optimizacijskom problemu u kojem je cilj minimizirati varijancu portfelja,

$$\sigma^2_{mvp} = w'_{mvp}\Sigma w_{mvp} \tag{5.1}$$

i koji ima ograničenje da suma svih težina mora biti jedan ($w'_{mvp}\mathbf{1} = 1$, gdje je $\mathbf{1}$ $N \times 1$ vektor jedinica) [23]. Rješenje optimizacijskog problema može se matrično izračunati:

$$w_{mvp} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}'\Sigma^{-1}\mathbf{1}} \tag{5.2}$$

gdje je Σ^{-1} inverz kovarijacijske matrice. U istraživanju se koristeći procijenjene kovarijacijske matrice, u svakom pomaku klizećeg prozora, kreira i portfelj minimalne varijance.

Modeli strojnog učenja se uspoređuju s obzirom na to koliko dobro mogu iskoristiti informacije iz prošlosti u predikciji korelacijskog koeficijenta i izračunu latentnih faktora. Faktori se zatim koriste za procjenu buduće kovarijance financijske imovine i izračun mjera performansi. Uz to je isprobano slaganje portfelja minimalne varijance te su modeli uspoređeni i na temelju standardne devijacije njihovih povrata.

U istraživanju se u procjeni koristi deset različitih modela nad 3 različita skupa podataka. U tablici 5.1. u stupcu "model" popisane su sve korištene kombinacije modela i podataka (u slučaju modela neuronskih mreža i sve kombinacije s optimizatorima). $Beta_i$ predstavlja osnovne modele koji faktorske bete procjenjuju njihovim povijesnim vrijednostima (odjeljak 3.1.), a broj u indeksu im označuje duljinu vremenskog prozora

model	L	$\left\|\hat{\Sigma} - S\right\|_F$	$\sigma_p(\%)$
$Beta_7$	86.55	0.0904	41.47
$Beta_{30}$	1011.17	0.0578	33.59
$Beta_{90}$	1101.39	0.04217	33.61
$Beta_{180}$	1117.39	**0.0319***	32.19
OLS-beta	1015.78	0.0491	32.92
OLS	1014.34	0.0564	35.29
OLS-stand	1015.00	0.0557	35.70
PolyReg-beta	1013.95	0.0500	33.72
PolyReg	1012.30	0.0626	33.62
PolyReg-stand	1005.53	0.0662	37.75
RFR-beta	1012.70	0.0527	34.12
RFR	1011.13	0.0569	34.42
RFR-stand	1011.28	0.0582	34.80
ReLU-beta-SGD	1131.14	0.0506	**30.14***
ReLU-beta-Adam	**1131.89***	0.0502	30.23
ReLU-stand-SGD	1075.27	0.0579	42.13
ReLU-stand-Adam	1097.77	0.0523	38.32
SIGM-beta-SGD	1130.43	0.0541	30.97
SIGM-beta-Adam	1128.73	0.0527	30.41
SIGM-SGD	1101.63	0.0698	35.85
SIGM-Adam	1121.02	0.0599	31.75
SIGM-stand-SGD	1123.57	0.0559	32.33
SIGM-stand-Adam	1111.32	0.0556	33.68

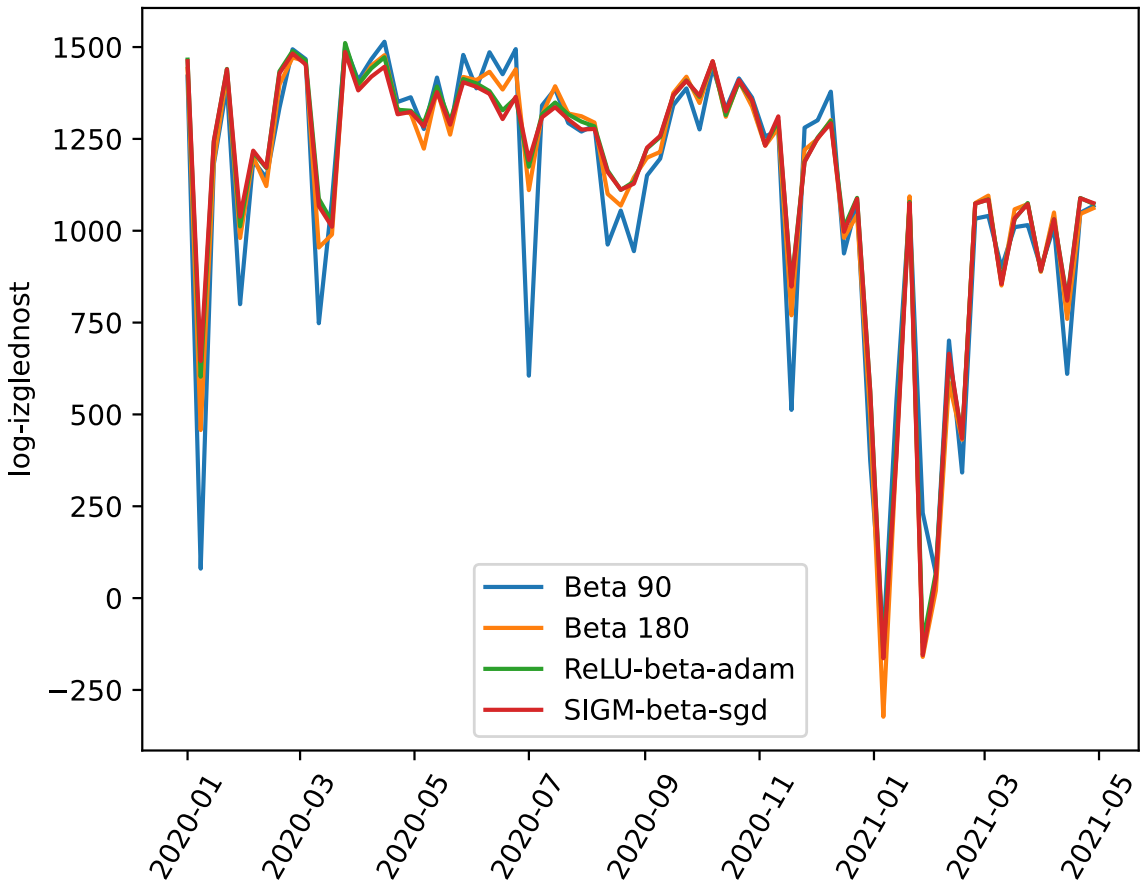
Tablica 5.1. Rezultati mjera po kojima se uspoređuju procjene latentnih faktora. U stupcima su redom s lijeva na desno: nazivi modela u kombinaciji s korištenim skupom podataka (i u slučaju neuronskim mreža optimizatorom), log-izglednost procjene kovarijacijske matrice izračunate iz procjene faktora, Frobeniusova norma razlike procijenjene i uzoračke kovarijacijske matrice, anualizirana standardna devijacija povrata portfelja minimalne varijance izražena u postocima.

nad kojim su izračunate. Kod ostalih naziva prva riječ je ime modela strojnog učenja koji se koristio. Iznimka su modeli neuronskih mreža koji na tom mjestu imaju naziv aktivacijske funkcije koja se koristila u skrivenom sloju. Nastavci u imenima imaju iduća značenja:

- **beta** Model je treniran nad skupom podataka koji sadržava samo vrijednosti povijesnih faktorskih β.
- **stand** Za treniranje je korišten skup podataka sa svim ulaznim parametrima koji su pri dohvaćanju standardizirani.
- **ništa** Ako u nazivu nema nastavka vezanog uz skup podataka znači da je model

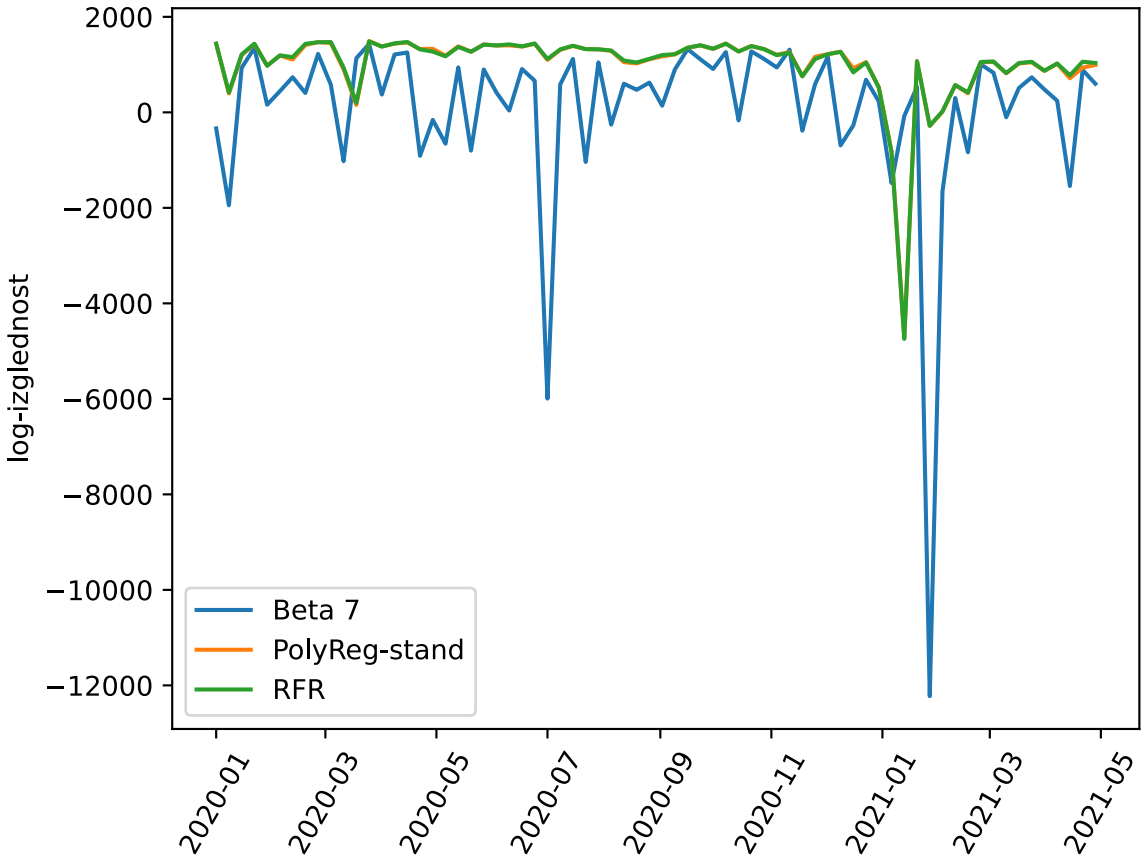
treniran nad svim ulaznim parametrima koji su pri dohvaćanju prošli drugi tip transformacija opisan u poglavlju 4.

- **SGD** U treniranju neuronske mreže korišten je SGD optimizator.
- **Adam** U treniranju neuronske mreže korišten je Adam optimizator.



Slika 5.1. Log-izglednost na razdoblju za testiranje.

Mjerama performanse (odjeljak 4.2.) opisana je kvaliteta procjene kovarijacijske matrice. U tablici 5.1. u stupcu "L" dana je prosječna log-izglednost procjene (prosječna zato što se pri testiranju koristio klizeći prozor) kovarijacijske matrice modela nad podacima za testiranje. Magnituda ove mjere nema neko posebno značenje zato što je podložna različitim faktorima, a najviše broju uzoraka koji se koristio u izračunu. Svejedno, kod uspoređivanja se smije reći da je model ostvario bolji ili gori rezultat u odnosu na neki drugi. Najveću prosječnu log-izglednost, iznosa 1131.89, je ostvario model ReLU-beta-Adam. Svi modeli neuronskih mreža su ostvarili jako dobre rezultate te iako je najbolji rezultat ostvarila mreža s ReLU aktivacijskom funkcijom, u prosjeku su mreže sa sigmo-

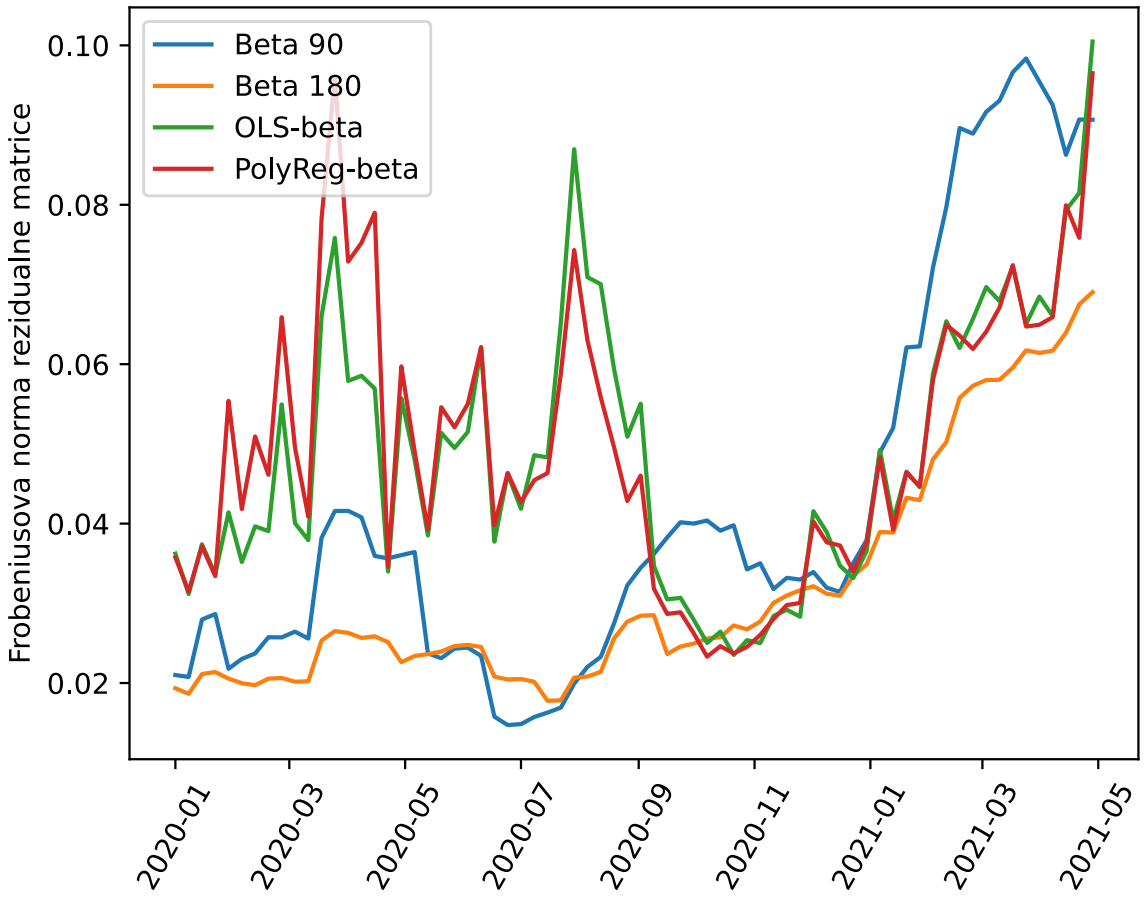


Slika 5.2. Modeli s najgorom prosječnom log-izglednošću.

idalnom aktivacijskom funkcijom u skrivenom sloju postigle bolje rezultate. Dodatno, moguće je primijetiti da u tablici 5.1. nema zapisa o neuronskim mrežama koje koriste ReLU i nemaju oznaku o podacima (stavka "ništa" u gornjem popisu). To je zato što njihovo treniranje nije bilo uspješno i nisu mogle dati procjenu kovarijacijske matrice koja nije singularna. Još su se dva modela istaknula visokim vrijednostima log-izglednosti, a to su povijesne faktorske β izračunate nad prozorima od 90 i 180 dana. Tržišne povijesne β su se općenito pokazale kao dobar izvor informacija svim modelima u predikciji. U prilog tomu ide činjenica da je svaki tip modela ostvario najvišu vrijednost log-izglednosti kada je bio treniran nad skupom podataka koji je uključivao samo njih.

Na slici 5.1. prikazano je kretanje vrijednosti log-izglednosti tijekom razdoblja za testiranje nad sto kriptovaluta. Slika uključuje modele koji imaju visoki prosječni iznos log-izglednosti. S druge strane, slikom 5.2. prikazano je kretanje log-izglednosti modela s njenim najgorim prosjekom.

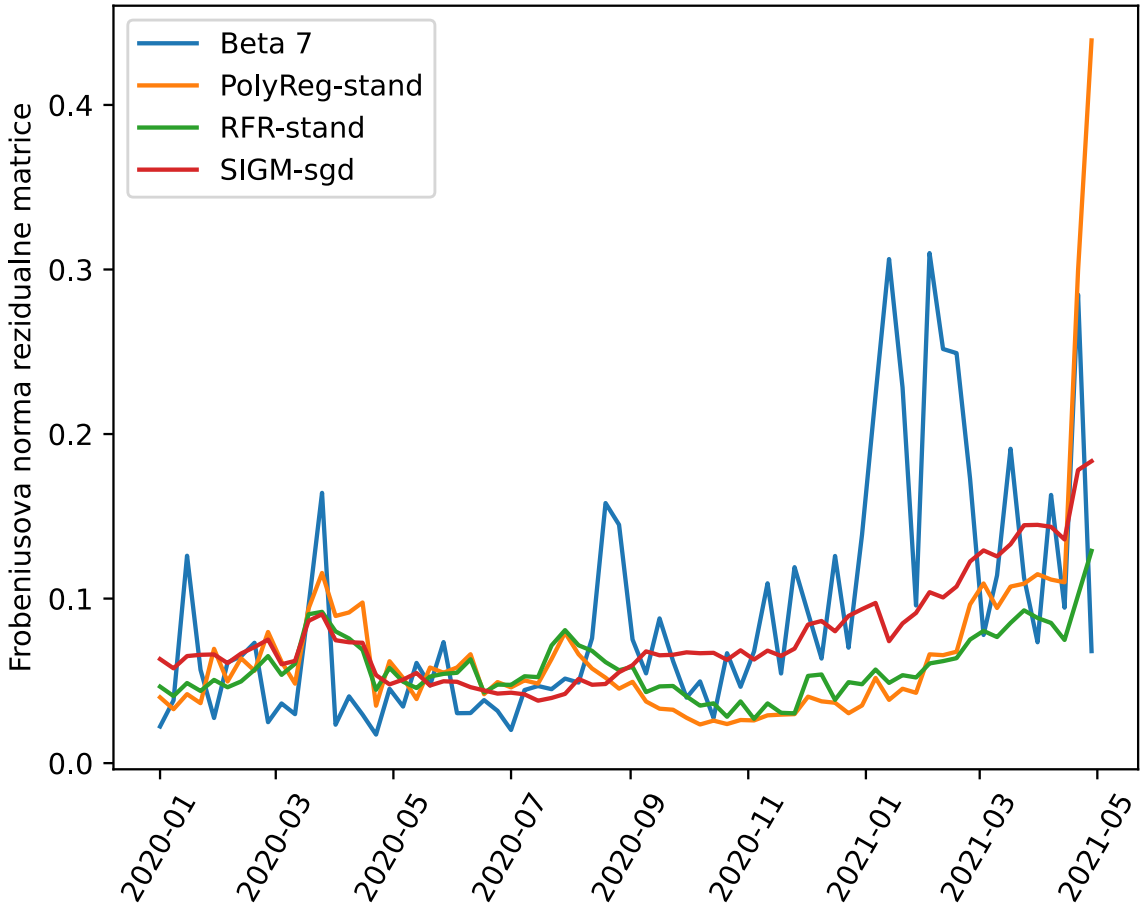
Prosječna Frobeniusova norma razlike procijenjene i uzoračke kovarijacijske matrice



Slika 5.3. Modeli s najboljom prosječnom vrijednosti Frobeniusove norme rezidualne matrice u testnom razdoblju.

dana je stupcem "$\left\|\Sigma - S\right\|_F$" tablice 5.1. Optimalno je da vrijednost ove mjere bude što manja zato što to znači da su pojedini elementi procijenjene matrice, po vrijednostima, bliži stvarnima. Najmanju prosječnu vrijednost rezidualnih matrica ostvario je model $Beta_{180}$. Zanimljivo je što je najgori rezultat ostvario model $Beta_7$. Među ostalima ističu se još linearna regresija trenirana skupu faktorskih β i $Beta_{90}$ zato što su jedini modeli čije su rezidualne matrice ostvarile prosječnu vrijednost norme manju od 0.05, a slijedi ih polinomijalni regresijski model PolyReg-beta. Iznenađenje su rezultati neuronskih mreža koje se, iako imaju najbolje rezultate log-izglednosti, ovdje nisu na isti način istaknule. Vrijednosti Frobeniusove norme razlike procijenjene i uzoračke kovarijacijske matrice za najbolje i najgore modele prikazane su slikama 5.3. i 5.4.

Kako su odluke regresora slučajnih šuma interpretabilne, njegova implementacija u knjižnici Scikit Learn sadrži atribut u kojem je zapisana važnost svakog ulaznog parametra pri donošenju odluka u stablima. Važnost je izražena u postocima i za regresor



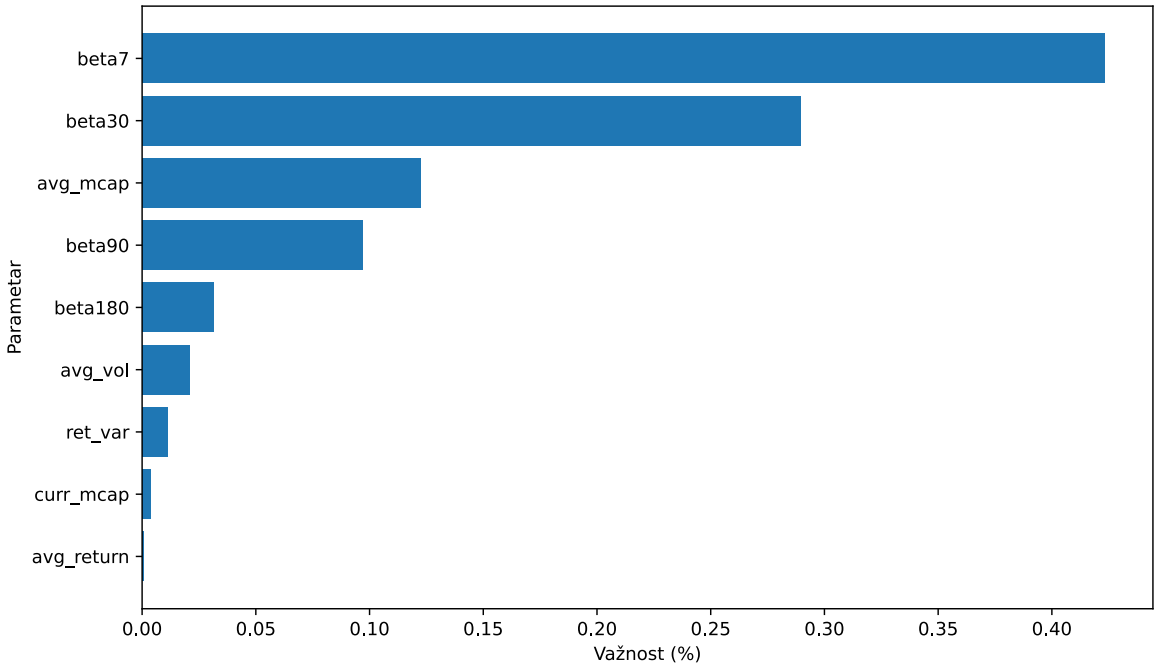
Slika 5.4. Modeli s najgorom prosječnom vrijednosti Frobeniusove norme rezidualne matrice u testnom razdoblju.

treniran na svim parametrima (skupu koji nije standardiziran nego ima drugi tip transformacija) prikazana je na slici 5.5. Rezultati na slici potvrđuju status tržišnih β kao najkorisnijih parametara. Na prvom mjestu su sedmodnevne faktorske bete koje se, kao zaseban model, zapravo ističu jako lošim rezultatima. To je potencijalan razlog zašto i regresor slučajnih šuma nema dobre rezultate.

Posljednjim stupcem "σ_p" u tablici 5.1. popisane su vrijednosti anualizirane standardne devijacije povrata portfelja minimalne varijance izražene u postocima. Portfelji su izgrađeni na temelju procjene kovarijacijske matrice odgovarajućih modela metodama opisanim na početku poglavlja. Optimalno je da standardna devijacija povrata bude što manja zato što to znači i da su fluktuacije (raspršenost) povrata manje. Posljedično je i rizik ulaganja u takav portfelj manji.

Najmanji rizik ulaganja, prema tablici 5.1., ima portfelj nastao iz procjene modela

ReLU-beta-SGD, a slijedi ga isti model s Adam optimizatorom. Ova dva modela imaju i najveću log-izglednost procjene. Poveznica visokog iznosa log-izglednosti i malene varijabilnosti portfelja minimalne varijance ima smisla zato što su ti portfelji napravljeni s kovarijacijskim matricama koje najbolje prikazuju odnose povrata kriptovaluta.



Slika 5.5. Važnost ulaznih parametara kod odlučivanja RFR modela.

6. Zaključak

U prethodnim poglavljima prezentirano je istraživanje kojim se ispituje mogu li modeli strojnog učenja procijeniti buduće latentne faktore u financijskim vremenskim nizovima. Uz to je provedena i usporedba koji od isprobanih modela daje najbolju procjenu (5.). S faktorima se može izračunati procjena kovarijacijske matrice kojom se opisuje međuodnos financijske imovine i koja se može iskoristiti za modeliranje portfelja minimalne varijance.

Procjena kovarijance u nekim slučajevima rezultirala je singularnom kovarijacijskom matricom. Iz tog razloga faktori se ne procjenjuju izravno, već se standardizacijom podataka vremenskih nizova provodi procjena korelacijskih koeficijenata između imovine i tržišta iz kojih se računaju koeficijenti latentnih faktora. Konačna implementacija se bazira na podacima kriptovaluta te programskom jeziku Python i njegovim knjižnicama.

Rezultati istraživanja pokazuju da najizglednije procjene kovarijacijske matrice i najmanje volatilni portfelji minimalne varijance dolaze iz modela neuronskih mreža. U njihovim izgradnjama korištene su aktivacijske funkcije ReLU i sigmoida te optimizatori SGD i Adam. Najkorisnijim ulaznim parametrima pokazale su se povijesne tržišne β od kojih su napravljeni osnovni modeli procjene. Među njima se istaknuo model kojeg čine β izračunate na vremenskom prozoru od 180 dana i koji je ostvario najmanju vrijednost Frobeniusove norme razlike procijenjene i uzoračke kovarijacijske matrice.

Literatura

[1] R. Walpole, R. Myers, S. Myers, i K. Ye, *Probability and Statistics for Engineers and Scientists*. Boston: Pearson Education, Inc., 2007.

[2] Taboga, Marco (2021), "Random vectors", Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix. https://www.statlect.com/fundamentals-of-probability/random-vectors, [datum nastanka: (2021); datum pristupa: svibanj 2024.].

[3] ——, "Covariance matrix", Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix. https://www.statlect.com/fundamentals-of-probability/covariance-matrix, [datum nastanka: (2021); datum pristupa: svibanj 2024.].

[4] Rama Cont, "Empirical properties of asset returns: stylized facts and statistical issues", http://rama.cont.perso.math.cnrs.fr/pdf/empirical.pdf, [datum nastanka: 28. listopad 2000.; datum pristupa: lipanj 2024.].

[5] Drobetz, Wolfgang and Hollstein, Fabian and Otto, Tizian and Prokopczuk, Marcel, "Estimating stock market betas via machine learning", Journal of Financial and Quantitative Analysis, forthcoming, Available at SSRN: https://ssrn.com/abstract=3933048, [datum nastanka: 29. rujan 2021.; datum pristupa: travanj 2024.].

[6] R. Johnson i D. Wichern, *Applied Multivariate Statistical Analysis*. Harlow: Pearson Education Limited, 2014.

[7] Will Kenton, "What beta means for investors", https://www.investopedia.com/terms/b/beta.asp, [datum nastanka: svibanj 2024.; datum pristupa: svibanj 2024.].

[8] Geek3 (https://commons.wikimedia.org/wiki/User:Geek3), "File:hyperbolic tangent.svg", https://commons.wikimedia.org/wiki/File:Hyperbolic_Tangent.svg, [datum nastanka: 5. rujan 2009.;datum pristupa: lipanj 2024.].

[9] Jan Šnajder and Bojana Dalbelo Bašić, "Strojno učenje", https://www.fer.unizg.hr/_download/repository/StrojnoUcenje[2].pdf, [datum nastanka: 2014; datum pristupa: lipanj 2024.].

[10] AnalytixLabs, "Random forest regression — how it helps in predictive analytics?" https://medium.com/@byanalytixlabs/random-forest-regression-how-it-helps-in-predictive-analytics-01c31897c1d4, [datum nastanka: listopad 2023.; datum pristupa: svibanj 2024.].

[11] Scikit Learn, "Decision trees", https://scikit-learn.org/stable/modules/tree.html#tree, [datum pristupa: lipanj 2024.].

[12] Cristian Leo, "The math behind the adam optimizer", https://towardsdatascience.com/the-math-behind-adam-optimizer-c41407efe59b, [datum nastanka: 30. siječanj 2024.;datum pristupa: lipanj 2024.].

[13] A. Uddin, X. Tao, C.-C. Chou, i D. Yu, "Methodology", u *Nonlinear Tensor Completion Using Domain Knowledge: An Application in Analysts' Earnings Forecast*, 11 2020., str. 377–384. https://doi.org/10.1109/ICDMW51313.2020.00059

[14] Statistical Modeling and Forecasting, "The pooled ols regression model for panel data sets", https://timeseriesreasoning.com/contents/pooled-ols-regression-models-for-panel-data-sets/, [datum pristupa: lipanj 2024.].

[15] Jason Fernando, "Market capitalization: What it means for investors", https://www.investopedia.com/terms/m/marketcapitalization.asp, [datum nastanka: ožujak 2024.; datum pristupa: svibanj 2024.].

[16] Adam Hayes, "What is volume of a stock, and why does it matter to investors?" https://www.investopedia.com/terms/v/volume.asp, [datum nastanka: svibanj 2023.; datum pristupa: svibanj 2024.].

[17] TensorFlow, “tf.keras.layers.dense”, https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense, [datum posljednjeg osvježavanja: 7. lipanj 2024.; datum pristupa: lipanj 2024.].

[18] Sagar Sharma, “Activation functions in neural networks: Sigmoid, tanh, softmax, relu, leaky relu explained !!!” https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6, [datum nastanka: 6. rujan 2017.;datum pristupa: lipanj 2024.].

[19] Jan Šnajder, “14. procjena parametara ii”, https://www.fer.unizg.hr/_download/repository/SU1-2022-P14-ProcjenaParametara2.pdf, [datum nastanka: 2022; datum pristupa: svibanj 2024.].

[20] Taboga, Marco (2021), “Log-likelihood”, Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix. https://www.statlect.com/glossary/log-likelihood, [datum nastanka: (2021); datum pristupa: svibanj 2024.].

[21] ——, “Multivariate normal distribution - maximum likelihood estimation”, Lectures on probability theory and mathematical statistics. Kindle Direct Publishing. Online appendix. https://www.statlect.com/fundamentals-of-statistics/multivariate-normal-distribution-maximum-likelihood, [datum nastanka: ; datum pristupa: svibanj 2024.].

[22] W. Ford, “Chapter 7 - vector and matrix norms”, u *Numerical Linear Algebra with Applications*, W. Ford, Ur. Boston: Academic Press, 2015., str. 119–144. https://doi.org/https://doi.org/10.1016/B978-0-12-394435-1.00007-7

[23] Clarke, R. and De Silva, H. and Thorley, S., “Minimum-variance portfolio composition”, Journal of Portfolio Management, Vol. 37, No. 2, pp. 31-45 (Winter 2011) https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1549949, [datum nastanka: 1. lipanj 2010.; datum pristupa: svibanj 2024.].

Sažetak

Procjena latentnih faktora u financijskim vremenskim nizovima zasnovana na strojnom učenju

Ivan Linardić

Rizik u financijama predstavlja varijabilnost imovine u portfelju. Modelira ga se kovarijacijskom matricom povrata koja uz to pruža informacije o međuodnosu imovine. Poželjno je znati kakva će biti kovarijacijska matrica u budućnosti kako bi se znalo modelirati budući rizik. U financijskim vremenskim nizovima postoje zajednički latentni faktori povrata koji su opisani faktorskim modelom. Procjenom budućih faktora moguće je rekonstruirati buduću kovarijancu imovine što je korisno u upravljanju rizikom i optimizaciji portfelja. Glavna ideja ovog rada je istražiti mogu li različiti modeli strojnog učenja procijeniti buduće latentne faktore iz dostupnih podataka i usporediti njihove rezultate. Najboljim procjeniteljima pokazale su se neuronske mreže i povijesne tržišne β.

Ključne riječi: latentni faktori; kovarijanca; modeli strojnog učenja

Abstract

Estimation of latent factors in financial time series based on machine learning

Ivan Linardić

In the finances risk represents the variability of the assets in a portfolio. It is modeled by the covariance matrix of returns which also gives information about the relation between the assets. It is desirable to know the value of the future covariance matrix so one could know how to model the future risk. There are common latent factors in the financial time series data that are described by the factor model. By the estimation of the future factors it is possible to reconstruct the future covariance matrix of the assets which is useful in the risk management and portfolio optimization. The main idea of this paper is to conduct a research if different machine learning models can estimate future latent factors using the available data and to compare their results. Neural networks and historical market β showed to be the best estimators.

Keywords: latent factors; covariance; machine learning models