

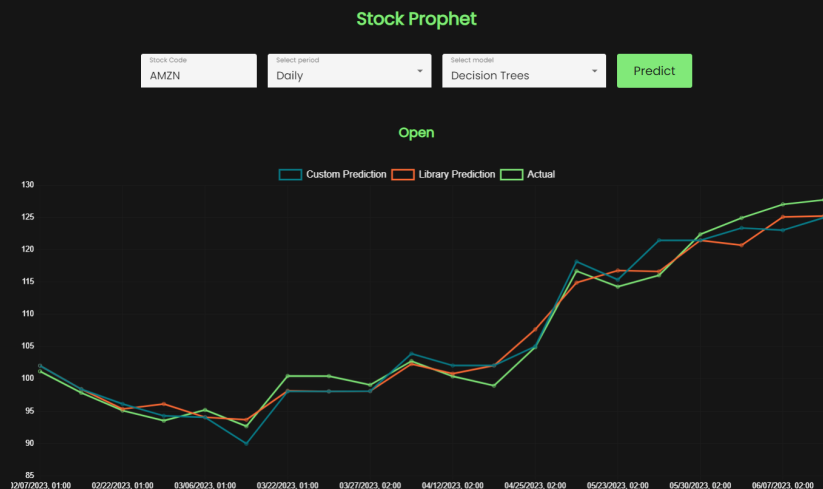
Stock Prophet

Predviđanje cena akcija

Miloš Čuturić
Luka Đorđević
Marko Janošević

Predviđanje cena akcija korišćenjem mašinskog učenja je popularan i izazovan problem u finansijama i data-science-u.

Cilj ovog projekta je korišćenje istorijskih cena akcija i povezanih podataka za obuku modela mašinskog učenja koji može precizno predvideti buduće cene akcija. Ovo može pomoći investitorima da donesu bolje informisane odluke o kupovini ili prodaji akcija i potencijalno da povećaju svoje prinose. Ideja je da se koriste različiti algoritmi i da se porede njihova predviđanja kako bi se dobili bolji i precizniji rezultati.



U projektu su implementirana i analizirana četiri ključna algoritma: Linearna regresija, Stabla odluke, Support Vector Machine i Neuronske mreže. Koristeći iste skupove testnih podataka, izvršili smo direktnu komparaciju između naših implementacija ovih algoritama i njihovih ekvivalenta iz Sklearn i Keras biblioteka, omogućujući nam da ocenimo efikasnost i preciznost našeg koda u realnom kontekstu predviđanja tržišta.

- **Linearna regresija**

Naša implementacija linearne regresije obuhvata inicijalizaciju parametara nagiba (m) i preseka (c) linije. Za obuku modela, koristi se formula minimalnih kvadrata da bi se izračunao optimalni nagib (m) i presek (c) za linearni regresijski model.

- **Neuronske mreže**

Naša neuronska mreža je inicijalizovana sa dva sloja težina i dva biasa, sa definisanim hiperparametrima za stopu učenja i L2 regularizaciju. Težine su inicijalizovane koristeći He inicijalizaciju, dok su biasi inicijalizovani na nulu. Funkcija aktivacije koju koristimo je Leaky ReLU, koja pomaže u ublažavanju problema nestajućih gradijenata. Ova funkcija se koristi u unaprednom prolazu, uzimajući u obzir težine i biasa za svaki sloj. Težine i biasi se ažuriraju u metodi unazadnog prolaza, koristeći gradijent gubitka u odnosu na težine, smanjen za stopu učenja i povećan za L2 regularizaciju. Kada obučavamo mrežu, koristimo mini-batch gradijentni spust, gde se ažuriranja vrše na manjem podskupu podataka, a ne na celom skupu podataka. Ova metoda je efikasna i omogućava brže konvergencije.

- **Stabla odluke**

Izgradili smo stablo odluke koristeći rekursivnu funkciju. U okviru ove funkcije, najpre pokušavamo da pronađemo najbolje mesto za podelu skupa podataka, a zatim rekursivno gradimo leve i desne grane stabla. Ovaj proces se nastavlja sve dok ne dostignemo maksimalnu dubinu stabla ili minimalan broj uzoraka za podelu. Kada izgradimo stablo, prolazimo kroz njega sve do listova, gde donosimo odluku na osnovu vrednosti u listu. Ovaj proces se ponavlja za sve ulaze, što rezultira listom predviđanja.

- **Support Vector Machine**

Naš model koristi linearnu klasifikaciju. Počinjemo sa inicijalizacijom težina na nulu i postavljanjem odgovarajućih hiperparametara. Prolazi se kroz skup za obuku i ažuriraju se težine na osnovu gubitka. Ako je uzorak pravilno klasifikovan, izvršava se ažuriranje koje smanjuje težinu. Međutim, ako uzorak nije pravilno klasifikovan, izračunava se gubitak na osnovu ciljne vrednosti i vrednosti predikcije i težina se ažurira na osnovu ovog gubitka. SVM koristi "hinge" gubitak koji kažnjava pogrešno klasifikovane tačke i tačke koje su blizu odlučujuće granice. Finalni rezultat predikcije se pretvara u oznaku klase koristeći funkciju znaka.