Komparativna analiza metoda iz mašinskog učenja za predikcije na berzi

Rijad Mutapčić, Emin Mulaimović

Februar 2024

Sažetak

U ovom radu koristit ćemo različite vrste modela dubokog učenja kako bismo predvidjeli kretanja zatvarajućih cijena(closing price) dionica koje se nalaze na Nasdaqu, koristeći podatke iz knjige naloga(order book) i završne aukcije(closing auction) dionica. Predviđat ćemo sintetički cilj koji predstavlja budući pomak od 60 sekundi u težinska prosječna cijena(wap) dionice, umanjen za budući pomak od 60 sekundi sintetičkog indeksa. Korišteni modeli su neuralna mreža (NN), konvolucijska neuralna mreža (CNN), rekurentna neuralna mreža (RNN), light-gradient boost (LGB) i catboost.

1 Uvod

Gotovo 10% dnevne trgovine na Nasdaq se izvršava na završnoj aukciji.Berza je dinamično i visokorizično okruženje gdje svaka sekunda ima važnost. Intenzitet se povećava kako se dan trgovanja bliži kraju, dostižući vrhunac u kritičnih posljednjih deset minuta. U zadnjih 10 minuta trgovci spajaju podatke o knjige naloga i završne aukcije, koji je ključan za pružanje najboljih cijena svim učesnicima na tržištu.

Predviđanje kretanja dionice na tržištu pružilo bi ogromnu prednost onima koji imaju pristup istim. Vrlo je teško predviđati zbog nepredvidljivosti i volatilnosti na berzama. Predviđati ćemo sintetički cilj koji predstavlja budući pomak od 60 sekundi od težinska prosječna cijene(wap) dionice, umanjen za budući pomak od 60 sekundi sintetičkog indeksa.

U prvom dijelu rada objasnili smo problem na kojem ćemo vršiti komparativnu analizu. Sljedeći dio je metodologija u kojoj ćemo objasniti sve metode mašinskog učenja koje ćemo koristiti u ovom radu. Na kraju ćemo prikazati rezultate tih mreža i izvesti zaključak.

2 Metodologija

Podaci koji ćemo koristiti u ovom problemu dolaze iz Kaggle takmičenja (Optiver - Trading at the Close). Naš cilj kao i cilj takmičenja je da razvijemo model sposoban za predviđanje kretanja završnih cijena na vise od stotinu dionica. U svrhu toga ćemo koristiti podatke iz knjige naloga. Ovaj skup podataka sadrži historijske podatke zadnjih deset minuta završnu aukcije na berzi dionica iz NASDAQ. U podacima se nalazi 180 dionica gdje o svakoj dionci imamo informacije zadnjih 10 minuta u intervalima od 10 sekundi. Ove podatke imamo za 480 dana i za svaki interval imamo sljedeće kolone:

- **id dionice:** Identifikator za svaku dionicu
- **id datuma:** Identifikator za svaki datum
- **velicina imbalansa:** Broj dionica na zatvaranju koji traže dodatnu likvidnost za trenutnu referentnu cijenu.
- **oznaka imbalansa:** Predstavlja da li je više kupljenih od prodatih, suprotno ili nema promjene:
  - više kupljenih nego prodatih: 1
  - više prodatih nego kupljenih: -1
  - isto prodatih i kupljenih (ovo su uparene dionice): 0

- **referentna cijena:** Cijena na kojoj su uparene dionice maksimizirane, neuravnoteženost je minimizirana i udaljenost od srednjeg cjenovnog nivoa kupovine i prodaje je minimizirana, tim redom.
- **bliska cijena:** Cijena koja će maksimizirati broj uparenih dionica na osnovu aukcijskih i tržišnih naloga.
- **daleka cijena:** Cijena prelaska koja će maksimizirati broj uparenih dionica na osnovu samo aukcijskog interesa.
- **kupovna cijena:** Najveća cijena za koju je kupljena dionica u tom vremenskom intervalu.
- **prodajna cijena:** Najveća cijena za koju je prodata dionica u tom vremenskom intervalu.
- **kupovna velicina:** Najveća količina dionica kupljena u tom vremenskom intervalu.
- **prodajna velicina:** Najveća količina dionica prodana u tom vremenskom intervalu.
- **wap (weighted average price):** Cijena koja se dobije sa (1).

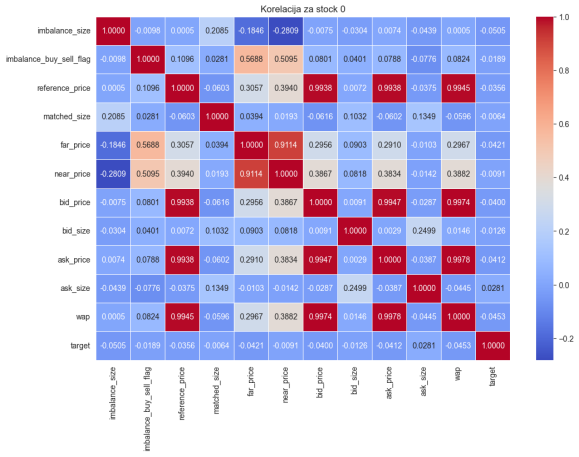
$$wap = \frac{prodajna\_cijena \times kupovna\_velicina + kupovna\_cijena \times prodajna\_velicina}{kupovna\_velicina + prodajna\_velicina}. \quad (1)$$

- **broj sekundi u kanti:** Broj sekundi nakon što je počela otvorena aukcija.
- **target:** Ovo predstavlja kako će wap izgledati za 60 sekundi i dobija se sa odnosom wapa sada i za 60 sekundi. Ovu vijednost mi želimo predvidjeti sa modelom.

Prije nego što se krenemo fokusirati na neke od tehnika mašinskog učenja uraditi ćemo neke osnovne pripreme za podatke. Prvo ćemo sortirati sve podatke po id-u dionice, zatim po id-u datum pa finalno po broju sekundi u kanti. Ovime su podaci poredani tako da prvo za jednu dionicu imamo sve podatke poredane hronološki zatim za drugu dionicu i tako dalje.

Primjetili smo da bliska i daleka cijena nisu određene za prvih pet minutna aukcije pa ćemo ih zamjeniti sa srednjom vrijednost od prodajne i kupovne cijene.

Sada ćemo sve ove kolone predstaviti u toplinskoj mapi korelacija za dionicu sa id-jem 0. Mapu vidite na slici 1.



Slika 1 Toplinska mapa atributa dionice 0

Na slici 1 primjetimo da je svaka osobina vezana sa nekom drugom osobinom pa nemamo viška kolona. Sada možemo početi koristiti neke metode mašinskog učenja da riješimo ovaj problem.

2.1 Obična Neuralna mreža

Obična neuralna mreža je model mašinskog učenja koji je inspirisan strukturom ljudskog mozga i njegovih neuralnih veza. Sastoji se od slojeva neurona koji su povezani težinama tako da svaka dva neurona u dva susjedna sloja su međusobno povezani. Prvi sloj je ulazni sloj koji prima podatke a zadnji sloj je onaj sloj koji vrši klasifikaciju ili u našem slučaju predikciju. Dok podaci prolaze kroz slojeve provode se različite

matematičke operacije između slojeva kako bi obradili ulazne podatke i generirali odgovarajuće izlazne rezultate.

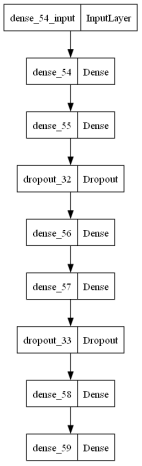
Prije nego što podaci pređu sa sloja na sloj koristi se aktivacijska funkcija da bi transformirala izlazne vrijednosti neurona iz sloja u odgovarajuće izlazne vrijednosti koje se šalju sljedećem sloju. Najčešće korištena aktivacijska funckija je ReLu koju vidite na slici 2a

U našem slučaju target može biti i negativna vrijednost pa moramo koristiti neku aktivacijsku funkciju koja može vratiti negativnu vrijednost pa ćemo koristiti swish kao aktivacijsku funckiju koju možete vidjeti na slici 2b. Ovu aktivacijsku funkciju ćemo koristiti i za sve ostale neuralne mreže. Ovu aktivacijsku funckiju ćemo malo modificirati da joj kodomen bude sličniji našem datasetu.

(a) Relu **(b)** Swish

Slika 2 Relu i Swish aktivacijske funckije

Podatke ćemo prvo normalizirati, zatim ćemo praviti svoje grupe za treniranje. Grupe se u neuralnim mrežama koriste da bi se ubrzalo treniranje to jest ne optimizujemo mrežu nakon što prođemo kroz sve podatke (što se zove epoha) nego kad prođemo kroz grupu podataka. Mi ćemo konstruisati grupe tako da prva grupa prestavlja podatke svih dionica za prvi dan, druga grupa podatke svih dionica za drugi dan i tako dalje. Ovakve grupe ćemo konstruisati za treniranje svih ostalih neuralnih mreža. Naša neuralna mreža izgleda kao na slici 3



Slika 3 Model obične neuralne mreže

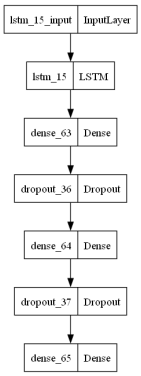
Dropoute koristimo da se mreža ne bi previše prilagodila treningu. Ovakve probleme gdje podaci koji prolaze kroz mrežu zavise od podataka koji su prošli kroz mrežu prije nije prirodno rješavati običnom neuralnom mrežom, jer ona takve stvari ne uči. Ovo nas motiviše da koristimo Rekurentne Neuralne mreže.

2.2 Naivna Rekurentna Neuralna mreža

Rekurentna neuralna mreža (RNN) je vrsta neuralnih mreža koja je dizajnirana za obradu sekvencijalnih podataka. Ključna karakteristika RNN-a je to što imaju unutrašnju petlju koja omogućava prenošenje informacija iz prethodnih koraka u naredne korake obrade. To im omogućuje da efikasno obrađuju podatke koji su organizirani u sekvence, kao što su naši podaci koji su organizirani u vremenske serije.

Serije se prave tako što više redova šaljemo od jednom i pošto nam target zavisi od sljedećih 60 sekundi nama

će sekvence biti svi podaci u 60 sekundi gdje ćemo za slučajeve gdje ne možemo popuniti sve podatke slati 0. Naša rekurentna mreža izgleda kao na slici 4.



Slika 4 Model naivne rekurentne mreže

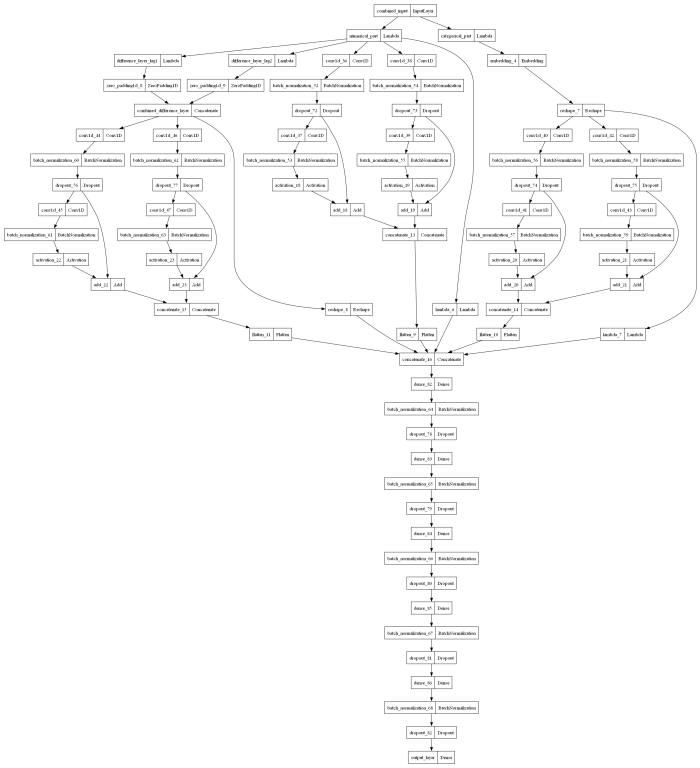
Long short-term memory (LSTM) je tip RNN-a koji rješava problem gdje RNN-ovi često brzo zaborave podatke koji su nekad na početku došli a oni su nam nekada bitni kasnije, pa pored toga što uči kako predviđati ona uči i kakve podatke treba pamtiti.

Ako pretpostavimo da nam dionice ne zavise jedna od druge ima smisla trenirati 180 neuralnih mreža, jedan za svaku dionicu. Ovo ćemo uraditi iako prelazi granice memorije dozvoljene na takmičenju, a koristiti ćemo isti model kao na slici 4.

Sada ćemo posmatrati mrežu koja je trenutno jedna od boljih na takmičenju.

2.3 Konvoluciona Neuralna mreža

Konvolucijska neuralna mreža (CNN) je vrsta neuralne mreže koja je posebno dizajnirana za obradu strukturnih podataka poput slika. Glavna karakteristika CNN-a je korištenje konvolucijskih slojeva koji se pomjeraju



Slika 5 Model konvolucione mreže sa takmičenja

kroz isti podatak i izdvajaju bitne karakteristike iz djelova podataka. CNN-i su izuzetno efikasni u obradi

vizualnih podataka kao što su slike, ali ih također možemo koristiti za obradu podataka u drugim domenima kao što je korišteno na ovom takmičenju.

Prije no što krenemo sa strukturom mreže oblikovati ćemo podatke tako da će nam ulaz u CNN uz orginalne atribute imate neke vrijednosti iz prijašnjih dana. U podatke od jednog vremenskog intervala ćemo uzeti target u to isto vrijeme u prethodna 3 dana. Pored ovoga ćemo računati imbalanse zadate u (2) od gotovo svake dvije kolone.

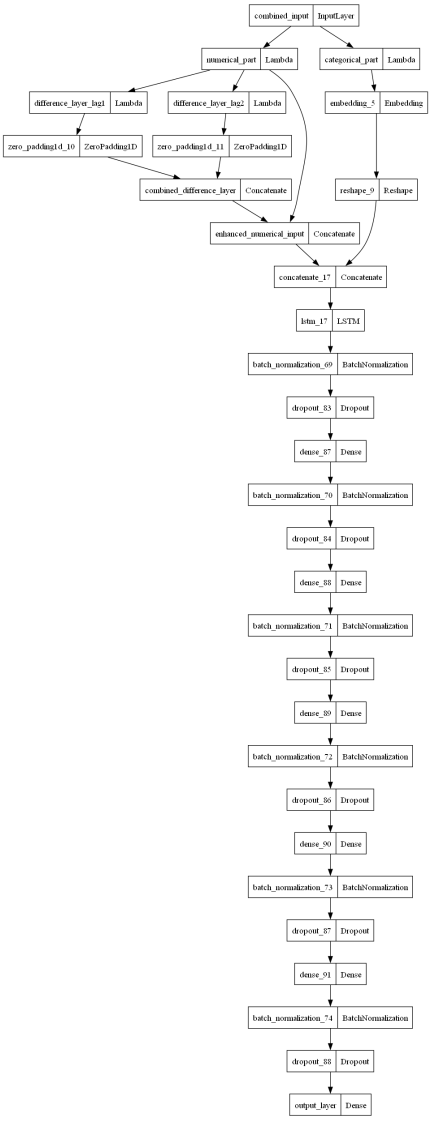
$$a > b, \quad imbalance(a,b) = \frac{a-b}{a+b} \tag{2}$$

Napraviti ćemo jedan kompleksan model u kojem ćemo raditi razne modifikacije sa atributima i preskakanje nekih veza (da bi izbjegli problem dubokih mreža gdje gradijenti ili idu u 0 ili u beskonačno). Ovaj model je kopiran sa takmičenja i njegova struktura je data na slici 5.

Koristeći prethodni model kao inspiraciju pokušati ćemo napraviti jednostavniji model koji koristi LSTM.

2.4 Rekurentna Neuralna mreža

Podatke ćemo isto oblikovati kao i kod Konvolucione Neuralne mreže samo ćemo ih još dodatno staviti u serije kao i predhodni LSTM. Strukturu vidite na slici 6.



Slika 6 Model naše rekurentne mreže

Prvo ulaz podijelimo na dva dijela kategorički i numerički (numerical_part, categorical_part). Sve cijene i veličine prolaze kroz numerički dio a oznaka imbalansa i broj sekundi ide u kategorički dio.

Kategorički dio transformišemo u brojeve (embedding_5), ovo smo uradili unutar mreže da bi ona učila kako najbolje transformisati te podatke. Zatim oblikujemo (reshape_9) za lakše spajanje sa numeričkim dijelom.

Numeričkom dijelu dodamo atribute od prije nekoliko dana (difference_layer_lag_1, difference_layer_lag_2), odvojeno posmatramo veličine i cijene a ostale podatke proslijedimo u (enhanced_numerical_input). Naša mreža zapravo uči koliko dana prije bi bilo idealno da posmatra atribute koji predstavljaju cijenu a koliko dana prije atribute koji predstavljaju veličinu. U slučaju da neke atribute ne može naći popunimo ih nulama (zero_padding_1d_10, zero_padding_1d_11).

Zatim sve spojimo zajedno, i to prvo cijena i veličine (combined_difference_layer), zatim cijeni i veličine sa ostalim (enhanced_numerical_input), te sve ovo sa kategoričkim (concatinate_17). Ovako dobijene podatke ubacimo u LSTM i nakon LSTM-a koristimo običnu neuralnu mrežu da predvidimo target.

2.5 Boost algoritmi

Boost algoritmi su vrsta algoritama mašinskog učenja koji se koriste za poboljšanje prediktivnih modela. Oni rade tako što iterativno treniraju niz slabih klasifikatora ili regresora (u našem slučaju regresora), pri čemu svaki sljedeći klasifikator/regresor pokušava popraviti greške prethodnih. LightGBM (LGB) i CatBoost su dva popularna boost algoritma koje ćemo mi koristiti u ovom radu.

2.5.1 Oblikovanje Atributa

Pošto Boost algoritmi za razliku od ostalih algoritama ne zahtjevaju normalizaciju podataka taj dio nećemo raditi ali ćemo napraviti što je moguće više atributa od postojećih (eng. Feature engineering, FE). Ovo smo već uradili sa imbalansima i dodavanjem atributa od prethodnih dana ali ćemo sada dodati još neke atribute. Mi ćemo od 14 bitnih atributa koji su nam dati napraviti 172. Neki od atributa koje smo napravili su:

- **predikcija wap-a sljedećih par dana**, ovu predikciju računamo pomoću Linearne regresije na prethodne wap-ove.
- **imbalanse**, imbalanse skoro svake dvije kolone opisane u (2).
- **trostruki imbalanse**, imbalanse skoro svake tri kolone opisane u (3).

$$a > b > c, \quad triple\_imbalance(a,b,c) = \frac{a-b}{b-c} \tag{3}$$

- **srednja cijena**, cijenu koju dobijemo kao prosjek kupljenje i prodajne cijene.
- **srednja vrijednost atributa**, srednja vrijednost nekih atributa te dionice od prije.
- **devijacija atributa**, devijacija nekih atributa te dionice od prije.
- **kurtoza atributa**, kurtoza nekih atributa te dionice od prije.
- **lag-ovi**, vrijednosti nekih atributa od prije 1, 2, 3, 6, i 10 dana.
- **vrijeme promjene imbalansa**, prije koliko vremena je imbalans bio promijenjen.
- **kvantila atributa**, u kojoj kvantili pripada određeni atribut.

2.5.2 LightGBM

LightGBM (LGB) je boost algoritam zasnovan na stablima odlučivanja, koji je razvila kompanija Microsoft. On radi na principu gradient boosting framework-a, gdje se više stabla odlučivanja trenira iterativno kako bi se postigla bolja predikcija. Algoritam počinje sa jednim stablom odlučivanja a zatim se dodaju dodatna stabla da bi se poboljšala predikcija.

2.5.3 CatBoost

CatBoost (CTB) je boost algoritam, specijaliziran za rad s kategoričkim varijablama pa nije dizajniran za regresione probleme kao naš preblem ali se može koristiti u te svrhe. Osim toga, CatBoost je poznat po svojoj sposobnosti automatskog eliminiranja nepotrebnih atributa, što može biti korisno za smanjenje prenaučenosti i poboljšanje predikcije modela. Naš CatBoost model će sa 172 atributa spasti na 100 atributa.

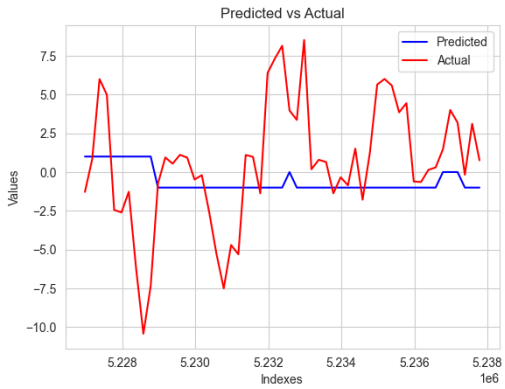
3 Rezultati

Svaku od mreža smo trenirali 10 epoha, za preciznost smo koristili apsolutnu grešku zadatu u (4).

$$Loss = \sum_{i=1}^{n} (y\_true[i] - y\_pred[i]) \tag{4}$$

U formuli (4) $y\_true[i]$ predstavlja pravu vrijednost targeta u i-tom vremenskom intervalu a $y\_pred[i]$ vrijednost targeta u tom intervalu koju je naš model predvidio. Zadnji dan koristimo za testiranjem, na tom danu naravno neuralne mreže nisu trenirane, a zadnji dan dionice sa id-jem 0 ćemo koristiti za grafiranje.

Trivijalno rješenje ovog problema je da stavimo da je target jednak imbalansu, to jest ako se više prodaje dionica nego kupuje onda očekujemo da će cijena opasti, a ako se više kupuje dionica nego što se prodaje tada očekujemo da će rasti. Ovako dobijamo rješenje sa kojim možemo uporediti model ima apsolutnu grešku od 4.9009 i grafik je dat na slici 7.

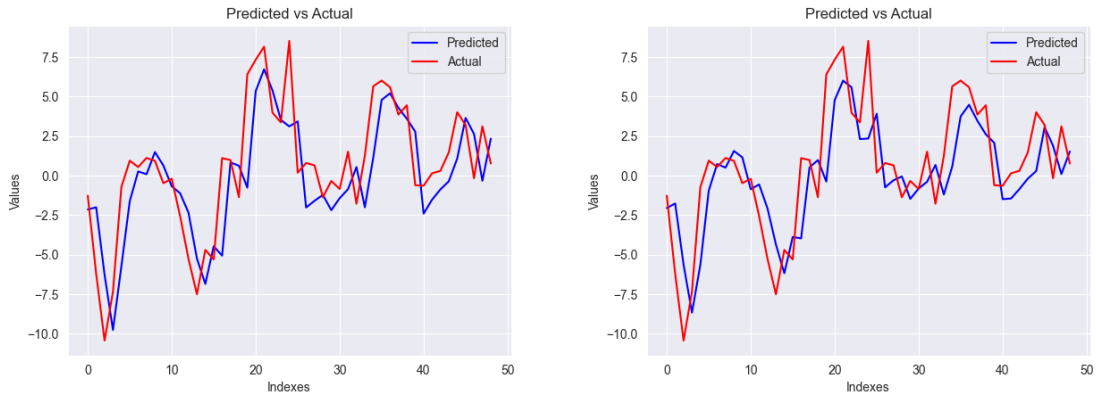


Slika 7 Grafik trivijalnog rješenja

Obična neuralna mreža ima absolutnu grešku od 4.8766 što znači da je malo bolja od trivijalnog rješenja. Ovo je očekivano jer kao što smo rekli obična neuralna mreža nije napravljena za podatke sekvencijalnog tipa. Grafik možemo vidjeti na slici 8.

Slika 8 Grafik obične neuralne mreže

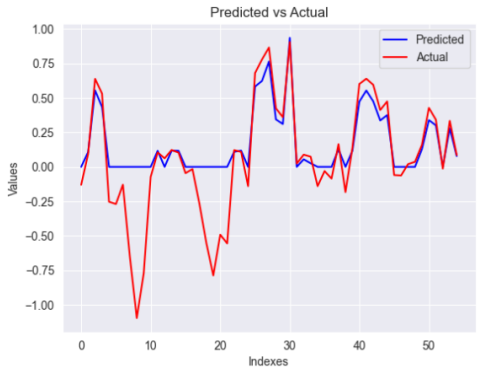
Rekuretne mreže su ipak napravljene za sekvencijalne podatke a to možemo vidjeti po rezultatima gdje obični LSTM ima apsolutnu grešku od 2.5826 a LSTM koji je treniran posebno za svaku dionicu ima apsolutnu grešku od 2.1438. Grafike ovih mreža možete vidjeti kao slike 9a i 9b redom.



(a) Graf običnog LSTM-a **(b)** Grafik LSTM-a po dionici

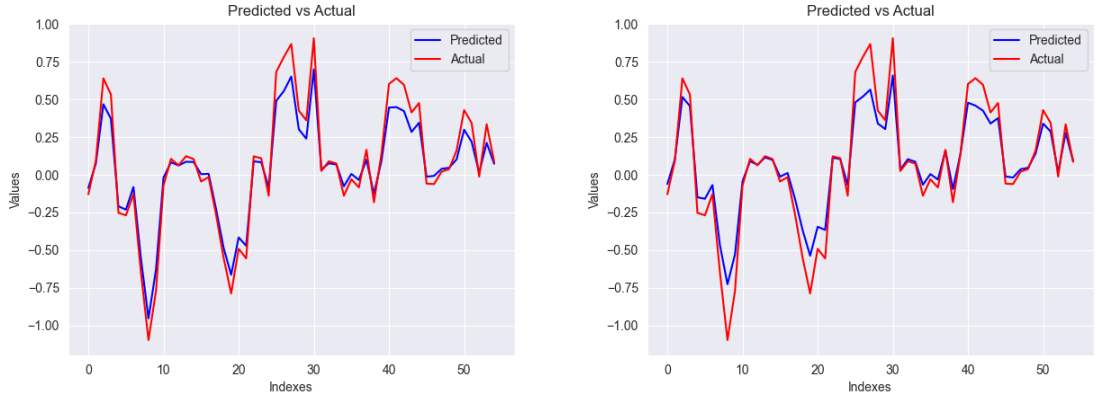
Slika 9 Grafovi naivnih rekurentnih mreža

Ako koristimo malo oblikovanja osobina i orginalni konvolucioni model sa takmičenja dobili smo graf kao na slici 10. Vidimo da model ne predviđa vrijednosti manje od 0 pa zbog toga smo modifikovali aktivacijsku funkciju.



Slika 10 Grafik konvolucione bez naše aktivacijske

Nakon promjene aktivacijske funkcije konvolucioni model dolazi do apsolutno greške od 0.1251. Ako malo pojednostavimo taj model i koristimo rekurentu mrežu naša apsolutna greška postaje 0.1485. Grafove ovih modela možete vidjeti na slikama 11a i 11b redom.

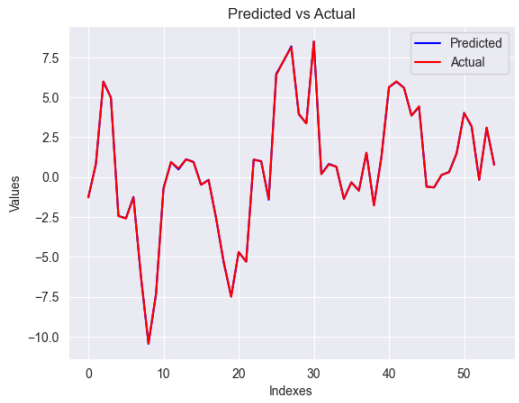


(a) Grafik Konvolucione mreže **(b)** Grafik našeg LSTM-a

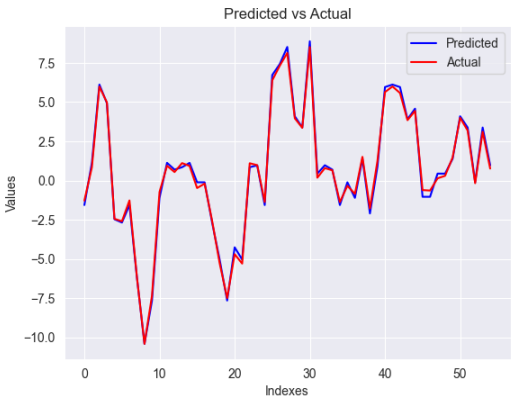
Slika 11 Grafovi konvolucione i našeg LSTM-a

Boost algoritmi sa mnogo oblikovanja osobina su se pokazali kao najbolji modeli, gdje za LGB ima apsolutnu

grešku od 0.0035 a za *CTB* grešku od 0.0746. Grafove ovih Boost algoritama vidite na slikama 12a i 12b redom.



(a) Grafik LGB-a



(b) Grafik CTB-a

Slika 12 Grafovi konvolucione i našeg LSTM-a

Komparativnu analizu svih ovih metoda iz mašinskog učenja vidite na tablici 1.

Metoda	Apsolutna greška
Trivijalno	4.9009
Obični NN	4.8766
Naivni LSTM 1	2.5826
Naivni LSTM 2	2.1438
CNN	0.1251
Naš LSTM	0.1485
LGB	0.0035
CTB	0.0746

Tablica 1 Tabela apsolutne greške kod metoda mašinskog učenja

4 Zaključak

U zaključku, ovaj rad istražio je primjenu različitih modela dubokog učenja za predviđanje kretanja cijena dionica na zatvaranju Nasdaqa korištenjem podataka iz knjige naloga i završne aukcije.

Rezultati su sažeti u tablici 1 koja prikazuje apsolutnu grešku svake metode, vidimo da su neki modeli bili mnogo uspješniji od drugih u predviđanju cilja.

Među testiranim modelima, LGB i CTB su se pokazali kao najbolje metode sa greškama od 0.0035 i 0.0746 respektivno. Ovi rezultati sugerišu da su ensamblerske metode veoma efektivni u predikciji ovog sintetičkog cilja. Sa druge strane, vidimo da Naivni LSTM iako je bolji od trivijlanog i obične NN nisu se značanje rezultati dobili sve dok se nisu koristili naprednije modele i tehnike dubokog učenja.

Iako su ensamblerske metode dali veoma dobre rezultate, uvijek ima mjesta da poboljšamo rezultate kao dodavanje više feature-a pomoću feature engineering-a ili hyperparameter fine-tuning. Kroz čitav rad smo govorili kako je predviđanje na berzi veoma dinamičan, a naši podaci su veoma statični, da prilagodimo naše modele više realnom svijetu možemo da koristimo tehniku re-training koji ponovo trenira mrežu na određen broj dana.