

# Estimacija cijene dionica primjenom ansambl metoda strojnog učenja

---

Ančić, Edi

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:491365>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-07-24**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



**SVEUČILIŠTE U RIJECI**  
**TEHNIČKI FAKULTET**

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**ESTIMACIJA CIJENE DIONICA PRIMJENOM ANSAMBL  
METODA STROJNOG UČENJA**

Rijeka, rujan 2023.

Edi Ančić  
0069086096

SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

**ESTIMACIJA CIJENE DIONICA PRIMJENOM ANSAMBL  
METODA STROJNOG UČENJA**

Mentor: prof dr. sc. Zlatan Car

Komentor: v. asist. dr. sc. Nikola Anđelić

Rijeka, rujan 2023.

Edi Ančić  
0069086096

Rijeka, 14. srpnja 2023.

Zavod: **Zavod za automatiku i elektroniku**  
Predmet: **Primjena umjetne inteligencije**  
Grana: **2.03.06 automatizacija i robotika**

## ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Edi Ančić (0069086096)**  
Studij: Sveučilišni diplomski studij elektrotehnike  
Modul: **Automatika**

Zadatak: **Estimacija cijene dionica primjenom ansambl metoda strojnog učenja**

### Opis zadatka:

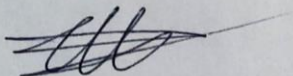
Napraviti pregled postojećih istraživanja u kojima je korištena umjetna inteligencija za estimaciju cijene dionica. Na javno dostupnom skupu podataka napraviti statističku analizu i predobradu podataka. Ispitati koje osnovne metode strojnog učenja daju najbolju estimaciju cijene dionica. Ispitati utjecaj različitih metoda skaliranja i normalizacije na točnost estimacije cijene dionica. Pri treniranju metoda strojnog učenja koristiti unakrsnu validaciju i metodu odabira nasumičnih hiperparametara. Kombinirati najbolje metode strojnog učenja u ansambl te ispitati da li se točnost estimacije poboljšala u usporedbi s rezultatima koji su postignuti sa osnovnim metodama strojnog učenja.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

*Edi Ančić*

Zadatak uručen pristupniku: 14. srpnja 2023.

Mentor:

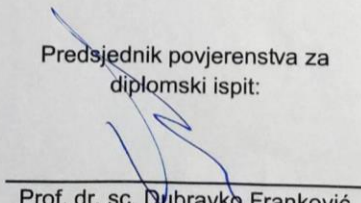


Prof. dr. sc. Zlatan Car

*Nikola Anđelić*

Dr. sc. Nikola Anđelić (komentor)

Predsjednik povjerenstva za  
diplomski ispit:



Prof. dr. sc. Dubravko Franković

Rijeka, rujan 2023.

### **IZJAVA**

Ja, Edi Ančić, rođen 03.08.1999. u Rijeci, potvrđujem da sam rad pod naslovom „Estimacija cijene dionica primjenom ansambl metoda strojnog učenja“ osobno napisao te da sam njegov jedini autor. Rad sam izradio samostalno pod mentorstvom prof dr. sc. Zlatana Cara i komentora dr. sc. Nikole Anđelića. Koristio sam svoje stečeno znanje tijekom obrazovanja. U njegovoj izradi nisam se koristio drugim izvorima do onih koji su navedeni u radu.

---

Edi Ančić

## ZAHVALA

*Zahvaljujem se obitelji i prijateljima na podršci tijekom studiranja. Zahvaljujem se mentoru prof. dr. sc. Zlatanu Caru i komentoru dr. sc. Nikoli Anđeliću na pomoći i usmjerenju pri izradi diplomskog rada. Ponajviše bi se htio zahvaliti curi koja mi je bila najveća potpora kroz sve godine studiranja.*

## Sadržaj

1. Uvod .....	1
2. Pregled postojećeg istraživanja .....	2
2.1. Vrste istraživanja na području umjetne inteligencije .....	2
2.2. Istraživanje s globalnim podacima .....	3
2.3. Predviđanje vrijednosti dionica u više koraka .....	4
2.4. Hibridni model .....	5
3. Materijali i metode .....	6
3.1. Opis dataset-a .....	6
3.2. Korelacijska analiza.....	9
3.3. Oblikovanje dataseta za primjenu modela strojnog učenja .....	14
3.4. Korišteni modeli .....	15
3.4.1. Linearna regresija .....	15
3.4.2. Višeslojni perceptron.....	16
3.4.3. Regresija najmanjeg kuta .....	16
3.4.4. Operator najmanjeg apsolutnog skupljanja i odabira i Lars Lasso regresija.....	16
3.4.5. Bayesova grebenasta regresija i automatsko određivanje relevantnosti.....	17
3.5. Metode skaliranja i normalizacije .....	17
3.5.1. MaxAbsScaler .....	18
3.5.2. MinMaxScaler .....	18
3.5.3. Normalizer.....	18
3.5.4. RobustScaler.....	18
3.5.5. StandardScaler.....	19
3.5.6. PowerTransformer.....	19
3.6. Evaluacijske metode.....	19
3.6.1. Koeficijent determinacije .....	19
3.6.2. Srednja apsolutna pogreška .....	20
3.6.3. Korijen srednje kvadratne pogreške .....	20

4. Rezultati primjenjenih modela .....	22
4.1. AMD dataset.....	22
4.2. Apple dataset .....	24
4.3. HPQ dataset.....	26
4.4. IBM dataset .....	28
4.5. Intel dataset.....	30
5. Optimiziranje modela nasumičnim odabirom hiperparametara .....	32
5.1. Hiperparametri modela.....	32
5.1.1. Hiperparametri linearnog modela.....	32
5.1.2. Hiperparametri MLP modela.....	33
5.1.3. Hiperparametri LARS modela.....	33
5.1.4. Hiperparametri Lasso modela.....	34
5.1.5. Hiperparametri LassoLars modela .....	34
5.1.6. Hiperparametri ARD regresije .....	35
5.1.7. Hiperparametri BayesianRidge regresije.....	35
5.2. Rezultati modela s optimalnim hiperparametrima .....	35
6. Unakrsna validacija .....	43
6.1. Rezultati unakrsne validacije.....	43
7. Ansambl metode.....	51
7.1. Rezultati StackingRegressor ansambl metode.....	53
7.2. Rezultati VotingRegressor ansambl metode .....	57
8. Zaključak .....	62
9. Literatura .....	64
10. Sažetak.....	67
11. Summary .....	68
Dodatak A. Tablice sa hiperparametrima modela .....	69
A.1. AMD dataset.....	69
A.2. AAPL dataset.....	78



A.3 HPQ dataset .....	87
A.4. IBM.....	96
A.5. INTC .....	105
Dodatak B. Primjer programskog koda. ....	115
B.1. Primjer MLP modela za AMD dataset .....	115
B.2. Primjer MLP modela s nasumičnim traženjem hiperparametara pri power skaliranju na AMD datasetu .....	116
B.3. Primjer MLP modela s unakrsnom validacijom pri power skaliranju na AMD datasetu	119
B.4. Primjer StackingRegressor ansambla na AMD datasetu .....	123
B.5. Primjer VotingRegressor ansambla na AMD datasetu .....	130

# 1. Uvod

Trgovina dionicama služi kao prilika za investiranje novca u razne tvrtke kako bi se ostvarila zarada. Kako bi se bolje ogradili od gubitaka investitori mogu uložiti novce u razne fondove koji prate tržišne indekse. Tržišni indeks je indikator koji se temelji na odabranom portfelju dionica. Indeksi mogu pratiti velik dio tržišta ili neki određeni tržišni sektor [1].

S obzirom na to razni investitori analiziraju tržište dionica kako bi mogli zaključiti koje dionice će im dovesti najveći profit. Rastom popularnosti podatkovne znanosti (engl. Data science) i strojnog učenja (engl. Machine learning) za analizu dionica se koriste različiti modeli i metode strojnog učenja.

U sklopu ovog rada izvršit će se analiza 5 dionica iz Standard and Poor's 500 indeksa baziranog na 500 najvećih tvrtki u Sjedinjenim Američkim Državama te će se na njima ispitati sljedeće hipoteze:

- Je li moguće primjeniti različite algoritme strojnog učenja i s njima dobiti visoku točnost estimacije cijene dionica?
- Mijenja li se točnost estimacija cijene dionica ako se na ulazne varijable primjene različite metode skaliranja i normalizacije?
- Je li moguće primjenom metode nasumičnog pretraživanja hiperparametara pronaći optimalnu kombinaciju hiperparametara i na taj način povećati točnost estimacije?
- Je li moguće primjenom metode unakrsne validacije postići visoku točnost estimacije cijene dionica?
- Hoće li se izradom stacking ili voting ansambla povećati točnost estimacije cijena dionica?

U sljedećem poglavlju opisati će se postojeća istraživanja na temu estimacije dionica. U trećem poglavlju opisati će se sam dataset, algoritmi strojnog učenja te metode skaliranja i normalizacije koje će se koristiti u radu. U četvrtom poglavlju opisani su rezultati modela iz trećeg poglavlja prije i nakon skaliranja. U petom poglavlju se opisuje i prikazuje potraga za optimalnim hiperparametrima svakog modela. Zatim, u šestom poglavlju primjenjuje se metoda unakrsne validacije kako bi se dalje poboljšala točnost estimacije i robustnost. U sedmom poglavlju prijašnje modele se kombinira u dvije vrste ansambla čiji je cilj da se dobije novi model koji je točniji od ulaznih modela. U osmom poglavlju je dan zaključak koji služi kao odgovor na prijašnje hipoteze.

## 2. Pregled postojećeg istraživanja

Predviđanje cijena dionica je problem star koliko i burza dionica. Na tu temu su napisani brojni radovi od kojih će se neki razmotriti ovdje. Razmotrit će se generalno razvrstavanje istraživanja na području umjetne inteligencije, korištenje modela na više različitih burza, korištenje višestupanjskih modela i korištenje hibridnih modela.

### 2.1. Vrste istraživanja na području umjetne inteligencije

Jedno istraživanje se fokusiralo na kategorizaciju različitih metoda razvijanja umjetne inteligencije te su razmatrali više istraživačkih radova uz određene kriterije kako bi došli do zaključka. Kriteriji su bili da se u istraživanju koriste jedna ili više metoda strojnog učenja, mora se ciljati predviđanje indeksa, a ne individualnih dionica te istraživanje nije smjelo biti starije od 20 godina.

Prema tom istraživanju svi radovi bi se mogli svrstati u ove četiri kategorije: istraživanje korištenjem Umjetnih neuronskih mreža (engl. Artificial neural network, skraćeno ANN), istraživanja s Metodom potpornih vektora (engl. Support vector machine, skraćeno SVM), istraživanja pomoću genetskih algoritama kombiniranih s drugim tehnikama i istraživanja gdje se koriste hibridni ili ostali pristupi s umjetnom inteligencijom.

ANN-ovi su računalni modeli bazirani na biološkim neuronskim mrežama. U neuronskoj mreži su čvorovi grupirani po slojevima koji počinju od ulaznog sloja te završavaju s izlaznim slojem. Signali prolaze kroz slojeve te se sustav poboljšava tako da uči na svojim greškama. To čine tako da se regulira „težina“ veze koje su spojene između čvorova. Signali koji više utječu na rezultat će imati veću težinu od onih koji ne utječu na rezultat.

Pregledom više istraživanja zaključilo se je da ANN najbolje primijeniti za predviđanje vrijednosti dionica te da duboka neuronska mreža može izvući dodatne informacije iz reziduala autoregresivnog modela koje poboljšavaju točnost.

SVM se koristi za predviđanje dionica putem kategorizacije. U treningu je potrebno primjere podataka podijeliti na skupine, na primjer rast ili pad. U takvoj je kategorizaciji cilj istrenirati model da predvidi hoće li cijena dionica rasti ili padati.

Istraživanje Schumaker i Chen (2009) je kombiniralo SVM zajedno s analizom vijesti vezane uz dionice te su tako uspjeli napraviti precizan model testiran na simulaciji trgovine dionica koji je predviđao rast vrijednosti dionice te njenu cijenu.

Zaključak istraživanja je da je SVM najbolje primijeniti za klasifikacijske probleme kao što su hoće li cijena indeksa pasti ili narasti.

Genetski algoritam je vrsta evolucijskog algoritma. Evolucijski proces počinje tako da se nasumično generiraju rezultati. Rezultati se onda mjere objektivnom funkcijom koja opisuje koji rezultat najviše odgovara zadanim podacima. Rezultati koji najbolje opisuju podatke se zadržavaju i kombiniraju kako bi stvorili novu generaciju rješenja. Ovaj proces se ponavlja dok se ne postigne zadovoljavajuća razina točnosti ili dok se ne dostigne neki određeni broj generacija.

U istraživanjima su se genetski algoritmi koristili u kombinaciji sa SVM-om i ANN-ovima. U rezultatima pokusa su otkrili da genetski algoritmi smanjuju potrebne ulazne podatke optimizacijom parametara.

U kombinaciji genetskog algoritma i SVM-a, genetski algoritam je služio za namještanje utjecaja ulaznih varijabli dok je SVM predviđao dinamike cijena dionica.

Zaključak istraživanja je bio da su genetski algoritmi korisni za zaključivanje koje dionice uključiti u portofolio [1].

## **2.2. Istraživanje s globalnim podacima**

U ovom istraživanju je predloženo korištenje globalnih podataka korištenjem više algoritama kako bi se vidjelo kako koje tržište utječe jedno na drugo. Kao podaci se ne koriste samo cijene dionica već i valute te cijena zlata, srebra i drugih vrijednih metala.

Cilj umjetne inteligencije u ovom radu je predvidjeti da li će vrijednost dionice narasti ili pasti. Podaci su obrađeni tako da se oduzmu dvije dnevne cijene dionica te se vrijednosti normaliziraju kako bi se mogle uspoređivati s drugim tržištima.

Rezultati korelacije predlažu snažnu povezanost između dioničnih indeksa Sjedinjenih Američkih Država i globalnog tržišta pri početku trgovanja na Američkoj burzi.

Za rješenja regresije koristi se SVM. Kao i u prijašnjem istraživanju cilj algoritma je da klasificira pad i rast cijene dionica. Nakon toga se slažu modeli koji će predlagati kako trgovati s dionicama.

Prvi model predlaže kupovinu na otvorenju burze te prodavanje na zatvaranju. Drugi predlaže kupovinu dionica čiji je rast očekivan te njihovu prodaju sljedeći dan. Treći model koristi predviđanja SVM algoritma da se prodaje tek onda kad SVM predvidi pad vrijednosti.

U rezultatima korišteni modeli daju pozitivan povrat novca te bi godišnji povrat uloženog novca bio oko 30%, ali to je sve na simulaciji te nije uračunato kako bi samo trgovanje prema algoritmu utjecalo na tržište [3].

### **2.3. Predviđanje vrijednosti dionica u više koraka**

Kod pristupa s dva stupnja, kod prvog stupnja regresija potpornih vektora (engl. Support vector regression, skraćeno SVR) algoritam priprema ulaze, dok se algoritam za predviđanje koristi u drugom stupnju.

U istraživanju se koristio dataset sa 10 godina podataka koji sadrži dva indeksa. Podaci koji se koriste u algoritmu su izračunati iz close, high, low, i open stupaca.

Kao mjere evaluacije koriste se Apsolutna postotna srednja greška (engl. Mean Absolute Percentage Error skraćeno MAPE), Apsolutna srednja greška (engl. Mean Absolute Error skraćeno MAPE), srednja kvadratna pogreška (engl. Mean Squared Error skraćeno MSE), korijen srednje kvadratne pogreške (engl. Relative Root Mean Square Error skraćeno rRMSE).

Za jednostupanjske modele se uzimaju ANN, SVR i stablo odluke (engl. Random forest, skraćeno RF). Za dvostupanjske algoritme se koristi SVR-ANN, SVR-SVR i SVR-RF te se testiraju na rasponu od 1-10, 15 i 30 dana unaprijed.

U prvom datasetu do 4. dana predviđanja unaprijed najbolje rezultate daje jednostupanjski SVR model. Na 4. danu se poprilično izjednačava sa SVR-SVR modelom. Na 15. danu su sve dvostupanjske metode bolje od jednostupanjskih te su po rRMSE metrici slični. Na 30. danu je SVR-ANN model najviše točan s time da kako se više dana u naprijed predviđa to su rezultati lošiji. U drugom datasetu su rezultati generalno isti.

Ako se gleda prosjek rezultata, najbolja tri modela ispadaju SVR-SVR, SVR-ANN te jednostupanjski SVR [4].

## 2.4. Hibridni model

Metoda najmanjih korijena potpornih vektora (engl. Least Square Support Vector Machine, skraćeno LS-SVM) je nastavak na običan SVM i koriste se za klasifikacijsku i regresijsku analizu.

Algoritam optimizacije roja čestica (engl. Particle Swarm Optimization, skraćeno PSO) je relativno nedavna heuristička metoda pretraživanja bazirana na ponašanju socijalnih grupa kao što su skupine ptica ili riba. Cilj PSO-a je da iterativno optimizira rješenje tako da za svaku iteraciju koju napravi dobiva povratnu informaciju o njenoj kvaliteti. Sličan je evolucijskim algoritmima iako ne eliminira manje točna rješenja.

U istraživanju je predložen model koji koristi PSO za optimizaciju ulaznih podataka te LS-SVM koji predviđa dnevne cijene dionica. Za usporedbu s tim modelom se koristi Levenberg-Marquardt (skraćeno LM) koji služi za rješavanje nelinearnih problema najmanjih korijena i još jedan ANN model.

Ovi modeli su trenirani i testirani na podacima 13 dionica. U svakom slučaju je PSO-LS-SVM model imao najbolje rezultate, nakon njega je najbolje rezultate imao LS-SVM osim u dva slučaja u kojem je ANN imao malo bolje rezultate [5].

### 3. Materijali i metode

Kako bi se pravilno izvršila analiza potrebno je prvo analizirati dataset kako bi se on kasnije mogao pravilno obraditi i skalirati. Nakon što su podaci iz dataseta obrađeni, koristi se više modela estimacije kako bi se skupilo što više rezultata koji će se ocijeniti evaluacijskim metodama.

#### 3.1. Opis dataset-a

Koristi se dataset s Kaggle-a koji sadrži podatke o svim tvrtkama koje se nalaze u Standard and Poor's 500 indeksu [6]. S&P 500 je indeks koji prati 500 najvećih tvrtki na burzi u Sjedinjenim Američkim Državama te služi kao pokazatelj performansi burze dionica i ekonomski indikator za SAD. Dioničari ne investiraju u sam indeks već u razne fondove koje ga prate. Indeks pokriva razne sektore, ali za ovaj rad je odabrano 5 dionica iz IT sektora. Te dionice su: AMD, Apple, HP, Intel i IBM. U datasetu svaka dionica ima redove koji sadrže izmjerene vrijednosti te stupce u kojima su varijable: Date, Open, High, Low, Close, Volume, Dividends, Stock Splits. Date je datum na kojeg su očitane vrijednosti drugih stupaca. Open je cijena dionice pri otvaranju burze. High i low su najveća i najniža cijena neke dionice na taj dan. Close je cijena dionice pri zatvaranju burze. Volume je broj dionica koje su se razmijenile na taj dan. Dividends je broj koji označava koliko je dolara isplaćeno po dionici dioničarima. Stock Splits označava da se jedna dionica rastavila na više njih te joj se proporcionalno smanjila cijena npr. ako je stock split 2 onda će se broj dionica koja neka osoba posjeduje udvostručiti. Cilj ovog rada će biti ugađati Close pomoću ostalih varijabli. Prednost korištenja ovog dataseta je što cijena dionica uglavnom raste te budući da se cijena iz dana u dan tipično ne mijenja previše nema toliko naglih skokova. Kako bi lakše obradili podatke prvo se na njima vrši statistička analiza uz pomoć pandas python knjižnice kao što je prikazano na tablicama: 2.1, 2.2, 2.3, 2.4, 2.5.

Tablica 3.1 Statistička analiza AMD dataseta

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
<b>count</b>	10454							
<b>mean</b>	5226,5	15,3295	15,9045	15,2199	15,5634	17331904	0	0,0005261
<b>std</b>	3017,95	20,711	20,9839	20,088	20,5407	26966746	0	0,0313098
<b>min</b>	0	0	1,69	1,61	1,62	0	0	0
<b>25%</b>	2613,25	5,125	5,5	5,25	5,375	1295850	0	0
<b>50%</b>	5226,5	9,75	9,94375	9,5	9,75	6689100	0	0
<b>75%</b>	7839,75	15,7197	15,99	15,375	15,6856	21247600	0	0
<b>max</b>	10453	163,28	164,46	156,1	161,91	325058400	0	2

Tablica 3.2 Statistička analiza Apple dataseta

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
<b>count</b>	10454							
<b>mean</b>	5226,5	13,954	14,1086	13,8	13,9601	331173835	0,00061	0,001626
<b>std</b>	3017,95	31,2953	31,6633	30,9329	31,3128	338951271	0,01012	0,085812
<b>min</b>	0	0,03876	0,03876	0,03833	0,03833	0	0	0
<b>25%</b>	2613,25	0,23538	0,24044	0,22991	0,23529	123904300	0	0
<b>50%</b>	5226,5	0,39027	0,39743	0,38355	0,39061	218612800	0	0
<b>75%</b>	7839,75	12,5891	12,7402	12,4343	12,5683	411367600	0	0
<b>max</b>	10453	182,13	182,439	178,63	181,512	7421640800	0,23	7



Tablica 3.3 Statistička analiza HP dataseta

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
<b>count</b>	10454							
<b>mean</b>	5226,5	8,35433	8,47348	8,24727	8,36117	20033707	0,00144	0,00117
<b>std</b>	3017,95	7,38916	7,49059	7,30136	7,39589	18350294	0,06983	0,04874
<b>min</b>	0	0,45391	0,46946	0,44769	0,45236	820906	0	0
<b>25%</b>	2613,25	1,4948	1,51646	1,47406	1,49527	9315891,5	0	0
<b>50%</b>	5226,5	6,834	6,91825	6,72208	6,82399	14675890	0	0
<b>75%</b>	7839,75	12,9121	13,0694	12,7416	12,9186	25151960	0	0
<b>max</b>	10453	39,73	41,47	38,99	40,06	442012304	7,0797	2,202

Tablica 3.4 Statistička analiza IBM dataseta

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
<b>count</b>	10454							
<b>mean</b>	5226,5	55,2593	55,7698	54,7754	55,2771	6957078	0,00805	0,00048
<b>std</b>	3017,95	44,1598	44,4861	43,8501	44,1732	4489306	0,08918	0,02949
<b>min</b>	0	4,00789	4,05888	3,9467	3,99769	0	0	0
<b>25%</b>	2613,25	12,2488	12,3412	12,1488	12,2488	4158295	0	0
<b>50%</b>	5226,5	49,1249	49,5985	48,7694	49,1856	5872087	0	0
<b>75%</b>	7839,75	102,98	104,086	101,877	102,958	8424223	0	0
<b>max</b>	10453	143,383	143,729	142,664	143,662	72639156	1,65	2

Tablica 3.5 Statistička analiza INTEL dataseta

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
<b>count</b>	10454							
<b>mean</b>	5226,5	15,0888	15,2871	14,8932	15,0895	51449692	0,00153	0,00129
<b>std</b>	3017,95	15,3335	15,5219	15,1498	15,3359	35030074	0,01917	0,0501
<b>min</b>	0	0,12949	0,12949	0,12795	0,12795	0	0	0
<b>25%</b>	2613,25	0,84283	0,85555	0,82433	0,84225	27682800	0	0
<b>50%</b>	5226,5	13,0872	13,266	12,9251	13,0984	45506950	0	0
<b>75%</b>	7839,75	19,9469	20,2276	19,6809	19,9614	65531750	0	0
<b>max</b>	10453	65,8536	66,1337	64,3376	65,9116	567708800	0,365	2

Iz ovih se tablica vidi broj podataka u svakom stupcu, njihova srednja vrijednost, standardna devijacija, minimum itd. Iz ovih tablica se vidi da je vrijednost dividends i stock splita za svih 5 dionica većinom 0. Također se vidi koliki je broj dana mjerena svaka dionica, budući da Apple-ova dionica ima najmanji broj dana, druge dionice će se skalirati radi lakšeg uspoređivanja podataka. Podaci o srednjoj vrijednosti nam govore kakvu skalu očekivati odnosno koje magnitude će biti podaci. Vrijednosti u redu s 25% označuju da 25% podataka u stupcu ima točno tu vrijednost ili neku nižu od te. Vrijednosti u redu s 75% označuju da 75% podataka u stupcu ima točno tu vrijednost ili neku nižu od te.

### 3.2. Korelacijska analiza

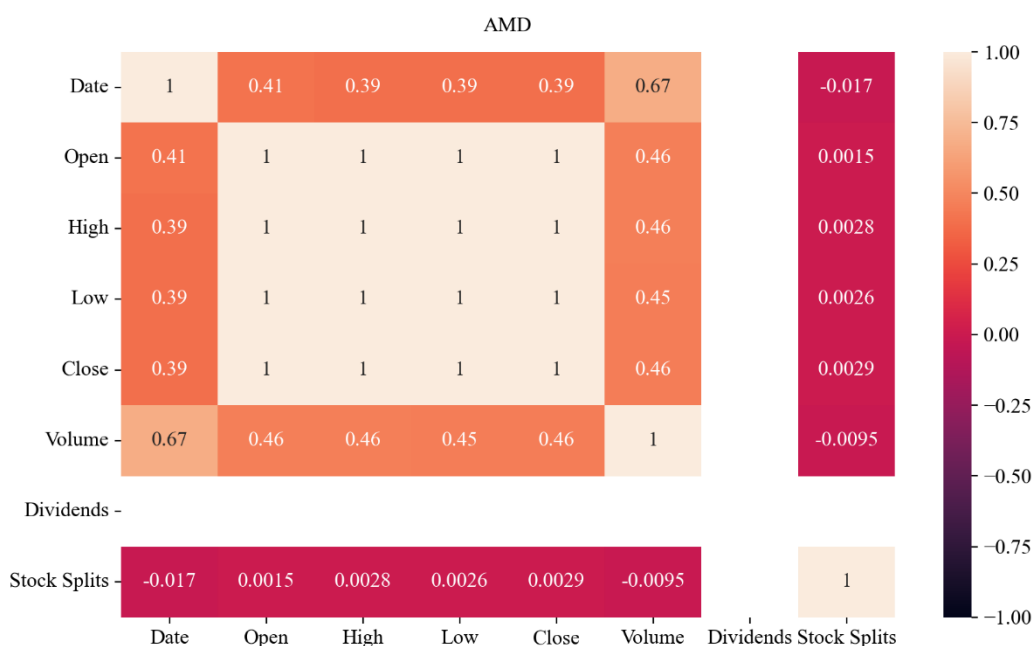
Kako bi se vidjelo koliko će koja varijabla utjecati na rezultat modela strojnog učenja izvršava se korelacijska analiza. Za to će se koristiti Pearsonov korelacijski koeficijent koji mjeri linearnu vezu između varijabli. Za korištenje Pearsonove korelacije bitno je da je dataset normalno distribuiran, bez nedostajućih podataka te varijable moraju biti nezavisne jedna od druge. Mana ove metode korelacije je osjetljivost na odstupajuće podatke i šum te prikazuje samo linearnu korelaciju, ako postoji nelinearna korelacija između varijabli ovom metodom je nije moguće otkriti [12]. Formula za Pearsonov korelacijski koeficijent dana je s :

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \quad (3.1)$$

Gdje su x i y varijable između kojih se traži korelacija[13].

Osim Pearsonove, preko pandas knjižnice u pythonu se mogu koristiti još i Kendall rank koja otkriva nelinearnu korelaciju i Spearman's rank koja također služi za otkrivanje nelinearne korelacije, ali je uz to manje osjetljiva na odstupajuće podatke.

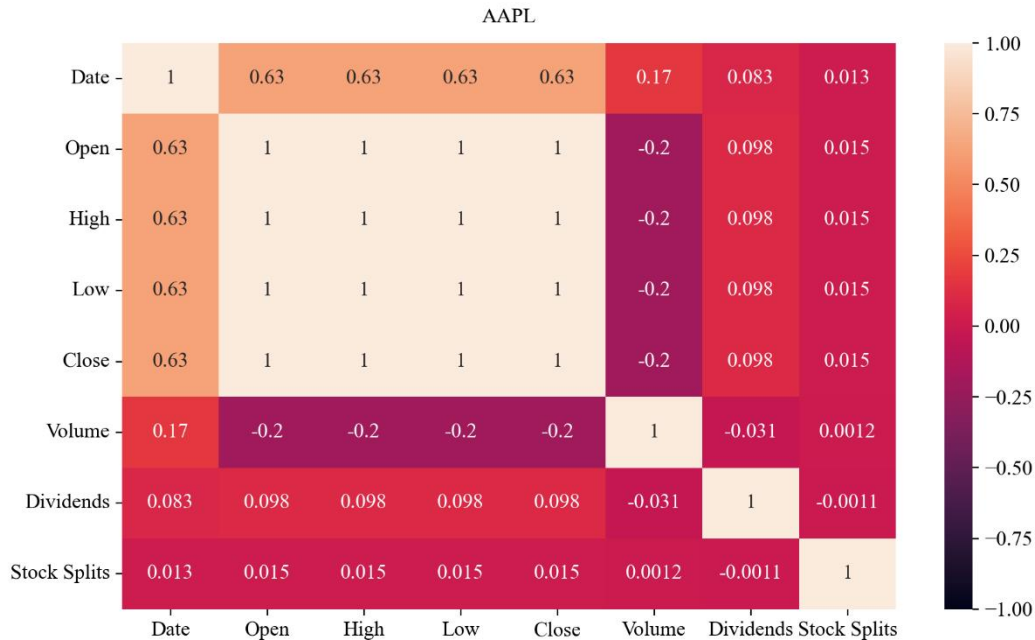
Rezultat Pearsonove korelacije je ocjena na skali od -1 do 1. Ta ocjena predstavlja korelacijski koeficijent koji može biti pozitivan ili negativan. Ako je koeficijent pozitivan, rastom jedne varijable će rasti i druga. Za negativan korelacijski koeficijent vrijedi obrnuto. Apsolutna vrijednost korelacijskog koeficijenta govori koliko jedna varijabla utječe na drugu. Ako je koeficijent od 0 do 0.5 to znači da varijable slabo utječu jedna na drugu, a ako je koeficijent od 0.5 do 1 to znači da varijable snažno utječu jedna na drugu. Korelacija je bitna jer se tako vidi koje varijable najviše utječu na rezultat te bi se onda mogle izbaciti one varijable sa slabim utjecajem. Na slikama: 2.1, 2.2, 2.3, 2.4, 2.5 su prikazani rezultati korelacijske analize napravljeni uz pomoć Seaborn knjižnice.



Slika 3.1 Korelacijska analiza AMD dataseta

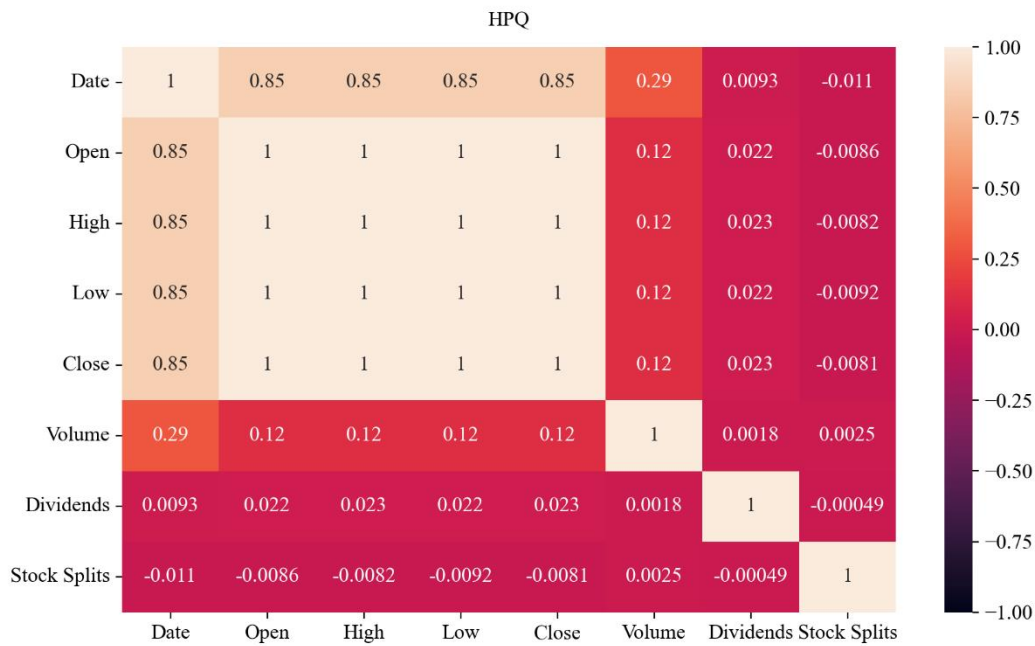
Na slici 3.1 prikazani su rezultati korelacijske analize AMD dataseta. Može se primjetiti da najveći utjecaj međusobno imaju Open, High, Low i Close sa ocjenom 1 što predlaže jaku pozitivnu korelaciju. Date ima slabiju korelaciju za Open, High, Low i Close, ali nešto jaču za Volume. Volume je kao i Date slabije koreliran s Open, High, Low i Close. Stock splits ima slabu korelaciju sa svim ostalim varijablama, praktički nema korelacije. Dividends u AMD datasetu konstantno

ima vrijednost 0 te zbog toga nema utjecaja na druge varijable pa zato njegov utjecaj nije prikazan na slici.



Slika 3.2 Korelacijska analiza AAPL dataseta

Na slici 3.2 prikazani su rezultati korelacijske analize AAPL dataseta. Može se primjetiti da najveći utjecaj međusobno imaju Open, High, Low i Close sa ocijenom 1 što predlaže jaku pozitivnu korelaciju. Date ima donekle jaku korelaciju za Open, High, Low i Close, ali slabiju za Volume, Dividends i Stock Splits. Volume i Stock Splits su slabo korelirani s ostalim varijablama, ocjena im je blizu 0 što ukazuje da gotovo nema korelacije.



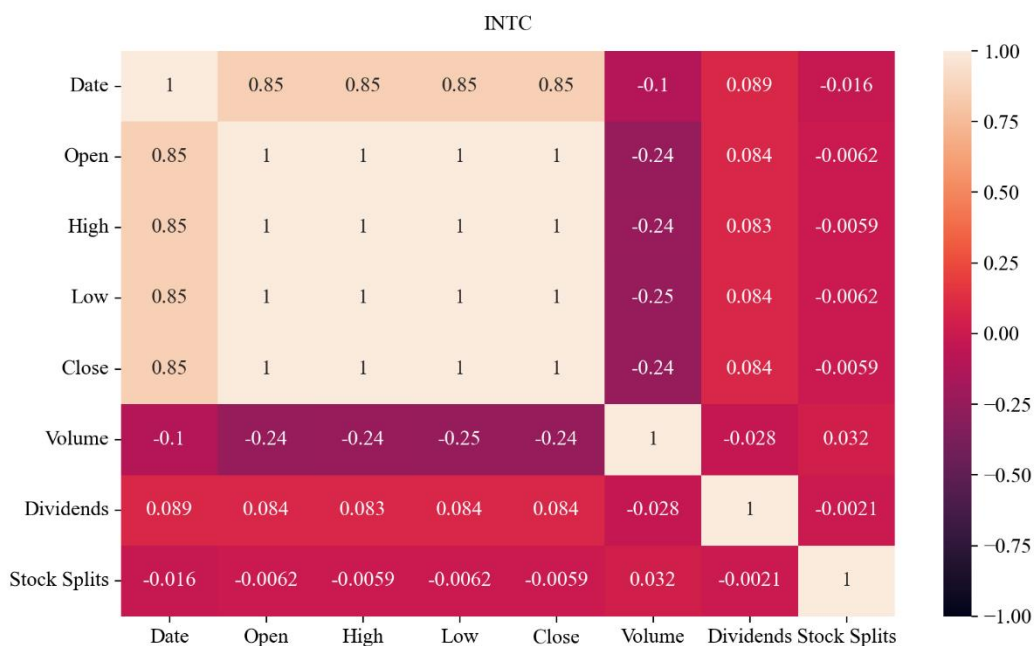
*Slika 3.3 Korelacijska analiza HPQ dataseta*

Na slici 3.3 prikazani su rezultati korelacijske analize HPQ dataseta. Može se primjetiti da najveći utjecaj međusobno imaju Open, High, Low i Close sa ocijenom 1 što predlaže jaku pozitivnu korelaciju. Date ima jaku korelaciju za Open, High, Low i Close, ali slabiju za Volume, Dividends i Stock Splits. Volume i Stock Splits su slabo korelirani s ostalim varijablama, ocjena im je blizu 0 što ukazuje da gotovo nema korelacije.



*Slika 3.4 Korelacijska analiza IBM dataseta*

Na slici 3.4 prikazani su rezultati korelacijske analize IBM dataseta. Može se primjetiti da najveći utjecaj međusobno imaju Open, High, Low i Close sa ocijenom 1 što predlaže jaku pozitivnu korelaciju. Date ima jaku korelaciju za Open, High, Low i Close, ali slabu za Volume, Dividends i Stock Splits. Volume i Stock Splits su slabo korelirani s ostalim varijablama, ocjena im je blizu 0 što ukazuje da gotovo nema korelacije.



*Slika 3.5 Korelacijska analiza INTC dataseta*

Na slici 3.5 prikazani su rezultati korelacijske analize INTC dataseta. Može se primjetiti da najveći utjecaj međusobno imaju Open, High, Low i Close sa ocijenom 1 što predlaže jaku pozitivnu korelaciju. Date ima jaku korelaciju za Open, High, Low i Close, ali slabiju za Volume, Dividends i Stock Splits. Volume i Stock Splits su slabo korelirani s ostalim varijablama, ocjena im je blizu 0 što ukazuje da gotovo nema korelacije.

Iz ovih rezultata se vidi da dividends i stock splits imaju najmanje utjecaja na Close. Volume je utjecajniji od Dividends i Stock Splitsa, ali ne previše. Kroz svih 5 datasetova najutjecajniji na Close su Open, Low i High.

### 3.3. Oblikovanje dataseta za primjenu modela strojnog učenja

Prvi korak je da se izbrišu podaci stariji od 10454 dana na svim datasetima osim Appleovog kako bi im se izjednačila veličina. Nakon toga datumi se pretvaraju u oblik integera. Close se izvlači iz dataseta te se stavlja pod varijablu y dok se ostatak dataseta postavlja pod varijablu X. Varijabla X su podaci koji ulaze u model te se njegova previđanja predstavljaju sa varijablom y. Varijable X i y se dijele na 70% za treniranje modela, a 30% će biti za testiranje točnosti. Dijeljenjem na taj

način se dobivaju nove varijable:  $X_{train}$ ,  $X_{test}$ ,  $y_{train}$ ,  $y_{test}$ . Varijable  $X_{train}$  i  $y_{test}$  služe za treniranje modela. Nakon što je model istreniran kao ulaz mu se stavlja  $X_{test}$  te rezultat modela se uspoređuje sa varijablom  $y_{test}$  kako bi se vidjelo koliko je model točan.

### 3.4. Korišteni modeli

U datasetu se nalaze mjerenja koje će se koristiti za predviđanje cijene pri zatvaranju burze. To je regresijski problem te će se za to koristiti regresijski modeli strojnog učenja iz „scikit-learn“ python knjižnice. Unutar te iste knjižnice se nalaze i metode skaliranja i normalizacije te korištene evaluacijske metode [14]. Cilj regresije je da pronađe funkciju koja je „najbliža“ mjerenjima [7].

#### 3.4.1. Linearna regresija

Linearna regresija je model koji opisuje vezu između jedne ili više varijabli označene s  $y$  te jedne ili više varijabli označene s  $x$ . Jednadžba za pravac dana je sa:

$$y = a_0 + a_1 \cdot x. \quad (3.2)$$

Odstupanje pravca od točaka dano je sa:

$$S = \sum_{i=1}^N (y_i - (a_0 + a_1 x_i))^2. \quad (3.3)$$

Gdje su  $a_0$  i  $a_1$  koeficijenti regresijskog pravca,  $a_0$  predstavlja presjek pravca s  $y$  osi, a  $a_1$  nagib pravca. Najbolji regresijski pravac će biti onaj sa najmanjom sumom odstupanja  $S$ .

Linearna regresija imati će loš rezultat ako su  $x$  i  $y$  varijable nelinearno vezane te da bi se to spriječilo potrebno je skalirati podatke [15].



### 3.4.2. Višeslojni perceptron

Višeslojni perceptron (engl. Multilayer Perceptron-MLP) je oblik regresije koja koristi Umjetnu neuronsku mrežu (engl. Artificial neural network) sa 3 ili više slojeva perceptrona. Prvi sloj je ulazni sloj, drugi sloj je skriveni sloj te njih može biti više od jedan, zadnji sloj je izlazni sloj. Informacije dolaze na ulazni sloj te se procesiraju na skrivenim slojevima čiji se „neuroni“ treniraju preko povratne veze, a estimacija se izvršava preko zadnjeg sloja [16]. U skrivenom sloju se odrađuje svo računanje, svakom neuronu se dodjeljuje uteg (engl. weight) koji određuje koliko će utjecati na rezultat, vrijednost utega se mijenja preko povratne veze. Kako bi se „neuron“ uključio koristi se aktivacijska funkcija koja i ujedino može pridodati nelinearnosti. Neke od aktivacijskih funkcija su: tanh, sigmoid, ReLU [17]. Za razliku od linearne regresije izlaz ne mora biti u obliku pravca.

### 3.4.3. Regresija najmanjeg kuta

Regresija najmanjeg kuta ili Lars (engl. Least angle regression-Lars) je model koji se koristi na podacima s većim brojem varijabli. Lars funkcionira tako da pronađe varijablu koji ima najveću korelaciju sa ciljanom vrijednošću te se onda rezultatna funkcija kreće prema toj varijabli, ako je više varijabli sa istom korelacijom onda se izlazna vrijednost orijentira prema njihovoj srednjoj vrijednosti. Algoritam radi tako da prvo normalizira sve vrijednosti na način da im je srednja vrijednost i varijanca 0, zatim pregledava korelaciju između varijabli te onda pomiče regresijsku liniju u njenom smjeru sve dok se ne dostigne druga varijabla s istom ili većom korelacijom [18]. Na taj način se postiže visoka razina optimizacije. Mana lars regresije je to što je osjetljiva na šum.

### 3.4.4. Operator najmanjeg apsolutnog skupljanja i odabira i Lars Lasso regresija

Operator najmanjeg apsolutnog skupljanja i odabira (engl. Least Absolute Shrinkage and Selection Operator-Lasso) je tip regresije koji funkcionira slično kao linearna regresija, ali ima dodatnu varijablu alfa (u nekoj literaturi se koristi lambda) čijim povećanjem se mijenja osjetljivost prema nezavisnim varijablama. Postoji mogućnost da će se neke varijable u potpunosti zanemariti. Cilj ovog algoritma je da se smanji broj korištenih varijabli iz dataseta. Za slučaj da je alfa 0 onda će izlaz biti isti kao kod linearne regresije. Alfa je suma apsolutnih vrijednosti varijable koje se

estimira. Cilj povećavanja alfa varijable je eliminiranje nezavisnih varijabli koje ne utječu na rezultat. Ovaj princip korištenja alfa za uklanjanje utjecaja varijabli se naziva L1 regularizacija [19].

Lars Lasso regresija kombinira zadnje dvije regresije. Model prati lars algoritam, ali ima dodatnu varijablu alfa s kojom se primjenjuje L1 regularizacija [20].

### 3.4.5. Bayesova grebenasta regresija i automatsko određivanje relevantnosti

Bayesova grebenasta regresija (engl. Bayesian Ridge ) je tip regresije utemeljen na Bayesovom teoremu vjerojatnosti. Bayesov teorem daje vezu apriorne i aposteriorne vjerojatnosti nakon ažuriranja informacija. Koristi se za estimiranje nepoznatih parametara koji su bazirani na poznatim parametrima [21]. Apriorne vrijednosti težina su postavljene preko Gausove sferične distribucije. Koristeći L2 regularizaciju težine parametara su generalno niske vrijednosti, ali nikad 0.

Probleme kao što su nedovoljan broj podataka i loša distribucija podataka rješava stvaranjem linearne regresije tako da se postave vjerojatnosti preko koeficijenata preciznosti suma i težina .

Automatsko određivanje relevantnosti (engl. Automatic Relevance Determination-ARD) je linearna regresija slična BayesianRidge regresiji. Koristi centriranu eliptičnu Gausovu distribuciju za postavljanje težina. Kao što ime nalaže, algoritam sam određuje koje su varijable najbitnije te ostalima zadaje težinu 0.

## 3.5. Metode skaliranja i normalizacije

Podaci u datasetu su mjereni na različitim skalama pa zbog toga ne pridonose jednako modelu što može uzrokovati da model strojnog učenja bude opredijeljen prema korištenju specifičnog raspona vrijednosti. Ako se npr. koristi model linearne regresije na podacima koji za to nisu prikladni, rezultati modela mogu biti nezadovoljavajući te mogu uzrokovati povećano vrijeme treniranja modela [22]. Ako model nije ovisan o skaliranju podataka onda je skaliranje podataka nepotrebno.

Kako bi se riješio taj problem koristi se skaliranje i normalizacija. Pri skaliranju mijenja se raspon podataka dok se pri normalizaciji mijenja oblik distribucije podataka tipično tako da vrijednosti

budu u rasponu između 0 i 1 [22]. Pri standardizaciji se srednja vrijednost mijenja u 0, a standardna devijacije postaje 1.

Korištene metode su opisane u nastavku.

### 3.5.1. MaxAbsScaler

MaxAbsScaler skalira svaki stupac tako da njegova apsolutna maksimalna vrijednost bude 1, a ostale vrijednosti su podijeljene sa maksimalnom čime će novi podaci imati vrijednost manju od 1.

MaxAbsScaler je osjetljiv na odstupajuće podatke te se zbog njih podaci koji se nalaze oko srednje vrijednosti mogu „stisnuti“ no distribucija podataka ostaje ista [23].

### 3.5.2. MinMaxScaler

MinMaxScaler normalizira podatke tako da najmanju vrijednost podataka pretvara u 0 dok najveću vrijednost pretvara u 1, a ostale vrijednosti se dijele sa njihovom razlikom. Kao i MaxAbsScaler osjetljiv je na odstupajuće podatke [23].

### 3.5.3. Normalizer

Normalizer radi tako da prvo zbroji kvadrirane vrijednosti u retku te ih korjenjuje. Nove vrijednosti se dobivaju tako da se svaka vrijednost u retku zasebno podijeli s dobivenim brojem [24].

### 3.5.4. RobustScaler

RobustScaler skalira tako da od vrijednosti koju je potrebno skalirati oduzme medijan stupca te rezultat podijeli sa razlikom vrijednosti većeg (75%) i manjeg (25%) kvantila tog reda. Budući da je RobustScaler baziran na postocima, odstupajući podaci ne utječu na njega. Sami podaci su sličniji originalu u usporedbi s drugim metodama skaliranja [23].

### 3.5.5. StandardScaler

StandardScaler skalira podatke tako da prvo oduzme srednju vrijednost svih podataka te onda tu razliku podijeli sa standardnom deviacijom. Skaliranje smanjuje raspon veličina podataka te je osjetljivo na odstupajuće podatke [23].

### 3.5.6. PowerTransformer

PowerTransformer skalira podatke tako da ih prvo obradi da distribucija bude sličnija Gaussovoj. Nakon toga skalira podatke na isti način kao i StandardScaler. U hiperparametrima se mogu odabrati dvije metode skaliranja: „box-cox“ koja funkcionira samo sa pozitivnim vrijednostima i „yeo-johnson“ koja se koristi s pozitivnim i negativnim vrijednostima. PowerTransformer također standardizira podatke [23].

## 3.6. Evaluacijske metode

Kako bi se moglo ocjeniti različite modele koriste se 3 evaluacijske metode koje uspoređuju prave rezultate s onima dobivenih iz modela.

### 3.6.1. Koeficijent determinacije

Koeficijent determinacije označavan kao  $R^2$  je broj između 0 i 1 koji mjeri koliko točno neka statistička metoda predviđa točan rezultat [25]. Ako je koeficijent determinacije 1 to znači da model savršeno predviđa rezultate, a ako je koeficijent determinacije 0 to znači da model predviđa srednju vrijednost rezultata bez obzira na ulazne vrijednosti modela. Koeficijent determinacije također može biti negativan što znači da je korišteni model loš.

Koeficijent determinacije se računa pomoću formule:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}. \quad (3.4)$$

Gdje je  $\hat{y}_i$  vrijednost koju je predvidio korišteni model, a  $y_i$  je točna vrijednost iz dataseta. Varijabla  $n$  je sveukupan broj previđenih rezultata.

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i. \quad (3.5)$$

### 3.6.2. Srednja apsolutna pogreška

Srednja apsolutna pogreška (engl. Mean absolute error, skraćeno MAE) je prosječna vrijednost svih apsolutnih razlika između predviđenog rezultata i točnog mjerenja iz dataseta tako da je poželjno dobiti što manji broj. Ova evaluacijska metoda je ovisna o skali podataka na kojoj se koristi tako da nema smisla za usporedbu s drugim metodama ako se koriste različiti podaci. Srednju apsolutnu grešku računamo formulom:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i|. \quad (3.6)$$

Gdje je  $\hat{y}_i$  vrijednost koju je predvidio korišteni model, a  $y_i$  je točna vrijednost iz dataseta [26]. Varijabla  $n$  je sveukupan broj previđenih rezultata.

### 3.6.3. Korijen srednje kvadratne pogreške

Korijen srednje kvadratne pogreške (engl. Root mean square error, skraćeno RMSE) je još jedna metoda za procjenu greške, ali zbog kvadrata je puno osjetljiviji na velike brojke. Ako su greške relativno niske onda će MAE i RMSE biti približno jednaki, ali na datasetu s velikim greškama RMSE će biti veći od MAE. Formula za računanje RMSE:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}. \quad (3.7)$$

Gdje je  $\hat{y}_i$  vrijednost koju je predvidio korišteni model, a  $y_i$  je točna vrijednost iz dataseta [26].  
Varijabla  $n$  je sveukupan broj previđenih rezultata.

## 4. Rezultati primjenjenih modela

U sljedećim poglavljima su prikazani rezultati različitih modela primjenjenih na pet dataseta uključujući i različite metode skaliranja.

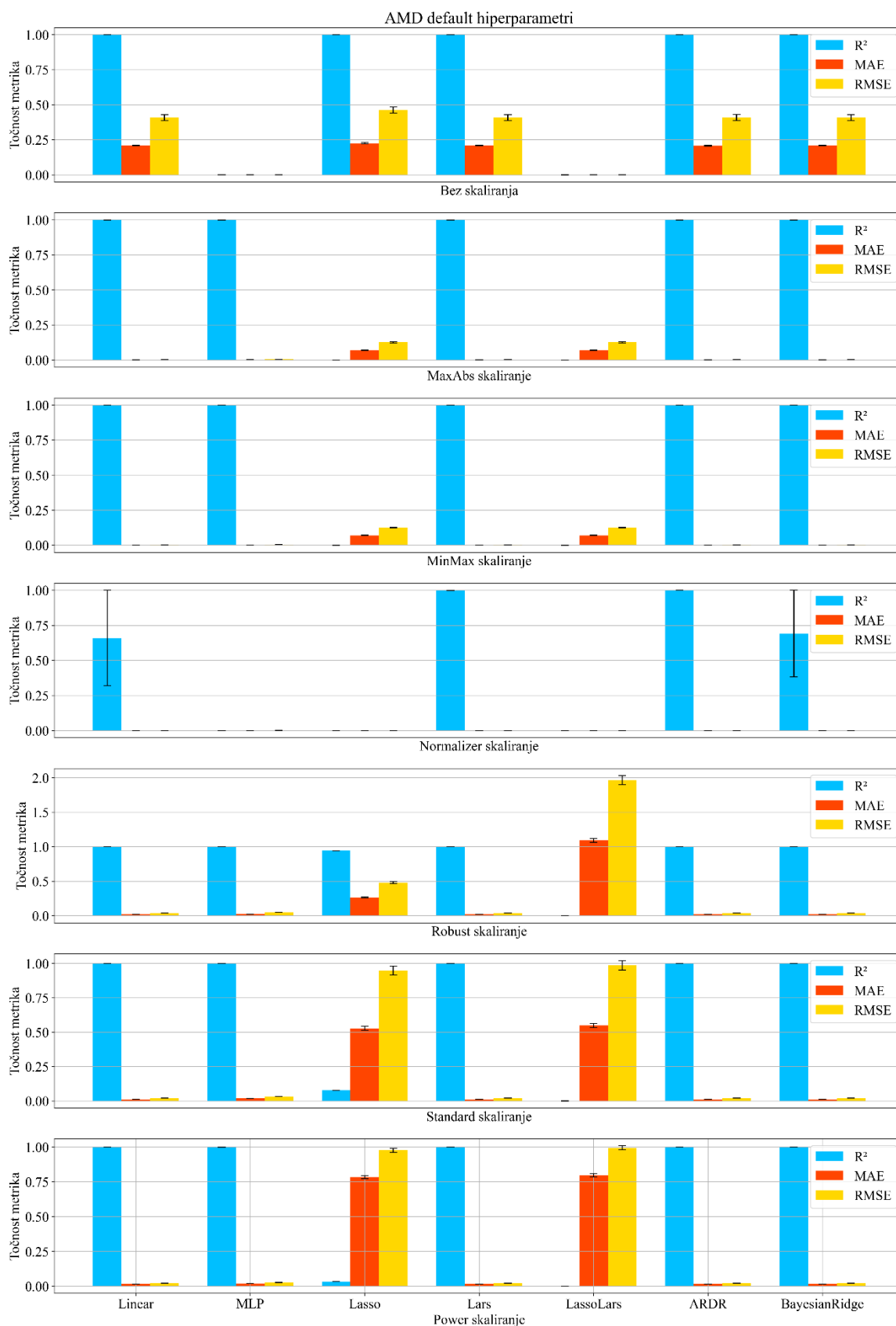
### 4.1. AMD dataset

Na slici 4.1 su prikazani rezultati modela korištenih na AMD datasetu.

Neke vrijednosti su bile toliko velike da su narušavale izgled grafa te su zbog toga uklonjene, to su  $R^2$ , MAE i RMSE MLP modela čije su vrijednosti redom: -847605.28, 10217.98, 18616.47, za LassoLars model te vrijednosti iznose: -0.00067, 11.26, 20.25, pri normalizer skaliranju iste vrijednosti za MLP model iznose: -53.71,  $3.79 \cdot 10^{-5}$ , 0.00092.

Rezultati  $R^2$  metrike su generalno dobri, ali lošiji se pojavljuju u normalizer skaliranju za linearni i Bayesian Ridge model dok loše rezultate imaju Lasso i LassoLars modeli pri standard i power skaliranju te slučajevi koji su uklonjeni iz grafa. Skaliranje maxabsom i minmaxom dobivaju se dobri rezultati osim za Lasso i LassoLars modele. Skaliranje normalizerom daje loše rezultate osim u slučaju linearnog i Lars modela. Robust skaliranje ima loš rezultat za LassoLars model.

Rezultati za MAE i RMSE metrike ukazuju da su greške uglavnom niske bez skaliranja te se nakon skaliranja povećavaju osobito za Lasso i LassoLars modele a s tim da power skaliranje ima najveće greške. Izuzetak ovome je normalizer skaliranje koje ima najmanje MAE i RMSE greške na svim modelima.



Slika 4.1 Rezultati modela primjenjenih na AMD dataset

Napomena: vrijednosti MLP i LassoLars modela bez skaliranja i MLP modela u normalizer skaliranju su uklonjene jer narušavaju izgled grafa



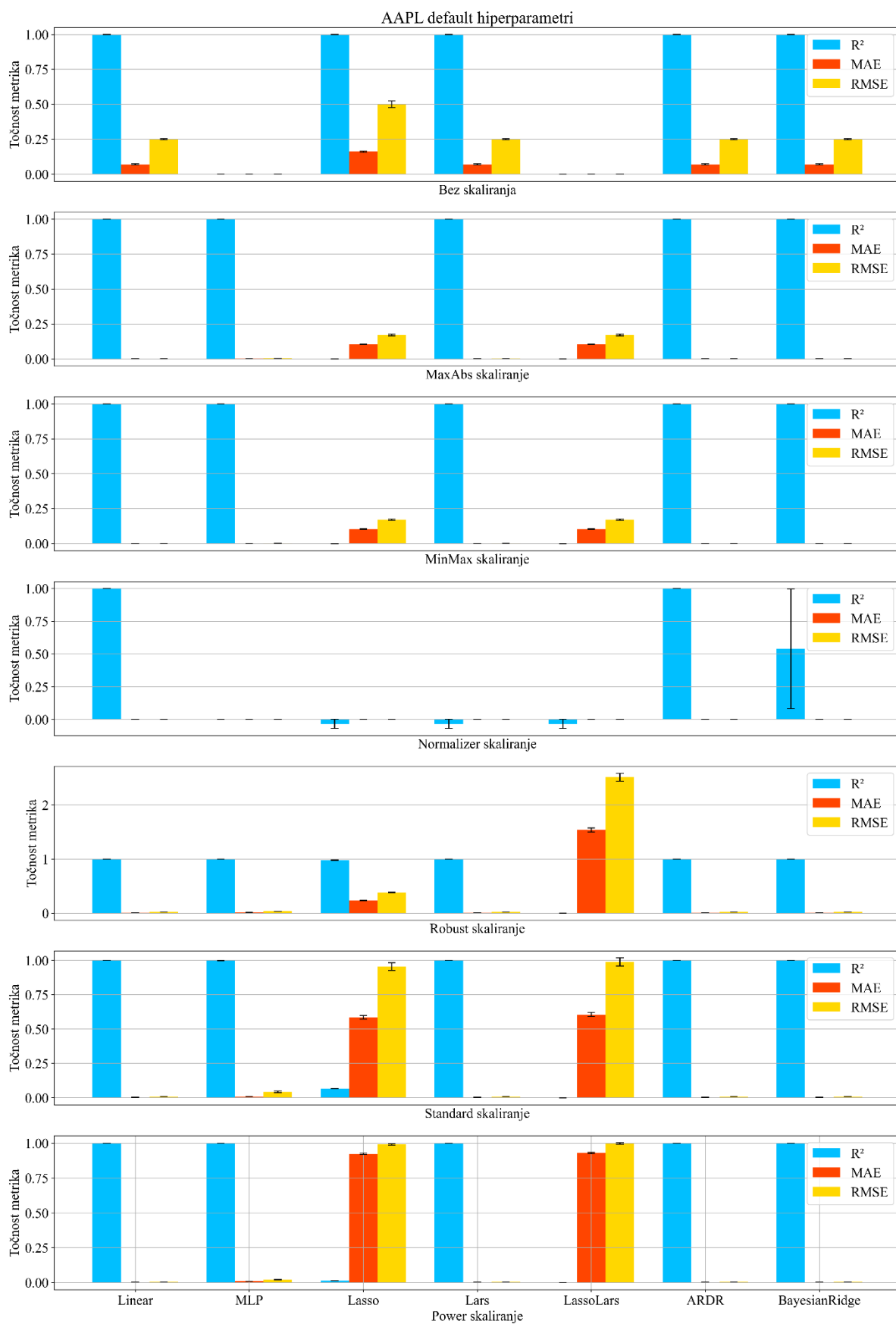
## 4.2. Apple dataset

Na slici 4.2 su prikazani rezultati modela korištenih na AAPL datasetu.

Neke vrijednosti su bile toliko velike da su narušavale izgled grafa te su zbog toga uklonjene, to su  $R^2$ , MAE i RMSE MLP modela čije su vrijednosti redom: -47702326.85, 148915.89, 213256.44, za LassoLars model te vrijednosti iznose: -0.000897, 18.94, 30.93, pri normalizer skaliranju iste vrijednosti za MLP model iznose: -5273.80,  $3.36 \cdot 10^{-5}$ , 0.00011.

Rezultati  $R^2$  metrike su generalno dobri, ali lošiji se pojavljuju u normalizer skaliranju za Bayesian Ridge model dok loše rezultate imaju Lasso i LassoLars modeli pri maxabs, minmax, standard i power skaliranju, LassoLars model pri robust skaliranju, Lasso, Lars i LassoLars modelu kod normalizer skaliranja te slučajevi koji su uklonjeni iz grafa.

Rezultati za MAE i RMSE metrike ukazuju da su greške najmanje u slučaju skaliranja normalizerom dok su bez skaliranja greške uglavnom niske. Kroz ostala skaliranja su greške niske osim za Lasso i LassoLars modele.



Slika 4.2 Rezultati modela primjenjenih na AAPL dataset  
 Napomena: vrijednosti MLP i LassoLars modela bez skaliranja i MLP modela u normalizer skaliranju su uklonjene jer narušavaju izgled grafa

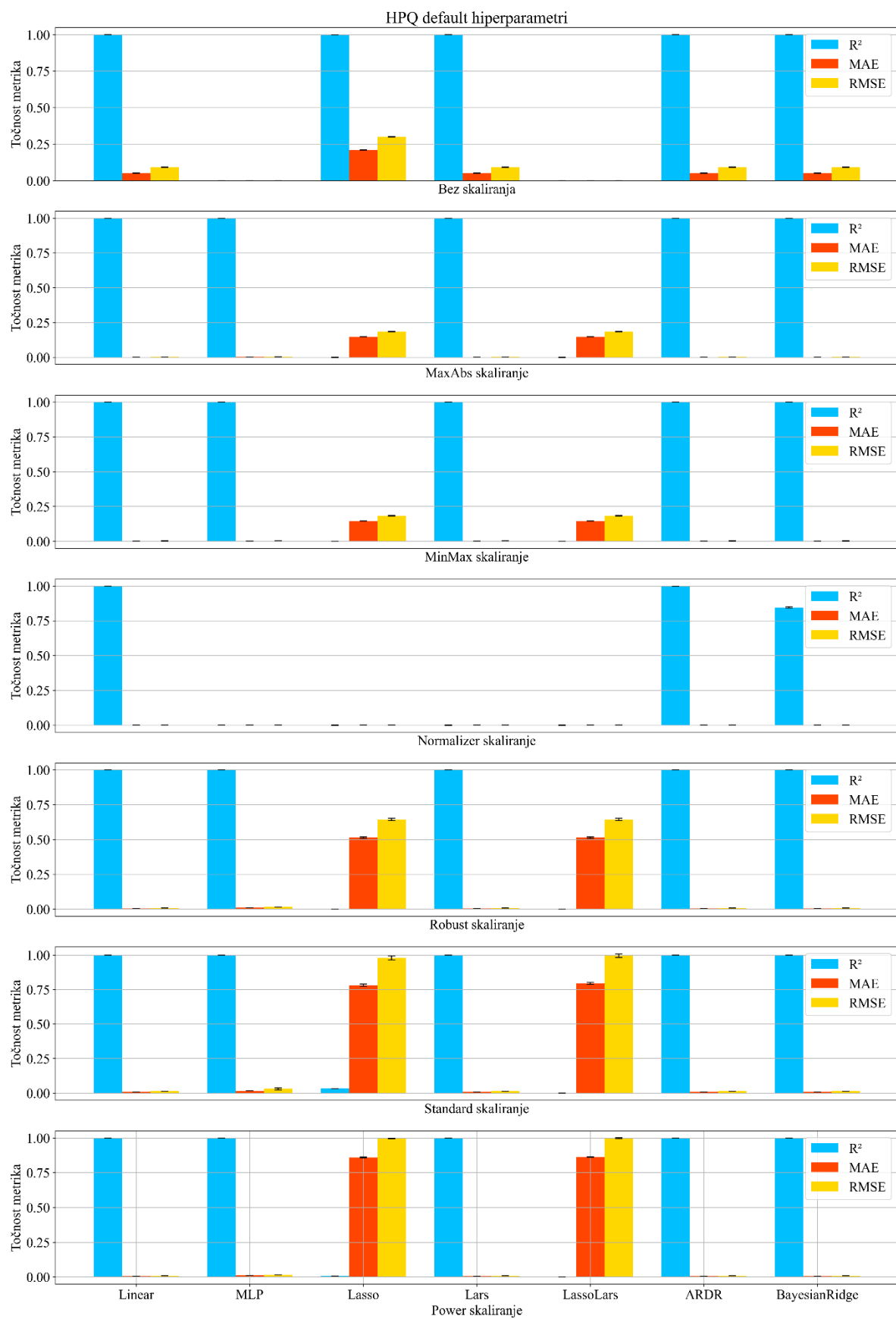
### 4.3. HPQ dataset

Na slici 4.3 su prikazani rezultati modela korištenih na HPQ datasetu.

Neke vrijednosti su bile toliko velike da su narušavale izgled grafa te su zbog toga uklonjene, to su  $R^2$ , MAE i RMSE MLP modela čije su vrijednosti redom: -143930.98, 2064.31, 2789.62, za LassoLars model te vrijednosti iznose -0.00057, 5.87, 7.35, pri normalizer skaliranju iste vrijednosti za MLP model iznose: -867.74,  $1.82 \cdot 10^{-5}$ ,  $2.45 \cdot 10^{-5}$ .

Rezultati  $R^2$  metrike su generalno dobri, ali loše rezultate imaju Lasso i LassoLars modeli pri robust, maxabs, minmax, standard i power skaliranju, Lasso, Lars i LassoLars modelu kod normalizer skaliranja te slučajevi koji su uklonjeni iz grafa.

Rezultati za MAE i RMSE metrike ukazuju da su greške najmanje u slučaju skaliranja normalizerom dok su bez skaliranja greške uglavnom niske. Kroz ostala skaliranja su greške niske osim za Lasso i LassoLars modele.



Slika 4.3 Rezultati modela primjenjenih na HPQ dataset  
 Napomena: vrijednosti MLP i LassoLars modela bez skaliranja i MLP modela u normalizer skaliranju su uklonjene jer narušavaju izgled grafa

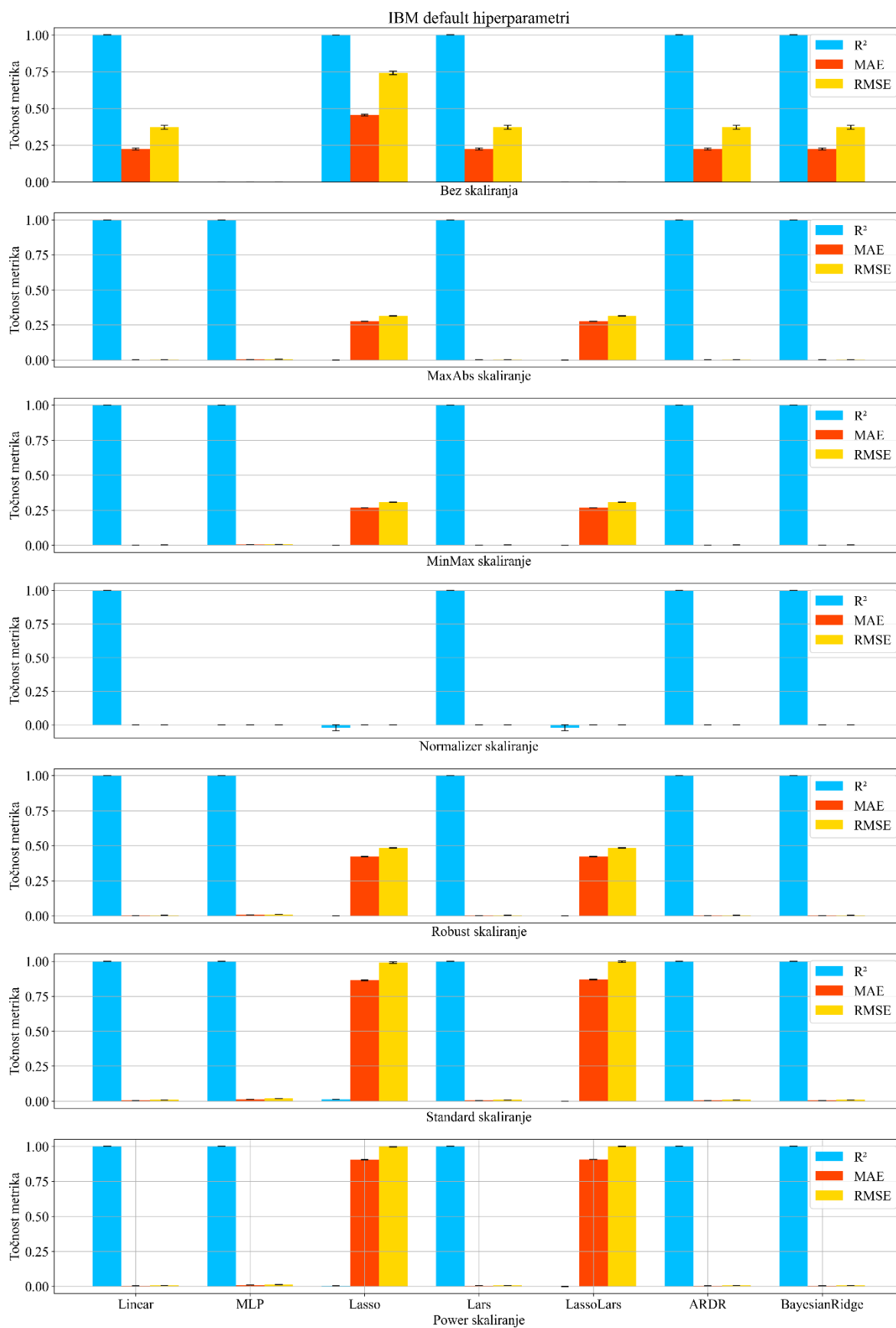
#### 4.4. IBM dataset

Na slici 4.4 su prikazani rezultati modela korištenih na IBM datasetu.

Neke vrijednosti su bile toliko velike da su narušavale izgled grafa te su zbog toga uklonjene, to su  $R^2$ , MAE i RMSE MLP modela čije su vrijednosti redom: 0.83, 14.82, 18.14, za LassoLars model te vrijednosti iznose -0.00058, 38.44, 44.07, pri normalizer skaliranju iste vrijednosti za MLP model iznose: -194.26, 0.00018, 0.00025.

Rezultati  $R^2$  metrike su generalno dobri te za razliku od prijašnjih slučajeva ovdje MLP bez skaliranja ima relativno visok rezultat, ali kao i prije loše rezultate imaju Lasso i LassoLars modeli pri maxabs, minmax, robust, standard i power skaliranju, MLP, Lasso i LassoLars modelu kod normalizer skaliranja te LassoLars model bez skaliranja.

Rezultati za MAE i RMSE metrike ukazuju da su greške najmanje u slučaju skaliranja normalizerom dok su bez skaliranja greške uglavnom niske. Kroz ostala skaliranja su greške niske osim za Lasso i LassoLars modele.



Slika 4.4 Rezultati modela primjenjenih na IBM dataset  
 Napomena: vrijednosti MLP i LassoLars modela bez skaliranja i MLP modela u normalizer skaliranju su uklonjene jer narušavaju izgled grafa

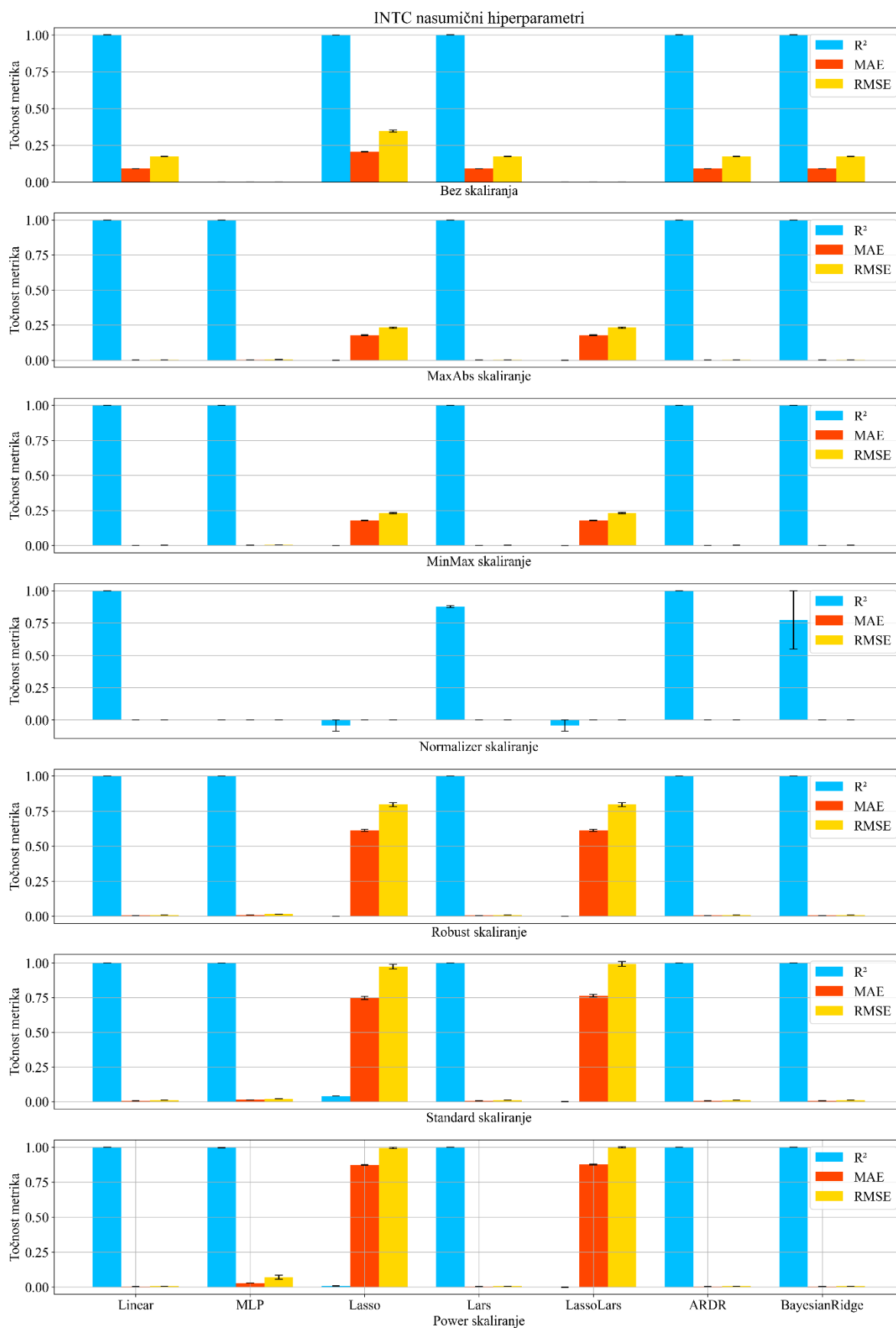
## 4.5. Intel dataset

Na slici 4.5 su prikazani rezultati modela korištenih na INTC datasetu.

Neke vrijednosti su bile toliko velike da su narušavale izgled grafa te su zbog toga uklonjene, to su  $R^2$ , MAE i RMSE MLP modela čije su vrijednosti redom: -23.76, 54.84, 75.63, za LassoLars model te vrijednosti iznose -0.00094, 11.70, 15.22, pri normalizer skaliranju iste vrijednosti za MLP model iznose: -1232.87,  $2.35 \cdot 10^{-5}$ ,  $3.26 \cdot 10^{-5}$ .

Rezultati  $R^2$  metrike su generalno dobri, ali loše rezultate imaju Lasso i LassoLars modeli pri svim skaliranjima te slučajevi koji su uklonjeni iz grafa.

Rezultati za MAE i RMSE metrike ukazuju da su greške najmanje u slučaju skaliranja normalizerom dok su bez skaliranja greške uglavnom niske. Kroz ostala skaliranja su greške niske osim za Lasso i LassoLars modele.



Slika 4.5 Rezultati modela primjenjenih na AAPL dataset  
 Napomena: vrijednosti MLP i LassoLars modela bez skaliranja i MLP modela u normalizer skaliranju su uklonjene jer narušavaju izgled grafa



## 5. Optimiziranje modela nasumičnim odabirom hiperparametara

Kako bi se povećala preciznost modela potrebno je podesiti njihove hiperparametre. Kako bi se pronašli najbolji hiperparametri nasumično će se isprobavati različite vrijednosti hiperparametara dok se ne dostigne željeni rezultat. U slučaju da se nije mogao potići bolji rezultat od početnog koristit će se default hiperparametri.

### 5.1. Hiperparametri modela

Svaki model ima zasebnu skupinu hiperparametara koji su opisani u nastavku.

#### 5.1.1. Hiperparametri linearnog modela

*Tablica 5.1 Opis hiperparametara linearnog modela*

Hiperparametar	Opis
Fit intercept	Ako je postavljen u „true“ stanje omogućava regresijskom pravcu da mu je ishodište iz točke koja najbolje opisuje regresijski pravac. Inače će regresijski pravac početi iz nule.
Copy_X	Ako je postavljen u „true“ stanje kopira ulaznu vrijednost X kako se ne bi promijenila, inače postoji šansa da se X promijeni nakon fit-a.
Positive	Kad je postavljen u „true“ stanje forsira koeficijente da budu pozitivni.

### 5.1.2. Hiperparametri MLP modela

Tablica 5.2 Opis hiperparametara MLP modela

Hiperparametar	Opis
Hidden_layer_sizes	Zadaje se broj koji određuje broj neurona u skrivenom sloju.
Activation	Bira se aktivacijska funkcija za skriveni sloj. Ona može biti „identity“ koja se ponaša linearno, „logistic“ koja daje sigmoidnu funkciju, „tanh“ koristi funkciju hiperbolnog tangensa, „relu“ na izlaz daje vrijednost ulaza dok god je ulaz pozitivan, inače daje nulu.
Solver	Služi za optimiziranje neurona. Različiti solveri su „lbfgs“ koji optimizira težine po Taylorovom nizu, „sgd“ koristi stohastički model postupnog opadanja, „adam“ koristi modificiranu verziju stohastičkog modela.
Alpha	Mijenjanjem ove varijable se mijenja osjetljivost regresijskog koeficijenta. Smanjenjem alfe se smanjuje šansa za overfitting.
Learning_rate	Koristi se samo kad je solver sgd, služi za ažuriranje težina.
Learning_rate_init	Broj koji određuje inicijalnu težinu te određuje veličinu koraka pri ažuriranju težina.
Warm_start	Ako je hiperparametar uključen koristi se zadnje rješenje pri novoj inicijalizaciji.
Beta_1,Beta_2	Eksponecijalne stope padanja za estimaciju prvog i drugog vektora momenta, koriste se kad je solver adam.

### 5.1.3. Hiperparametri LARS modela

Tablica 5.3 Opis hiperparametara LARS modela

Hiperparametar	Opis
Eps	Koristi se za regulaciju preciznosti u računanju Cheoleskyovih dijagonalnih faktora koji služe za rješavanje sustava linearnih jednadžbi.
n_nonzero_coefs	Određuje koliko će se koeficijenata koristiti, a da nisu nula.
Jitter	Dodaje neku vrijednost na y kako bi se povećala stabilnost.

#### 5.1.4. Hiperparametri Lasso modela

Tablica 5.4 Opis hiperparametara Lasso modela

Hiperparametar	Opis
Alpha	Koristi se za mijenjanje varijable alfa čijim povećanjem se mijenja osjetljivost prema nezavisnim varijablama.
Max_iter	Zadaje se broj maksimalnih iteracija koje model može provesti.
Warm_start	Ako je hiperparametar uključen koristi se zadnje rješenje pri novoj inicijalizaciji.
Positive	Kad je postavljen u „true“ stanje forsira koeficijente da budu pozitivni.
Tol	Određuje koliko blizu rješenja mora biti iteracija da bi se model prekinuo.

#### 5.1.5. Hiperparametri LassoLars modela

Tablica 5.5 Opis hiperparametara LassoLars modela

Hiperparametar	Opis
Alpha	Koristi se za mijenjanje varijable alfa čijim povećanjem se mijenja osjetljivost prema nezavisnim varijablama.
Max_iter	Zadaje se broj maksimalnih iteracija koje model može provesti.
Positive	Kad je postavljen u „true“ stanje forsira koeficijente da budu pozitivni.
Jitter	Dodaje neku vrijednost na y kako bi se povećala stabilnost.
Eps	Koristi se za regulaciju preciznosti u računanju Cheoleskyovih dijagonalnih faktora koji služe za rješavanje sustava linearnih jednadžbi.

### 5.1.6. Hiperparametri ARD regresije

Tablica 5.6 Opis hiperparametara ARDR modela

Hiperparametar	Opis
N_iter	Zadaje se broj maksimalnih iteracija koje model može provesti.
Tol	Određuje koliko blizu rješenja mora biti iteracija da bi se model prekinuo.
Alpha_1	Hiperparametar oblika za gama distribuciju nad hiperparametrom lambda.
Alpha_2	Obrnuto skalirani hiperparametar za gama distribuciju nad hiperparametrom lambda.
Lambda_1	Hiperparametar oblika za gama distribuciju nad hiperparametrom lambda.
Lambda_2	Obrnuto skalirani hiperparametar za gama distribuciju nad hiperparametrom lambda.
threshold_lambda	Koristi se za otklanjanje težina iz izračuna.

### 5.1.7. Hiperparametri BayesianRidge regresije

Tablica 5.7 Opis hiperparametara BayesianRidge modela

Hiperparametar	Opis
N_iter	Zadaje se broj maksimalnih iteracija koje model može provesti.
Tol	Određuje koliko blizu rješenja mora biti iteracija da bi se model prekinuo.
Alpha_1	Hiperparametar oblika za gama distribuciju nad hiperparametrom lambda.
Alpha_2	Obrnuto skalirani hiperparametar za gama distribuciju nad hiperparametrom lambda.
Lambda_1	Hiperparametar oblika za gama distribuciju nad hiperparametrom lambda.
Lambda_2	Obrnuto skalirani hiperparametar za gama distribuciju nad hiperparametrom lambda.

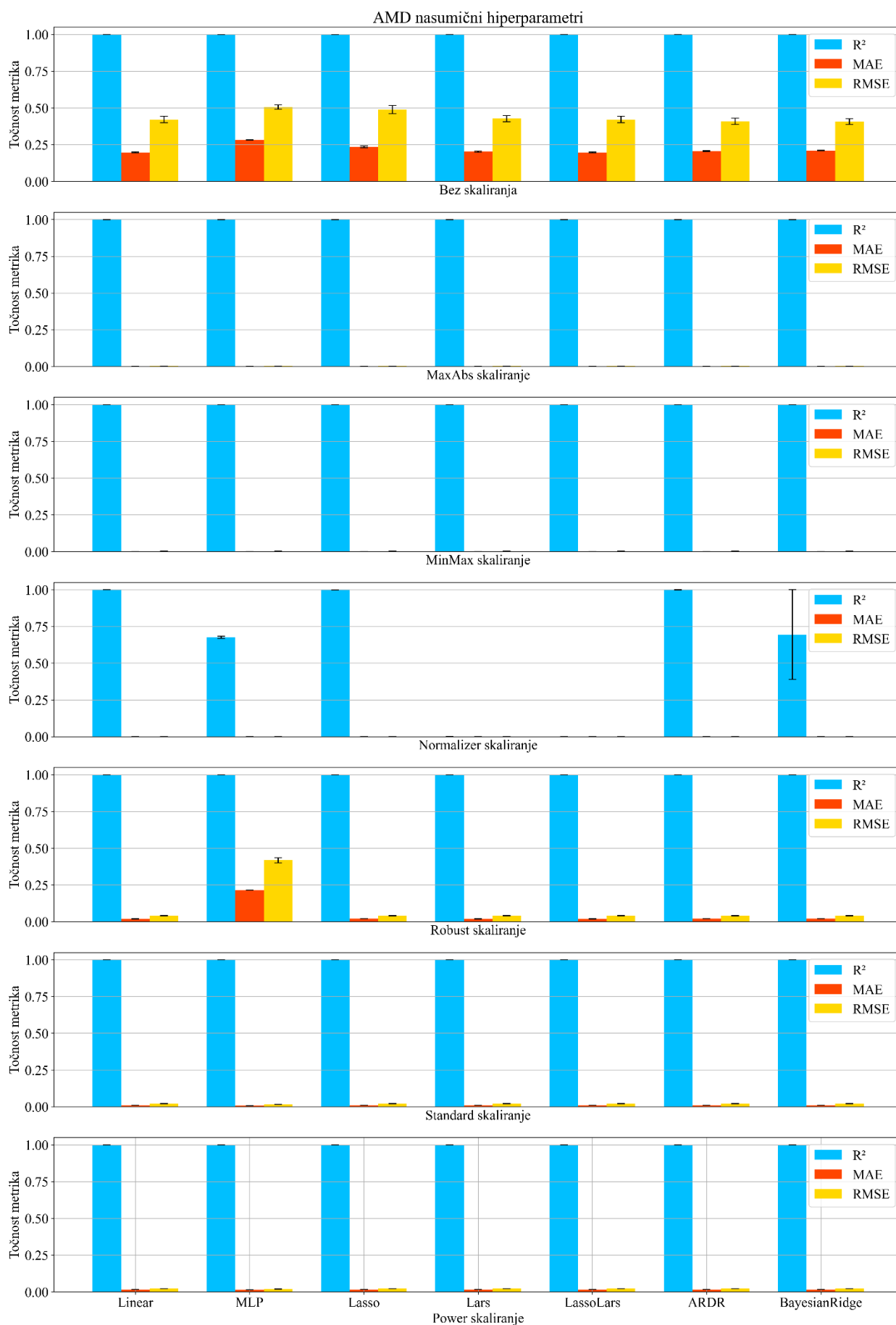
## 5.2. Rezultati modela s optimalnim hiperparametrima

Na slikama (5.1-5.5) prikazani su rezultati modela s optimalnim hiperparametrima za AMD, APPL, HPQ, IBM i INTC datasete.

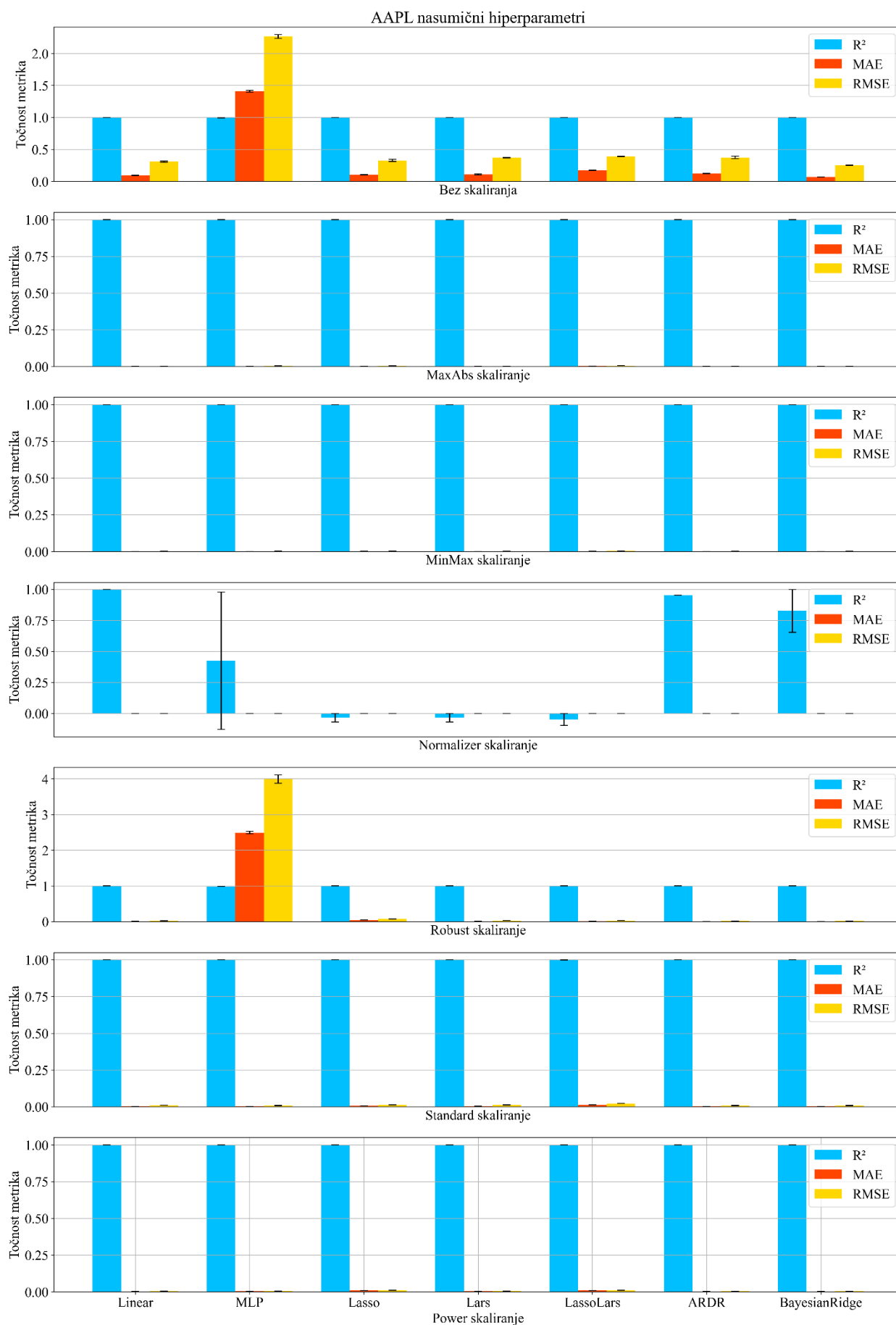
Rezultati su generalno slični, ali ipak ima nekih manjih razlika. Bez skaliranja većina modela ima odlične rezultate na  $R^2$  metrici. Što se tiče MAE i RMSE rezultati ukazuju na generalno male greške s tim da se na AAPL datasetu dobiva najveća greška i to za MLP model. Za maxabs i minmax skaliranja dobiveni su odlični rezultati za  $R^2$  metriku te sitne greške za MAE i RMSE metrike koje su praktički zanemarive. Pri skaliranju normalizerom MAE i RMSE metrike imaju odlične rezultate, ali rezultati  $R^2$  metrike su lošiji. Linearni model zadržava dobar rezultat, MLP je dobar za AMD, IBM i HPQ, a za AAPL i INTC ima rezultat od oko 0.4 s velikom standardnom devijacijom. Lasso ima dobar rezultat samo za AMD dok su ostali oko 0 ili niže. Lars daje dobar rezultat za IBM i donekle dobar za INTC dok su rezultati za ostale datasete poprilično loši. LassoLars ima jedini dobar rezultat od oko 0.7 na IBM datasetu dok je za ostale datasete poprilično loš. ARDR i BayesianRidge generalno daju dobre rezultate, ali BayesianRidge ima malo niže rezultate na AMD, AAPL i INTC datasetu. Za robust skaliranje  $R^2$  metrika ima dobre rezultate na svim modelima. MAE i RMSE ukazuju na niske greške, jedina iznimka tome je MLP gdje su greške više od ostalih modela pogotovo na AAPL datasetu gdje je greška velika. Standard i power skaliranja su poprilično slična.  $R^2$  metrika u svakom modelu ima dobar rezultat te su greške niske.

Tablice s hiperparametrima su dane u dodatku. Tablice (A.1-A.7, A.29-A.35, A.57-A.63, A.85-A.91, A.113-A.119) prikazuju hiperparametre za AMD, AAPL, HPQ, IBM i INTC modele s optimalnim hiperparametrima. U njima se vidi da za modele linearne regresije hiperparametri su uglavnom isti. Za MLP se vidi da ima velik broj neurona po sloju dok je broj slojeva više ovisan o datasetu. Aktivacijska funkcija je uglavnom identity na svim datasetovima i metodama skaliranja što predlaže da podaci imaju linearnu korelaciju. Solver je dominantno lbfgs. Alpha i batch\_size su uglavnom svi istog ili sličnog reda veličine. Learn\_rate nema toliko sličnosti, ali on nije bitan jer se on koristi samo ako je solver sgd što nije slučaj. Learn\_rate\_init se koristi samo kad je solver adam ili sgd te su brojke generalno istog reda veličine. Max\_iter je tipično viši od default vrijednosti te AAPL i INTC koji imaju slabije rezultate kod normalizera imaju identičan max\_iter. Tol su većinom slične. Warm\_start nema većih sličnosti među datasetovima. Za Lars modele eps je ili default ili se čini nepovezan. N\_nonzero\_coefs je većinom default ili kruži oko default. Jitter uvijek ima default odnosno none vrijednost. Pri Lasso modelu svi su hiperparametri generalno slični no zanimljivo je da pri normalizeru na svim datasetovima osim AAPL su za sve hiperparametre najbolje default vrijednosti, a dobre  $R^2$  rezultate dobivaju samo HPQ i INTC. Za LassoLars modele je slična situacija gdje su svi hiperparametri slični s tim da je alfa puno manja od svoje default verzije, a max\_iter značajno veći. Kod ARD regresije IBM i HPQ datasetovi imaju većinom default vrijednosti no nemaju lošije rezultate od ostalih datasetova. Ostale vrijednosti su uglavnom slične s tim da lambde i alfe imaju manje vrijednosti od defaulta, a n\_iter veće. Tol je

također manji od defaultne vrijednosti što bi moglo značiti da je modelu bilo lakše doći do dobrih rezultata. Za BayesianRidge regresiju je kao i kod ARD regresije najviše default vrijednosti kod IBM i HPQ dataseta dok su svi hiperparametri generalno slični. Najveća razlika je kod `n_iter` za AMD koji ima vrijednost samo 78 no svejedno postiže dobar rezultat.

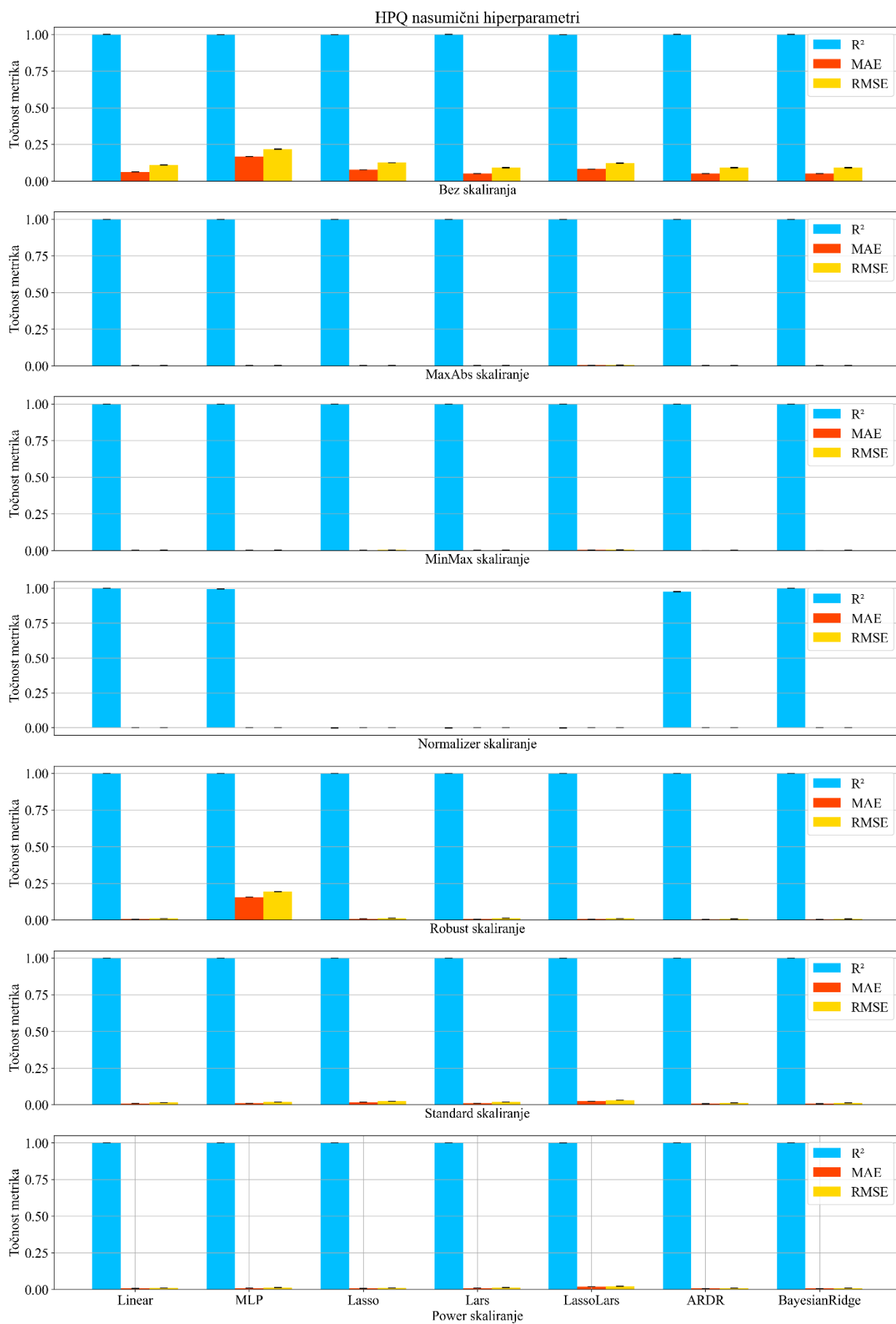


Slika 5.1 Rezultati modela s optimalnim hiperparametirma za AMD dataset.

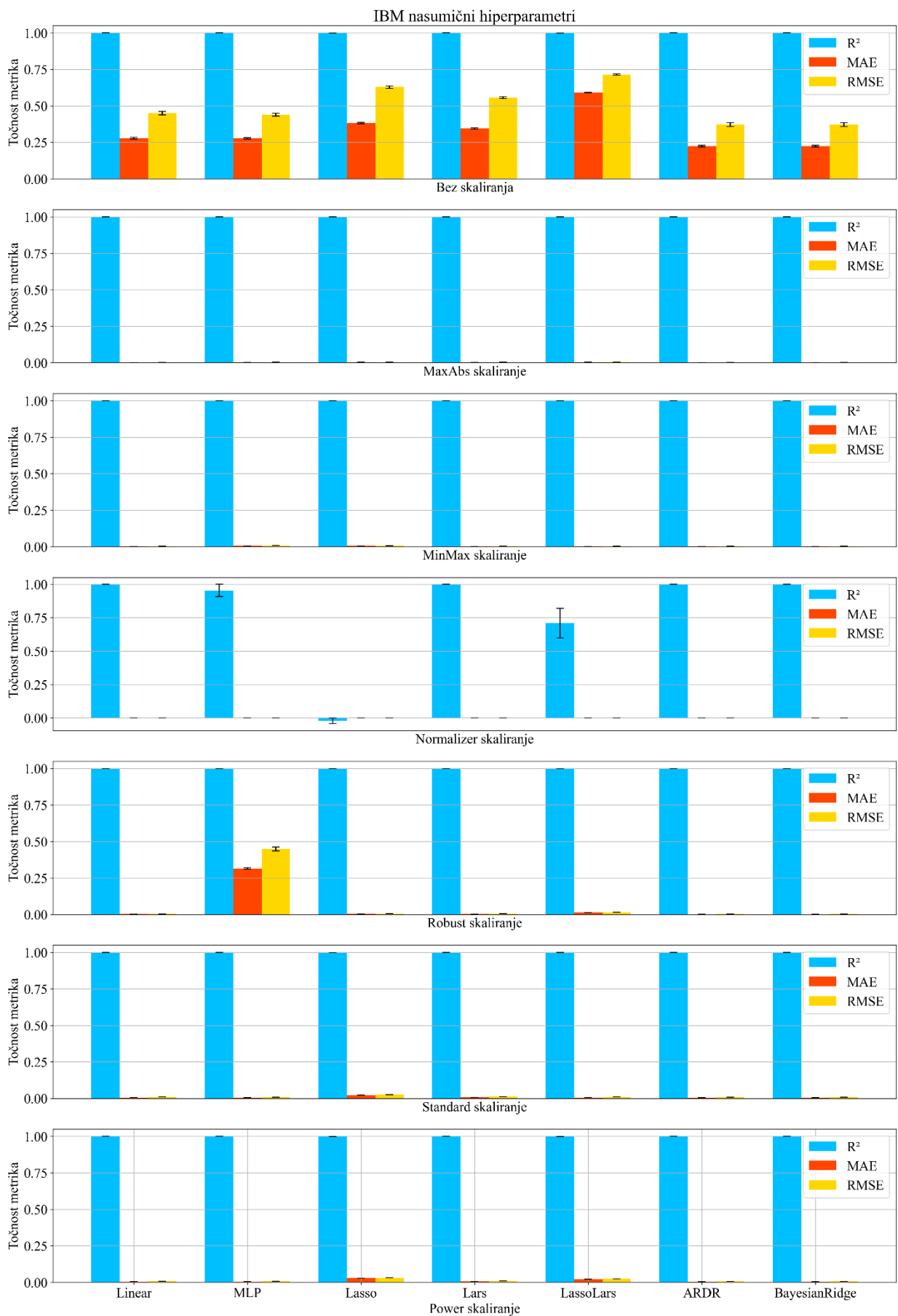


Slika 5.2 Rezultati modela s optimalnim hiperparametirma za AAPL dataset.

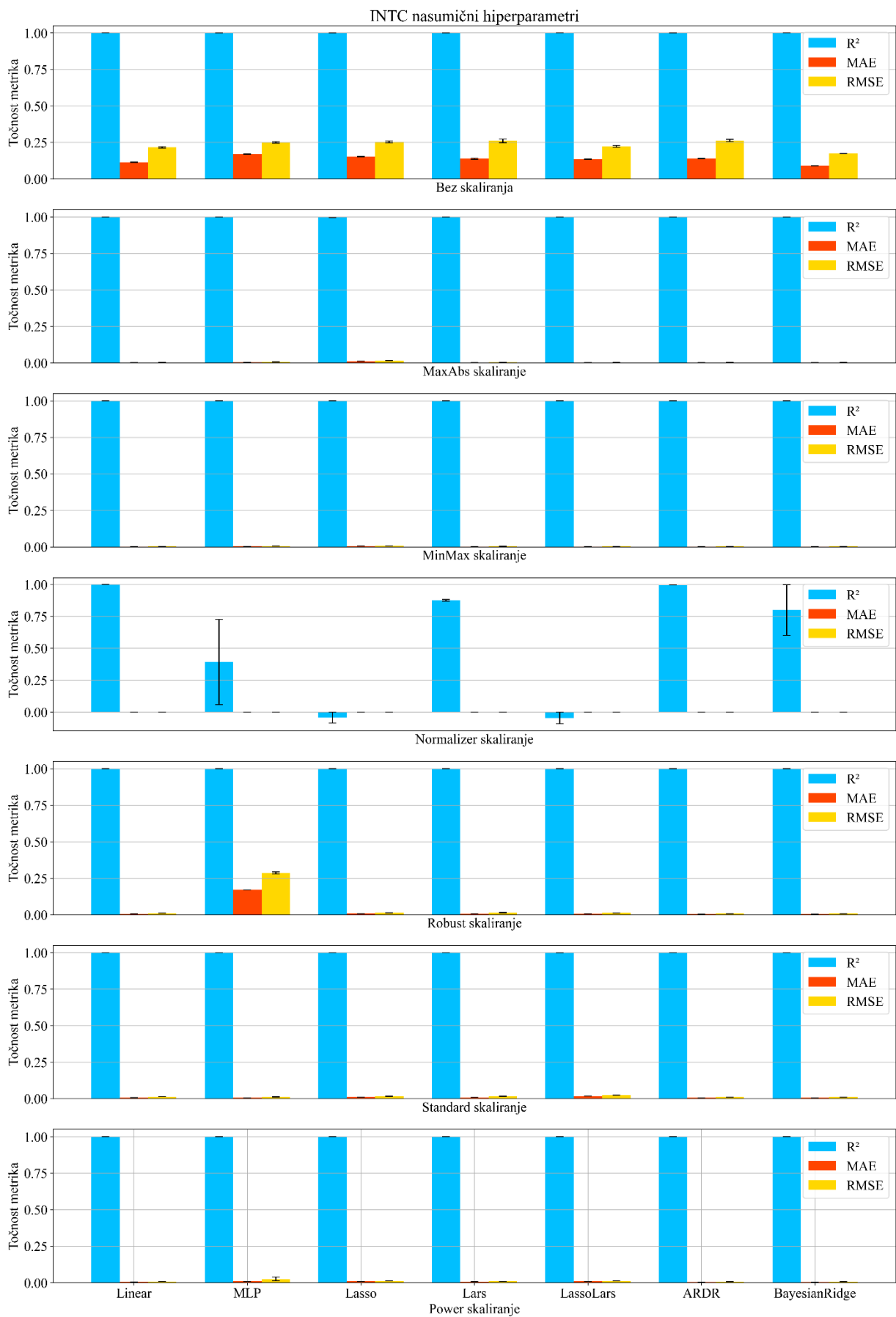




Slika 5.3 Rezultati modela s optimalnim hiperparametirma za HPQ dataset.



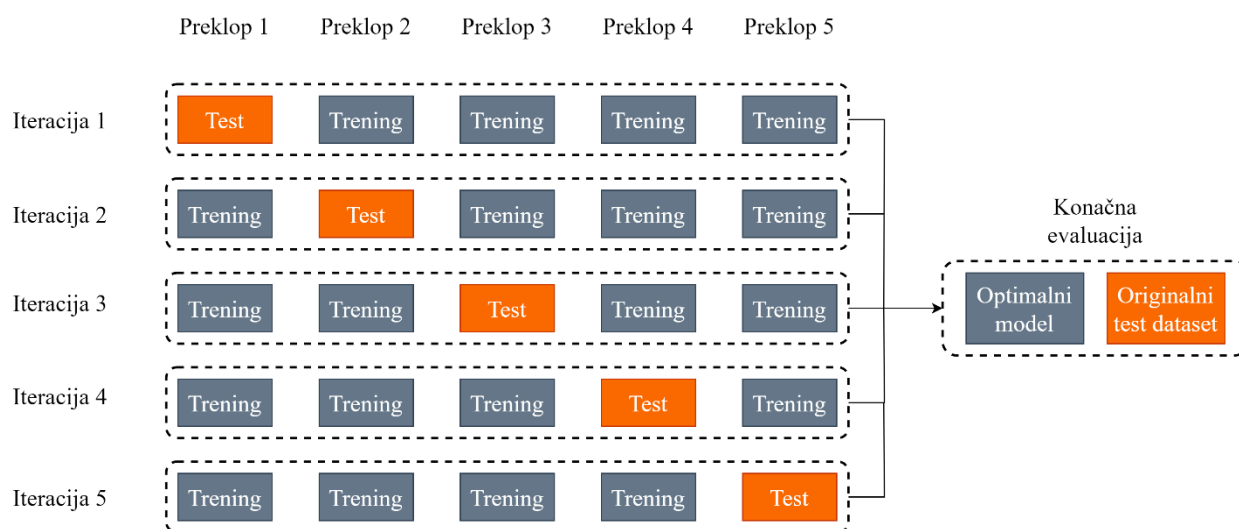
Slika 5.4 Rezultati modela s optimalnim hiperparametirma za IBM dataset.



Slika 5.5 Rezultati modela s optimalnim hiperparametirma za INTC dataset.

## 6. Unakrsna validacija

Unakrsna validacija (engl. Cross Validation) u kontekstu umjetne inteligencije odnosi se na metodu provjere performanse modela. Traženjem optimalnih hiperparametara za svaki dataset veća je šansa da će model patiti od prenaučivosti (engl. Overfitting). To znači da će za ulazne podatke iz trening dataseta imati odlične rezultate, ali da se modelu da neki novi ulazni podatak izlazni rezultat može biti loš. Kako bi se to spriječilo koristit će se k-struka unakrsna validacija (engl. k-folded cross validation). Ona funkcionira tako da se odrede nasumični hiperparametri za model, te se onda uzme dataset za trening i podijeli na k preklopa ( u ovom radu će k biti 5), k-1 od njih se koriste za trening dok se preostali koriste za test. Ovaj postupak se ponavlja k puta te se testni skup pomiče u svakom koraku. Ako su rezultati zadovoljavajući, model se testira na originalnom skupu podataka za testiranje. U suprotnom, proces se ponavlja s novim nasumičnim hiperparametrima. Ovaj proces je prikazan na slici 6.1. Korištenje unakrsne validacije na ovaj način pomaže osigurati da se model dobro prilagođava trenirajućim podacima i da je sposoban donositi valjane odluke kada se suoči s novim, nepoznatim podacima. Korištenjem unakrsne validacije rješava se problem prenaučivosti, povećava se robusnost modela te se poboljšavaju performanse [27].



Slika 6.1 Prikaz principa rada k-struke unakrsne validacije

### 6.1. Rezultati unakrsne validacije

Na slikama (6.2-6.6) prikazani su rezultati modela unakrsne validacije za AMD, APPL, HPQ, IBM i INTC datasete.

Za modele bez skaliranja se može vidjeti da je rezultat  $R^2$  metrike oko 1 što je odlično te da su greške uglavnom niske osim u slučaju MLP-a na svakom datasetu. Maxabs i minmax skaliranja daju odličan  $R^2$  rezultat i minimalne greške na svakom datasetu. Na normalizer skaliranju rezultati  $R^2$  metrike su dobri samo za linearni model na svim datasetima. MLP ima najbolje rezultate na IBM i AMD datasetu s tim da ti rezultati su ispod 0.5 tako da nisu dobri, a na ostalim datasetima su rezultati ispod 0. Lasso daje dobar rezultat jedino za AMD dataset, ostali su loši. Lars ima dobar rezultat na IBM datasetu te donekle dobar rezultat od 0.7 na INTC datasetu, za ostale datasete rezultat je loš. LassoLars ima loš rezultat na svim datasetovima. ARDR i BayesianRidge imaju međusobno slične rezultate s time da je BayesianRidge malo lošiji. Imaju dobre rezultate za AMD i IBM te malo gore za HPQ i INTC, a za AAPL dataset imaju rezultate malo više od 0.5 što nije zadovoljavajuće. RMSE i MAE greške su niske na svakom modelu za sve datasete. Na robust skaliranju su  $R^2$  rezultati odlični te ima manja greška na Lasso modelu za AAPL dataset. Pri standard i power skaliranju  $R^2$  metrika ima odličan rezultat te su sve greške minimalne.

Tablice s hiperparametrima su dane u dodatku. Tablice (A.8-A.14, A.36-A.42, A.64-A.70, A.92-A.98, A.120-A.126) prikazuju hiperparametre za AMD, AAPL, HPQ, IBM i INTC modele unakrsne validacije.

U njima se vidi da za modele linearne regresije hiperparametri su uglavnom isti.

Za MLP se vidi da ima velik broj neurona po sloju dok je broj slojeva veći u usporedbi s modelima s optimalnim hiperparametrima te su ovdje veće sličnosti. Najčešće aktivacijske funkcije su identity i relu s tim da se i tanh pojavljuje u malo manjem broju. Solver je dominantno lbfgs kao i kod modela s optimalnim hiperparametrima. Alfe su sve istog reda veličine. Max\_iter je tipično istog reda veličine, ali nema prevelikih sličnosti. Tol su većinom slične.

Za Lars modele eps ima slične vrijednosti. N\_nonzero\_coefs su slične default vrijednostima. Jitter uvijek ima default odnosno none vrijednost.

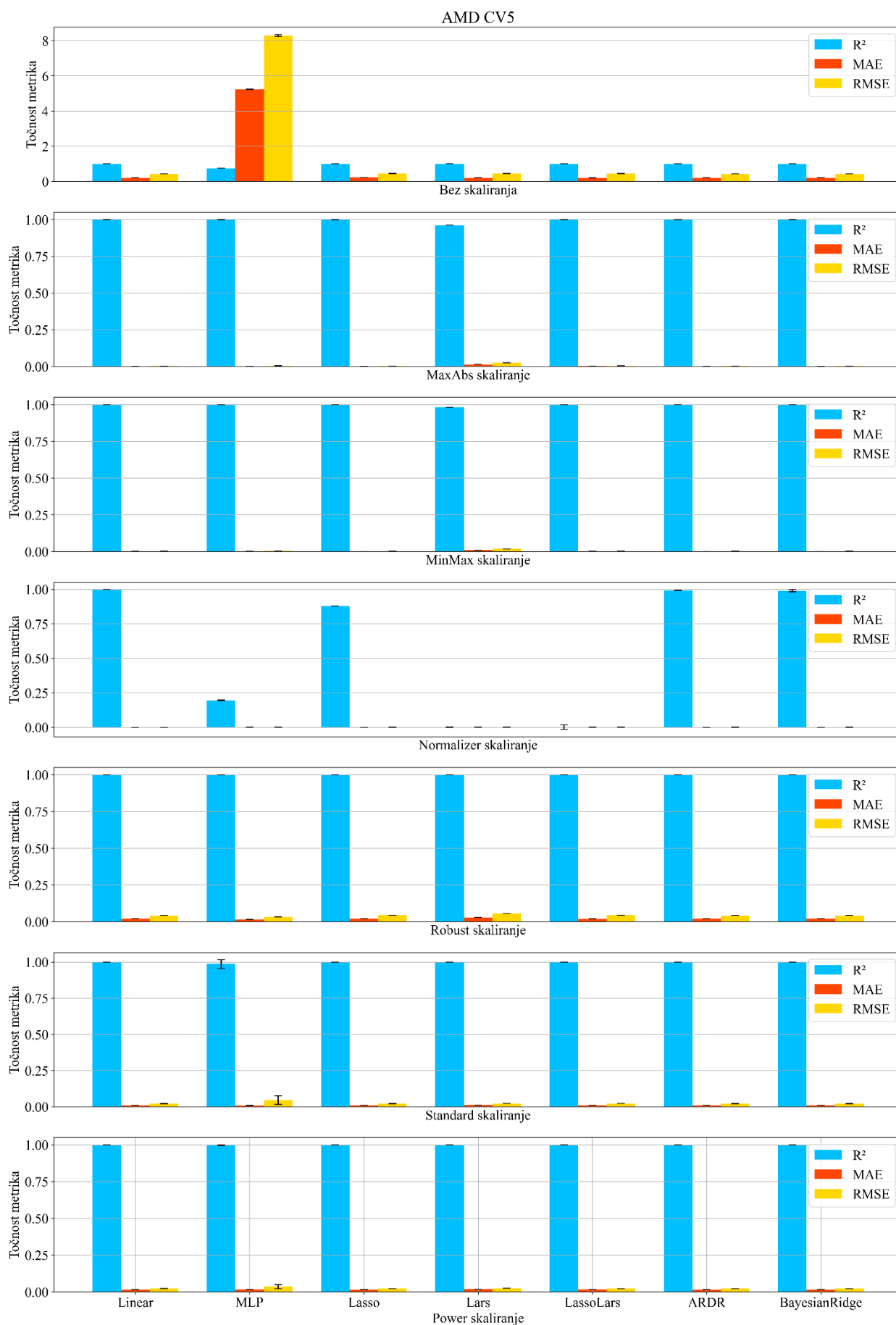
Pri Lasso alfa ima sličnu vrijednost kroz metode skaliranja i normalizacije na različitim datasetovima. Max\_iter je značajno veći od defaulta na svim modelima. Warm\_start i positive nemaju sličnosti između modela. Tol je sličan za sve modele.

Za LassoLars modele su svi hiperparametri slični u usporedbi s različitim modelima s tim da je alfa puno manja od svoje default verzije, a max\_iter značajno veći kao i kod modela s optimalnim hiperparametrima.

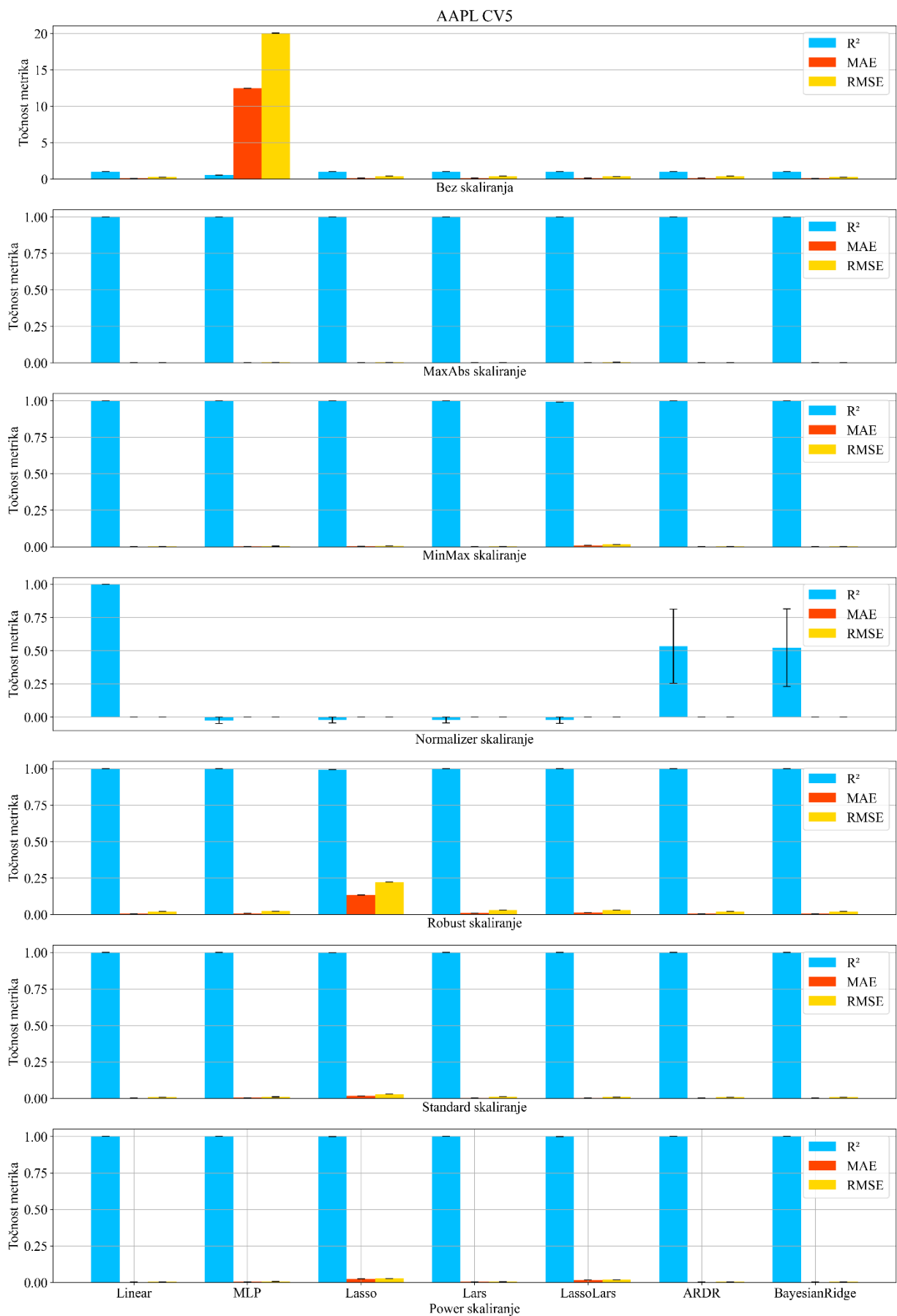
Kod ARD regresije vrijednosti su uglavnom slične za lambde i alfe s tim da imaju manje vrijednosti od defaulta. Tol je također manji od defaultne vrijednosti što bi moglo značiti da je

modelu bilo lakše doći do dobrih rezultata.  $N\_iter$  ima vrijednosti između 300 i 4000, ali nema nekih sličnosti između modela i metoda skaliranja i normalizacije.

Za BayesianRidge regresiju je kao i kod ARD regresije  $\alpha$ ,  $\lambda$  i  $\tau$  imaju slične vrijednosti između različitih modela, ali su vrijednosti niže od defaultnih. Slično je i za  $n\_iter$  gdje su sve vrijednosti u rasponu od 300 do 2000, ali se ne vidi neka sličnost između modela.

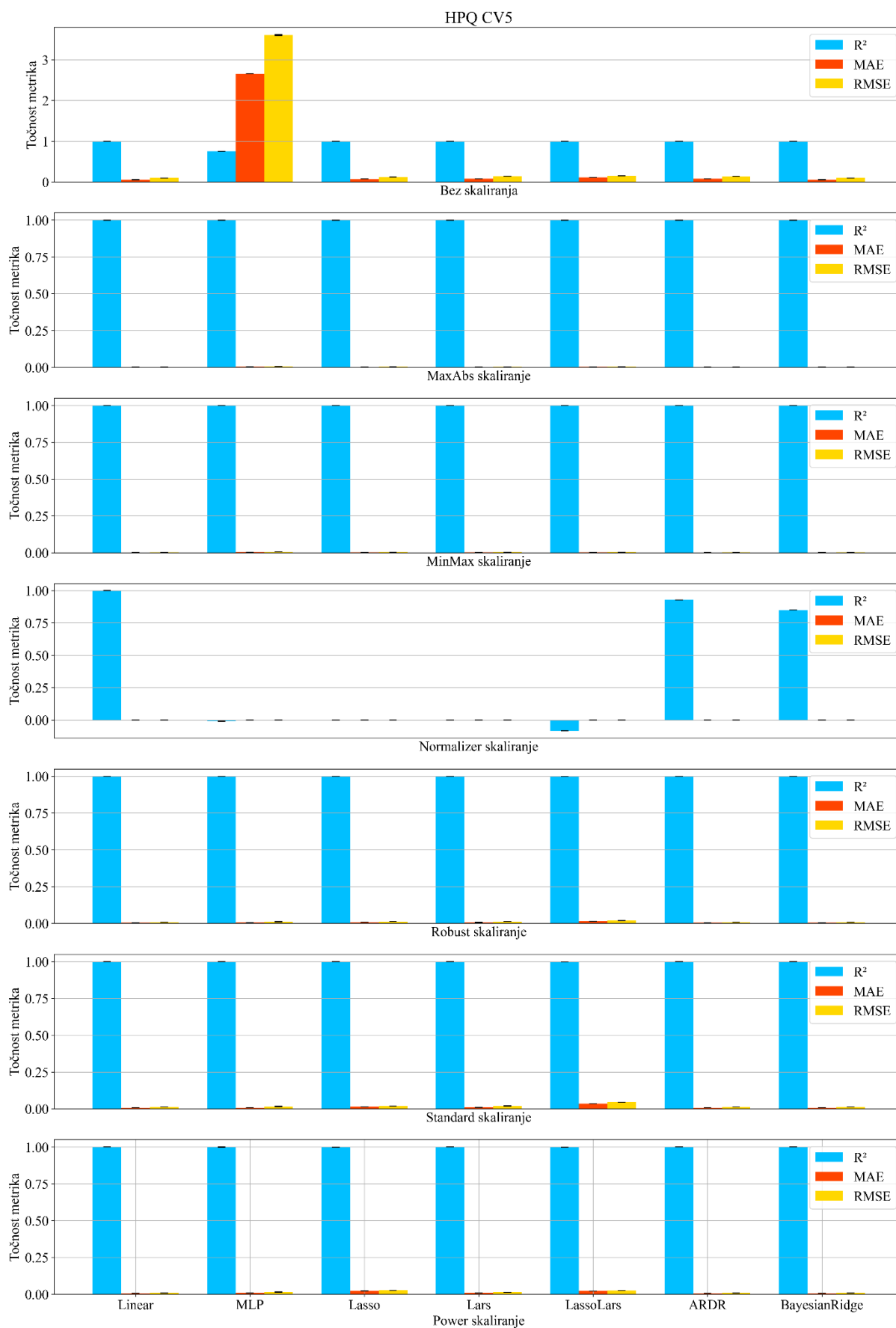


Slika 6.2 Rezultati modela s unakrsnom validacijom za AMD dataset.

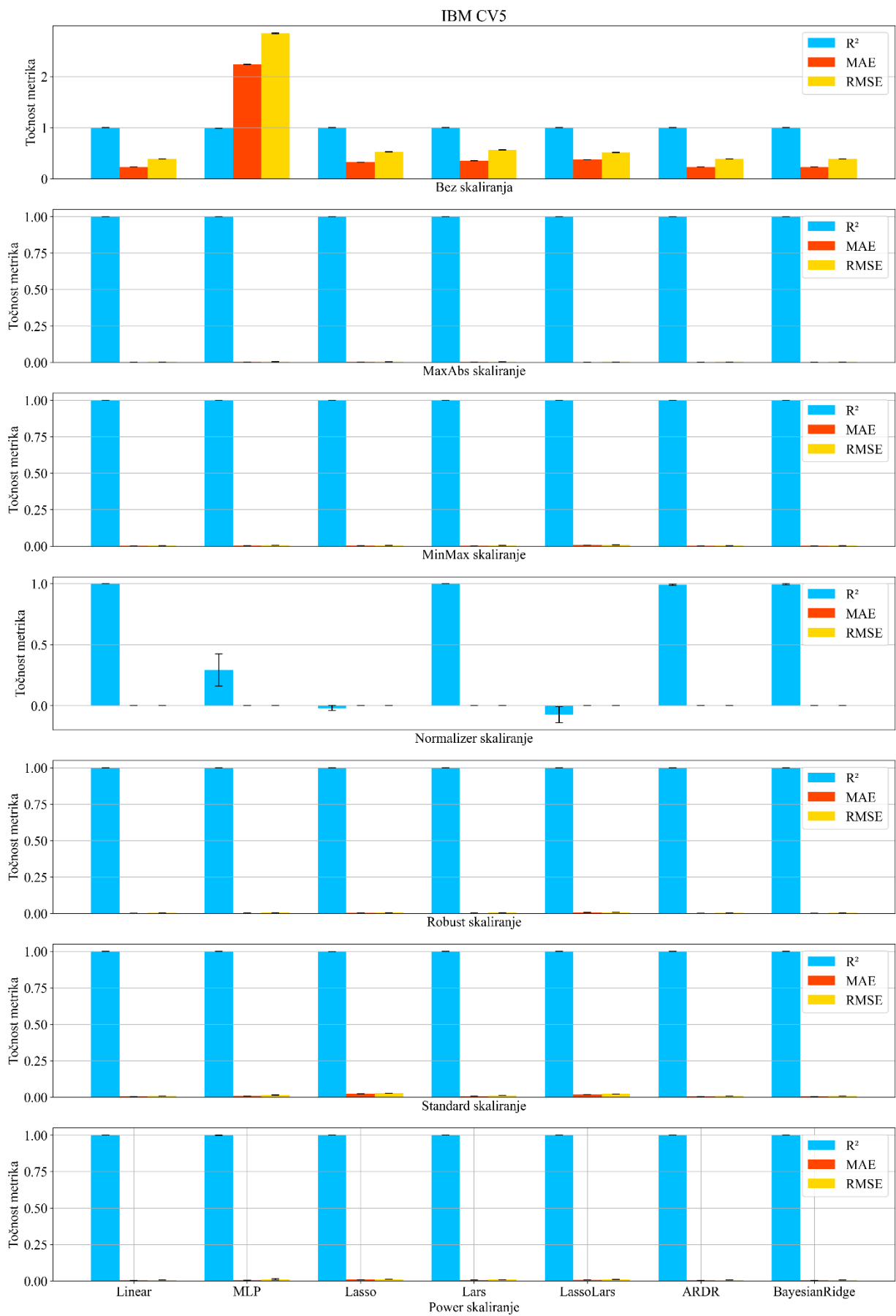


Slika 6.3 Rezultati modela s unakrsnom validacijom za AAPL dataset.

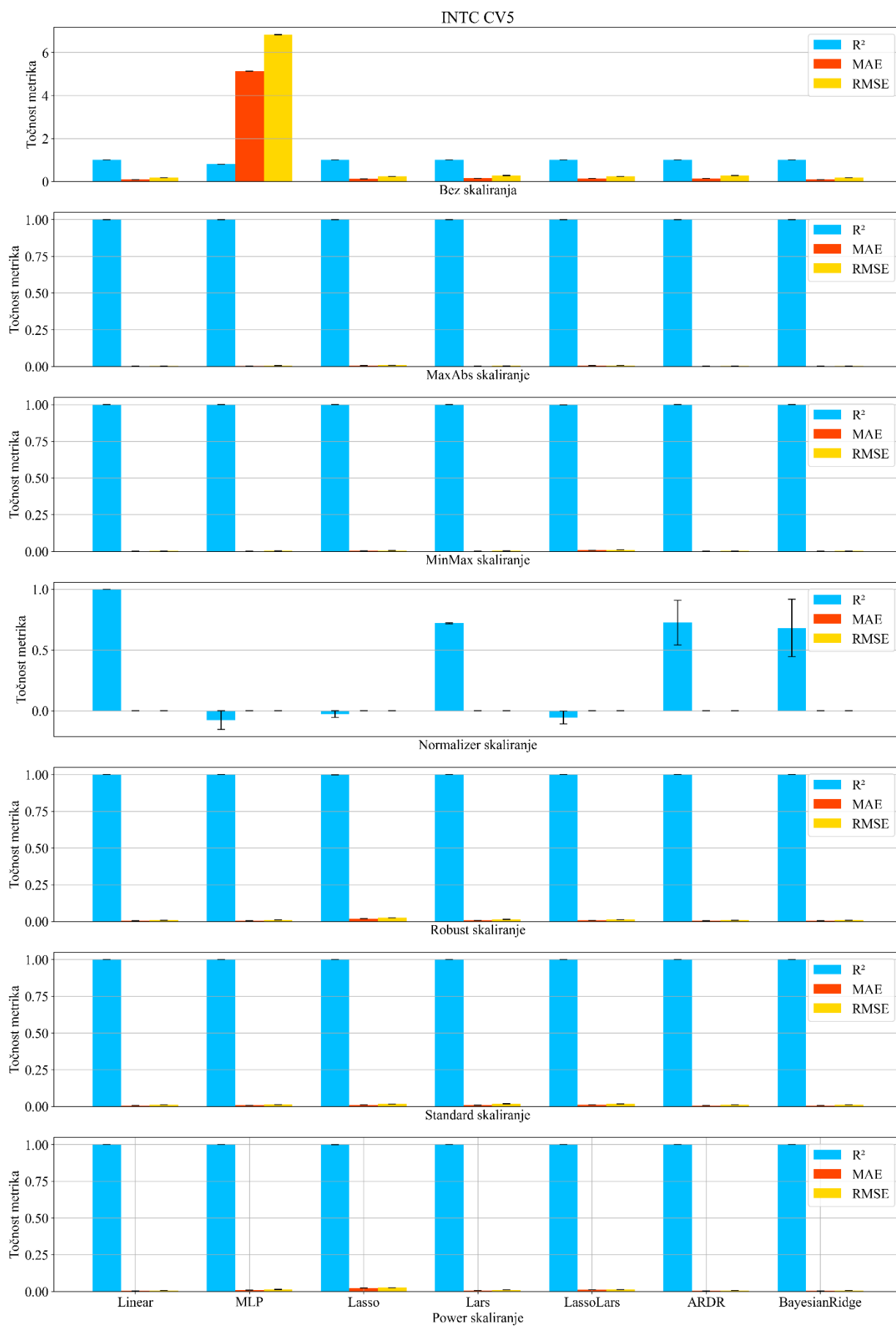




Slika 6.4 Rezultati modela s unakrsnom validacijom za HPQ dataset.



Slika 6.5 Rezultati modela s unakrsnom validacijom za IBM dataset.



Slika 6.6 Rezultati modela s unakrsnom validacijom za INTC dataset.

## 7. Ansambl metode

Cilj ansambl metoda je spojiti predikcije više modela kako bi se poboljšala robustnost. Kombinacijom više modela smanjuje se utjecaj šuma i varijabilnosti individualnih modela. Ansambl metode se generalno dijele na bagging, boosting i stacking[10].

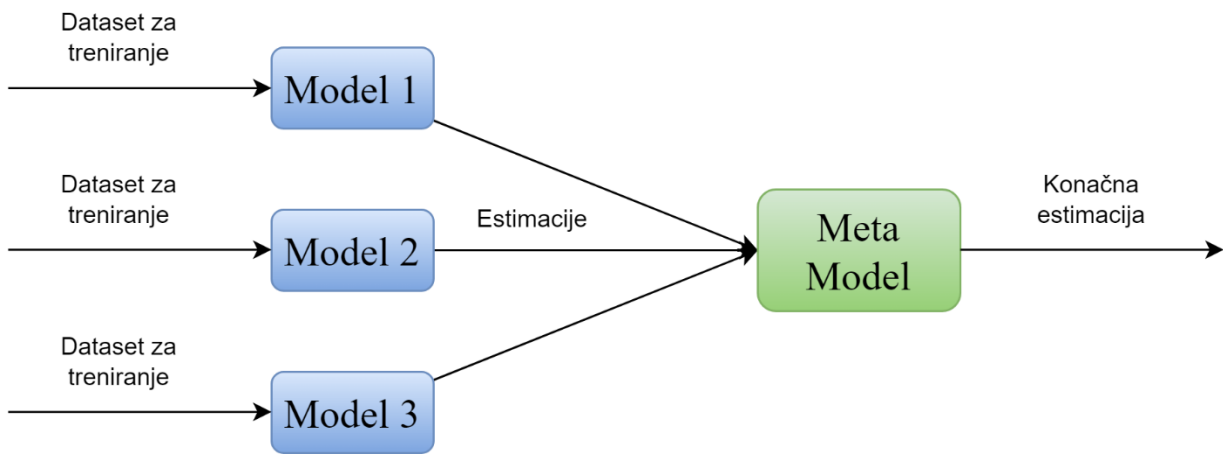
Kod averaging metoda koriste se više modela estimatora nezavisno te se njihova predviđanja usrednjuju. Na ovaj način se smanjuje varijanca te bi se trebao dobiti bolji rezultat od individualnih modela.

Kod boosting metoda modeli se koriste serijski te je cilj svakog modela poboljšati rezultat prijašnjeg. Na taj način se iz više slabijih modela stvara jedan snažan model.

Za razliku od boosting metode stacking metoda paralelno trenira više modela te se na kraju stvara novi model koji koristi prijašnje modele kao ulaze.

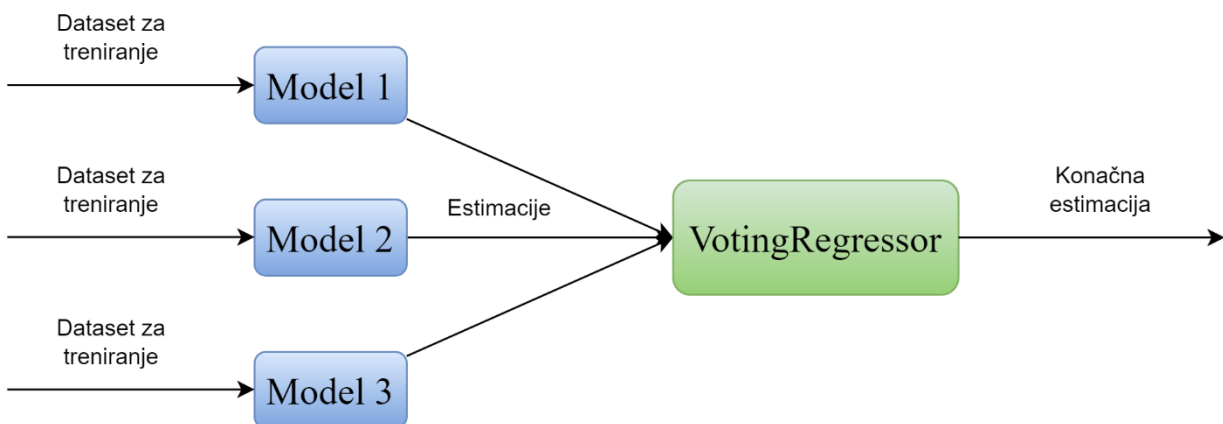
U ovom radu će se koristiti StackingRegressor [28] iz stacking skupine i VotingRegressor [29] iz bagging skupine.

StackingRegressor funkcionira tako da se skupi više običnih modela treniranih sa različitim algoritmima. U ovom radu će se za to koristiti modeli s nasumično pronađenim hiperparametrima i unakrsnom validacijom. Za slučaj da početni modeli nemaju dobre rezultate pokušava se ponovno s novim hiperparametrima. U sljedećem koraku se stvara meta model koji će koristiti estimacije prijašnjih modela da napravi konačnu estimaciju. Meta model je u ovom slučaju RidgeCV regresor. Meta model se također naziva level-1 model dok se početni modeli nazivaju level-0 modeli [31]. Meta model koristi prijašnje estimacije kao dataset za trening te se njegove estimacije evaluiraju pomoću testnog dataseta. Svrha meta modela je da odredi utege ulaznim estimacijama i da nađe njihovu optimalnu kombinaciju za konačnu estimaciju što dovodi do boljeg rezultata. Zbog korištenja više različitih ulaznih modela smanjuje se rizik overfittinga [30]. Cilj StackingRegressor je stvoriti novi model koji daje bolje rezultate od bilo kojeg početnog modela. Shema principa rada StackingRegressor je prikazana na slici 7.1.



Slika 7.1 Princip rada StackingRegressora

Za VotingRegressor je također potrebno imati početne modele treniranim na različitim algoritmima. VotingRegressor funkcioniira tako da uzme srednju vrijednost estimacija svih početnih modela [32]. Za razliku od StackingRegressora ovdje nema težina te zbog toga svaki model jednako pridonosi rezultatu. To također znači da će prisutnost loših modela narušavati rezultat te je najbolje da svi modeli daju slične rezultate [31]. Još jedna od razlika s StackingRegressorom je to da model VotingRegressor neće nužno dati bolje rezultate od bilo kojeg početnog modela. Shema principa rada VotingRegressor je prikazana na slici 7.2.



Slika 7.2 Princip rada VotingRegressora

## 7.1. Rezultati StackingRegressor ansambl metode

Na slikama (7.3-7.7) prikazani su rezultati stackingregressora za AMD, APPL, HPQ, IBM i INTC datasete.

Rezultati  $R^2$  metrike bez skaliranja su svi približno jedan na svim datasetovima, a greške relativno niske. Za skaliranja maxabs, minmax, robust, standard i power dobivaju se idealni rezultati gdje je  $R^2$  metrika približno jedan te greške MAE i RMSE su zanemarive. Pri normalizer skaliranju na svim datasetovima greške su niske osim u slučaju AAPL dataseta gdje RMSE greška iznosi 33% od ciljane vrijednosti. Rezultati  $R^2$  metrike varijaju, na AMD datasetu rezultat je 0.04 što je nisko, na AAPL datasetu rezultat je praktički nula što je još gore. Za HPQ, IBM i INTC datasetove rezultat varira od 0.15 do 0.2, ali zato je varijanca velika.

Tablice s hiperparametrima su dane u dodatku. Tablice (A.15-A.21, A.43-A.49, A.71-A.79, A.99-A.105, A.127-A.133) prikazuju hiperparametre za AMD, AAPL, HPQ, IBM i INTC modele StackingRegressora.

U njima se vidi da za modele linearne regresije hiperparametri su uglavnom isti.

Za MLP se vidi da ima veći broj slojeva i broj neurona za HPQ i AAPL nego prije, dok su ostali modeli sličniji modelima unakrsne validacije. Najčešće aktivacijske funkcije su identity i relu s tim da se tanh i logistic pojavljuje u malo manjem broju. Solver je dominantno lbfgs kao i kod prijašnjih modela. Alfe su sve istog reda veličine. Max\_iter je tipično istog reda veličine te su vrijednosti tipično veće od defaultnih. Tol su većinom slične.

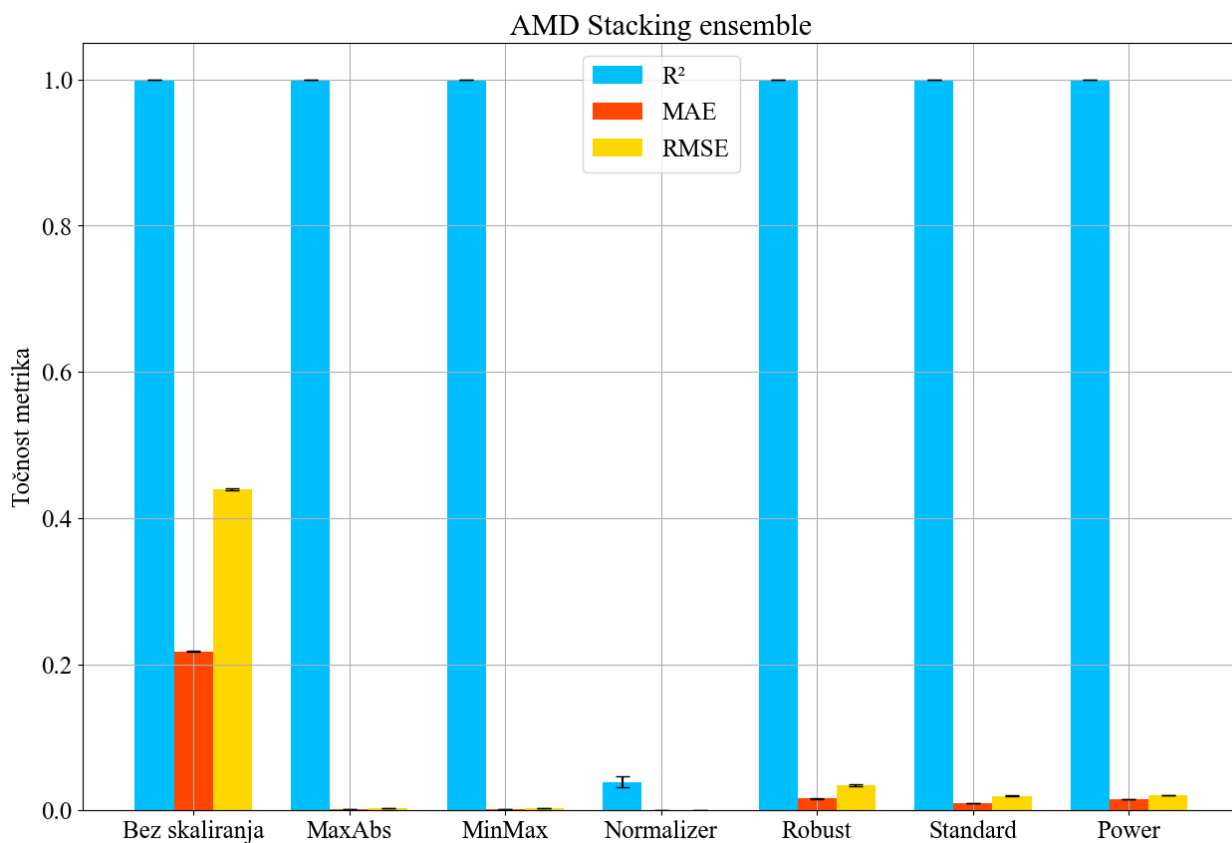
Za Lars modele kao i prije eps ima slične vrijednosti. N\_nonzero\_coefs su slične default vrijednostima. Jitter uvijek ima default odnosno none vrijednost.

Pri Lasso alfa ima sličnu vrijednost kroz metode skaliranja i normalizacije na različitim datasetovima. Max\_iter je značajno veći od defaulta na svim modelima. Warm\_start i positive nemaju sličnosti između modela. Tol je sličan za sve modele.

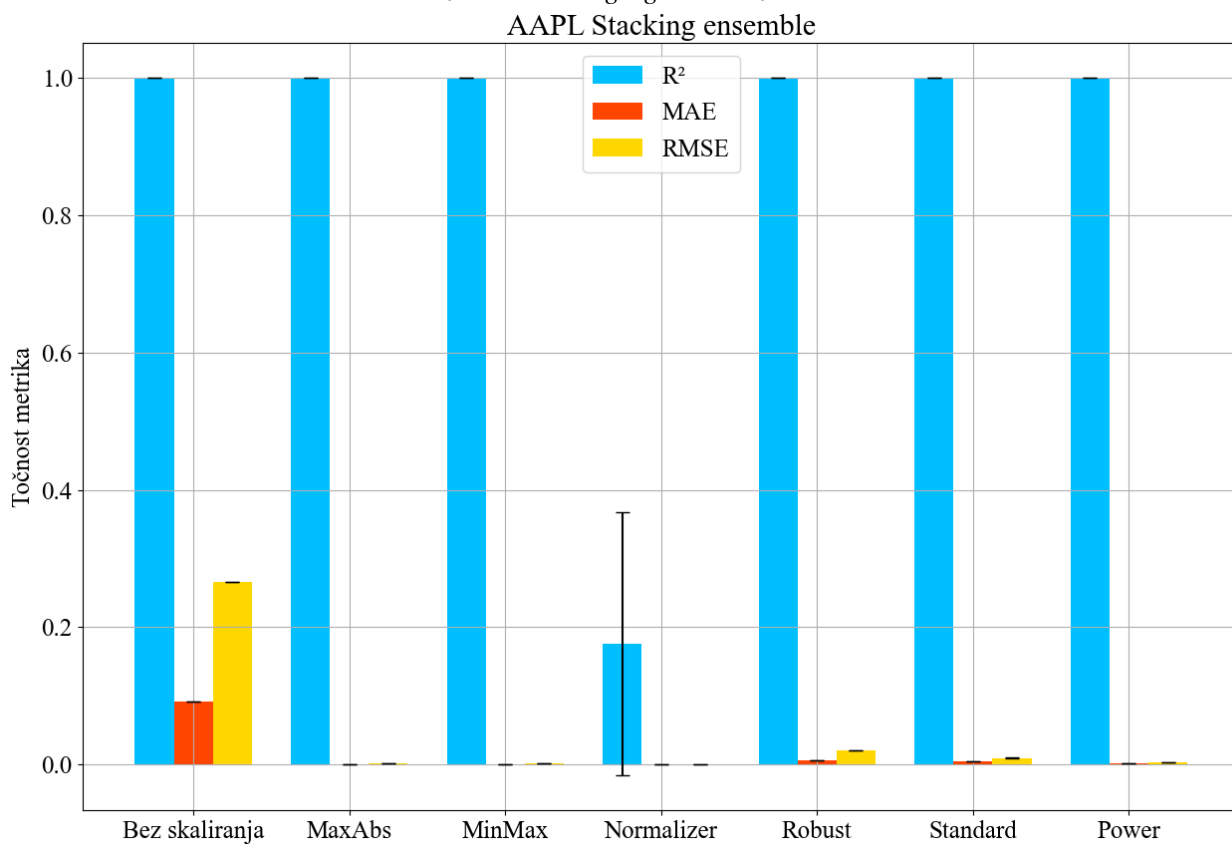
Za LassoLars modele su svi hiperparametri slični u usporedbi s različitim modelima s tim da je alfa puno manja od svoje default verzije, a max\_iter značajno veći kao i kod modela s optimalnim hiperparametrima. Positive se čini nasumičan, ali to je vjerojatno zbog različitih metoda skaliranja i normalizacije te zbog razlika između dataseta koje utječu na to da li će Close samo rasti ili će zbog prije navedenih razloga padati što bi moglo utjecati na korisnost positive hiperparametra.

Kod ARD regresije vrijednosti su uglavnom slične za  $\lambda$  i  $\alpha$  s tim da imaju manje vrijednosti od defaulta.  $\lambda_{\text{threshold}}$  je tipično viši u usporedbi s modelima unakrsne validacije.  $\text{Tol}$  je također manji od defaultne vrijednosti što bi moglo značiti da je modelu bilo lakše doći do dobrih rezultata.  $N_{\text{iter}}$  ima vrijednosti u sličnom rasponu kroz sve modele, ali nema većih sličnosti.

Za BayesianRidge regresiju je kao i kod ARD regresije  $\alpha$ ,  $\lambda$  i  $\text{Tol}$  imaju slične vrijednosti između različitih modela, ali su vrijednosti niže od defaultnih. Slično je i za  $n_{\text{iter}}$  gdje su sve vrijednosti u sličnom rasponu, ali se ne vidi sličnost između modela.

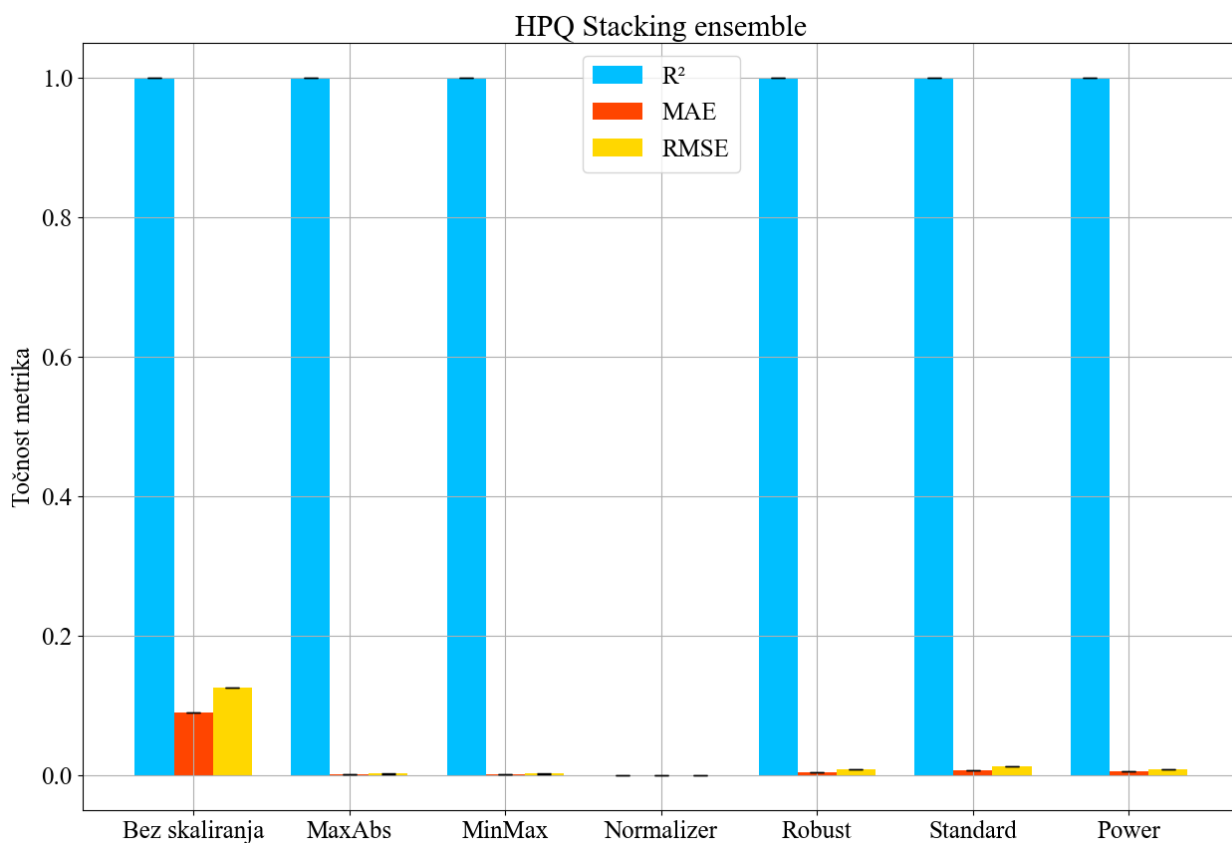


Slika 7.3 Rezultati stackingregressora za AMD dataset.

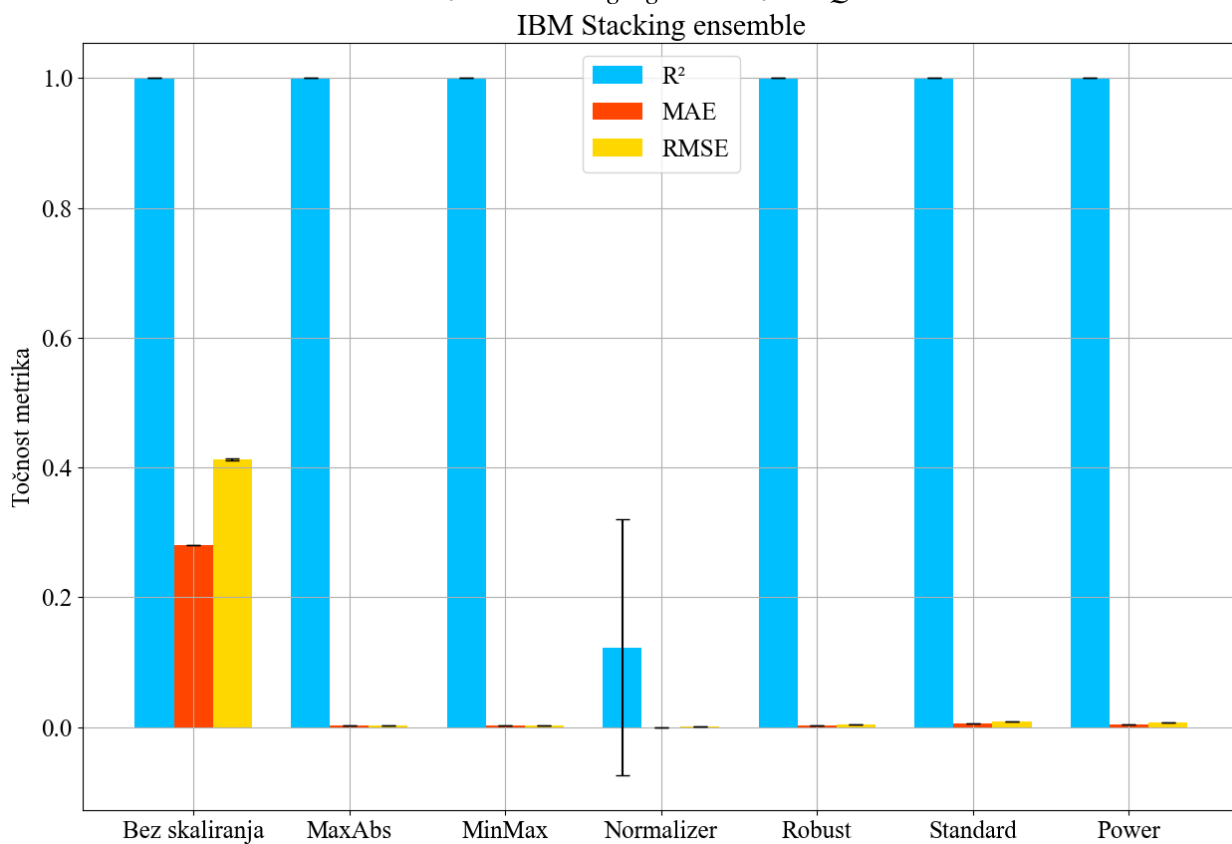


Slika 7.4 Rezultati stackingregressora za AAPL dataset.

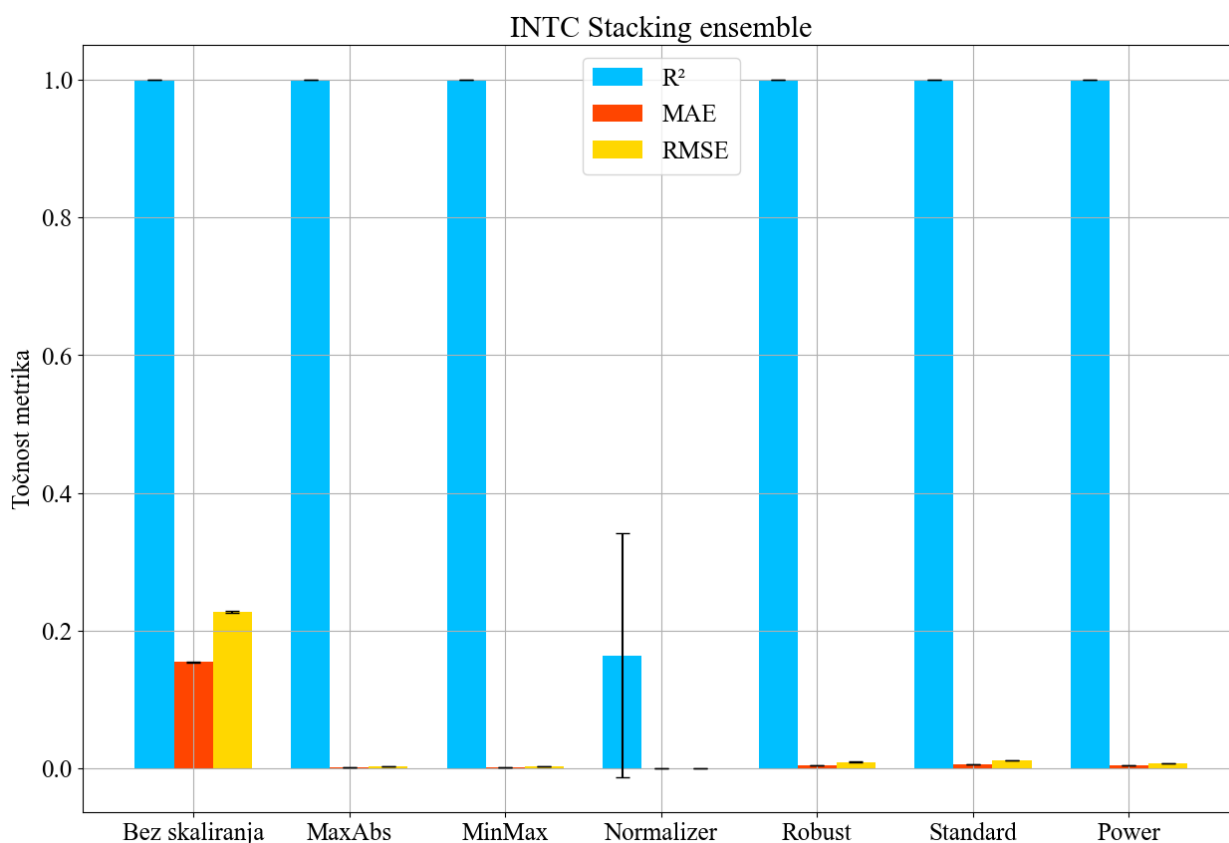




Slika 7.5 Rezultati stackingregressora za HPQ dataset.



Slika 7.6 Rezultati stackingregressora za IBM dataset.



Slika 7.7 Rezultati stackingregressora za INTC dataset.

## 7.2. Rezultati VotingRegressor ansambl metode

Na slikama (7.8-7.12) prikazani su rezultati votingregressora za AMD, APPL, HPQ, IBM i INTC datasete.

Bez skaliranja R<sup>2</sup> metrika daje dobre rezultate na svim datasetovima. Što se tiče MAE i RMSE grešaka jedino na HPQ datasetu su manje od jedan. Na maxabs, minmax, robust, standard i power skaliranjima R<sup>2</sup> metrika daje niže rezultate u usporedbi sa stackingregressorom, ali samo po sebi su zadovoljavajuće visoki. Za maxabs i mimax su greške poprilično niske dok su za power, standard i robust skaliranja veće, to je zato jer bi oni tipično imali neke sitne greške, ali zbog načina rada votingregressor ansambla one se ne mogu smanjiti jer nema boljeg rezultata. Normalizer skaliranje na svim datasetovima ima sitne RMSE i MAE greške s tim da kod AAPL dataseta se pojavljuje veća greška kao i kod stackingregressora. Rezultati R<sup>2</sup> metrike su bolji u usporedbi s stackingregressorom jer se uzimaju srednje vrijednosti svih modela. IBM dataset ima najbolje rezultate s rezultatom od 0.75 dok su ostali datasetovi od 0.4 do 0.5 s time da AMD i INTC imaju velike varijance.

Tablice s hiperparametrima su dane u dodatku. Tablice (A.22-A.28, A.50-A.56, A.78-A.84, A.106-A.112, A.134-A.140) prikazuju hiperparametre za AMD, AAPL, HPQ, IBM i INTC modele VotingRegressora.

U njima se vidi da za modele linearne regresije hiperparametri su uglavnom isti.

Za MLP se vidi da ima veći broj slojeva i broj neurona za INTC dok su modeli ostalih datasetova isti kao i prijašnji. Najčešće aktivacijske funkcije su identity i relu za INTC i IBM, AMD ima pretežno identity, AAPL pretežno logistic, a HPQ ima mješavinu relu, tanh i logistic. Solver je dominantno lbfgs kao i kod prijašnjih modela. Alfe su sve istog reda veličine. Max\_iter je tipično istog reda veličine te su vrijednosti tipično veće od defaultnih. Tol su većinom slične.

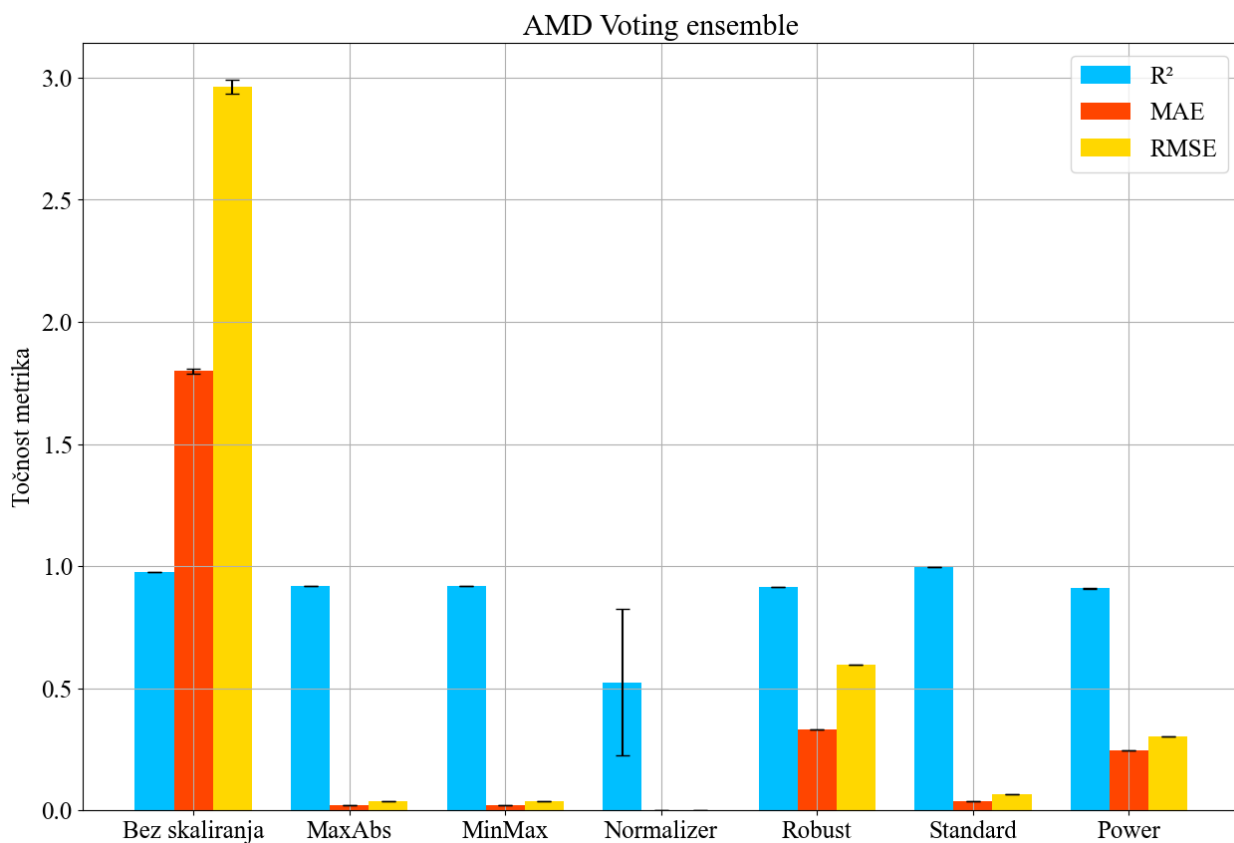
Za Lars modele kao i prije eps ima slične vrijednosti. N\_nonzero\_coefs su slične default vrijednostima. Jitter uvijek ima default odnosno none vrijednost.

Pri Lasso alfa ima sličnu vrijednost kroz metode skaliranja i normalizacije na različitim datasetovima. Max\_iter je značajno veći od defaulta na svim modelima. Warm\_start i positive nemaju sličnosti između modela. Tol je sličan za sve modele.

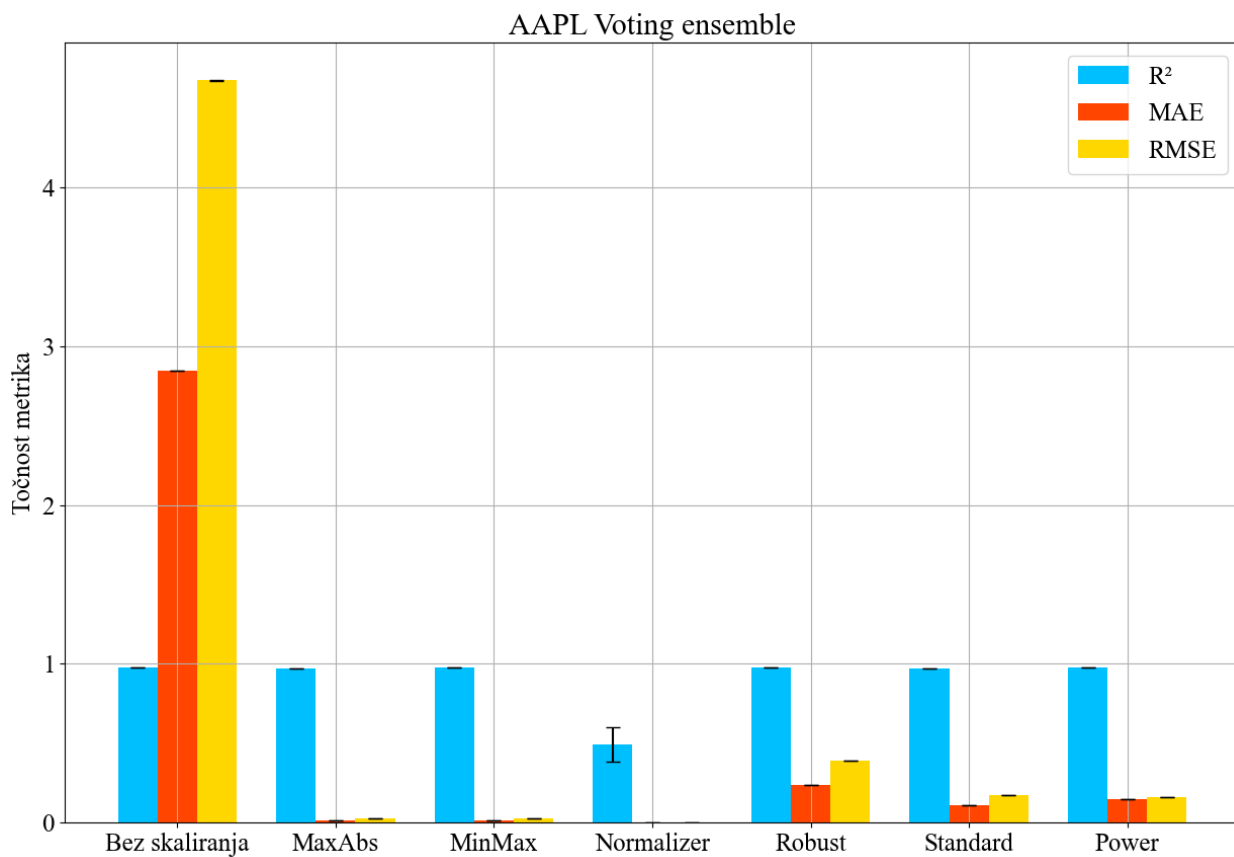
Za LassoLars modele su svi hiperparametri slični u usporedbi s različitim modelima s tim da je alfa puno manja od svoje default verzije, a max\_iter značajno veći. Positive se i dalje čini nasumičan.

Kod ARD regresije vrijednosti su uglavnom slične za lambde i alfe s tim da imaju manje vrijednosti od defaulta. Lambda\_threshold je tipično viši u usporedbi s modelima unakrsne validacije. Tol je također manji od defaultne vrijednosti što bi moglo značiti da je modelu bilo lakše doći do dobrih rezultata. N\_iter ima vrijednosti u sličnom rasponu kroz sve modele, ali nema većih sličnosti.

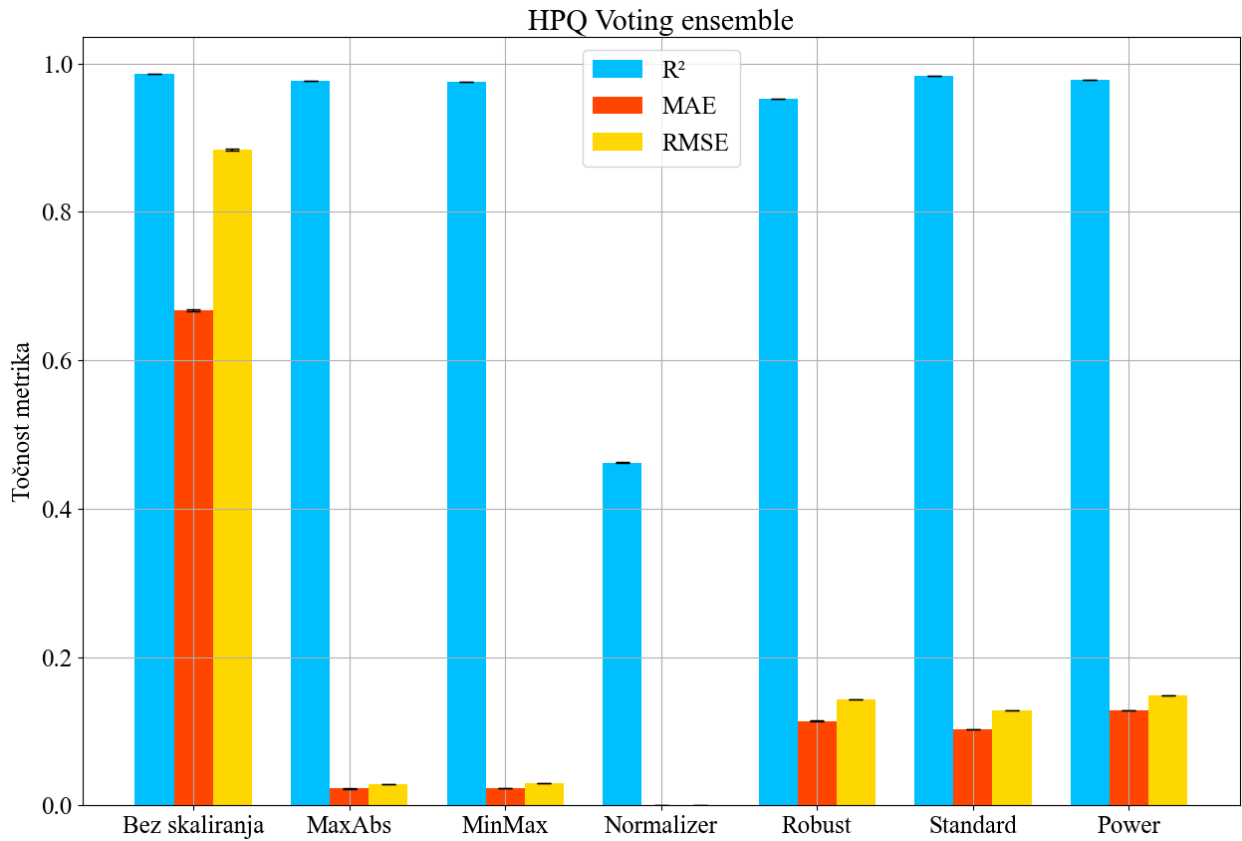
Za BayesianRidge regresiju je kao i kod ARD regresije alfe, labmde i tol imaju slične vrijednosti između različitih modela, ali su vrijednosti niže od defaultnih. Slično je i za n\_iter gdje su sve vrijednosti u sličnom rasponu, ali se ne vidi sličnost između modela.



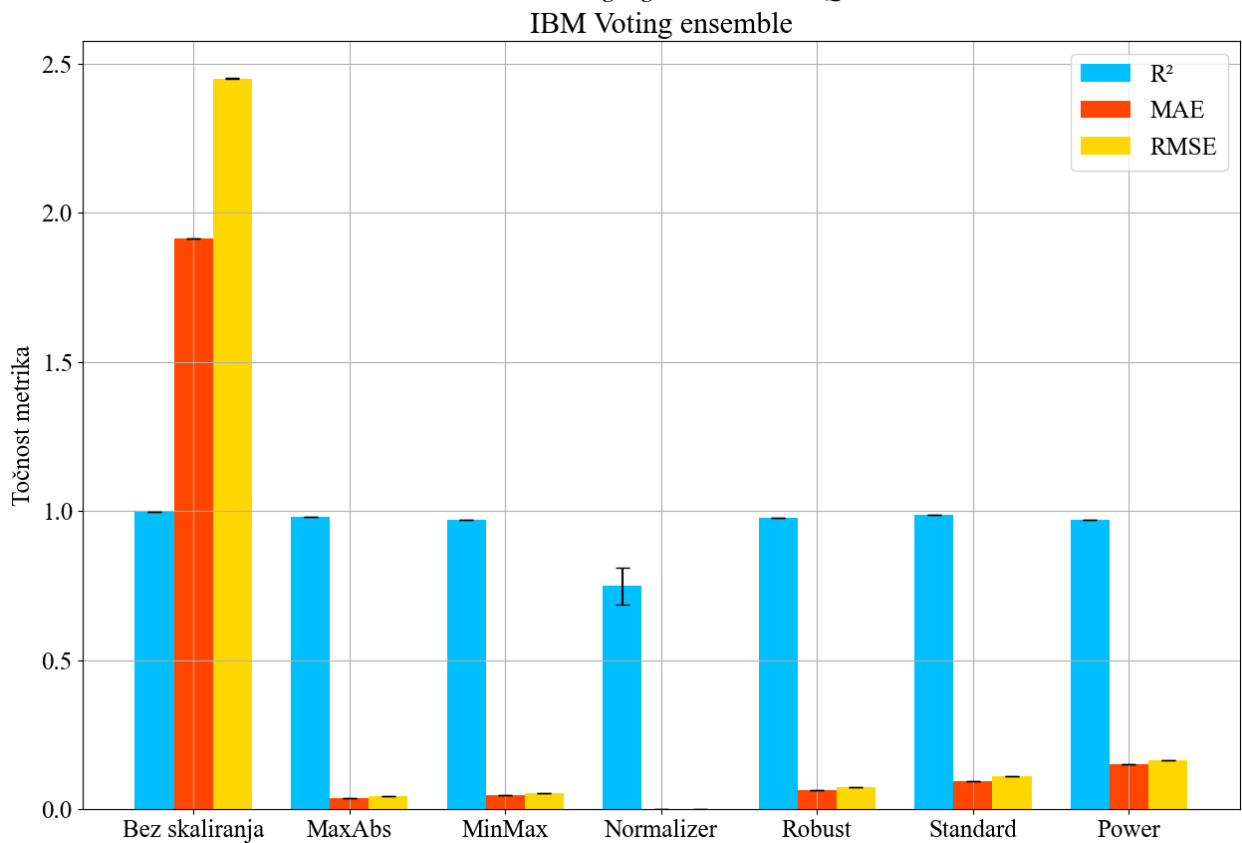
Slika 7.8 Rezultati votingregressora za AMD dataset.



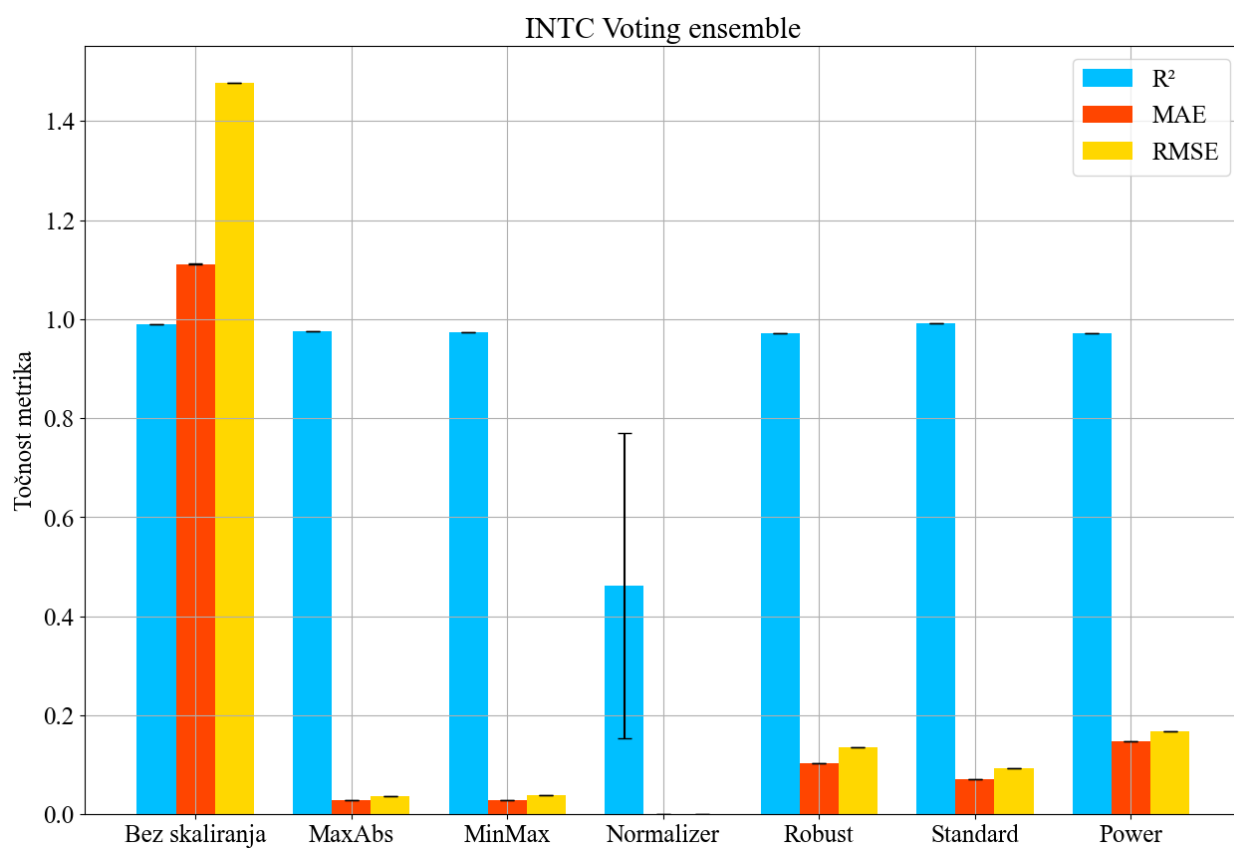
Slika 7.9 Rezultati votingregressora za AAPL dataset.



Slika 7.10 Rezultati votingregressora za HPQ dataset.



Slika 7.11 Rezultati votingregressora za IBM dataset.



*Slika 7.12 Rezultati votingregressora za INTC dataset.*

## 8. Zaključak

U ovom diplomskom radu primijenjeni su algoritmi strojnog učenja za estimaciju cijene dionica pri zatvaranju burze. U istraživanju je korišteno 5 različitih dataset-ova koje sadrže podatke o dionicama iz S&P 500 indeksa. Broj uzoraka u datasetovima prilagođeni su tako da svaki od datasetova ima jednak broj uzoraka. Korišteni modeli za rješavanje ovog problema su: linearna regresija, višeslojni perceptron, Lars, Lasso, LassoLars, Bayesova grebenasta regresija i ARD regresija. Ispitan je utjecaj različitih metoda skaliranja i normalizacije na točnost estimacije cijene dionica. Estimacije modela su se evaluirale uz pomoć koeficijenta determinacije, srednje apsolutne pogreške i korijena srednje kvadratne pogreške. Prvi modeli prikazuju rezultate s nepromijenjenim hiperparametrima te kako metode skaliranja djeluju na rezultate modela. Nakon toga se rezultat poboljšava tako što se koriste modeli s optimalnim hiperparametrima. Na hiperparametrima istih modela na različitim datasetima može se primjetiti da su neke vrijednosti poprilično slične te bi se moglo zaključiti da one najviše utječu na točnost estimacije. Takvi bi hiperparametri npr. bili alfa i lambda BayesianRidge i ARD regresora, alfa za Lasso modele, solver za MLP modele itd. Takvi modeli mogu biti skloni pretreniranju (engl. overfitting) te su zbog toga napravljeni modeli s unakrsnom validacijom kako bi im se povećala robustnost. Na kraju su korištene ansambl metode kako bi više različitih modela kombinirali u jedan u nadi da će se dobiti najbolji rezultat. Za to su korišteni StackingRegressor i VotingRegressor.

Na početku su postavljene hipoteze te se prema njima mogu izvući sljedeći zaključci:

- Moguće je primjeniti različite algoritme strojnog učenja i dobiti visoku točnost estimacije cijena dionica, ali ne daju svi algoritmi dobar rezultat (MLP i LassoLars).
- Skaliranjem ili normalizacijom se mogu poboljšati rezultati prijašnjih modela, ali u slučaju korištenja normalizera se dobivaju lošiji rezultati.
- Primjenom metode nasumičnog pretraživanja moguće je pronaći optimalne hiperparametre i tako drastično povećati točnost estimacija.
- Modeli na kojima je primijenjena unakrsna validacija imaju točnost estimacije usporednu s modelima s optimalnim hiperparametrima.
- Modeli sa stacking ansamblom imaju bolje rezultate nego prijašnji modeli, ali modeli sa voting ansamblom imaju lošije rezultate osim u slučajevima gdje se koristio normalizer.

Prednost ovog pristupa je jednostavna nadogradnja svakog modela tako da je svaki model bolji nego onaj prije. Primjenom različitih metoda novi modeli su kompleksniji u usporedbi s prijašnjima. Dok bi prvi modeli koristili obične algoritme te bi se onda trenirali i testirali,

modelima nakon se postepeno dodaje kompleksnost uz nasumičan odabir, unakrsnu validaciju i stvaranjem ansambla. Sve korištene dionice su s S&P 500 indeksa te imaju veliku količinu podataka što je pomoglo u točnosti modela.

Nedostatak je što zbog različitih metoda skaliranja i normalizacije točnost estimacija je teško prikazati na grafovima. Zbog prevelikog vremena računanja moralo se smanjiti raspon nasumičnih vrijednosti hiperparametara nekih modela.

Kako bi se za buduće istraživanje poboljšala čitljivost grafova i njihova interpretacija bilo bi dobro da se koristi metrika kao što je srednja apsolutna postotna greška. Nadalje bi bilo dobro na isti princip napraviti modele za dataset dionice koja nije na S&P 500 listi kako bi se testirala robustnost cijelog procesa.



## 9. Literatura

- [1] „Tržišni indeks - pregled, funkcije i primjeri“, s interneta, <https://hr.pharoskc.com/2335-what-is-market-index>, 9.8.2023.
- [2] Strader, T. J., Rozycki, J. J., Root, T. H., Huang, Y. H. J. : „Machine learning stock market prediction studies: Review and research directions“ *Journal of International Technology and Information Management*, 28(4), pp. 63-83. 2020
- [3] Shen, S., Jiang, H., & Zhang, T.: „Stock market forecasting using machine learning algorithms“. *Department of Electrical Engineering, Stanford University, Stanford, CA*, pp- 1-5, 2012.
- [4] Patel, J., Shah, S., Thakkar, P., & Kotecha, K.: „Predicting stock market index using fusion of machine learning techniques“. *Expert Systems with Applications*, 42(4), pp. 2162-2172. 2015
- [5] Hegazy, O., Soliman, O. S., & Salam, M. A.: A machine learning model for stock market prediction. *arXiv preprint arXiv:1402.7351*. 2014
- [6] Parikh R. (2021, May), S&P500 Stock Prices, Version 35, Retrieved May 31, 2021 from <https://www.kaggle.com/datasets/rprkh15/sp500-stock-prices>
- [7] Maćešić , S. :Predavanja Numerička i stohastička matematika
- [8] „Bayesian Ridge Regression“, s interneta , [https://scikitlearn.org/stable/modules/linear\\_model.html#bayesian-regression](https://scikitlearn.org/stable/modules/linear_model.html#bayesian-regression), 10.8.2023
- [9] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. . Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), pp. 2825–2830. 2011
- [10] Khandelwal Yash,“ Ensemble Stacking for Machine Learning and Deep Learning“, s interneta <https://www.analyticsvidhya.com/blog/2021/08/ensemble-stacking-for-machine-learning-and-deep-learning/> , 23.08.2023.
- [11] S interneta <https://libguides.library.kent.edu/SPSS/PearsonCorr> 04.09.2023.
- [12] S interneta <https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/>
- [13] McKinney, W., & others.: „Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference „, (Vol. 445, pp. 51–56). 2010
- [14] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. „Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*“, 2011
- [15] Thomas M.H. Hope: „Chapter 4 - Linear regression“ *Machine Learning* ,Academic Press, pp. 67-81, 2020

- [16] S. Abirami, P. Chitra : „Chapter Fourteen - Energy-efficient edge based real-time healthcare support system“, Advances in Computers, Elsevier, Volume 117, Issue 1, pp. 339-368, 2020
- [17] Harth, S: „Multi-Layer Perceptron Explained: A Beginner's Guide „, s internea <https://www.pycodemates.com/2023/01/multi-layer-perceptron-a-complete-overview.html> 06.08.2023
- [18] „Least Angle Regression (LARS)“, s interneta <https://www.geeksforgeeks.org/least-angle-regression-lars/> 15.08.2023.
- [19] Kumar ,D: “ A Complete understanding of LASSO Regression“, s interneta <https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/> 15.08.2023.
- [20] Efron, B., Hastie, T., Johnstone ,I. : „Least Angle Regression“, Statistics Department, Stanford University, pp. 3-4, 2003
- [21] „Implementation of Bayesian Regression“, s interneta <https://www.geeksforgeeks.org/implementation-of-bayesian-regression/> , 15.08.2023
- [22] Kumar, A.: „MinMaxScaler vs StandardScaler – Python Examples“, s interneta <https://vitalflux.com/minmaxscaler-standardscaler-python-examples/> 15.08.2023.
- [23] „Compare the effect of different scalers on data with outliers“, S interneta: [https://scikit-learn.org/stable/auto\\_examples/preprocessing/plot\\_all\\_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py](https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py) 20.08.2023
- [24] Verma, J.: „How to Normalize Data Using scikit-learn in Python“, s interneta <https://www.digitalocean.com/community/tutorials/normalize-data-in-python> 21.08.2023.
- [25] Turney, S.: “ Coefficient of Determination ( $R^2$ ) | Calculation & Interpretation“, s interneta, <https://www.scribbr.com/statistics/coefficient-of-determination/> , 21.08.2023.
- [26] Cort, J., Matsuura, K.: „Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance“, CLIMATE RESEARCH, Vol:30, pp. 79-82, 2005.
- [27] „Cross-validation: evaluating estimator performance „, s interneta [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html) , 25.08.2023.
- [28] „sklearn.ensemble.StackingRegressor“, s interneta <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html> , 29.08.2023
- [29] „sklearn.ensemble.VotingRegressor“, s interneta <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingRegressor.html> , 29.08.2023
- [30] Soni, B.: „Stacking to Improve Model Performance: A Comprehensive Guide on Ensemble Learning in Python „, s interneta [https://medium.com/@brijesh\\_soni/stacking-to-](https://medium.com/@brijesh_soni/stacking-to-)

[improve-model-performance-a-comprehensive-guide-on-ensemble-learning-in-python-9ed53c93ce28](#) 29.08.2023.

- [31] Witten, I., Eibe, F., Hall, M.: „Data Mining: Practical Machine Learning Tools and Techniques“, Morgan Kaufmann, 2011
- [32] Brownlee, J.: „How to Develop Voting Ensembles With Python“, s interneta <https://machinelearningmastery.com/voting-ensembles-with-python/> , 30.08.2023

## 10. Sažetak

Ovim radom prikazan je način izrade modela strojnog učenja za estimaciju cijene dionica. Objasnjena je obrada podataka prije stvaranje modela te su prikazani rezultati sa nepodešenim modelima. Rezultati su poboljšani korištenjem više metoda skaliranja. Nadalje se svakom modelu pronalaze optimalni hiperparametri kako bi se poboljšali rezultati. Traženje optimalnih hiperparametara može dovesti do pretreniranja za specifičan dataset, kako bi se to spriječilo model se je treniralo koristeći unakrsnu validaciju. Modeli su se onda uz pomoć dvije metode ansambla kombinirali kako bi se dobio bolji rezultat od svakog individualnog modela. Rezultati različitih modela su uspoređeni na svakom koraku. Ovaj pristup razvijanja modela strojnog učenja i poboljšavanja rezultata istih se može primijeniti i na druge datasetove.

Ključne riječi: Strojno učenje, regresija, skaliranje, nasumično traženje hiperparametara, unakrsna validacija, ansambl metode

## 11. Summary

This paper demonstrates how to make a machine learning model for estimating stock prices. Data preprocessing before the creation of the model is explained and the results with unoptimised models are shown. The results are improved using multiple scaling methods. Optimal hyperparameters are found for each model to improve the results. The search for optimal hyperparameters can lead to overfitting for a specific dataset, to prevent that the model was trained using cross-validation. The models were then combined using two ensemble methods as to achieve a better result than each individual model. Results for each model are compared at each step. This approach to developing machine learning models and improving their results can be applied on other datasets as well.

Key words: Machine learning, regression, scaling, random hyperparameter search, cross validation, ensemble methods

## A. Tablice sa hiperparametrima modela

### A.1. AMD dataset

Tablica A.1 Hiperparametri linearnog modela AMD dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	True	True	False	False	True	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.2 Hiperparametri MLP modela AMD dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(35, 92, 98, 57, 59)	(24, 84, 98, 90, 41, 75, 49)	(55, 48, 39, 97, 41, 74, 85, 13, 45)	(97, 64, 38, 96, 62, 10, 87, 63)	(21.)	(82, 38, 82)	(66, 82, 95, 88, 73, 54, 95, 71)
actType	identity	identity	identity	tanh	identity	relu	relu
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	6.68E-04	9.32E-04	1.19E-04	5.55E-04	9.64E-04	2.03E-04	1.39E-04
batch_size	877	1193	703	1990	1961	1132	1832
learn_rate	constant	constant	constant	adaptive	adaptive	invscaling	adaptive
learning_rate_init	6.995299	7.740298	6.150955	5.135227	5.615547	3.352395	9.808603
max_iter	1523	398	504	906	1045	870	1679
tol	8.70E-05	5.37E-05	1.65E-06	1.25E-05	9.66E-05	2.02E-05	7.87E-06
warm_start	False	True	False	False	True	True	True

Tablica A.3 Hiperparametri Lars modela AMD dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDAR D	POWER
eps	0.030781	0.027938	0.027967	0.916547	0.027946	0.027949	0.022287
n_nonzero_coefs	339	224	819	230	244	706	406
jitter	None	None	None	None	None	None	None

*Tablica A.4 Hiperparametri Lasso modela AMD dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	1.93E-03	3.88E-06	5.74E-05	3.89E-01	3.50E-04	2.90E-04	1.74E-04
<b>max_iter</b>	5511	4880	3512	8291	4356	4169	6054
<b>warm_start</b>	False	False	False	True	True	True	True
<b>positive</b>	False	False	True	False	True	False	True
<b>tol</b>	2.41E-05	5.27E-05	3.22E-05	2.18E-05	1.14E-05	4.32E-06	2.61E-05

*Tablica A.5 Hiperparametri LassoLars modela AMD dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	1.68E-07	2.18E-07	1.57E-06	4.73E-03	4.59E-07	1.39E-06	2.85E-06
<b>max_iter</b>	2130	1119	3647	4401	3724	2893	1271
<b>positive</b>	True	False	True	True	True	True	True
<b>jitter</b>	3.05E-06	1.71E-06	6.38E-06	2.43E-07	7.04E-06	6.10E-06	5.52E-06
<b>eps</b>	3.19E-05	9.85E-05	3.05E-06	6.61E-05	4.57E-05	5.92E-06	4.50E-05

*Tablica A.6 Hiperparametri ARDR modela AMD dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJ A</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUS T</b>	<b>STANDARD</b>	<b>POWER</b>
<b>n_iter</b>	1389	3866	2243	1072	3921	2746	3732
<b>tol</b>	3.76E-03	5.88E-03	6.44E-03	2.75E-03	3.34E-03	6.65E-03	2.06E-03
<b>alpha_1</b>	8.18E-03	6.07E-03	7.84E-03	1.10E-03	6.94E-03	5.18E-03	1.43E-03
<b>alpha_2</b>	4.98E-04	9.88E-03	6.65E-03	1.10E-03	9.48E-03	9.04E-03	5.98E-03
<b>lambda_1</b>	9.79E-03	7.74E-03	8.65E-04	8.13E-03	6.81E-03	6.12E-04	8.13E-03
<b>lambda_2</b>	1.34E-06	1.03E-05	7.24E-03	9.95E-06	1.76E-05	1.82E-05	6.54E-03
<b>threshold_lambda</b>	22,216	5,154	18,616	7,910	28,873	12,554	6,577

Tablica A.7 Hiperparametri BayesianRidge modela AMD dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	78	3544	1288	624	563	2908	2750
tol	1.31E-04	1.57E-03	2.12E-03	6.09E-03	2.16E-03	8.40E-03	9.11E-03
alpha_1	7.10E-03	9.14E-03	4.89E-03	8.27E-03	2.61E-03	7.39E-03	4.20E-03
alpha_2	3.25E-03	9.34E-03	8.77E-03	5.19E-07	9.94E-03	8.79E-03	6.04E-05
lambda_1	1.00E-02	3.88E-03	9.51E-03	4.46E-03	9.96E-03	4.69E-03	6.62E-04
lambda_2	1.04E-05	9.67E-03	1.21E-03	8.40E-03	2.34E-05	1.92E-03	9.97E-03

Tablica A.8 Hiperparametri linearnog modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	False	True	False	False	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	False	False	False	False

Tablica A.9 Hiperparametri MLP modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(12, 10, 7)	(63, 47, 49, 91)	(44, 57, 60, 24, 28, 50, 16, 47)	(10, 7, 10, 5, 7, 6, 11, 13, 6, 5, 8)	(74, 89, 51, 73)	(24, 79, 84, 97)	(85, 49, 60, 49, 94, 40, 51, 35, 63)
actType	identity	identity	identity	relu	relu	relu	identity
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	8.47E-04	4.80E-04	9.12E-04	5.42E-04	7.91E-05	6.31E-04	5.80E-04
batch_size	1356	562	368	253	1851	797	1983
learn_rate	constant	invscaling	invscaling	invscaling	invscaling	invscaling	invscaling
learning_rate_init	0.405745	7.219767	2.801689	7.242103	1.213649	8.721047	6.252764
max_iter	317	1353	1583	1927	1770	1094	625
tol	7.26E-05	1.27E-05	1.51E-05	5.20E-07	9.40E-06	2.92E-05	2.82E-05
warm_start	True	True	False	True	True	True	False



Tablica A.10 Hiperparametri Lars modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>eps</b>	0.03891233417134876	0.023041	0.018804	0.670936	0.034539	0.006742	0.017351
<b>n_nonzero_coefs</b>	6	274	570	284	479	236	119
<b>jitter</b>	None	None	None	None	None	None	None

Tablica A.11 Hiperparametri Lasso modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	4.08E-01	3.26E-03	2.30E-03	5.52E-01	6.84E-02	8.44E-03	1.16E-02
<b>max_iter</b>	2194	5684	4512	1275	3810	7119	5836
<b>warm_start</b>	True	True	False	True	True	False	False
<b>positive</b>	False	False	False	False	True	True	True
<b>tol</b>	1.15E-05	8.87E-05	3.62E-05	6.31E-05	5.35E-06	4.81E-05	3.85E-06

Tablica A.12 Hiperparametri LassoLars modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	4.31E-04	5.02E-05	1.75E-05	3.51E-02	9.49E-05	7.93E-05	6.42E-05
<b>max_iter</b>	3175	1902	4469	2711	3800	2948	3196
<b>positive</b>	True	False	True	False	False	True	False
<b>jitter</b>	8.79E-06	7.21E-06	6.32E-07	2.07E-07	2.85E-06	6.13E-06	9.85E-06
<b>eps</b>	3.53E-05	3.22E-05	7.03E-06	8.36E-05	2.63E-06	4.37E-05	2.73E-05

Tablica A.13 Hiperparametri ARDR modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2299	3926	2971	2763	2221	2279	676
tol	1.97E-03	7.99E-03	6.11E-03	9.78E-03	7.81E-03	7.00E-03	5.70E-03
alpha_1	5.48E-03	4.65E-03	4.42E-03	8.80E-03	2.72E-03	8.38E-03	7.02E-04
alpha_2	5.38E-03	6.03E-03	8.96E-03	2.36E-04	4.06E-03	9.93E-03	8.96E-03
lambda_1	8.96E-03	9.23E-03	6.24E-03	7.34E-03	6.23E-03	8.78E-03	9.43E-03
lambda_2	1.60E-05	1.69E-03	8.65E-03	1.03E-04	1.01E-03	7.00E-03	1.03E-03
threshold_lambda	16,378	98,902	54,645	1,449	48,875	45,797	32,500

Tablica A.14 Hiperparametri BayesianRidge modela AMD dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	789	1418	1596	561	414	1788	1154
tol	7.14E-03	8.47E-03	2.74E-03	8.31E-03	5.48E-03	7.44E-03	2.32E-03
alpha_1	9.48E-03	1.02E-03	9.33E-03	5.34E-03	3.09E-03	5.64E-03	6.92E-03
alpha_2	3.95E-03	2.87E-03	3.92E-04	2.92E-05	3.18E-03	1.18E-03	9.80E-03
lambda_1	8.69E-03	4.63E-03	7.72E-03	9.48E-03	9.84E-03	2.32E-03	3.50E-03
lambda_2	4.69E-03	9.23E-03	9.95E-03	8.72E-03	4.54E-03	9.70E-03	4.45E-03

Tablica A.15 Hiperparametri linearnog modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	True	False	False	True	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	False	False	False	False

Tablica A.16 Hiperparametri MLP modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>tupleHid</b>	(100, 94)	(77, 45, 23)	(36, 37, 68)	(28, 24, 24, 51, 37, 49)	(48, 88, 62)	(83, 46, 56, 51, 47)	(56, 71, 92, 52, 78)
<b>actType</b>	logistic	relu	logistic	identity	relu	relu	identity
<b>solver</b>	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
<b>alpha</b>	7.98E-04	7.75E-04	9.83E-05	3.67E-04	4.28E-04	3.09E-04	7.67E-04
<b>batch_size</b>	1444	1332	547	1071	1459	772	1351
<b>learn_rate</b>	adaptive	constant	constant	constant	constant	constant	invscaling
<b>learning_rate_init</b>	2.918768	4.717097	4.081575	5.925235	6.430684	3.726894	0.919687
<b>max_iter</b>	1308	1862	1075	459	1140	883	933
<b>tol</b>	4.70E-06	4.46E-05	1.37E-05	7.13E-05	3.75E-05	9.69E-05	8.56E-06
<b>warm_start</b>	False	True	True	True	True	False	True

Tablica A.17 Hiperparametri Lars modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDAR D	POWER
<b>eps</b>	0.920014	0.542738	0.455536	0.089517	0.380325	0.759035	0.269021
<b>n_nonzero_coefs</b>	429	315	860	1000	173	430	567
<b>jitter</b>	None	None	None	None	None	None	None

Tablica A.18 Hiperparametri Lasso modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	7.15E-01	4.65E-01	3.22E-01	2.29E-01	6.19E-01	1.87E-01	1.46E-02
<b>max_iter</b>	8412	8291	8623	5665	7806	6811	2506
<b>warm_start</b>	False	True	True	False	True	True	True
<b>positive</b>	True	False	True	True	True	False	True
<b>tol</b>	6.61E-06	7.53E-05	5.78E-05	2.85E-05	3.53E-05	6.25E-06	7.94E-05

Tablica A.19 Hiperparametri LassoLars modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.36E-02	8.92E-02	5.38E-02	4.43E-03	3.26E-02	2.18E-03	3.47E-02
max_iter	2014	3074	4679	2947	4643	2264	4293
positive	False	False	False	True	True	False	True
jitter	5.36E-06	3.41E-06	2.92E-06	6.92E-06	3.07E-06	2.66E-06	6.62E-06
eps	6.71E-05	6.82E-05	8.63E-05	4.53E-05	7.78E-06	3.35E-05	3.19E-05

Tablica A.20 Hiperparametri ARDR modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2432	2382	1843	409	933	1396	1999
tol	5.36E-03	5.88E-03	2.48E-04	9.42E-03	5.32E-04	4.87E-04	9.87E-03
alpha_1	3.07E-03	9.64E-03	1.81E-03	4.50E-03	2.53E-04	8.05E-03	3.31E-03
alpha_2	2.02E-03	3.93E-03	8.94E-03	8.80E-03	6.97E-03	7.22E-03	6.18E-03
lambda_1	3.87E-03	9.34E-03	1.37E-03	4.58E-03	3.03E-03	6.21E-03	9.63E-03
lambda_2	3.24E-04	7.45E-04	6.52E-03	8.72E-03	1.83E-03	2.04E-04	6.32E-03
threshold_lambda	75,408	4,546	5,444	45,505	48,424	73,124	6,356

Tablica A.21 Hiperparametri BayesianRidge modela AMD dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	877	1181	902	1008	1717	968	758
tol	6.48E-03	4.26E-03	9.41E-04	8.11E-03	9.48E-03	4.15E-03	3.45E-03
alpha_1	7.91E-03	1.86E-03	8.51E-03	9.91E-03	2.51E-04	9.62E-03	7.42E-03
alpha_2	6.19E-03	4.13E-03	4.12E-03	2.57E-03	1.88E-03	1.99E-03	1.85E-03
lambda_1	8.28E-03	4.58E-03	6.67E-03	9.05E-03	7.21E-03	6.33E-04	2.57E-03
lambda_2	7.01E-03	1.34E-03	5.36E-03	3.10E-03	9.14E-03	7.45E-03	1.90E-03

Tablica A.22 Hiperparametri linearnog modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	True	False	True	False	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	False	False	False	False

Tablica A.23 Hiperparametri MLP modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(30, 62, 28)	(81, 25, 98, 99, 95)	(51, 32)	(57, 89, 82, 68, 70, 18, 89)	(28,)	(27, 58, 92, 54, 23, 87, 97, 92)	(62, 90, 78, 18, 78)
actType	identity	identity	identity	logistic	identity	relu	identity
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	2.44E-04	5.65E-05	4.24E-04	7.78E-04	7.93E-05	1.87E-04	7.87E-04
batch_size	1320	1449	1941	376	1508	1649	534
learn_rate	invscaling	invscaling	constant	constant	adaptive	invscaling	constant
learning_rate_init	2.450909	3.651801	4.415749	7.021176	2.827586	9.370142	5.344363
max_iter	1910	1172	240	836	1092	1683	1966
tol	9.56E-05	4.27E-05	6.88E-05	3.15E-05	4.34E-05	1.10E-05	7.40E-05
warm_start	True	False	False	True	False	True	False

Tablica A.24 Hiperparametri Lars modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDAR D	POWER
eps	0.873501	0.662509	0.992057	0.667688	0.276425	0.002783	0.392673
n_nonzero_coefs	975	488	600	449	38	449	361
jitter	None	None	None	None	None	None	None

Tablica A.25 Hiperparametri Lasso modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.58E-01	2.00E-01	8.55E-04	7.13E-01	3.55E-01	1.05E-01	1.10E-01
max_iter	5139	7468	2633	8150	1261	8107	2236
warm_start	False	True	False	True	True	True	False
positive	True	True	False	False	False	False	False
tol	1.54E-05	2.86E-05	6.38E-06	6.65E-05	4.45E-05	4.75E-05	1.70E-05

Tablica A.26 Hiperparametri LassoLars modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.91E-02	1.56E-04	4.65E-02	7.71E-03	3.46E-03	4.30E-03	5.58E-02
max_iter	3186	2947	2223	1794	4073	4551	3112
positive	False	False	False	True	True	False	True
jitter	1.26E-06	6.78E-06	2.14E-06	1.67E-07	1.48E-06	1.06E-06	2.78E-06
eps	3.34E-05	7.76E-05	4.10E-05	2.41E-05	7.48E-05	3.68E-07	6.88E-09

Tablica A.27 Hiperparametri ARDR modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	1135	3347	3455	1606	3484	683	2451
tol	6.85E-03	4.09E-03	3.05E-03	5.85E-03	2.26E-03	2.35E-04	4.92E-03
alpha_1	3.18E-03	2.13E-03	6.88E-03	2.49E-03	4.29E-03	9.54E-05	4.12E-03
alpha_2	3.99E-03	7.40E-03	6.49E-03	1.24E-05	2.11E-03	1.82E-03	4.32E-03
lambda_1	3.10E-03	3.98E-04	9.30E-03	1.02E-03	3.35E-03	4.28E-03	5.04E-03
lambda_2	2.97E-03	5.48E-03	1.68E-03	1.07E-03	1.75E-03	6.92E-03	4.95E-03
threshold_lambda	80,507	18,872	94,260	31,176	41,803	7,752	18,682

Tablica A.28 Hiperparametri BayesianRidge modela AMD dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	544	619	1447	912	1762	1786	1137
tol	8.50E-03	8.27E-03	9.28E-03	1.80E-03	7.82E-03	9.82E-03	4.57E-03
alpha_1	7.52E-03	5.51E-03	4.08E-03	2.87E-03	1.12E-03	9.14E-03	7.08E-03
alpha_2	7.89E-03	2.02E-03	4.80E-03	5.81E-03	2.41E-03	1.92E-03	1.94E-03
lambda_1	1.09E-03	7.88E-03	9.00E-03	1.73E-03	6.30E-03	1.94E-03	5.75E-03
lambda_2	5.76E-03	2.91E-03	8.88E-03	3.84E-04	1.94E-04	8.46E-03	3.67E-03

## A.2.AAPL dataset

Tablica A.29 Hiperparametri linearnog modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	True	True	False	False	True	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.30 Hiperparametri MLP modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(82, 42)	(66, 55, 74, 34, 35, 47, 47, 85, 37)	(10, 17)	(46, 74, 58)	default	(61,)	(70, 83, 94, 71, 65)
actType	identity	identity	identity	logistic	default	identity	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	default	lbfgs	lbfgs
alpha	4.12E-04	1.50E-04	6.74E-04	1.58E-05	default	1.22E-04	2.97E-04
batch_size	1858	946	731	1914	default	580	1837
learn_rate	constant	invscaling	invscaling	constant	default	invscaling	constant
learning_rate_init	6.659718	4.83942	8.071584	5.277306	default	3.768192	2.563394
max_iter	1826	1733	486	673	default	1697	560
tol	2.44E-05	6.91E-06	2.18E-05	4.59E-05	default	2.02E-05	3.66E-05
warm_start	True	True	False	False	default	True	False

Tablica A.31 Hiperparametri Lars modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	default	default	default	default	9.85E-03	1.10E+16	3.64E-02
n_nonzero_coefs	default	default	default	default	248	200	407
jitter	default	default	default	default	None	None	None

Tablica A.32 Hiperparametri Lasso modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	1.36E-02	5.96E-04	4.77E-04	default	2.80E-01	6.50E-04	2.87E-03
max_iter	8842	7104	5209	default	1561	7024	2871
warm_start	False	True	True	default	False	False	True
positive	True	True	True	default	True	False	True
tol	3.95E-06	9.47E-05	1.44E-05	default	5.92E+09	7.95E-05	6.51E-05

Tablica A.33 Hiperparametri LassoLars modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	3.27E-06	1.91E-07	1.90E-06	default	2.85E-06	5.94E-07	4.07E-05
max_iter	4365	3608	2877	default	1633	3777	4969
positive	True	False	False	default	False	False	False
jitter	5.83E-06	3.44E-06	6.28E-06	default	1.90E-06	3.84E-06	7.72E-07
eps	2.39E-05	7.43E-05	3.19E-06	default	8.13E-06	3.95E-05	7.55E-05



Tablica A.34 Hiperparametri ARDR modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2313	525	3472	default	3697	973	3032
tol	9.50E-03	2.00E-03	1.72E-03	default	7.89E-03	7.53E-03	7.02E-04
alpha_1	6.04E-03	9.56E-03	8.54E-03	default	8.02E-03	4.27E-05	4.65E-03
alpha_2	8.18E-03	3.41E-03	9.92E-03	default	2.99E-03	7.85E-03	1.78E-03
lambda_1	3.30E-03	7.56E-03	1.39E-03	default	6.72E-03	6.98E-03	7.83E-03
lambda_2	1.32E-05	7.37E-03	9.01E-03	default	3.17E-03	9.47E-03	7.58E-03
threshold_lambda	23,068	23,086	11,920	default	29,137	25,737	15,300

Tablica A.35 Hiperparametri BayesianRidge modela AAPL dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1569	498	1946	3948	1516	2728	1295
tol	4.70E-03	1.50E-03	7.46E-03	3.74E-03	7.57E-03	4.44E-03	5.20E-03
alpha_1	6.97E-03	1.58E-04	2.18E-03	9.00E-03	6.99E-03	9.63E-03	4.26E-03
alpha_2	1.61E-03	7.60E-03	9.00E-04	1.20E-06	3.08E-03	7.44E-03	7.95E-03
lambda_1	5.24E-03	7.02E-04	3.94E-04	5.26E-03	4.33E-03	2.88E-03	7.16E-03
lambda_2	6.69E-03	5.19E-03	9.64E-03	9.32E-03	6.16E-03	6.11E-03	4.37E-03

Tablica A.36 Hiperparametri linearnog modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	False	False	False	False	True	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.37 Hiperparametri MLP modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>tupleHid</b>	(68, 11, 57, 25, 33, 78, 54, 53)	(25, 90, 55, 31)	(44, 40, 51, 29)	(58, 83, 46)	(98, 94, 92, 84, 52, 85, 90, 49)	(28, 70, 72, 37, 75, 61, 31)	(23, 96, 18, 28, 86)
<b>actType</b>	tanh	relu	relu	identity	relu	tanh	identity
<b>solver</b>	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
<b>alpha</b>	3.97E-04	2.47E-04	5.41E-04	9.83E-04	1.76E-04	6.10E-04	4.47E-04
<b>batch_size</b>	1660	907	471	1039	847	1680	949
<b>learn_rate</b>	adaptive	invscaling	adaptive	invscaling	invscaling	adaptive	adaptive
<b>learning_rate_init</b>	4.997215	3	1.252241	4.309074	2.459045	9.674855	2.239422
<b>max_iter</b>	923	545	1339	309	707	1208	1479
<b>tol</b>	9.49E-05	1.41E-07	3.00E-05	8.54E-06	4.26E-05	8.62E-05	6.55E-05
<b>warm_start</b>	False	True	False	True	True	False	False

Tablica A.38 Hiperparametri Lars modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>eps</b>	7.23E-01	9.42E-01	5.68E-01	1.21E-01	2.90E-01	4.59E-01	4.19E-02
<b>n_nonzero_coefs</b>	544	721	340	887	825	175	258
<b>jitter</b>	None	None	None	None	None	None	None

Tablica A.39 Hiperparametri Lasso modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	2.08E-01	3.78E-04	8.68E-04	2.77E-02	5.66E-01	2.73E-02	2.45E-02
<b>max_iter</b>	5910	5508	7878	8568	3493	4510	6496
<b>warm_start</b>	True	False	True	False	False	False	False
<b>positive</b>	True	True	True	False	True	True	True
<b>tol</b>	6.15E-05	3.53E-05	5.79E-05	7.09E-06	7.15E-05	3.04E-05	6.89E-05

Tablica A.40 Hiperparametri LassoLars modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	1.08E-03	4.53E-05	1.97E-04	1.07E-02	2.15E-04	2.44E-05	2.28E-04
max_iter	2264	3960	4454	2643	4985	2630	3126
positive	True	False	True	False	True	True	True
jitter	6.86E-06	9.94E-06	9.85E-06	7.17E-06	7.23E-06	2.11E-07	1.73E-06
eps	8.40E-05	9.42E-05	4.12E-05	3.57E-05	1.59E-05	1.67E-05	1.08E-05

Tablica A.41 Hiperparametri ARDR modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXAB S	MINMAX	NORM	ROBUS T	STANDAR D	POWER
n_iter	471	3125	3329	3593	1648	1825	1374
tol	8.05E-03	7.91E-03	2.10E-03	5.89E-03	8.98E-03	5.51E-03	5.62E-03
alpha_1	2.45E-03	2.28E-03	1.35E-03	0.004119	5.84E-03	2.57E-03	6.93E-03
alpha_2	6.59E-03	7.01E-04	6.92E-03	0.002991	7.11E-03	4.50E-03	4.84E-03
lambda_1	3.71E-03	5.10E-03	4.86E-03	0.000694	4.13E-03	7.25E-03	1.01E-03
lambda_2	3.64E-03	1.00E-03	9.55E-03	0.006038	5.94E-03	8.14E-03	9.80E-03
threshold_lambda	71,160	98,247	2,507	39,080	90,439	40,464	32,554

Tablica A.42 Hiperparametri BayesianRidge modela AAPL dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	654	1196	1925	708	1485	1505	1423
tol	3.07E-03	4.45E-03	2.30E-04	9.58E-03	3.67E-03	8.89E-03	8.07E-03
alpha_1	4.88E-03	4.09E-03	5.06E-03	1.84E-03	4.37E-03	7.04E-03	6.04E-03
alpha_2	4.76E-03	7.77E-03	5.47E-03	8.61E-03	9.73E-03	1.16E-03	7.51E-03
lambda_1	9.27E-04	2.90E-03	7.80E-03	6.97E-05	4.11E-03	4.10E-03	4.94E-03
lambda_2	8.92E-03	3.20E-03	2.26E-03	8.15E-03	7.92E-03	2.82E-03	2.47E-03

Tablica A.43 Hiperparametri linearnog modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	True	False	False	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.44 MLP linearnog modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(31, 64, 36, 11, 38, 37)	(75, 93, 43, 83, 89, 32, 67)	(56, 58, 11, 15, 94, 53, 76, 93, 18)	(54, 75, 100, 35, 88, 25, 72)	(60, 88, 79, 88, 94)	(70, 64, 80, 68, 98, 26, 73)	(97, 22, 55, 16, 64, 71)
actType	tanh	identity	relu	relu	relu	relu	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	4.40E-04	5.34E-04	5.59E-04	1.19E-04	9.20E-04	1.52E-04	5.50E-05
batch_size	1604	1607	1745	436	1178	1875	1141
learn_rate	invscaling	constant	constant	adaptive	constant	invscaling	invscaling
learning_rate_init	6.282219	3.583497	6.686367	4.132179	4.95124	9.487827	9.093334
max_iter	1118	821	507	1314	559	892	1685
tol	7.04E-05	1.39E-05	8.61E-05	6.29E-05	6.70E-06	4.27E-05	2.63E-05
warm_start	True	True	False	True	True	False	False

Tablica A.45 Hiperparametri Lars modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	7.44E-01	9.04E-02	6.80E-01	7.34E-01	3.13E-01	1.08E-01	8.55E-01
n_nonzero_coefs	447	843	318	780	679	747	724
jitter	None	None	None	None	None	None	None

Tablica A.46 Hiperparametri Lasso modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	3.03E-02	7.86E-01	4.31E-01	4.51E-01	7.04E-01	4.12E-01	8.88E-01
max_iter	6642	3783	6900	3158	3176	6800	2762
warm_start	False	True	True	False	False	True	True
positive	True	True	True	True	True	True	True
tol	9.52E-05	7.83E-05	4.46E-05	8.79E-05	1.41E-05	9.68E-05	7.47E-05

Tablica A.47 Hiperparametri LassoLars modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	7.68E-02	1.95E-02	4.24E-02	3.09E-02	8.66E-02	1.23E-02	7.51E-02
max_iter	4186	3601	2418	2899	1917	2697	3406
positive	False	False	False	True	False	True	False
jitter	8.04E-06	9.15E-06	9.42E-06	8.41E-06	6.69E-06	8.65E-06	7.82E-06
eps	9.43E-05	5.65E-05	7.76E-05	6.41E-05	3.45E-05	9.01E-05	8.22E-05

Tablica A.48 Hiperparametri ARDR modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	659	2605	2229	2205	3091	724	2478
tol	6.88E-03	9.24E-03	8.40E-04	1.32E-03	7.57E-03	8.96E-03	1.42E-03
alpha_1	2.19E-03	9.20E-03	2.21E-03	2.48E-03	2.56E-03	9.58E-03	8.06E-03
alpha_2	1.75E-03	9.16E-03	7.49E-03	5.62E-03	8.51E-03	9.50E-04	2.35E-03
lambda_1	6.63E-03	1.96E-03	1.36E-03	1.25E-03	3.87E-03	6.66E-04	3.85E-03
lambda_2	6.51E-03	4.81E-03	2.08E-03	3.20E-03	7.41E-03	5.63E-03	4.67E-03
threshold_lambda	86,259	46,149	38,150	30,400	18,677	85,117	68,787

Tablica A.49 Hiperparametri BayesianRidge modela AAPL dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1506	1265	1393	688	1226	1815	1521
tol	3.43E-03	3.70E-03	7.54E-03	3.93E-03	6.05E-03	1.43E-03	7.60E-03
alpha_1	9.49E-03	9.85E-03	6.09E-03	8.82E-03	3.48E-03	8.50E-03	5.33E-04
alpha_2	7.30E-04	5.70E-03	9.42E-05	6.26E-03	7.28E-03	5.36E-04	4.25E-03
lambda_1	9.51E-03	1.18E-03	9.69E-03	5.83E-03	2.54E-03	4.00E-03	8.90E-03
lambda_2	5.56E-03	2.50E-03	5.97E-03	1.13E-03	1.90E-03	8.11E-03	1.29E-03

Tablica A.50 Hiperparametri linearnog modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	False	True	False	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.51 Hiperparametri MLP modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(46, 57, 24, 21, 26)	(69, 35, 10, 90, 58)	(36, 75)	(32, 21, 22, 100)	(53, 37)	(30, 21, 65, 79)	(82, 69, 79, 41, 98, 39, 77, 15)
actType	identity	tanh	logistic	logistic	logistic	tanh	identity
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	8.88E-04	6.40E-04	4.53E-05	6.29E-04	6.90E-04	4.94E-04	6.74E-04
batch_size	938	1341	1066	794	942	237	455
learn_rate	invscaling	adaptive	constant	adaptive	constant	adaptive	adaptive
learning_rate_init	9.037909	5.129648	3.637922	8.083975	2.396526	2.217414	7.389298
max_iter	1639	262	648	1303	347	594	1066
tol	3.20E-05	6.83E-05	4.32E-05	5.61E-05	9.96E-05	3.99E-05	2.58E-05
warm_start	True	False	False	True	True	True	False

Tablica A.52 Hiperparametri Lars modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	7.37E-01	2.04E-01	8.93E-01	9.49E-01	1.64E-02	8.51E-01	1.98E-01
n_nonzero_coefs	295	190	484	360	781	289	828
jitter	None	None	None	None	None	None	None

Tablica A.53 Hiperparametri Lasso modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.89E-01	4.13E-03	1.58E-03	5.60E-01	8.70E-01	3.02E-01	4.56E-01
max_iter	3499	4467	1083	5999	7076	6445	2015
warm_start	True	True	False	False	True	False	False
positive	True	False	False	False	True	True	True
tol	8.22E-05	4.73E-05	1.83E-05	2.34E-05	1.30E-05	7.33E-05	8.34E-05

Tablica A.54 Hiperparametri LassoLars modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	7.07E-02	3.15E-02	1.79E-02	9.00E-02	2.52E-02	7.28E-03	8.86E-03
max_iter	2470	2456	1018	2049	3973	3398	4601
positive	True	False	True	False	False	False	True
jitter	2.78E-06	5.47E-06	9.91E-06	5.24E-07	3.22E-07	5.61E-06	1.85E-07
eps	9.89E-05	3.55E-05	4.24E-05	4.06E-05	9.69E-05	9.82E-06	2.07E-05

Tablica A.55 Hiperparametri ARDR modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2352	430	3037	677	3134	3784	2758
tol	2.63E-04	1.26E-03	4.54E-03	8.45E-03	8.24E-03	2.78E-03	7.68E-03
alpha_1	1.16E-03	6.66E-03	2.82E-03	8.13E-04	7.21E-03	3.81E-03	2.05E-03
alpha_2	8.31E-03	3.81E-03	4.67E-03	1.32E-04	4.15E-03	1.93E-03	6.75E-04
lambda_1	2.25E-03	1.46E-03	4.87E-03	4.04E-03	2.64E-03	4.97E-03	8.66E-03
lambda_2	6.91E-03	1.89E-04	9.25E-03	5.97E-04	7.82E-03	7.67E-03	4.00E-03
threshold_lambda	28,816	97,425	50,341	59,499	59,874	69,412	46,044

Tablica A.56 Hiperparametri BayesianRidge modela AAPL dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	816	931	1460	1437	1891	1351	1352
tol	1.24E-03	5.34E-03	4.92E-03	2.28E-03	8.54E-03	4.93E-03	3.36E-03
alpha_1	3.26E-03	2.81E-03	5.96E-03	2.29E-03	8.09E-03	7.89E-03	7.20E-03
alpha_2	1.27E-03	6.95E-03	7.28E-03	1.13E-03	3.82E-03	9.49E-03	3.09E-04
lambda_1	2.45E-03	3.51E-03	2.57E-03	5.55E-03	2.82E-03	1.62E-04	1.94E-03
lambda_2	7.45E-04	1.09E-03	3.39E-03	9.86E-03	2.11E-03	4.39E-04	2.88E-03

### A.3.HPQ dataset

Tablica A.57 Hiperparametri linearnog modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	True	False	False	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True



Tablica A.58 Hiperparametri MLP modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>tupleHid</b>	(33, 70)	default	(24, 62, 73, 74, 48, 70)	(65,)	default	(45, 14, 37, 75, 94)	(50,)
<b>actType</b>	identity	default	tanh	identity	default	relu	tanh
<b>solver</b>	lbfgs	default	lbfgs	lbfgs	default	lbfgs	lbfgs
<b>alpha</b>	2.33E-04	default	1.44E-04	1.59E-04	default	3.02E-04	7.47E-04
<b>batch_size</b>	567	default	1763	506	default	1992	368
<b>learn_rate</b>	constant	default	constant	constant	default	invscaling	constant
<b>learning_rate_init</b>	1.255197	default	0.292255	3.396424	default	3.912244	2.282869
<b>max_iter</b>	1838	default	404	1374	default	753	918
<b>tol</b>	6.04E-05	default	2.19E-06	3.89E-05	default	8.73E-05	5.12E-05
<b>warm_start</b>	False	default	True	True	default	False	False

Tablica A.59 Hiperparametri Lars modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>eps</b>	default	default	default	default	default	default	1.30E-02
<b>n_nonzero_coefs</b>	default	default	default	default	default	default	155
<b>jitter</b>	default	default	default	default	default	default	None

Tablica A.60 Hiperparametri Lasso modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	4.65E-03	3.31E-06	1.13E-04	default	4.19E-05	3.37E-04	1.78E-03
<b>max_iter</b>	5980	4803	4678	default	7935	7825	7801
<b>warm_start</b>	True	False	False	default	False	True	False
<b>positive</b>	False	True	False	default	True	True	True
<b>tol</b>	4.92E-06	9.90E-06	6.05E-05	default	8.62E-05	3.50E-05	1.86E-05

Tablica A.61 Hiperparametri LassoLars modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	4.10E-06	7.25E-07	1.32E-07	default	3.51E-06	2.70E-06	1.25E-06
max_iter	1309	2184	1381	default	2988	2892	4162
positive	True	False	True	default	True	True	True
jitter	3.20E-06	6.24E-07	8.38E-06	default	1.66E-06	2.05E-06	2.00E-06
eps	7.08E-05	5.22E-05	1.73E-05	default	1.13E-05	4.92E-05	6.34E-05

Tablica A.62 Hiperparametri ARDR modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	default	default	default	default	3697	default	default
tol	default	default	default	default	7.89E-03	default	default
alpha_1	default	default	default	default	8.02E-03	default	default
alpha_2	default	default	default	default	2.99E-03	default	default
lambda_1	default	default	default	default	6.72E-03	default	default
lambda_2	default	default	default	default	3.17E-03	default	default
threshold_lambda	default	default	default	default	29,137	default	default

Tablica A.63 Hiperparametri BayesianRidge modela HPQ dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1105	default	default	1584	1516	default	default
tol	6.32E-03	default	default	9.63E-03	7.57E-03	default	default
alpha_1	2.01E-03	default	default	1.01E-03	6.99E-03	default	default
alpha_2	2.57E-03	default	default	3.90E-08	3.08E-03	default	default
lambda_1	4.13E-03	default	default	9.02E-03	4.33E-03	default	default
lambda_2	8.54E-04	default	default	1.27E-03	6.16E-03	default	default

Tablica A.64 Hiperparametri linearnog modela HPQ dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	False	True	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.65 Hiperparametri MLP modela HPQ dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(65,)	(92, 84, 99)	(34,)	(36, 90, 77, 73, 83)	(31, 65)	(13, 33, 45)	(45, 46, 63, 42, 53, 40, 64)
actType	tanh	relu	relu	identity	identity	relu	relu
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	1.07E-04	9.74E-05	8.32E-04	8.74E-04	7.63E-04	4.75E-04	9.67E-04
batch_size	986	297	839	427	1585	411	803
learn_rate	adaptive	constant	constant	constant	invscaling	invscaling	constant
learning_rate_init	7.966832	2.209544	2.740442	1.047293	7.293535	2.343097	7.766568
max_iter	619	1746	517	1979	962	932	652
tol	5.67E-05	9.62E-05	8.19E-05	9.04E-05	6.69E-05	1.61E-05	6.11E-05
warm_start	False	False	True	True	True	False	True

Tablica A.66 Hiperparametri Lars modela HPQ dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	2.56E-01	5.02E-01	5.04E-01	1.07E-01	1.54E-01	8.73E-01	7.39E-01
n_nonzero_coefs	438	564	513	564	843	82	276
jitter	None	None	None	None	None	None	None

Tablica A.67 Hiperparametri Lasso modela HPQ datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	1.09E-01	2.62E-04	2.68E-04	7.56E-01	4.42E-03	1.09E-02	2.11E-02
<b>max_iter</b>	4399	6526	2891	1520	7536	8722	1472
<b>warm_start</b>	True	True	True	True	True	False	True
<b>positive</b>	False	False	False	False	False	False	True
<b>tol</b>	3.50E-05	1.34E-05	9.29E-05	8.58E-05	3.41E-05	7.41E-05	5.32E-05

Tablica A.68 Hiperparametri LassoLars modela HPQ datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	1.28E-03	3.76E-05	9.50E-06	5.09E-02	7.67E-02	2.55E-02	2.96E-04
<b>max_iter</b>	1760	4671	3293	1590	3896	2007	2018
<b>positive</b>	False	False	True	False	True	True	False
<b>jitter</b>	6.62E-07	4.19E-06	7.59E-06	4.41E-06	3.71E-06	3.10E-07	1.87E-06
<b>eps</b>	2.95E-05	8.43E-05	4.41E-05	3.06E-05	3.87E-05	7.28E-05	6.62E-05

Tablica A.69 Hiperparametri ARDR modela HPQ datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
<b>n_iter</b>	3434	3823	2567	3499	3819	1070	397
<b>tol</b>	7.79E-03	9.55E-03	7.17E-03	3.90E-03	2.84E-03	8.58E-03	4.93E-03
<b>alpha_1</b>	2.46E-03	1.89E-03	8.55E-04	7.57E-03	5.16E-03	9.55E-04	9.01E-03
<b>alpha_2</b>	1.44E-03	3.81E-03	4.46E-03	3.22E-03	9.23E-03	2.92E-03	3.93E-03
<b>lambda_1</b>	5.16E-03	9.70E-04	1.24E-03	9.57E-05	4.48E-03	5.61E-03	2.71E-03
<b>lambda_2</b>	2.17E-03	1.70E-03	3.86E-03	8.91E-04	6.11E-03	8.78E-03	8.35E-03
<b>threshold_lambda</b>	35,579	52,208	83,775	38,406	80,519	62,749	77,212

Tablica A.70 Hiperparametri BayesianRidge modela HPQ dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1236	1748	1765	1179	1171	1097	1217
tol	6.83E-03	1.99E-03	5.66E-04	8.39E-03	3.18E-03	2.54E-03	8.76E-03
alpha_1	1.99E-03	8.42E-03	6.80E-03	7.92E-03	6.21E-03	6.44E-03	3.67E-03
alpha_2	9.15E-03	6.17E-03	7.20E-03	3.42E-03	9.50E-03	3.85E-03	3.53E-03
lambda_1	3.76E-04	3.41E-03	8.98E-03	1.88E-04	3.99E-03	7.96E-03	1.75E-03
lambda_2	3.78E-03	6.52E-03	3.60E-03	9.92E-03	3.72E-03	5.10E-03	7.59E-03

Tablica A.71 Hiperparametri linearnog modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	False	True	True	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.72 Hiperparametri MLP modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(30, 38, 55, 69, 14, 94, 15)	(89, 68, 54, 62, 90, 64, 86, 91)	(34, 85, 33, 32, 96)	(34,)	(97, 42, 41, 51, 95, 66, 59)	(85, 30, 62, 18, 40, 58, 71, 37)	(84, 65, 33)
actType	tanh	identity	logistic	identity	tanh	relu	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	4.57E-04	1.22E-04	8.35E-04	4.15E-04	4.69E-04	2.64E-04	5.12E-04
batch_size	202	1699	1797	423	1178	1603	1396
learn_rate	adaptive	constant	invscaling	constant	constant	invscaling	adaptive
learning_rate_init	6.859704	3.261337	5.038686	6.959161	8.710001	7.370883	9.833686
max_iter	458	1027	825	1610	1305	1839	1032
tol	6.30E-05	4.63E-05	4.22E-05	9.73E-05	2.96E-05	4.36E-07	2.56E-05
warm_start	True	False	False	True	False	True	True

Tablica A.73 Hiperparametri Lars modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	4.70E-01	7.64E-01	9.11E-01	9.02E-01	1.43E-01	9.45E-01	5.68E-01
n_nonzero_coefs	295	370	936	630	593	972	798
jitter	None	None	None	None	None	None	None

Tablica A.74 Hiperparametri Lasso modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	1.97E-01	3.03E-01	9.68E-01	3.61E-02	6.64E-01	1.50E-01	1.04E-01
max_iter	3341	7652	7401	5588	6196	7934	8117
warm_start	False	True	True	False	False	True	True
positive	False	True	True	True	False	True	False
tol	5.11E-06	1.17E-05	7.94E-05	6.38E-05	4.84E-05	7.37E-05	9.68E-05

Tablica A.75 Hiperparametri LassoLars modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.46E-02	4.26E-02	3.25E-02	7.37E-02	2.79E-02	1.71E-02	6.27E-02
max_iter	1716	2380	1842	4410	4150	4900	1389
positive	False	False	True	False	True	False	False
jitter	9.17E-06	7.70E-07	4.63E-08	3.56E-06	2.49E-06	5.10E-06	8.75E-06
eps	6.93E-05	7.60E-05	4.09E-06	1.54E-05	8.45E-05	3.48E-05	3.00E-05

Tablica A.76 Hiperparametri ARDR modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
<b>n_iter</b>	3466	3708	3746	2692	320	1709	2881
<b>tol</b>	2.49E-03	9.45E-03	9.74E-03	4.07E-03	6.37E-03	8.95E-03	4.41E-03
<b>alpha_1</b>	8.69E-03	7.05E-03	7.58E-03	1.28E-03	9.53E-03	7.81E-03	6.88E-03
<b>alpha_2</b>	8.58E-03	1.59E-03	1.21E-03	7.39E-03	8.30E-03	6.27E-03	5.97E-03
<b>lambda_1</b>	9.90E-03	1.50E-03	9.12E-03	3.74E-03	7.34E-03	5.27E-03	6.24E-03
<b>lambda_2</b>	1.32E-03	2.89E-03	4.20E-03	9.78E-03	8.99E-03	1.74E-04	1.85E-04
<b>threshold_lambda</b>	61,187	4,388	44,483	21,838	31,661	36,576	10,539

Tablica A.77 Hiperparametri BayesianRidge modela HPQ dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>n_iter</b>	1889	355	623	619	553	771	1474
<b>tol</b>	7.60E-03	3.62E-03	7.32E-03	1.04E-03	6.08E-03	3.98E-03	3.97E-03
<b>alpha_1</b>	9.80E-03	7.12E-03	6.93E-03	9.67E-03	2.92E-03	2.53E-03	1.55E-03
<b>alpha_2</b>	4.77E-03	3.49E-03	9.17E-03	3.32E-03	7.49E-03	9.30E-03	4.31E-04
<b>lambda_1</b>	3.19E-03	4.36E-03	4.96E-03	5.10E-03	4.69E-04	1.82E-03	9.93E-03
<b>lambda_2</b>	7.81E-03	8.39E-03	1.68E-03	8.36E-03	7.28E-03	5.93E-03	9.09E-03

Tablica A.78 Hiperparametri linearnog modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>fit_intercept</b>	False	False	True	True	True	False	True
<b>copy_X</b>	True	True	True	True	True	True	True
<b>positive</b>	False	False	False	True	True	True	True

Tablica A.79 Hiperparametri MLP modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>tupleHid</b>	(38, 51, 14, 62, 23, 77, 76, 85, 43)	(95.)	(83, 60, 30, 96, 42, 91, 97, 49)	(62, 83, 95, 85, 48, 32, 12, 65)	(99, 46)	(38, 74, 57, 70, 33, 46, 94, 72)	(88, 36)
<b>actType</b>	relu	tanh	relu	logistic	tanh	relu	logistic
<b>solver</b>	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
<b>alpha</b>	3.91E-04	3.71E-04	3.68E-04	8.64E-04	9.98E-04	9.58E-04	1.36E-04
<b>batch_size</b>	553	299	1369	564	203	1934	1907
<b>learn_rate</b>	constant	invscaling	constant	invscaling	invscaling	constant	constant
<b>learning_rate_init</b>	9.328082	2.704707	9.188755	1.140565	4.843734	5.714525	2.605852
<b>max_iter</b>	1052	1145	1822	641	476	969	1285
<b>tol</b>	9.34E-05	6.88E-06	8.98E-05	9.78E-05	3.67E-06	9.26E-05	7.27E-05
<b>warm_start</b>	False	False	False	True	False	False	True

Tablica A.80 Hiperparametri Lars modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	4.57E-01	9.85E-01	2.93E-01	9.08E-01	8.52E-01	3.00E-01	8.51E-01
n_nonzero_coefs	878	533	158	735	501	913	351
jitter	None	None	None	None	None	None	None

Tablica A.81 Hiperparametri Lasso modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	2.84E-01	5.61E-01	6.89E-01	7.44E-01	8.39E-02	8.35E-01	3.04E-02
<b>max_iter</b>	1479	5464	7125	4391	8733	8014	4043
<b>warm_start</b>	False	False	False	True	True	True	True
<b>positive</b>	True	True	False	False	True	True	False
<b>tol</b>	1.27E-05	6.63E-05	1.91E-05	3.61E-05	4.77E-05	3.05E-05	7.54E-05



Tablica A.82 Hiperparametri LassoLars modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	3.69E-02	1.54E-04	2.13E-04	6.14E-02	8.08E-02	8.60E-04	2.12E-02
max_iter	3752	1453	2605	2360	1092	2805	3379
positive	False	False	True	False	True	False	False
jitter	1.42E-06	9.72E-06	6.16E-07	2.70E-06	1.18E-06	6.85E-07	7.01E-06
eps	8.92E-05	3.76E-05	5.99E-05	3.06E-05	3.20E-05	1.00E-04	6.70E-05

Tablica A.83 Hiperparametri ARDR modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	3067	3255	1910	2556	364	2301	1141
tol	3.20E-05	3.96E-03	1.12E-03	5.82E-04	6.70E-03	8.12E-03	6.19E-03
alpha_1	4.56E-03	3.22E-03	8.47E-04	5.50E-04	1.32E-03	5.74E-03	2.62E-03
alpha_2	2.72E-04	1.15E-03	8.28E-03	2.00E-05	4.08E-04	8.94E-03	4.70E-05
lambda_1	3.65E-03	6.23E-03	1.66E-03	3.97E-03	3.04E-04	5.61E-03	7.19E-03
lambda_2	9.52E-04	4.32E-03	7.52E-03	6.56E-04	6.69E-03	3.71E-03	3.92E-03
threshold_lambda	3,956	49,267	74,424	64,478	85,040	44,258	76,409

Tablica A.84 Hiperparametri BayesianRidge modela HPQ dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1370	1978	1875	1587	1836	1820	666
tol	5.31E-03	7.45E-03	4.70E-03	6.54E-03	9.50E-03	4.77E-03	3.18E-03
alpha_1	2.88E-03	1.17E-03	1.77E-03	1.36E-04	1.03E-03	2.09E-03	1.24E-03
alpha_2	2.40E-03	9.49E-03	8.07E-03	6.58E-03	9.45E-03	1.24E-03	8.65E-03
lambda_1	8.89E-03	2.54E-04	5.04E-03	4.95E-04	5.24E-03	6.73E-03	7.47E-04
lambda_2	3.15E-03	5.70E-03	2.88E-03	3.21E-03	5.50E-03	5.31E-03	5.09E-03

#### A.4.IBM

Tablica A.85 Hiperparametri linearnog modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	True	False	False	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.86 Hiperparametri MLP modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(77, 93, 13, 88, 62, 77, 36, 82, 15)	(13, 66, 88, 15, 48)	(57,)	(51,)	(32, 67, 85, 72, 61, 41, 19, 13, 67)	(44,)	(10,)
actType	identity	identity	tanh	logistic	identity	tanh	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	8.85E-04	3.70E-04	5.59E-04	3.83E-04	5.22E-04	2.02E-04	4.14E-04
batch_size	810	1142	945	1333	959	1831	1457
learn_rate	adaptive	constant	adaptive	invscaling	invscaling	constant	invscaling
learning_rate_init	1.081821	3.572241	9.788295	4.586868	4.421049	3.189941	6.029272
max_iter	424	332	1873	1551	1949	1229	871
tol	3.10E-05	2.05E-05	4.70E-06	6.69E-05	8.05E-05	2.09E-06	9.29E-05
warm_start	True	False	True	True	True	True	False

Tablica A.87 Hiperparametri Lars modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	default	default	default	default	default	default	default
n_nonzero_coefs	default	default	default	default	default	default	default
jitter	default	default	default	default	default	default	default

Tablica A.88 Hiperparametri Lasso modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	9.37E-02	5.69E-04	3.06E-05	default	3.48E-04	5.31E-04	3.54E-04
<b>max_iter</b>	8940	5957	8023	default	8884	8948	8026
<b>warm_start</b>	False	False	False	default	False	True	False
<b>positive</b>	False	True	True	default	True	False	True
<b>tol</b>	3.07E-05	3.59E-05	6.41E-05	default	2.24E-05	3.18E-05	4.30E-05

Tablica A.89 Hiperparametri LassoLars modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	8.07E-07	5.53E-06	1.99E-07	1.01E-06	5.96E-06	3.08E-06	4.25E-05
<b>max_iter</b>	3033	1937	3462	1006	3027	1618	1746
<b>positive</b>	False	True	True	True	False	True	True
<b>jitter</b>	3.52E-06	7.32E-06	7.44E-07	9.63E-06	8.41E-06	1.66E-06	5.23E-06
<b>eps</b>	1.92E-06	3.08E-06	7.72E-06	6.16E-05	7.73E-05	1.30E-05	9.79E-05

Tablica A.90 Hiperparametri ARDR modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
<b>n_iter</b>	default	default	default	default	default	default	default
<b>tol</b>	default	default	default	default	default	default	default
<b>alpha_1</b>	default	default	default	default	default	default	default
<b>alpha_2</b>	default	default	default	default	default	default	default
<b>lambda_1</b>	default	default	default	default	default	default	default
<b>lambda_2</b>	default	default	default	default	default	default	default
<b>threshold_lambda</b>	default	default	default	default	default	default	default

Tablica A.91 Hiperparametri BayesianRidge modela IBM dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	2269	default	default	1522	default	default	default
tol	4.53E-03	default	default	5.94E-03	default	default	default
alpha_1	8.14E-03	default	default	1.77E-03	default	default	default
alpha_2	1.00E-03	default	default	4.59E-07	default	default	default
lambda_1	2.55E-03	default	default	4.09E-03	default	default	default
lambda_2	9.36E-03	default	default	1.50E-03	default	default	default

Tablica A.92 Hiperparametri linearnog modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	True	True	False	False	False	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.93 Hiperparametri MLP modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(98, 49, 65, 13)	(30, 52, 49, 80)	(51, 71, 87, 18)	(57, 16, 59)	(54, 20, 90, 40, 99, 30, 31, 83)	(73, 40)	(60, 96)
actType	identity	identity	relu	tanh	identity	relu	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	5.26E-04	2.23E-04	6.62E-04	5.07E-04	8.83E-04	6.14E-04	7.36E-04
batch_size	1245	1039	1143	675	1997	336	515
learn_rate	adaptive	adaptive	adaptive	invscaling	adaptive	invscaling	adaptive
learning_rate_init	9.866733	3.877764	8.069357	3.426736	3.23545	4.393481	0.769069
max_iter	727	479	1000	1635	1459	1955	858
tol	8.44E-05	9.00E-05	6.67E-05	7.86E-05	2.25E-05	6.43E-05	2.74E-05
warm_start	False	True	True	True	False	True	True

Tablica A.94 Hiperparametri Lars modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	3.63E-01	8.14E-01	6.73E-01	4.03E-01	4.54E-01	5.63E-01	8.36E-01
n_nonzero_coefs	633	133	352	274	123	332	497
jitter	None	None	None	None	None	None	None

Tablica A.95 Hiperparametri Lasso modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	6.34E-01	6.38E-04	8.26E-04	3.59E-01	1.81E-03	2.30E-02	2.30E-03
max_iter	6096	8063	8341	1000	7852	2136	3225
warm_start	True	False	False	True	True	False	False
positive	False	False	False	True	False	True	True
tol	2.34E-06	4.81E-05	9.28E-05	9.96E-05	2.53E-05	9.43E-05	4.31E-05

Tablica A.96 Hiperparametri LassoLars modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	2.94E-03	4.53E-07	9.06E-05	8.65E-03	8.34E-05	2.54E-04	7.82E-05
max_iter	2498	1562	3522	4194	3526	4033	3466
positive	True	True	False	True	True	True	False
jitter	7.21E-06	2.14E-06	4.83E-06	1.01E-07	4.99E-06	3.78E-07	3.45E-06
eps	5.17E-05	8.75E-05	9.54E-05	6.14E-05	7.96E-06	5.34E-05	4.33E-05

Tablica A.97 Hiperparametri ARDR modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2970	2150	2264	3748	2500	3946	3207
tol	9.67E-03	7.73E-03	6.54E-03	7.70E-03	4.28E-03	6.15E-03	9.75E-03
alpha_1	2.70E-03	1.36E-03	3.28E-03	5.38E-03	6.57E-03	1.73E-03	8.66E-03
alpha_2	3.73E-03	3.45E-03	2.63E-03	1.42E-04	7.56E-03	3.56E-03	1.90E-03
lambda_1	1.45E-04	1.98E-03	1.77E-03	3.07E-03	4.20E-03	3.54E-03	5.99E-05
lambda_2	5.47E-03	8.33E-03	5.98E-03	8.89E-03	8.23E-03	5.88E-03	4.65E-03
threshold_lambda	95,670	48,137	24,838	2,824	87,969	35,625	43,807

Tablica A.98 Hiperparametri BayesianRidge modela IBM dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1464	1803	1296	897	1197	1918	432
tol	3.77E-03	9.55E-03	5.99E-03	2.79E-03	5.12E-03	7.29E-04	1.67E-03
alpha_1	1.46E-03	4.20E-03	7.85E-03	3.49E-03	8.20E-03	1.42E-03	2.48E-03
alpha_2	1.42E-03	7.06E-03	8.54E-03	6.99E-06	1.46E-03	7.33E-03	8.41E-04
lambda_1	9.18E-04	8.85E-04	7.10E-03	1.49E-03	7.19E-03	2.93E-03	5.15E-03
lambda_2	6.93E-03	9.84E-03	3.87E-03	2.28E-03	5.18E-04	4.76E-04	1.24E-03

Tablica A.99 Hiperparametri linearnog modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	True	False	False	False	False	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.100 Hiperparametri MLP modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(69, 49)	(32, 20)	(61, 29, 41)	(58, 50, 22, 42, 91, 82, 95, 28, 64)	(99, 84, 36, 30)	(73, 54)	(67, 56, 90, 17, 11, 82, 100, 58)
actType	identity	relu	relu	relu	relu	identity	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	3.58E-04	2.78E-04	8.68E-04	7.57E-04	5.19E-04	1.21E-04	6.97E-04
batch_size	1477	1490	1576	621	1461	200	1515
learn_rate	constant	constant	adaptive	constant	constant	invscaling	invscaling
learning_rate_init	2.212602	0.333747	6.96817	6.802933	4.18018	5.710356	2.033778
max_iter	1875	1568	1296	1344	967	1898	1284
tol	6.57E-05	4.40E-07	3.64E-05	4.19E-05	8.78E-05	6.21E-05	5.78E-05
warm_start	True	True	False	True	False	True	False

Tablica A.101 Hiperparametri Lars modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	7.65E-01	9.42E-01	6.00E-01	1.73E-01	9.76E-01	9.41E-02	7.87E-01
n_nonzero_coefs	799	78	160	23	710	176	207
jitter	None	None	None	None	None	None	None

Tablica A.102 Hiperparametri Lasso modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	9.44E-01	6.93E-01	1.78E-01	4.00E-02	7.65E-01	9.65E-01	1.14E-01
max_iter	1814	1758	7625	7112	4246	2801	2108
warm_start	True	True	True	True	True	True	False
positive	False	False	False	True	True	True	False
tol	4.25E-05	7.82E-05	1.71E-05	1.33E-05	2.83E-05	2.28E-05	8.20E-05

Tablica A.103 Hiperparametri LassoLars modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	9.92E-02	9.53E-02	1.83E-02	9.55E-02	8.56E-02	2.75E-02	3.34E-02
max_iter	3133	4292	3738	2133	4456	2354	4621
positive	True	True	False	False	False	False	True
jitter	6.05E-06	5.04E-06	5.78E-06	3.60E-07	9.39E-06	8.39E-06	1.78E-06
eps	5.60E-05	1.15E-06	8.11E-06	8.72E-05	5.99E-05	6.13E-05	9.87E-05

Tablica A.104 Hiperparametri ARDR modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	1790	755	3769	1736	707	3215	2047
tol	1.70E-03	8.11E-04	4.86E-03	2.81E-03	5.67E-03	3.85E-03	2.21E-03
alpha_1	5.64E-03	8.26E-03	9.26E-03	5.39E-04	5.34E-03	2.93E-03	5.44E-03
alpha_2	9.67E-03	2.23E-03	3.31E-03	4.68E-03	4.61E-03	3.48E-03	1.18E-03
lambda_1	7.85E-03	4.38E-03	7.50E-03	6.93E-03	5.91E-04	3.50E-03	5.77E-03
lambda_2	2.81E-03	6.70E-03	1.56E-03	8.81E-03	3.98E-03	4.94E-04	3.65E-03
threshold_lambda	95,678	23,207	6,996	39,031	26,212	35,505	11,610

Tablica A.105 Hiperparametri BayesianRidge modela IBM dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1489	629	1418	519	1442	1953	736
tol	5.47E-03	8.80E-03	4.04E-03	4.94E-03	7.77E-03	9.31E-03	9.70E-03
alpha_1	2.53E-03	4.15E-03	1.24E-03	3.94E-03	3.67E-03	8.17E-03	8.60E-03
alpha_2	3.70E-03	2.93E-03	5.65E-03	9.15E-03	8.11E-03	7.51E-03	8.42E-03
lambda_1	4.60E-03	6.68E-03	7.23E-03	6.07E-03	5.40E-03	4.56E-04	4.80E-03
lambda_2	9.38E-05	2.58E-03	6.43E-03	8.93E-03	1.04E-04	8.12E-03	9.36E-04

Tablica A.106 Hiperparametri linearnog modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	False	False	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.107 Hiperparametri MLP modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(61, 55, 90, 29)	(76,)	(27, 24)	(23, 38, 53, 79, 63)	(45, 48, 34, 97, 65)	(85, 43, 47)	(37, 44, 49, 17, 52, 65, 39, 13, 19)
actType	relu	relu	logistic	relu	identity	identity	identity
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	6.78E-05	1.08E-04	7.67E-05	8.15E-04	7.64E-04	8.82E-04	8.90E-04
batch_size	611	1037	1520	551	1246	682	1312
learn_rate	adaptive	constant	constant	constant	constant	constant	invscaling
learning_rate_init	9.012201	3.995287	9.23597	7.273396	1.381279	5.182832	8.148752
max_iter	307	1724	1925	765	1637	1167	1589
tol	2.27E-05	7.58E-05	9.56E-05	6.49E-05	1.43E-06	7.43E-05	1.39E-05
warm_start	True	True	True	True	True	False	False



Tablica A.108 Hiperparametri Lars modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	3.22E-01	4.55E-01	3.97E-01	2.03E-01	5.41E-01	2.32E-01	5.36E-01
n_nonzero_coefs	6	239	821	230	695	868	532
jitter	None	None	None	None	None	None	None

Tablica A.109 Hiperparametri Lasso modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	8.51E-01	1.99E-04	3.66E-01	1.60E-02	1.18E-01	5.58E-02	1.62E-01
max_iter	1989	5109	1620	4421	8350	2920	5855
warm_start	True	False	True	False	True	True	True
positive	True	True	True	True	True	True	True
tol	7.54E-05	3.73E-05	4.64E-05	2.75E-06	9.19E-05	8.48E-05	4.30E-05

Tablica A.110 Hiperparametri LassoLars modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	9.91E-02	5.04E-02	7.24E-04	4.84E-02	3.61E-03	9.32E-03	5.67E-02
max_iter	2310	3605	1607	2467	4450	1279	1510
positive	False	False	False	True	False	False	False
jitter	2.49E-06	8.48E-06	9.23E-06	8.74E-06	8.83E-06	1.97E-06	1.00E-05
eps	7.35E-05	9.03E-05	3.16E-06	4.62E-05	6.74E-05	9.91E-05	6.83E-05

Tablica A.111 Hiperparametri ARDR modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2297	2112	3257	1667	2421	2932	1888
tol	2.05E-03	5.56E-03	9.21E-03	5.00E-03	3.05E-03	2.48E-03	8.05E-03
alpha_1	6.80E-03	4.55E-03	8.20E-03	5.72E-03	8.09E-03	1.44E-03	1.93E-03
alpha_2	2.32E-03	9.76E-03	5.39E-03	6.70E-03	6.63E-03	3.74E-04	8.74E-03
lambda_1	5.10E-03	8.85E-03	9.69E-03	4.64E-03	1.59E-03	3.66E-03	9.71E-03
lambda_2	2.77E-03	6.12E-03	3.74E-03	5.05E-03	1.62E-03	1.91E-03	2.89E-03
threshold_lambda	84,440	6,199	12,011	71,787	41,116	91,542	52,573

Tablica A.112 Hiperparametri BayesianRidge modela IBM dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	842	1565	824	461	933	601	1753
tol	6.52E-03	5.10E-04	4.47E-03	1.84E-03	7.06E-03	7.73E-03	1.81E-03
alpha_1	6.80E-03	1.19E-03	2.68E-03	4.30E-03	9.05E-03	8.10E-03	6.91E-03
alpha_2	2.50E-03	3.87E-04	7.53E-03	8.07E-03	6.90E-03	5.06E-03	6.84E-03
lambda_1	8.42E-03	3.93E-03	5.90E-03	1.79E-03	9.01E-03	4.46E-03	5.40E-03
lambda_2	4.83E-03	5.56E-04	2.76E-03	9.14E-03	8.89E-03	4.15E-03	6.64E-03

## A.5.INTC

Tablica A.113 Hiperparametri linearnog modela INTC dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	True	False	False	False	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.114 Hiperparametri MLP modela INTC dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(82, 42)	(66, 55, 74, 34, 35, 47, 47, 85, 37)	(10, 17)	(46, 74, 58)	default	(61,)	(70, 83, 94, 71, 65)
actType	identity	identity	identity	logistic	default	identity	tanh
solver	lbfgs	lbfgs	lbfgs	lbfgs	default	lbfgs	lbfgs
alpha	4.12E-04	1.50E-04	6.74E-04	1.58E-05	default	1.22E-04	2.97E-04
batch_size	1858	946	731	1914	default	580	1837
learn_rate	constant	invscaling	invscaling	constant	default	invscaling	constant
learning_rate_init	6.659718	4.83942	8.071584	5.277306	default	3.768192	2.563394
max_iter	1826	1733	486	673	default	1697	560
tol	2.44E-05	6.91E-06	2.18E-05	4.59E-05	default	2.02E-05	3.66E-05
warm_start	True	True	False	False	default	True	False

*Tablica A.115 Hiperparametri Lars modela INTC dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJ A</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>eps</b>	2.76E-03	default	default	default	9.85E-03	1.10E-02	3.64E-02
<b>n_nonzero_coefs</b>	90	default	default	default	248	200	407
<b>jitter</b>	None	default	default	default	None	None	None

*Tablica A.116 Hiperparametri Lasso modela INTC dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	1.36E-02	5.96E-04	4.77E-04	default	2.80E-01	6.50E-04	2.87E-03
<b>max_iter</b>	8842	7104	5209	default	1561	7024	2871
<b>warm_start</b>	False	True	True	default	False	False	True
<b>positive</b>	True	True	True	default	True	False	True
<b>tol</b>	3.95E-06	9.47E-05	1.44E-05	default	9.59E-05	7.95E-05	6.51E-05

*Tablica A.117 Hiperparametri LassoLars modela INTC dataseta pri metodi nasumičnog traženja hiperparametara*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	3.27E-06	1.91E-07	1.90E-06	default	2.85E-06	5.94E-07	4.07E-05
<b>max_iter</b>	4365	3608	2877	default	1633	3777	4969
<b>positive</b>	True	False	False	default	False	False	False
<b>jitter</b>	5.83E-06	3.44E-06	6.28E-06	default	1.90E-06	3.84E-06	7.72E-07
<b>eps</b>	2.39E-05	7.43E-05	3.19E-06	default	8.13E-06	3.95E-05	7.55E-05

Tablica A.118 Hiperparametri ARDR modela INTC dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	2313	525	3472	default	3697	973	3032
tol	9.50E-03	2.00E-03	1.72E-03	default	7.89E-03	7.53E-03	7.02E-04
alpha_1	6.04E-03	9.56E-03	8.54E-03	default	8.02E-03	4.27E-05	4.65E-03
alpha_2	8.18E-03	3.41E-03	9.92E-03	default	2.99E-03	7.85E-03	1.78E-03
lambda_1	3.30E-03	7.56E-03	1.39E-03	default	6.72E-03	6.98E-03	7.83E-03
lambda_2	1.32E-05	7.37E-03	9.01E-03	default	3.17E-03	9.47E-03	7.58E-03
threshold_lambda	23,068	23,086	11,920	default	29,137	25,737	15,300

Tablica A.119 Hiperparametri BayesianRidge modela INTC dataseta pri metodi nasumičnog traženja hiperparametara

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1569	498	1946	3948	1516	2728	1295
tol	4.70E-03	1.50E-03	7.46E-03	3.74E-03	7.57E-03	4.44E-03	5.20E-03
alpha_1	6.97E-03	1.58E-04	2.18E-03	9.00E-03	6.99E-03	9.63E-03	4.26E-03
alpha_2	1.61E-03	7.60E-03	9.00E-04	1.20E-06	3.08E-03	7.44E-03	7.95E-03
lambda_1	5.24E-03	7.02E-04	3.94E-04	5.26E-03	4.33E-03	2.88E-03	7.16E-03
lambda_2	6.69E-03	5.19E-03	9.64E-03	9.32E-03	6.16E-03	6.11E-03	4.37E-03

Tablica A.120 Hiperparametri linearnog modela INTC dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	True	False	True	False	True
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.121 Hiperparametri MLP modela INTC datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>tupleHid</b>	(30, 33, 56, 10, 43, 83, 80)	(33, 96, 11, 25, 84, 29, 38)	(18, 15)	(54, 15, 34, 43, 89, 22)	(47, 90, 21, 92, 86, 78, 79, 35, 34)	(21, 19, 73)	(23, 40, 15, 100, 57, 46, 48, 69, 90)
<b>actType</b>	relu	relu	identity	relu	identity	relu	relu
<b>solver</b>	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
<b>alpha</b>	9.63E-04	2.53E-04	7.16E-04	1.34E-04	4.65E-04	6.57E-04	8.98E-04
<b>batch_size</b>	1469	1344	720	210	681	796	500
<b>learn_rate</b>	constant	invscaling	invscaling	constant	invscaling	invscaling	constant
<b>learning_rate_init</b>	7.233881	4.773745	7.823963	6.402862	7.196343	2.964997	6.098857
<b>max_iter</b>	1167	330	801	1329	450	1429	210
<b>tol</b>	6.20E-05	5.84E-05	1.62E-06	8.27E-05	3.64E-05	5.31E-05	7.41E-05
<b>warm_start</b>	False	True	False	False	False	False	False

Tablica A.122 Hiperparametri Lars modela INTC datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>eps</b>	4.86E-01	5.43E-01	2.42E-01	6.39E-01	4.11E-01	8.35E-02	2.13E-01
<b>n_nonzero_coefs</b>	534	861	991	22	63	346	428
<b>jitter</b>	None	None	None	None	None	None	None

Tablica A.123 Hiperparametri Lasso modela INTC datseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
<b>alpha</b>	1.26E-01	1.72E-03	1.24E-03	9.27E-02	1.75E-02	6.73E-03	2.26E-02
<b>max_iter</b>	7807	2551	4073	7461	4939	5032	4162
<b>warm_start</b>	True	True	True	False	False	False	True
<b>positive</b>	True	True	True	True	True	True	False
<b>tol</b>	9.20E-05	2.49E-05	1.89E-05	2.72E-05	7.83E-05	3.42E-05	5.75E-05

Tablica A.124 Hiperparametri LassoLars modela INTC dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	7.37E-04	7.68E-05	1.29E-04	5.87E-03	5.43E-05	1.19E-04	1.32E-04
max_iter	2822	3137	1076	3551	2872	1910	4888
positive	False	True	False	False	False	True	False
jitter	2.01E-06	2.90E-06	6.09E-06	1.62E-07	5.30E-06	5.35E-06	7.87E-06
eps	5.58E-05	7.95E-05	7.58E-05	1.65E-05	6.52E-05	5.14E-05	5.36E-05

Tablica A.125 Hiperparametri ARDR modela INTC dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	323	867	3107	2721	2319	686	3725
tol	3.08E-03	1.88E-03	5.29E-03	7.08E-03	5.17E-03	2.00E-03	4.05E-03
alpha_1	5.84E-03	6.65E-04	8.97E-03	2.25E-04	8.67E-03	7.15E-03	6.91E-03
alpha_2	6.82E-03	4.58E-03	6.97E-03	9.10E-03	1.77E-04	3.86E-03	4.23E-03
lambda_1	2.08E-03	7.76E-03	9.04E-03	5.95E-03	7.61E-04	7.38E-03	5.95E-03
lambda_2	4.10E-03	4.26E-03	3.60E-03	3.79E-03	8.26E-03	7.20E-03	9.03E-03
threshold_lambda	51,359	54,339	49,246	78,230	30,913	76,756	52,957

Tablica A.126 Hiperparametri BayesianRidge modela INTC dataseta pri metodi unakrsne validacije

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	1251	456	1707	324	323	1185	1341
tol	1.78E-04	7.88E-03	8.17E-04	8.37E-03	6.07E-03	7.41E-03	3.79E-03
alpha_1	1.90E-03	5.44E-03	3.13E-03	8.98E-03	3.52E-03	4.71E-03	1.46E-03
alpha_2	3.57E-04	1.56E-03	8.58E-03	5.90E-03	3.59E-04	7.82E-03	6.58E-03
lambda_1	5.46E-03	5.11E-03	9.76E-03	6.48E-03	1.36E-03	9.07E-03	5.62E-03
lambda_2	3.24E-03	3.32E-03	7.50E-03	3.85E-03	8.49E-03	2.33E-03	8.77E-03

Tablica A.127 Hiperparametri linearnog modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	False	True	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.128 Hiperparametri MLP modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(27, 39, 66, 88, 61)	(60, 97, 31, 77, 84, 46, 44, 66, 42)	(47, 20, 45, 43)	(91, 35, 38, 82, 75, 99, 67, 29)	(14, 93)	(15, 59)	(38, 40, 74, 38, 70, 46, 45, 38, 50)
actType	tanh	relu	identity	identity	tanh	logistic	logistic
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	6.32E-04	5.73E-04	7.77E-04	9.37E-04	2.09E-04	6.61E-04	6.88E-04
batch_size	1617	487	1256	1437	1464	1789	1644
learn_rate	constant	constant	adaptive	constant	invscaling	invscaling	invscaling
learning_rate_init	4.115399	3.540497	1.898929	4.103074	9.701375	6.108533	7.240081
max_iter	1994	1795	1344	1157	1661	1705	1416
tol	8.15E-05	2.67E-05	8.37E-05	4.85E-05	8.26E-05	1.70E-05	3.45E-05
warm_start	True	False	True	True	False	False	False

Tablica A.129 Hiperparametri Lars modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
eps	7.59E-01	7.69E-01	8.79E-01	1.19E-01	5.81E-01	3.83E-01	5.13E-01
n_nonzero_coefs	983	294	607	529	160	933	776
jitter	None	None	None	None	None	None	None

Tablica A.130 Hiperparametri Lasso modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	1.50E-01	4.52E-01	7.71E-01	7.73E-01	8.02E-01	9.97E-01	7.48E-01
max_iter	5037	3224	8032	5129	8336	5688	4281
warm_start	False	True	False	True	False	True	False
positive	False	False	False	True	False	True	False
tol	9.49E-05	5.79E-05	1.00E-05	1.72E-05	2.03E-05	7.08E-05	3.05E-05

Tablica A.131 Hiperparametri LassoLars modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
alpha	9.76E-02	6.30E-02	9.69E-02	3.36E-02	8.34E-02	3.12E-02	2.52E-02
max_iter	1297	1548	3028	4536	4857	3792	2811
positive	False	True	True	True	False	False	False
jitter	2.52E-06	2.84E-06	1.89E-06	5.00E-06	3.45E-06	5.82E-06	1.58E-07
eps	2.54E-05	1.78E-06	8.16E-05	8.98E-05	3.46E-05	7.70E-05	3.11E-05

Tablica A.132 Hiperparametri ARDR modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJ A	MAXABS	MINMAX	NORM	ROBUS T	STANDARD	POWER
n_iter	3161	1837	2862	3498	3932	3501	820
tol	7.76E-03	6.33E-03	3.08E-04	4.85E-03	8.85E-04	4.21E-03	7.05E-03
alpha_1	7.52E-03	2.90E-03	8.57E-03	8.47E-03	3.56E-03	7.62E-03	6.13E-03
alpha_2	4.86E-03	1.12E-03	4.97E-04	9.55E-03	2.52E-03	8.42E-03	5.78E-03
lambda_1	6.27E-03	2.35E-03	4.96E-03	7.85E-03	8.06E-03	1.89E-03	9.94E-03
lambda_2	4.65E-03	5.72E-03	8.82E-03	2.64E-03	8.62E-03	9.04E-03	5.27E-03
threshold_lambda	67,842	39,053	94,113	53,676	33,112	21,352	40,718



Tablica A.133 Hiperparametri Bayesian ridge modela INTC dataseta pri stackingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
n_iter	563	1595	1235	1875	1426	343	375
tol	9.89E-03	7.48E-03	4.20E-03	5.85E-03	6.55E-03	8.04E-03	8.44E-03
alpha_1	5.81E-03	5.28E-03	2.21E-03	7.67E-03	5.41E-03	7.24E-03	6.41E-03
alpha_2	2.03E-05	5.59E-03	4.98E-03	3.59E-03	2.69E-03	8.37E-03	3.46E-03
lambda_1	5.26E-03	9.70E-03	6.59E-03	8.30E-03	9.12E-03	9.57E-03	6.07E-03
lambda_2	1.35E-04	3.48E-03	4.23E-03	4.67E-03	9.67E-03	5.64E-03	2.40E-03

Tablica A.134 Hiperparametri linearnog modela INTC dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
fit_intercept	False	False	False	True	False	True	False
copy_X	True	True	True	True	True	True	True
positive	False	False	False	True	True	True	True

Tablica A.135 Hiperparametri MLP modela INTC dataseta pri votingregressor ansamblu

Parametri	BEZ SKALIRANJA	MAXABS	MINMAX	NORM	ROBUST	STANDARD	POWER
tupleHid	(41,)	(46, 51, 35, 70, 78, 83)	(29, 87, 86, 35, 19, 49, 74, 60)	(91, 10, 37, 90, 94, 63, 93, 62)	(88, 13, 81, 49, 85)	(81, 79, 49, 32, 96, 59)	(95,)
actType	identity	identity	relu	relu	relu	identity	identity
solver	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs	lbfgs
alpha	4.09E-05	4.06E-04	6.11E-04	4.00E-04	9.42E-04	6.81E-04	5.47E-04
batch_size	1660	1631	366	1992	1518	1439	463
learn_rate	adaptive	invscaling	adaptive	adaptive	invscaling	invscaling	constant
learning_rate_init	2.766008	6.176139	3.976297	5.904712	4.847644	3.290305	4.956706
max_iter	1609	617	850	1484	668	853	1530
tol	3.44E-05	4.50E-06	1.69E-05	2.89E-05	4.46E-05	1.56E-05	3.25E-05
warm_start	True	True	True	True	True	True	False

*Tablica A.136 Hiperparametri Lars modela INTC dataseta pri votingregressor ansamblu*

<b>Parametri</b>	<b>BEZ SKALIRANJ A</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>eps</b>	8.17E-01	6.50E-01	4.78E-01	4.61E-01	9.58E-01	1.53E-01	9.34E-02
<b>n_nonzero_coefs</b>	570	611	213	182	208	558	390
<b>jitter</b>	None	None	None	None	None	None	None

*Tablica A.137 Hiperparametri Lasso modela INTC dataseta pri votingregressor ansamblu*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	3.94E-01	4.97E-01	7.96E-03	9.77E-02	1.13E-01	4.85E-01	1.73E-01
<b>max_iter</b>	2580	6510	1124	2606	5944	1151	2735
<b>warm_start</b>	False	False	False	True	True	True	False
<b>positive</b>	True	True	True	False	True	True	False
<b>tol</b>	6.03E-05	4.04E-05	1.87E-05	5.18E-05	3.48E-05	8.80E-05	3.69E-06

*Tablica A.138 Hiperparametri LassoLars modela INTC dataseta pri votingregressor ansamblu*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>alpha</b>	5.88E-02	3.07E-04	2.78E-02	5.03E-02	5.94E-02	2.24E-03	1.61E-02
<b>max_iter</b>	1322	3750	2768	2718	2779	3879	4798
<b>positive</b>	True	True	False	False	True	True	True
<b>jitter</b>	7.94E-06	6.63E-07	7.08E-06	1.81E-06	3.72E-06	4.45E-06	6.90E-06
<b>eps</b>	5.46E-05	3.98E-05	9.08E-06	5.18E-05	5.15E-05	2.51E-05	6.76E-05

*Tablica A.139 Hiperparametri ARDR modela INTC dataseta pri votingregressor ansamblu*

<b>Parametri</b>	<b>BEZ SKALIRANJ A</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUS T</b>	<b>STANDARD</b>	<b>POWER</b>
<b>n_iter</b>	3900	3418	3516	1793	3339	2799	454
<b>tol</b>	1.89E-03	7.32E-04	2.44E-04	6.96E-03	5.62E-03	3.55E-03	5.68E-04
<b>alpha_1</b>	9.97E-03	7.55E-03	9.37E-03	1.66E-04	8.24E-03	6.41E-03	5.36E-03
<b>alpha_2</b>	3.96E-03	6.77E-03	3.35E-04	3.79E-03	4.74E-03	8.02E-03	9.59E-03
<b>lambda_1</b>	2.35E-03	9.56E-03	3.52E-03	8.33E-03	7.01E-03	9.10E-04	8.33E-03
<b>lambda_2</b>	5.00E-03	4.42E-03	9.62E-03	9.54E-03	6.44E-03	6.41E-03	9.66E-04
<b>threshold_lambda</b>	21,994	16,644	67,411	35,546	10,499	8,547	53,920

*Tablica A.140 Hiperparametri BayesianRidge modela INTC dataseta pri votingregressor ansamblu*

<b>Parametri</b>	<b>BEZ SKALIRANJA</b>	<b>MAXABS</b>	<b>MINMAX</b>	<b>NORM</b>	<b>ROBUST</b>	<b>STANDARD</b>	<b>POWER</b>
<b>n_iter</b>	1236	1327	1862	1823	813	1635	833
<b>tol</b>	2.68E-03	6.29E-03	5.03E-03	3.66E-03	4.31E-03	7.08E-03	5.87E-03
<b>alpha_1</b>	9.76E-03	7.85E-04	2.41E-04	2.24E-03	2.55E-03	4.51E-03	6.31E-03
<b>alpha_2</b>	6.82E-03	9.43E-03	4.05E-03	5.28E-03	4.66E-03	3.61E-04	6.03E-03
<b>lambda_1</b>	2.48E-03	9.85E-03	8.79E-03	1.14E-03	4.74E-03	6.55E-03	8.60E-03
<b>lambda_2</b>	1.94E-03	7.56E-03	6.51E-03	2.69E-03	9.56E-03	1.24E-03	8.36E-03

## B. Primjer programskog koda.

### B.1. Primjer MLP modela za AMD dataset

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns #Za korelacijsku mapu
import numpy as np
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

data_AMD= pd.read_csv('AMD.csv')#.sample(frac=1)
data_AMD.drop(          # Brisanje redova tako da sve pocinje od istog
datuma
    labels=range(0, (len(data_AMD)-10454)),
    axis=0,          # 0 znaci da se brisu redovi
    inplace=True    # True znaci da zamijenjuje dataset, inace se
stvara nova varijabla ili tako nes
)
data_AMD=data_AMD.reset_index(drop=True) #restart indexa od 0
data_AMD["Date"]= np.linspace(0,len(data_AMD["Date"])-
1,len(data_AMD["Date"]))
data_AMD['Date'] = data_AMD['Date'].astype(int)#datum u integer

y = data_AMD.pop("Close")
X = data_AMD
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

#Mlp
from sklearn.neural_network import MLPRegressor
regr_mlp = MLPRegressor()
regr_mlp.fit(X_train, y_train)

print('MLP')
y_pred_mlp=regr_mlp.predict(X_test)

print('Vrijednosti iz treninga')
print(r2_score(y_train, regr_mlp.predict(X_train)), 'r2')
print(mean_absolute_error(y_train, regr_mlp.predict(X_train)), 'mean absolute
error')
print(mean_squared_error(y_train, regr_mlp.predict(X_train),
squared=False), 'RMSE')
print('')

print('Vrijednosti na testu')
print(r2_score(y_test, y_pred_mlp), 'r2')
print(mean_absolute_error(y_test, y_pred_mlp), 'mean absolute error')
print(mean_squared_error(y_test, y_pred_mlp, squared=False), 'RMSE')
print('')

#standardna deviacija i mean arrayevi koji idu u np.mean,np.std
array_mlp_r2=[r2_score(y_train, regr_mlp.predict(X_train)) , r2_score(y_test,
y_pred_mlp)]
```

```

array_mlp_mae=[mean_absolute_error(y_train, regr_mlp.predict(X_train)) ,
mean_absolute_error(y_test, y_pred_mlp)]
array_mlp_rmse=[mean_squared_error(y_train, regr_mlp.predict(X_train),
squared=False) , mean_squared_error(y_test, y_pred_mlp, squared=False)]

print('r2 srednja vrijednost i devijacija')
standard_mean_mlp_r2=np.mean(array_mlp_r2,dtype=float)
standard_std_mlp_r2=np.std(array_mlp_r2,dtype=float)
print('srednja vrijednost ',standard_mean_mlp_r2)
print('standardna devijacija',standard_std_mlp_r2)
print('')
print('mae srednja vrijednost i devijacija')
standard_mean_mlp_mae=np.mean(array_mlp_mae,dtype=float)
standard_std_mlp_mae=np.std(array_mlp_mae,dtype=float)
print('srednja vrijednost ',standard_mean_mlp_mae)
print('standardna devijacija',standard_std_mlp_mae)
print('')
standard_mean_mlp_rmse=np.mean(array_mlp_rmse,dtype=float)
standard_std_mlp_rmse=np.std(array_mlp_rmse,dtype=float)
print('rmse srednja vrijednost i devijacija')
print('srednja vrijednost ',standard_mean_mlp_rmse)
print('standardna devijacija',standard_std_mlp_rmse)
print('')
print('')
print('')

```

## B.2. Primjer MLP modela s nasumičnim traženjem hiperparametara pri power skaliranju na AMD datasetu

```

import pandas as pd
import random
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

data_AMD= pd.read_csv('AMD.csv')#.sample(frac=1)
data_AMD.drop(          # Brisanje redova tako da sve pocinje od istog
datuma
    labels=range(0, (len(data_AMD)-10454)),
    axis=0,          # 0 znaci da se brisu redovi
    inplace=True    # True znaci da zamijenjuje dataset, inace se
stvara nova varijabla
)
data_AMD=data_AMD.reset_index(drop=True) #restart indexa od 0
data_AMD["Date"]= np.linspace(0,len(data_AMD["Date"])-
1,len(data_AMD["Date"]))
data_AMD['Date'] = data_AMD['Date'].astype(int)#datum u integer

#scaler
from sklearn.preprocessing import PowerTransformer
transformer = PowerTransformer().fit(data_AMD)
data2 = transformer.transform(data_AMD)
data_AMD= pd.DataFrame(data=data2, columns=data_AMD.columns)

```

```

y = data_AMD.pop("Close")
X = data_AMD
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,
random_state = 42)

def RandHypSearch():
    parameters = []
    numHidLayers = random.randint(2,10)
    hidLayers = []
    for i in range(1,numHidLayers,1):
        hidLayers.append(random.randint(10,100))
    tupleHid = tuple(hidLayers)
    print(type(tupleHid))
    actType =random.choice(['identity', 'logistic', 'tanh', 'relu'])
    solver = random.choice(['lbfgs'])
    alpha = random.uniform(0 , 0.001)
    batch_size = random.randint(200,2000)
    learn_rate = random.choice(['constant', 'invscaling', 'adaptive'])
    learning_rate_init=random.uniform(0.001,10)
    max_iter = random.randint(200,2000)
    tol = random.uniform(0,1e-4)
    warm_start=random.choice(['True','False'])

    parameters = [tupleHid,
                  actType,
                  solver,
                  alpha,
                  batch_size,
                  learn_rate,
                  learning_rate_init,
                  max_iter,
                  tol,
                  warm_start,]

    print("paramters = {}".format(parameters))

file1.write("{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\n".format(parameters[0],
                                                                    parameters[1],
                                                                    parameters[2],
                                                                    parameters[3],
                                                                    parameters[4],
                                                                    parameters[5],
                                                                    parameters[6],
                                                                    parameters[7],
                                                                    parameters[8],
                                                                    parameters[9],
                                                                    ))

    return parameters

def MLPR(parameter, X_train, y_train, X_test, y_test):
    mlpr = MLPRegressor(hidden_layer_sizes=parameter[0],
                        activation=parameter[1],
                        solver=parameter[2],
                        alpha= parameter[3],
                        batch_size=parameter[4],
                        learning_rate=parameter[5],
                        learning_rate_init=parameter[6],
                        max_iter= parameter[7],
                        tol = parameter[8],
                        warm_start=parameter[9],)
    mlpr.fit(X_train, y_train)

    R2Train = r2_score(y_train, mlpr.predict(X_train))

```

```

R2Test = r2_score(y_test, mlpr.predict(X_test))

MAETrain = mean_absolute_error(y_train, mlpr.predict(X_train))
MAETest = mean_absolute_error(y_test, mlpr.predict(X_test))

RMSETrain = mean_squared_error(y_train, mlpr.predict(X_train),
squared=False)
RMSETest = mean_squared_error(y_test, mlpr.predict(X_test),
squared=False)

MeanR2 = np.mean([R2Train, R2Test])
MeanMAE = np.mean([MAETrain, MAETest])
MeanRMSE = np.mean([RMSETrain, RMSETest])
STDR2 = np.std([R2Train, R2Test])
STDMAE = np.std([MAETrain, MAETest])
STDRMSE = np.std([RMSETrain, RMSETest])
print("R2Train = {}".format(R2Train))
print("MAETrain = {}".format(MAETrain))
print("RMSETrain = {}".format(RMSETrain))
print("R2Test = {}".format(R2Test))
print("MAETest = {}".format(MAETest))
print("RMSETest = {}".format(RMSETest))
print("MeanR2 = {}".format(MeanR2))
print("MeanMAE = {}".format(MeanMAE))
print("MeanRMSE = {}".format(MeanRMSE))
print("STDR2 = {}".format(STDR2))
print("STDMAE = {}".format(STDMAE))
print("STDRMSE = {}".format(STDRMSE))
print("R2Train = {}".format(R2Train))
print("MAETrain = {}".format(MAETrain))
file2.write("R2Train = {}\n".format(R2Train))
file2.write("MAETrain = {}\n".format(MAETrain))
file2.write("RMSETrain = {}\n".format(RMSETrain))
file2.write("R2Test = {}\n".format(R2Test))
file2.write("MAETest = {}\n".format(MAETest))
file2.write("RMSETest = {}\n".format(RMSETest))
file2.write("MeanR2 = {}\n".format(MeanR2))
file2.write("MeanMAE = {}\n".format(MeanMAE))
file2.write("MeanRMSE = {}\n".format(MeanRMSE))
file2.write("STDR2 = {}\n".format(STDR2))
file2.write("STDMAE = {}\n".format(STDMAE))
file2.write("STDRMSE = {}\n".format(STDRMSE))
file2.write("R2Train = {}\n".format(R2Train))
file2.write("MAETrain = {}\n".format(MAETrain))
return MeanR2

name = "MLPHYPESearch"
file1 = open("{}_Hyperparameters.dat".format(name), "w")
file2 = open("{}_Results.dat".format(name), "w")
k = 0
while True:
    print("iteration number = {}".format(k))
    file1.write("#####")
    file2.write("#####")
    file1.write("iteration number = {}\n".format(k))
    file2.write("iteration number = {}\n".format(k))
    try:
        param = RandHypSearch()
        res = MLPR(param, X_train, y_train, X_test, y_test)
        file1.write("MeanR2 = {}\n".format(res))
        k += 1
        if res > 0.9999:
            print("Solution is found!")

```

```

        break
    except ZeroDivisionError:
        print("dijeljenje s nulom!")
    except ValueError:
        print('Overflow')

file1.close()
file2.close()

```

### B.3. Primjer MLP modela s unakrsnom validacijom pri power skaliranju na AMD datasetu

```

import numpy as np
import pandas as pd
import os
import random
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import cross_validate
data_AMD = pd.read_csv('AMD.csv')#.sample(frac=1)
data_AMD.drop(
    # Brisanje redova tako da sve pocinje od istog
    # datuma
    labels=range(0, (len(data_AMD)-10454)),
    axis=0,
    # 0 znaci da se brisu redovi
    inplace=True
)
data_AMD = data_AMD.reset_index(drop=True) #restart indexa od 0
data_AMD["Date"] = np.linspace(0, len(data_AMD["Date"])-
1, len(data_AMD["Date"]))
data_AMD['Date'] = data_AMD['Date'].astype(int)#datum u integer

#scaler
from sklearn.preprocessing import PowerTransformer
transformer = PowerTransformer().fit(data_AMD)
data2 = transformer.transform(data_AMD)
data_AMD = pd.DataFrame(data=data2, columns=data_AMD.columns)

y = data_AMD.pop("Close")
X = data_AMD
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state = 42)

def MLPParRandomSearch():
    parameters = []
    numHidLayers = random.randint(2,10)
    hidLayers = []
    for i in range(1, numHidLayers, 1):
        hidLayers.append(random.randint(10,100))
    tupleHid = tuple(hidLayers)

```



```

print(type(tupleHid))
actType =random.choice(['identity', 'logistic', 'tanh', 'relu'])
solver = random.choice(['lbfgs'])
alpha = random.uniform(0 , 0.001)
batch_size = random.randint(200,2000)
learn_rate = random.choice(['constant', 'invscaling', 'adaptive'])
learning_rate_init=random.uniform(0.001,10)
max_iter = random.randint(200,2000)
tol = random.uniform(0,1e-4)
warm_start=random.choice(['True','False'])
parameters.append(tupleHid)
parameters.append(actType)
parameters.append(solver)
parameters.append(alpha)
parameters.append(batch_size)
parameters.append(learn_rate)
parameters.append(learning_rate_init)
parameters.append(max_iter)
parameters.append(tol)
parameters.append(warm_start)

file0.write("{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\t{}\n".format(parameters[0],
parameters[1],
parameters[2],
parameters[3],
parameters[4],
parameters[5],
parameters[6],
parameters[7],
parameters[8],
parameters[9]))
file0.flush()
return parameters

def MLPRegCV(parameters, X_train,y_train,X_test, y_test):
    model = MLPRegressor(hidden_layer_sizes=parameters[0],
        activation=parameters[1],
        solver=parameters[2],
        alpha= parameters[3],
        batch_size=parameters[4],
        learning_rate=parameters[5],
        learning_rate_init=parameters[6],
        max_iter= parameters[7],
        tol = parameters[8],
        warm_start=parameters[9])
    cvmodel = cross_validate(model, X_train, y_train, cv=5,
        scoring = ("r2",
            "neg_mean_absolute_error",
            "neg_root_mean_squared_error"),
        return_train_score=True)

print("#####")
print("# Results from CV 5 Cross Validation Using Multiple Metric")

```

```

print("#####")
    print("All Scores From CV5 = {}".format(cvmodel))

#####
#Train Test Scores
#####
    file1.write("R2 Train Scores = {}\n".format(cvmodel['train_r2']))
    file1.write("R2 Test Scores = {}\n".format(cvmodel['test_r2']))
    file1.write("MAE Train Scores =
{}\n".format(abs(cvmodel['train_neg_mean_absolute_error'])))
    file1.write("MAE Test Scores =
{}\n".format(abs(cvmodel['test_neg_mean_absolute_error'])))
    file1.write("RMSE Train Scores =
{}\n".format(abs(cvmodel['train_neg_root_mean_squared_error'])))
    file1.write("RMSE Test Scores =
{}\n".format(abs(cvmodel['test_neg_root_mean_squared_error'])))

print("#####")
    print("# Calculate Mean and Standard Deviation of Metric values ")

print("#####")
    AvrR2ScoreTrain = np.mean(cvmodel['train_r2'])
    StdR2ScoreTrain = np.std(cvmodel['train_r2'])
    AvrR2ScoreTest = np.mean(cvmodel['test_r2'])
    StdR2ScoreTest = np.std(cvmodel['test_r2'])
    AvrAllR2Score = np.mean([AvrR2ScoreTrain,AvrR2ScoreTest])
    StdAllR2Score = np.std([AvrR2ScoreTrain, AvrR2ScoreTest])

    AvrMAEScoreTrain = np.mean(abs(cvmodel['train_neg_mean_absolute_error']))
    StdMAEScoreTrain = np.std(abs(cvmodel['train_neg_mean_absolute_error']))
    AvrMAEScoreTest = np.mean(abs(cvmodel['test_neg_mean_absolute_error']))
    StdMAEScoreTest = np.std(abs(cvmodel['test_neg_mean_absolute_error']))
    AvrAllMAEScore = np.mean([AvrMAEScoreTrain, AvrMAEScoreTest])
    StdAllMAEScore = np.std([AvrMAEScoreTrain, AvrMAEScoreTest])

    AvrRMSEScoreTrain =
np.mean(abs(cvmodel['train_neg_root_mean_squared_error']))
    StdRMSEScoreTrain =
np.std(abs(cvmodel['train_neg_root_mean_squared_error']))
    AvrRMSEScoreTest =
np.mean(abs(cvmodel['test_neg_root_mean_squared_error']))
    StdRMSEScoreTest =
np.std(abs(cvmodel['test_neg_root_mean_squared_error']))
    AvrAllRMSEScore = np.mean([AvrRMSEScoreTrain, AvrRMSEScoreTest])
    StdAllRMSEScore = np.std([AvrRMSEScoreTrain, AvrRMSEScoreTest])

    print("CV-R^2 Score = {}".format(AvrAllR2Score))
    print("CV-STD R^2 Score = {}".format(StdAllR2Score))
    print("CV-MAE Score = {}".format(AvrAllMAEScore))
    print("CV-STD MAE Score = {}".format(StdAllMAEScore))
    print("CV-RMSE Score ={}".format(AvrAllRMSEScore))
    print("CV-STD RMSE Score = {}".format(StdAllRMSEScore))

file1.write("#####\n
"+
    "AvrR2Score Train = {}\n".format(AvrR2ScoreTrain)+\
    "StdR2Score Train = {}\n".format(StdR2ScoreTrain)+\
    "AvrR2Score Test = {}\n".format(AvrR2ScoreTest)+\
    "StdR2Score Test = {}\n".format(StdR2ScoreTest)+\
    "AvrAllR2Score = {}\n".format(AvrAllR2Score)+\
    "StdAllR2Score = {}\n".format(StdAllR2Score)+\
    "AvrMAEScore Train = {}\n".format(AvrMAEScoreTrain)+\

```

```

        "StdMAEScore Train = {}".format(StdMAEScoreTrain)+\
        "AvrMAEScore Test = {}".format(AvrMAEScoreTest)+\
        "StdMAEScore Test = {}".format(StdMAEScoreTest)+\
        "AvrAllMAEScore = {}".format(AvrAllMAEScore)+\
        "StdAllMAEScore = {}".format(StdAllMAEScore)+\
        "AvrRMSEScore Train = {}".format(AvrRMSEScoreTrain)+\
        "StdRMSEScore Train = {}".format(StdRMSEScoreTrain)+\
        "AvrRMSEScore Test = {}".format(AvrRMSEScoreTest)+\
        "StdRMSEScore Test = {}".format(StdRMSEScoreTest)+\
        "AvrAllRMSEScore = {}".format(AvrAllRMSEScore)+\
        "StdAllRMSEScore = {}".format(StdAllRMSEScore)+\

#####\n")
    if AvrAllR2Score > 0.99:

print("#####")
    print(" Final Evaluation")

print("#####")

file1.write("#####\n")
    file1.write(" Final Evaluation\n")

file1.write("#####\n")
    model.fit(X_train,y_train)
    R2Test = model.score(X_test,y_test)
    MAETest = mean_absolute_error(y_test, model.predict(X_test))
    RMSETest = np.sqrt(mean_squared_error(y_test, model.predict(X_test)))
    print("R^2 Test = {}".format(R2Test))
    print("MAE Test = {}".format(MAETest))
    print("RMSE Test = {}".format(RMSETest))

file1.write("#####\n")
    file1.write("model R^2 Test = {}".format(R2Test))
    file1.write("MAE Test = {}".format(MAETest))
    file1.write("RMSE Test = {}".format(RMSETest))

file1.write("#####\n")
    file1.flush()
    return R2Test
else:
    return AvrAllR2Score

name = "MLP_CV5"
file0 = open("{}_parameters.data".format(name), "w")
file1 = open("{}_results.data".format(name), "w")
test = MLPParRandomSearch()
print(test)
k = 0
while True:
    print("Current Iteration = {}".format(k))
    Param = MLPParRandomSearch()
    test = MLPRegCV(Param,X_train,y_train, X_test, y_test)
    k+=1
    if test > 0.999:
        print("Solution is Found!!")
        file1.write("Solution is Found!")
        file1.flush()
        break
    else:
        continue

```

```
file0.close()
file1.close()
```

#### B.4. Primjer StackingRegressor ansambla na AMD datasetu

```
import numpy as np
import pandas as pd
import os
import random
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
from sklearn.linear_model import ARDRegression
from sklearn.model_selection import cross_validate

from sklearn.linear_model import LassoLars
from sklearn.linear_model import Lars
from sklearn.linear_model import ARDRegression
from sklearn.linear_model import BayesianRidge
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import StackingRegressor

data_AMD= pd.read_csv('AMD.csv')#.sample(frac=1)
data_AMD.drop(          # Brisanje redova tako da sve pocinje od istog
datuma
    labels=range(0, (len(data_AMD)-10454)),
    axis=0,          # 0 znaci da se brisu redovi
    inplace=True
)
data_AMD=data_AMD.reset_index(drop=True) #restart indexa od 0
data_AMD["Date"]= np.linspace(0,len(data_AMD["Date"])-
1,len(data_AMD["Date"]))
data_AMD['Date'] = data_AMD['Date'].astype(int)#datum u integer

y = data_AMD.pop("Close")
X = data_AMD
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,
random_state = 42)

def StackingEnsamble(X_train,X_test, y_train, y_test):
    def ARDRandomizedGridSearch():
        parametersList = []
        NumIter = random.randint(300,4000)
        tolerance = random.uniform(1e-6,1e-2)
        alpha1 = random.uniform(1e-6,1e-2)
        alpha2 = random.uniform(1e-6,1e-2)
        lambda1 = random.uniform(1e-6,1e-2)
        lambda2 = random.uniform(1e-6,1e-2)
        thresholdLambda = random.randint(1000,100000)
        parametersList = [NumIter,
            tolerance,
            alpha1,
            alpha2,
            lambda1,
            lambda2,
            thresholdLambda]
```

```

    print("ARDR list of randomly chosen parameters =
{}").format(parametersList))
    file0.write("ARD_PARAMETERS = " + str(NumIter)+"\t"+\
        str(tolerance)+"\t"+\
        str(alpha1)+"\t"+\
        str(alpha2)+"\t"+\
        str(lambda1)+"\t"+\
        str(lambda2)+"\t"+\
        str(thresholdLambda)+"\n")
    file0.flush()
    return parametersList
def BayesianRandomizedSearch():
    parameters = []
    n_iter = random.randint(300,2000)
    tol= random.uniform(0,1e-2)
    alpha_1 = random.uniform(0,1e-2)
    alpha_2 = random.uniform(0,1e-2)
    lambda_1 = random.uniform(0,1e-2)
    lambda_2 = random.uniform(0,1e-2)

    parameters = [n_iter,
        tol,
        alpha_1,
        alpha_2,
        lambda_1,
        lambda_2]
    print("Bayes Chosen parameters = {}".format(parameters))
    file0.write("Bayes_PARAM = " + str(n_iter)+"\t"+\
        str(tol)+"\t"+\
        str(alpha_1)+"\t"+\
        str(alpha_2)+"\t"+\
        str(lambda_1)+"\t"+\
        str(lambda_2)+"\n")
    file0.flush()
    return parameters

def LassoRandomParam():
    parameters = []
    alpha = random.uniform(0,1)
    max_iter= random.randint(1000,9000)
    warm_start = random.choice(['True','False'])
    positive = random.choice(['True','False'])
    tol=random.uniform(1e-7,1e-4)
    parameters = [alpha,
        max_iter,
        warm_start,
        positive,
        tol]
    print("Lasso Chosen parameters = {}".format(parameters))
    file0.write("Lass_PARAM = " + str(alpha)+"\t"+\
        str(max_iter)+"\t"+\
        str(warm_start)+"\t"+\
        str(positive)+"\t"+\
        str(tol)+"\n")
    file0.flush()
    return parameters

def LinearRandomizedGridSearch():
    parameters = []
    fitIntercept = random.choice([True, False])
    parameters = [fitIntercept]
    print("Linear Chosen Parameters = {}".format(parameters[0]))
    file0.write("Linear = " + str(fitIntercept)+"\n")

```

```

file0.flush()
return parameters

def MLPParRandomSearch():
    parameters = []
    numHidLayers = random.randint(2,10)
    hidLayers = []
    for i in range(1,numHidLayers,1):
        hidLayers.append(random.randint(10,100))
    tupleHid = tuple(hidLayers)
    print(type(tupleHid))
    actType =random.choice(['identity', 'logistic', 'tanh', 'relu'])
    solver = random.choice(['lbfgs'])
    alpha = random.uniform(0 , 0.001)
    batch_size = random.randint(200,2000)
    learn_rate = random.choice(['constant', 'invscaling', 'adaptive'])
    learning_rate_init=random.uniform(0.001,10)
    max_iter = random.randint(200,2000)
    tol = random.uniform(0,1e-4)
    warm_start=random.choice(['True', 'False'])

    parameters.append(tupleHid)
    parameters.append(actType)
    parameters.append(solver)
    parameters.append(alpha)
    parameters.append(batch_size)
    parameters.append(learn_rate)
    parameters.append(learning_rate_init)
    parameters.append(max_iter)
    parameters.append(tol)
    parameters.append(warm_start)

    print("MLP Chosen parameters = {}".format(parameters))
    file0.write(" MLP_PARAM =
    {}\\t{}\\t{}\\t{}\\t{}\\t{}\\t{}\\t{}\\t{}\\n".format(parameters[0],
parameters[1],
parameters[2],
parameters[3],
parameters[4],
parameters[5],
parameters[6],
parameters[7],
parameters[8],
parameters[9]))
    file0.flush()
    return parameters
def LARSRandomizedSearch():
    parametersList = []
    eps = random.uniform(0,1)
    n_nonzero_coefs= random.randint(1,1000)
    jitter = random.choice(['None', 'None'])
    parametersList = [eps,

```

```

        n_nonzero_coefs,
        jitter]
    print("LARS list of randomly chosen parameters =
{}").format(parametersList)
    file0.write("Lars_PARAM = " + str(eps)+"\t"+\
                str(n_nonzero_coefs)+"\t"+\
                str(jitter)+"\n")
    file0.flush()
    return parametersList

def LASSLARSRandomizedSearch():
    parameters = []
    alpha = random.uniform(0,0.1)
    max_iter= random.randint(1000,5000)
    positive = random.choice(['True','False'])
    jitter = random.uniform(0,0.00001)
    eps = random.uniform(0,0.0001)
    parameters = [alpha,
                  max_iter,
                  positive,
                  jitter,
                  eps]
    print("LassoLars Chosen parameters = {}".format(parameters))
    file0.write("LassLars_PARAM = " + str(alpha)+"\t"+\
                str(max_iter)+"\t"+\
                str(positive)+"\t"+\
                str(jitter)+"\t"+\
                str(eps)+"\n")
    file0.flush()
    return parameters

```

```

ARDParam = ARDRandomizedGridSearch()
BayesianParam = BayesianRandomizedSearch()
LassoParam = LassRandomParam()
LinParam = LinearRandomizedGridSearch()
MLPParam = MLPParRandomSearch()
LarsParam = LARSRandomizedSearch()
LassLarsParam = LASSLARSRandomizedSearch()

```

```

estimators = [('ARD', ARDRegression(n_iter = ARDParam[0],
                                   tol = ARDParam[1],
                                   alpha_1 = ARDParam[2],
                                   alpha_2 = ARDParam[3],
                                   lambda_1 = ARDParam[4],
                                   lambda_2 = ARDParam[5],
                                   threshold_lambda = ARDParam[6])),
              ('Br', BayesianRidge(n_iter=BayesianParam[0],
                                   tol=BayesianParam[1],
                                   alpha_1=BayesianParam[2],
                                   alpha_2=BayesianParam[3],
                                   lambda_1=BayesianParam[4],
                                   lambda_2=BayesianParam[5])),
              ('Lr', Lasso(alpha=LassoParam[0],
                           max_iter=LassoParam[1],
                           warm_start=LassoParam[2],
                           positive=LassoParam[3],
                           tol=LassoParam[4])),
              ('LinR', LinearRegression(fit_intercept=False)),
              ('MLP', MLPRegressor(hidden_layer_sizes=MLPParam[0],

```

```

        activation=MLPParam[1],
        solver=MLPParam[2],
        alpha= MLPParam[3],
        batch_size=MLPParam[4],
        learning_rate=MLPParam[5],
        learning_rate_init=MLPParam[6],
        max_iter= MLPParam[7],
        tol = MLPParam[8],
        warm_start=MLPParam[9])),
    ('Lars', Lars (eps=LarsParam[0],
                  n_nonzero_coefs=LarsParam[1],
                  jitter=None)),
    ('LassoLars', LassoLars (alpha=LassLarsParam[0],
                             max_iter=LassLarsParam[1],
                             positive=LassLarsParam[2],
                             jitter=LassLarsParam[3],
                             eps=LassLarsParam[4]))])

#Parametri algoritma

model = StackingRegressor(estimators=estimators,
                          final_estimator = None,
                          # cv='prefit'
                          )
cvmodel = cross_validate(model, X_train, y_train, cv=5,
                         scoring = ("r2",
                                    "neg_mean_absolute_error",
                                    "neg_root_mean_squared_error"),
                         return_train_score=True)

print("#####")
print("# Results from CV 5 Cross Validation Using Multiple Metric")

print("#####")
print("All Scores From CV5 = {}".format(cvmodel))

#Train Test Scores

file1.write("R2 Train Scores = {}\n".format(cvmodel['train_r2']))
file1.write("R2 Test Scores = {}\n".format(cvmodel['test_r2']))
file1.write("MAE Train Scores =
{}\n".format(abs(cvmodel['train_neg_mean_absolute_error'])))
file1.write("MAE Test Scores =
{}\n".format(abs(cvmodel['test_neg_mean_absolute_error'])))
file1.write("RMSE Train Scores =
{}\n".format(abs(cvmodel['train_neg_root_mean_squared_error'])))
file1.write("RMSE Test Scores =
{}\n".format(abs(cvmodel['test_neg_root_mean_squared_error'])))

print("#####")
print("# Calculate Mean and Standard Deviation of Metric values ")

print("#####")
AvrR2ScoreTrain = np.mean(cvmodel['train_r2'])
StdR2ScoreTrain = np.std(cvmodel['train_r2'])
AvrR2ScoreTest = np.mean(cvmodel['test_r2'])
StdR2ScoreTest = np.std(cvmodel['test_r2'])
AvrAllR2Score = np.mean([AvrR2ScoreTrain,AvrR2ScoreTest])
StdAllR2Score = np.std([AvrR2ScoreTrain, AvrR2ScoreTest])

AvrMAEScoreTrain = np.mean(abs(cvmodel['train_neg_mean_absolute_error']))
StdMAEScoreTrain = np.std(abs(cvmodel['train_neg_mean_absolute_error']))
AvrMAEScoreTest = np.mean(abs(cvmodel['test_neg_mean_absolute_error']))

```



```

StdMAEScoreTest = np.std(abs(cvmodel['test_neg_mean_absolute_error']))
AvrAllMAEScore = np.mean([AvrMAEScoreTrain, AvrMAEScoreTest])
StdAllMAEScore = np.std([AvrMAEScoreTrain, AvrMAEScoreTest])

AvrRMSEScoreTrain =
np.mean(abs(cvmodel['train_neg_root_mean_squared_error']))
StdRMSEScoreTrain =
np.std(abs(cvmodel['train_neg_root_mean_squared_error']))
AvrRMSEScoreTest =
np.mean(abs(cvmodel['test_neg_root_mean_squared_error']))
StdRMSEScoreTest =
np.std(abs(cvmodel['test_neg_root_mean_squared_error']))
AvrAllRMSEScore = np.mean([AvrRMSEScoreTrain, AvrRMSEScoreTest])
StdAllRMSEScore = np.std([AvrRMSEScoreTrain, AvrRMSEScoreTest])

print("CV-R^2 Score = {}".format(AvrAllR2Score))
print("CV-STD R^2 Score = {}".format(StdAllR2Score))
print("CV-MAE Score = {}".format(AvrAllMAEScore))
print("CV-STD MAE Score = {}".format(StdAllMAEScore))
print("CV-RMSE Score = {}".format(AvrAllRMSEScore))
print("CV-STD RMSE Score = {}".format(StdAllRMSEScore))

file1.write("#####\n"
"+\
    AvrR2Score Train = {}\n".format(AvrR2ScoreTrain)+\
    StdR2Score Train = {}\n".format(StdR2ScoreTrain)+\
    AvrR2Score Test = {}\n".format(AvrR2ScoreTest)+\
    StdR2Score Test = {}\n".format(StdR2ScoreTest)+\
    AvrAllR2Score = {}\n".format(AvrAllR2Score)+\
    StdAllR2Score = {}\n".format(StdAllR2Score)+\
    AvrMAEScore Train = {}\n".format(AvrMAEScoreTrain)+\
    StdMAEScore Train = {}\n".format(StdMAEScoreTrain)+\
    AvrMAEScore Test = {}\n".format(AvrMAEScoreTest)+\
    StdMAEScore Test = {}\n".format(StdMAEScoreTest)+\
    AvrAllMAEScore = {}\n".format(AvrAllMAEScore)+\
    StdAllMAEScore = {}\n".format(StdAllMAEScore)+\
    AvrRMSEScore Train = {}\n".format(AvrRMSEScoreTrain)+\
    StdRMSEScore Train = {}\n".format(StdRMSEScoreTrain)+\
    AvrRMSEScore Test = {}\n".format(AvrRMSEScoreTest)+\
    StdRMSEScore Test = {}\n".format(StdRMSEScoreTest)+\
    AvrAllRMSEScore = {}\n".format(AvrAllRMSEScore)+\
    StdAllRMSEScore = {}\n".format(StdAllRMSEScore)+\

#####\n")
if AvrAllR2Score > 0.9999:

print("#####")
    print(" Final Evaluation")

print("#####")

file1.write("#####\n"
n")
    file1.write(" Final Evaluation\n")

file1.write("#####\n"
\n")
    model.fit(X_train,y_train)
    R2Test = model.score(X_test,y_test)
    MAETest = mean_absolute_error(y_test, model.predict(X_test))

```

```

RMSETest = np.sqrt(mean_squared_error(y_test, model.predict(X_test)))
print("R^2 Test = {}".format(R2Test))
print("MAE Test = {}".format(MAETest))
print("RMSE Test = {}".format(RMSETest))

file1.write("#####\n")
file1.write("Final R^2 Test = {}\n".format(R2Test))
file1.write("Final MAE Test = {}\n".format(MAETest))
file1.write("Final RMSE Test = {}\n".format(RMSETest))

file1.write("#####\n")
file1.flush()
return R2Test
else:
return AvrAllR2Score
name = "AMD_bez_skaliranja"
#####
file0 = open("{}_CV5_Param.dat".format(name), "w")
file1 = open("{}_CV5_score.dat".format(name), 'w')

k = 0
res_best=0
while True:
print("Current Iteration = {}".format(k))
k+=1
res = StackingEnsamble(X_train, X_test, y_train, y_test)
if res>res_best:
res_best=res
print('Najbolji rez do sad:',res_best, 'Trenutni rez',res)

if res > 0.999:
print("Solution is Found!")
break
else:
continue
file1.close()

```

## B.5. Primjer VotingRegressor ansambla na AMD datasetu

```
import numpy as np
import pandas as pd
import os
import random
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
from sklearn.linear_model import ARDRegression
from sklearn.model_selection import cross_validate

from sklearn.linear_model import LassoLars
from sklearn.linear_model import Lars
from sklearn.linear_model import ARDRegression
from sklearn.linear_model import BayesianRidge
from sklearn.linear_model import Lasso
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.ensemble import VotingRegressor

data_AMD= pd.read_csv('AMD.csv')#.sample(frac=1)
data_AMD.drop(          # Brisanje redova tako da sve pocinje od istog
datuma
    labels=range(0, (len(data_AMD)-10454)),
    axis=0,          # 0 znaci da se brisu redovi
    inplace=True
)
data_AMD=data_AMD.reset_index(drop=True) #restart indexa od 0
data_AMD["Date"]= np.linspace(0,len(data_AMD["Date"])-
1,len(data_AMD["Date"]))
data_AMD['Date'] = data_AMD['Date'].astype(int)#datum u integer

y = data_AMD.pop("Close")
X = data_AMD
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,
random_state = 42)

def VotingEnsamble(X_train,X_test, y_train, y_test):
    def ARDRandomizedGridSearch():
        parametersList = []
        NumIter = random.randint(300,4000)
        tolerance = random.uniform(1e-6,1e-2)
        alpha1 = random.uniform(1e-6,1e-2)
        alpha2 = random.uniform(1e-6,1e-2)
        lambda1 = random.uniform(1e-6,1e-2)
        lambda2 = random.uniform(1e-6,1e-2)
        thresholdLambda = random.randint(1000,100000)
        parametersList = [NumIter,
            tolerance,
            alpha1,
            alpha2,
            lambda1,
            lambda2,
            thresholdLambda]

        print("ARDR list of randomly chosen parameters =
{}".format(parametersList))
        file0.write("ARD_PARAMETERS = " + str(NumIter)+"\t"+\
```

```

        str(tolerance)+"\t"+\
        str(alpha1)+"\t"+\
        str(alpha2)+"\t"+\
        str(lambda1)+"\t"+\
        str(lambda2)+"\t"+\
        str(thresholdLambda)+"\n")
file0.flush()
return parametersList
def BayesianRandomizedSearch():
parameters = []
n_iter = random.randint(300,2000)
tol= random.uniform(0,1e-2)
alpha_1 = random.uniform(0,1e-2)
alpha_2 = random.uniform(0,1e-2)
lambda_1 = random.uniform(0,1e-2)
lambda_2 = random.uniform(0,1e-2)

parameters = [n_iter,
              tol,
              alpha_1,
              alpha_2,
              lambda_1,
              lambda_2]

print("Bayes Chosen parameters = {}".format(parameters))
file0.write("Bayes_PARAM = " + str(n_iter)+"\t"+\
           str(tol)+"\t"+\
           str(alpha_1)+"\t"+\
           str(alpha_2)+"\t"+\
           str(lambda_1)+"\t"+\
           str(lambda_2)+"\n")

file0.flush()
return parameters

def LassRandomParam():
parameters = []
alpha = random.uniform(0,1)
max_iter= random.randint(1000,9000)
warm_start = random.choice(['True', 'False'])
positive = random.choice(['True', 'False'])
tol=random.uniform(1e-7,1e-4)
parameters = [alpha,
              max_iter,
              warm_start,
              positive,
              tol]

print("Lasso Chosen parameters = {}".format(parameters))
file0.write("Lass_PARAM = " + str(alpha)+"\t"+\
           str(max_iter)+"\t"+\
           str(warm_start)+"\t"+\
           str(positive)+"\t"+\
           str(tol)+"\n")

file0.flush()
return parameters

def LinearRandomizedGridSearch():
parameters = []
fitIntercept = random.choice([True, False])
parameters = [fitIntercept]
print("Linear Chosen Parameters = {}".format(parameters[0]))
file0.write("Linear = " + str(fitIntercept)+"\n")
file0.flush()
return parameters

```

```

def MLPParRandomSearch():
    parameters = []
    numHidLayers = random.randint(2,10)
    hidLayers = []
    for i in range(1,numHidLayers,1):
        hidLayers.append(random.randint(10,100))
    tupleHid = tuple(hidLayers)
    print(type(tupleHid))
    actType =random.choice(['identity', 'logistic', 'tanh', 'relu'])
    solver = random.choice(['lbfgs'])
    alpha = random.uniform(0 , 0.001)
    batch_size = random.randint(200,2000)
    learn_rate = random.choice(['constant', 'invscaling', 'adaptive'])
    learning_rate_init=random.uniform(0.001,10)
    max_iter = random.randint(200,2000)
    tol = random.uniform(0,1e-4)
    warm_start=random.choice(['True', 'False'])

    parameters.append(tupleHid)
    parameters.append(actType)
    parameters.append(solver)
    parameters.append(alpha)
    parameters.append(batch_size)
    parameters.append(learn_rate)
    parameters.append(learning_rate_init)
    parameters.append(max_iter)
    parameters.append(tol)
    parameters.append(warm_start)

    print("MLP Chosen parameters = {}".format(parameters))
    file0.write(" MLP_PARAM =
    {} \t {} \t {} \t {} \t {} \t {} \t {} \t {} \t {} \n".format(parameters[0],
parameters[1],
parameters[2],
parameters[3],
parameters[4],
parameters[5],
parameters[6],
parameters[7],
parameters[8],
parameters[9]))
    file0.flush()
    return parameters
def LARSRandomizedSearch():
    parametersList = []
    eps = random.uniform(0,1)
    n_nonzero_coefs= random.randint(1,1000)
    jitter = random.choice(['None', 'None'])
    parametersList = [eps,
n_nonzero_coefs,
jitter]

```

```

    print("LARS list of randomly chosen parameters =
{}".format(parametersList))
    file0.write("Lars_PARAM = " + str(eps)+"\t"+\
                str(n_nonzero_coefs)+"\t"+\
                str(jitter)+"\n")
    file0.flush()
    return parametersList

def LASSLARSRandomizedSearch():
    parameters = []
    alpha = random.uniform(0,0.1)
    max_iter= random.randint(1000,5000)
    positive = random.choice(['True','False'])
    jitter = random.uniform(0,0.00001)
    eps = random.uniform(0,0.0001)
    parameters = [alpha,
                  max_iter,
                  positive,
                  jitter,
                  eps]
    print("LassoLars Chosen parameters = {}".format(parameters))
    file0.write("LassLars_PARAM = " + str(alpha)+"\t"+\
                str(max_iter)+"\t"+\
                str(positive)+"\t"+\
                str(jitter)+"\t"+\
                str(eps)+"\n")
    file0.flush()
    return parameters

```

```

ARDParam = ARDRandomizedGridSearch()
BayesianParam = BayesianRandomizedSearch()
LassoParam = LassRandomParam()
LinParam = LinearRandomizedGridSearch()
MLPParam = MLPParRandomSearch()
LarsParam = LARSRandomizedSearch()
LassLarsParam = LASSLARSRandomizedSearch()

```

```

estimators = [('ARD', ARDRegression(n_iter = ARDParam[0],
                                   tol = ARDParam[1],
                                   alpha_1 = ARDParam[2],
                                   alpha_2 = ARDParam[3],
                                   lambda_1 = ARDParam[4],
                                   lambda_2 = ARDParam[5],
                                   threshold_lambda = ARDParam[6])),
              ('Br', BayesianRidge(n_iter=BayesianParam[0],
                                   tol=BayesianParam[1],
                                   alpha_1=BayesianParam[2],
                                   alpha_2=BayesianParam[3],
                                   lambda_1=BayesianParam[4],
                                   lambda_2=BayesianParam[5])),
              ('Lr', Lasso(alpha=LassoParam[0],
                           max_iter=LassoParam[1],
                           warm_start=LassoParam[2],
                           positive=LassoParam[3],
                           tol=LassoParam[4])),
              ('LinR', LinearRegression(fit_intercept=False)),
              ('MLP', MLPRegressor(hidden_layer_sizes=MLPParam[0],
                                   activation=MLPParam[1],
                                   solver=MLPParam[2],

```

```

        alpha= MLPParam[3],
        batch_size=MLPParam[4],
        learning_rate=MLPParam[5],
        learning_rate_init=MLPParam[6],
        max_iter= MLPParam[7],
        tol = MLPParam[8],
        warm_start=MLPParam[9])),
    ('Lars', Lars(eps=LarsParam[0],
                 n_nonzero_coefs=LarsParam[1],
                 jitter=None)),
    ('LassoLars', LassoLars(alpha=LassLarsParam[0],
                           max_iter=LassLarsParam[1],
                           positive=LassLarsParam[2],
                           jitter=LassLarsParam[3],
                           eps=LassLarsParam[4]))]

#Parametri algoritma

model = VotingRegressor(estimators=estimators)
cvmodel = cross_validate(model, X_train, y_train, cv=5,
                        scoring = ("r2",
                                   "neg_mean_absolute_error",
                                   "neg_root_mean_squared_error"),
                        return_train_score=True)

print("#####")
print("# Results from CV 5 Cross Validation Using Multiple Metric")

print("#####")
print("All Scores From CV5 = {}".format(cvmodel))

#Train Test Scores

file1.write("R2 Train Scores = {}\n".format(cvmodel['train_r2']))
file1.write("R2 Test Scores = {}\n".format(cvmodel['test_r2']))
file1.write("MAE Train Scores =
{}\n".format(abs(cvmodel['train_neg_mean_absolute_error'])))
file1.write("MAE Test Scores =
{}\n".format(abs(cvmodel['test_neg_mean_absolute_error'])))
file1.write("RMSE Train Scores =
{}\n".format(abs(cvmodel['train_neg_root_mean_squared_error'])))
file1.write("RMSE Test Scores =
{}\n".format(abs(cvmodel['test_neg_root_mean_squared_error'])))

print("#####")
print("# Calculate Mean and Standard Deviation of Metric values ")

print("#####")
AvrR2ScoreTrain = np.mean(cvmodel['train_r2'])
StdR2ScoreTrain = np.std(cvmodel['train_r2'])
AvrR2ScoreTest = np.mean(cvmodel['test_r2'])
StdR2ScoreTest = np.std(cvmodel['test_r2'])
AvrAllR2Score = np.mean([AvrR2ScoreTrain,AvrR2ScoreTest])
StdAllR2Score = np.std([AvrR2ScoreTrain, AvrR2ScoreTest])

AvrMAEScoreTrain = np.mean(abs(cvmodel['train_neg_mean_absolute_error']))
StdMAEScoreTrain = np.std(abs(cvmodel['train_neg_mean_absolute_error']))
AvrMAEScoreTest = np.mean(abs(cvmodel['test_neg_mean_absolute_error']))
StdMAEScoreTest = np.std(abs(cvmodel['test_neg_mean_absolute_error']))
AvrAllMAEScore = np.mean([AvrMAEScoreTrain, AvrMAEScoreTest])
StdAllMAEScore = np.std([AvrMAEScoreTrain, AvrMAEScoreTest])

```

```

    AvrRMSEScoreTrain =
np.mean(abs(cvmodel['train_neg_root_mean_squared_error']))
    StdRMSEScoreTrain =
np.std(abs(cvmodel['train_neg_root_mean_squared_error']))
    AvrRMSEScoreTest =
np.mean(abs(cvmodel['test_neg_root_mean_squared_error']))
    StdRMSEScoreTest =
np.std(abs(cvmodel['test_neg_root_mean_squared_error']))
    AvrAllRMSEScore = np.mean([AvrRMSEScoreTrain, AvrRMSEScoreTest])
    StdAllRMSEScore = np.std([AvrRMSEScoreTrain, AvrRMSEScoreTest])

    print("CV-R^2 Score = {}".format(AvrAllR2Score))
    print("CV-STD R^2 Score = {}".format(StdAllR2Score))
    print("CV-MAE Score = {}".format(AvrAllMAEScore))
    print("CV-STD MAE Score = {}".format(StdAllMAEScore))
    print("CV-RMSE Score = {}".format(AvrAllRMSEScore))
    print("CV-STD RMSE Score = {}".format(StdAllRMSEScore))

file1.write("#####\n
"+\
            "AvrR2Score Train = {}\n".format(AvrR2ScoreTrain)+\
            "StdR2Score Train = {}\n".format(StdR2ScoreTrain)+\
            "AvrR2Score Test = {}\n".format(AvrR2ScoreTest)+\
            "StdR2Score Test = {}\n".format(StdR2ScoreTest)+\
            "AvrAllR2Score = {}\n".format(AvrAllR2Score)+\
            "StdAllR2Score = {}\n".format(StdAllR2Score)+\
            "AvrMAEScore Train = {}\n".format(AvrMAEScoreTrain)+\
            "StdMAEScore Train = {}\n".format(StdMAEScoreTrain)+\
            "AvrMAEScore Test = {}\n".format(AvrMAEScoreTest)+\
            "StdMAEScore Test = {}\n".format(StdMAEScoreTest)+\
            "AvrAllMAEScore = {}\n".format(AvrAllMAEScore)+\
            "StdAllMAEScore = {}\n".format(StdAllMAEScore)+\
            "AvrRMSEScore Train = {}\n".format(AvrRMSEScoreTrain)+\
            "StdRMSEScore Train = {}\n".format(StdRMSEScoreTrain)+\
            "AvrRMSEScore Test = {}\n".format(AvrRMSEScoreTest)+\
            "StdRMSEScore Test = {}\n".format(StdRMSEScoreTest)+\
            "AvrAllRMSEScore = {}\n".format(AvrAllRMSEScore)+\
            "StdAllRMSEScore = {}\n".format(StdAllRMSEScore)+\

            "#####\n")
    if AvrAllR2Score > 0.9999:

print("#####")
    print(" Final Evaluation")

print("#####")

file1.write("#####\n")
    file1.write(" Final Evaluation\n")

file1.write("#####\n")
    model.fit(X_train,y_train)
    R2Test = model.score(X_test,y_test)
    MAETest = mean_absolute_error(y_test, model.predict(X_test))
    RMSETest = np.sqrt(mean_squared_error(y_test, model.predict(X_test)))
    print("R^2 Test = {}".format(R2Test))
    print("MAE Test = {}".format(MAETest))
    print("RMSE Test = {}".format(RMSETest))

```



```

file1.write("#####\n")
    file1.write("Final R^2 Test = {}\n".format(R2Test))
    file1.write("Final MAE Test = {}\n".format(MAETest))
    file1.write("Final RMSE Test = {}\n".format(RMSETest))

file1.write("#####\n")
    file1.flush()
    return R2Test
else:
    return AvrAllR2Score

name = "AMD_bez_skaliranje"

#####

file0 = open("{}_CV5_Param.dat".format(name), "w")
file1 = open("{}_CV5_score.dat".format(name), 'w')
k = 0
res_best=0
while True:
    print("Current Iteration = {}".format(k))
    k+=1
    res = VotingEnsamble(X_train, X_test, y_train, y_test)
    if res>res_best:
        res_best=res
    print('Najbolji rez do sad:',res_best, 'Trenutni rez',res)

    if res > 0.99:
        print("Solution is Found!")
        break
    else:
        continue
file1.close()

```